An adaptive domain-based POD/ECM hyper-reduced modeling framework without offline training

Rocha, I. B.C.M.; van der Meer, F. P.; Sluys, L. J.

DOI
10.1016/j.cma.2019.112650

Publication date
2020

Document Version
Final published version

Published in
Computer Methods in Applied Mechanics and Engineering

Available online at www.sciencedirect.com
ScienceDirect

Comput. Methods Appl. Mech. Engrg. 358 (2020) 112650

Computer methods in applied mechanics and engineering

www.elsevier.com/locate/cma

An adaptive domain-based POD/ECM hyper-reduced modeling framework without offline training

I.B.C.M. Rocha, F.P. van der Meer*, L.J. Sluys

Delft University of Technology, Faculty of Civil Engineering and Geosciences, P.O. Box 5048, 2600GA Delft, The Netherlands

Received 16 March 2019; received in revised form 7 August 2019; accepted 21 September 2019
Available online xxxx

Abstract

This work presents a reduced-order modeling framework that precludes the need for *offline* training and adaptively adjusts its lower-order solution space as the analysis progresses. The analysis starts with a fully-solved step and elements are clustered based on their strain response. Elements with the highest strains are solved with a local/global approach in which degrees of freedom from elements undergoing the highest amount of nonlinearity are fully-solved and the rest is approximated by a Proper Orthogonal Decomposition (POD) reduced model with full integration. Elements belonging to the remaining clusters are subjected to a hyper-reduction step using the Empirical Cubature Method (ECM). *Online* error estimators are used to trigger a retraining process once the reduced solution space becomes inadequate. The performance of the framework is assessed through a series of numerical examples featuring a material model with pressure-dependent plasticity.

Keywords: Reduced-order modeling; Adaptive reduction; Local/global approach; Hyper-reduction

1. Introduction

The search for efficient and reliable acceleration techniques for numerical analysis is currently the subject of extensive research. A number of applications in computational mechanics rely on employing high-fidelity numerical techniques that quickly become computational bottlenecks. Concurrent multiscale analysis (FE^2) is one such technique [1,2], as it involves embedding a lower-scale finite element mesh at each higher-scale integration point. Even when relatively coarse meshes are used at the macroscale, the associated execution times make the approach infeasible for most practical applications [3]. Mechanical models that require a large number of time steps (*e.g*. dynamics or fatigue analysis) also become computationally infeasible, in particular if complex material models are employed [4] and when processes at a slower time scale are also being modeled (*e.g*. aging or slow cyclic damage accumulation) [5,6]. Finally, material or structural design through optimization algorithms or response surfaces involves a large number of model executions with different input parameters [7,8], a process that is rendered inefficient if single model executions are computationally taxing.

A popular approach for mitigating the computational effort of these so-called *many-query* applications consists in substituting the full-order models by reduced-order surrogates with a similar level of fidelity but with

* Corresponding author.
E-mail address: f.p.vandermeer@tudelft.nl (F.P. van der Meer).

https://doi.org/10.1016/j.cma.2019.112650

lower computational complexity. The underlying assumption behind this approach is that the full-order solution (*e.g.* nodal displacements) can be posed in a lower-dimensional solution manifold (*e.g.* global displacement modes) with limited loss of accuracy. The problem of constructing a reduced-order model therefore lies in finding a suitable lower-dimensional manifold. This is commonly done before the analysis (*offline* phase) through a training process that collects solution snapshots and extracts a reduced solution basis through data compression or pattern recognition techniques — *e.g.* Proper Orthogonal Decomposition (POD) [9] or Proper Generalized Decomposition (PGD) [10]. These techniques are often combined with *hyper-reduction* methods, *e.g.* the Discrete Empirical Interpolation Method (DEIM) [11] or the Empirical Cubature Method (ECM) [12], that reduce the number of constitutive model computations and allow for further acceleration.

Although these techniques work remarkably well for linear models, they become inefficient for models with nonlinear time- and history-dependent responses (*e.g.* (visco)-plasticity or damage). For these cases, the size of the parameter space to be spanned by the reduced-order model increases dramatically [13], making the construction of an efficient reduced model a difficult task for a number of reasons. Firstly, choosing the correct scenarios for training becomes a complex task — in a highly nonlinear model, subtle changes in loading direction can lead to significantly different post-localization behavior. Secondly, the size of the reduced solution space necessary to maintain a reasonable level of accuracy becomes unacceptably large (*i.e.* the singular values of the snapshot matrix decay at a slow rate), negating the acceleration obtained by the reduction process. Alternative approaches and additional model components are therefore necessary.

One solution, proposed in [13–15], consists in constructing multiple reduced solution spaces and using them at different moments during the analysis. Although this approach overcomes the issue of the slow decay in singular values, the training process to obtain the bases is still complex (see [16] for an efficient greedy algorithm for training). Another solution is to limit the training set to only elastic load cases, use the snapshots to divide the domain into element clusters and solve for the reduced problem under the assumption of constant strain inside each cluster [17]. Alternatively, only the POD reduction can be trained and an adaptive wavelet approximation can be used for computing internal forces [18]. Finally, the reduced model can be made adaptive by partitioning the global equilibrium system of equations (local/global approach) in order to fully solve for degrees of freedom in parts of the mesh where the sources of nonlinearity are located and add this new information to the basis [19]. This framework can also be expanded by using domain decomposition to divide the mesh into domains and use different POD bases for each [20].

The solution explored in this work can be seen as a combination of a local/global approach [19] with the idea of clustering proposed in [17]. An adaptive hyper-reduction framework is proposed which eliminates the need for an *offline* training phase by starting from a fully-solved step and consecutively constructing reduced and hyper-reduced versions of the model during the two following steps. The strains obtained during the fully-solved step are used to divide the finite elements into clusters. Elements belonging to the cluster with highest average strain are solved with POD while the rest of the domains are solved with a combination of POD and ECM. Inside the POD domain, the level of reduction is further tweaked by fully solving the degrees of freedom (DOFs) of elements with the highest amount of nonlinearity. Three levels of reduction (full, reduced and hyperreduced) can therefore coexist on the same finite element mesh. *Online* error measurements are used to predict when the reduced basis and reduced integration sets should be updated. The model then switches back to a full step, cluster topology is updated and the domains are retrained. The framework is used in combination with a pressure-dependent plasticity model in a number of numerical examples in order to assess its precision and level of acceleration.

1.1. Mathematical notation

In this work, vectors are represented by boldfaced lower-case symbols (*e.g.* $\mathbf{v}$) and matrices are given by boldfaced upper-case symbols (*e.g.* $\mathbf{M}$). Sets of indices are represented in upper-case calligraphic script (*e.g.* $\mathcal{S}$). The subscript h (*e.g.* $\mathbf{v}_{\mathrm{h}}$) indicates an entity in the full solution space, while reduced-order entities are not marked in order to keep the notation compact.

A subscript with a set between parentheses (*e.g.* $\mathbf{v}_{(\mathcal{S})}$) indicates a selection operation: lines of the operand that correspond to the indices in the set are extracted from the full-size operand. If two sets are used in the selection operation (*e.g.* $\mathbf{M}_{(\mathcal{S},\mathcal{T})}$), a submatrix is extracted from the operand with rows corresponding to the indices in $\mathcal{S}$ and columns corresponding to the indices in $\mathcal{T}$. When a single set is used with a matrix, all columns are selected (*i.e.* $\mathbf{M}_{(\mathcal{S})} \equiv \mathbf{M}_{(\mathcal{S},\mathrm{all})}$).

2. Reduced-order modeling

In this section, the reduced-order modeling techniques that constitute the main ingredients of the present framework are introduced, before it is shown how they are combined in the adaptive framework of Section 3. The formulation starts from a full-order model and the two levels of reduction (Galerkin projection and hyper-reduction) are applied consecutively. The formulations are meant to be self-contained but only the essential steps are included in order to keep the discussion compact.

2.1. Full-order equilibrium problem

The numerical problem at hand consists in finding the weak-form equilibrium of a volume Ω subjected to Dirichlet constraints at the boundary Γ_u and Neumann boundary conditions at Γ_f ($\Gamma_\mathrm{u} \cap \Gamma_\mathrm{f} = \varnothing$) using the Finite Element Method. In the full-order solution space, this equilibrium problem can be written as:

$$\mathbf{r}_\mathrm{h} = \mathbf{f}_\mathrm{h}^{\Gamma} - \mathbf{f}_\mathrm{h}^{\Omega}(\mathbf{u}_\mathrm{h}) = \mathbf{0} \tag{1}$$

where $\mathbf{r}_\mathrm{h} \in \mathbb{R}^N$ is a force residual defined as the difference between internal ($\mathbf{f}_\mathrm{h}^{\Omega}$) and external ($\mathbf{f}_\mathrm{h}^{\Gamma}$) forces and $\mathbf{u}_\mathrm{h} \in \mathbb{R}^N$ contains displacement values at the N mesh nodes. The global internal force vector $\mathbf{f}_\mathrm{h}^{\Omega}$ is computed through numerical integration of the force contributions at M integration points arranged throughout the mesh:

$$\mathbf{f}_\mathrm{h}^{\Omega} = \int_{\Omega} \mathbf{f}_\mathrm{h} d\Omega \approx \sum_{i}^{M} \mathbf{f}_\mathrm{h}(\mathbf{x}_i) w_i \tag{2}$$

where w_i is the integration weight of the point located at coordinates $\mathbf{x}_i$. These force contributions are obtained from the stresses $\boldsymbol{\sigma}$ at each integration point:

$$\mathbf{f}_\mathrm{h}\left(\mathbf{x}_i\right) = \mathbf{B}^\mathrm{T}\left(\mathbf{x}_i\right) \boldsymbol{\sigma}\left(\boldsymbol{\varepsilon}_i, \boldsymbol{\mu}_i\right) \tag{3}$$

where the matrix $\mathbf{B}$ converts nodal displacements to local strains and the stress $\boldsymbol{\sigma}$ is a function of strains $\boldsymbol{\varepsilon}$ and material history $\boldsymbol{\mu}$. The pressure-dependent plasticity constitutive model formulated in [21] is used here to demonstrate the proposed framework, but it is in principle suitable for any type of constitutive behavior.

Since material response is nonlinear, Eq. (1) is solved iteratively using the Newton–Raphson method. At each iteration, the approximation for the displacement field $\mathbf{u}$ is improved by an increment $\delta\mathbf{u}$ given by:

$$\delta\mathbf{u}_\mathrm{h} = -\mathbf{K}_\mathrm{h}^{-1}\mathbf{r}_\mathrm{h} \tag{4}$$

where $\mathbf{K}_\mathrm{h} = \partial\mathbf{f}_\mathrm{h}^{\Omega}/\partial\mathbf{u}_\mathrm{h}$ is the global tangent stiffness matrix. The volume is considered to be in equilibrium when:

$$\frac{\left\|\mathbf{f}_\mathrm{h}^{\Gamma} - \mathbf{f}_\mathrm{h}^{\Omega}(\mathbf{u}_\mathrm{h})\right\|}{\lambda} < \epsilon_\mathrm{solver} \tag{5}$$

where ϵ_solver is a predefined tolerance and λ is a scale factor computed from $\mathbf{f}_\mathrm{h}^{\Omega}$ and $\mathbf{f}_\mathrm{h}^{\Gamma}$ which corresponds to the magnitude of the external loads applied to Ω if Neumann boundary conditions are used.

2.2. Proper orthogonal decomposition (POD)

The first reduction technique consists in substituting the original equilibrium problem of size N by the reduced problem of solving for $n \ll N$ mode contributions $\boldsymbol{\alpha}$. After solving for $\boldsymbol{\alpha}$, the full-order displacement field is in turn recovered as a linear combination of these modes:

$$\mathbf{u}_\mathrm{h} = \boldsymbol{\Phi}\boldsymbol{\alpha} \tag{6}$$

where $\boldsymbol{\Phi} \in \mathbb{R}^{N\times n}$ is a basis matrix with a column-wise arrangement of orthonormal displacement modes ϕ_i:

$$\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\phi}_1 & \boldsymbol{\phi}_2 & \cdots & \boldsymbol{\phi}_n \end{bmatrix} \tag{7}$$

The basis matrix $\boldsymbol{\Phi}$ is commonly obtained through a set of numerical experiments (*offline training*): the full-order model is executed and the resulting P displacement snapshots are stored as columns in a matrix $\mathbf{X}_\mathrm{u} \in \mathbb{R}^{N\times P}$. In order to extract an optimum basis from this set of snapshots, an orthonormal set of basis vectors is computed from $\mathbf{X}_{\mathrm{u}}$ through a Singular Value Decomposition (SVD) operation:

$$\mathbf{X}_{\mathrm{u}} = \mathbf{\Phi}\mathbf{S}\mathbf{V}^{\mathrm{T}} \tag{8}$$

where $\mathbf{S}$ is a diagonal matrix with singular values sorted from the highest to the lowest and $\mathbf{V}$ is the right-singular matrix of $\mathbf{X}_{\mathrm{u}}$. Depending on the rate of decay of the singular values, the SVD operation can be truncated to the first $n \leq P$ basis vectors associated to the highest n singular values. The SVD guarantees that the reduced model will contain as much information about the full-order behavior as possible for a model with n degrees of freedom.

Finally, the full-order equilibrium problem of Eq. (1) can be reduced by ensuring orthogonality between the residual and the basis vectors (Galerkin projection):

$$\mathbf{\Phi}^{\mathrm{T}}\mathbf{r}_{\mathrm{h}} = \mathbf{0} \tag{9}$$

which results in reduced versions of the force vectors and stiffness matrix:

$$\mathbf{f}^{\Omega} = \mathbf{\Phi}^{\mathrm{T}}\mathbf{f}_{\mathrm{h}}^{\Omega} \qquad \mathbf{f}^{\Gamma} = \mathbf{\Phi}^{\mathrm{T}}\mathbf{f}_{\mathrm{h}}^{\Gamma} \qquad \mathbf{K} = \mathbf{\Phi}^{\mathrm{T}}\mathbf{K}_{\mathrm{h}}\mathbf{\Phi} \tag{10}$$

2.3. Empirical Cubature Method (ECM)

Although the size of the equilibrium problem is greatly reduced by using POD, the full-order force vector $\mathbf{f}_{\mathrm{h}}^{\Omega}$ and stiffness matrix $\mathbf{K}_{\mathrm{h}}$ of Eq. (10) must still be computed at every time step. Computing these integrals often make up a substantial portion of the total analysis time, especially if complex material models are used.

The Empirical Cubature Method (ECM) [12] aims to eliminate this bottleneck by finding a reduced set of $m \ll M$ integration points $\mathcal{Z}$ and modified weights $\boldsymbol{\varpi}$ that minimizes the error of the following approximation:

$$\mathbf{f}^{\Omega} = \mathbf{\Phi}^{\mathrm{T}}\left(\sum_{i=1}^{M}\mathbf{f}_{\mathrm{h}}(\mathbf{x}_i)w_i\right) \approx \mathbf{\Phi}^{\mathrm{T}}\left(\sum_{j=1}^{m}\mathbf{f}_{\mathrm{h}}(\mathbf{x}_j)\varpi_j\right) \tag{11}$$

This substantial reduction in the number of integration points is made possible by the reduced size of $\mathbf{f}^{\Omega}$ compared to the original $\mathbf{f}_{\mathrm{h}}^{\Omega}$. ECM is therefore a second reduction technique (*hyper-reduction*) built atop POD.

Computing $\mathcal{Z}$ and $\boldsymbol{\varpi}$ starts by running the POD-reduced model for a second set of P training steps during which stresses at every integration point are stored in a snapshot matrix $\mathbf{X}_{\sigma} \in \mathbb{R}^{sM\times P}$ ($\boldsymbol{\sigma} \in \mathbb{R}^{s}$). A basis matrix for $\boldsymbol{\sigma} \in \mathbb{R}^{sM\times q}$ is computed through SVD (with $q \leq P$):

$$\mathbf{X}_{\sigma} = \mathbf{\Psi}\mathbf{S}\mathbf{V}^{\mathrm{T}} \tag{12}$$

which is, in turn, used to compute a basis matrix $\mathbf{\Lambda} \in \mathbb{R}^{M\times nq}$ for the error caused by the approximation of Eq. (11):

$$\mathbf{\Lambda} = \begin{bmatrix}\mathbf{\Lambda}_1 & \mathbf{\Lambda}_2 & \cdots & \mathbf{\Lambda}_q\end{bmatrix} \tag{13}$$

where each submatrix $\mathbf{\Lambda}_j \in \mathbb{R}^{M\times n}$ corresponding to one of the q stress modes can be written as:

$$\mathbf{\Lambda}_j = \begin{bmatrix} \sqrt{w_1}\left(\mathbf{f}_j^{1}\left(\mathbf{x}_1\right) - \frac{1}{\Omega}\mathbf{f}_j^{\Omega}\right) \\ \sqrt{w_2}\left(\mathbf{f}_j^{1}\left(\mathbf{x}_2\right) - \frac{1}{\Omega}\mathbf{f}_j^{\Omega}\right) \\ \vdots \\ \sqrt{w_M}\left(\mathbf{f}_j^{1}\left(\mathbf{x}_M\right) - \frac{1}{\Omega}\mathbf{f}_j^{\Omega}\right) \end{bmatrix} \tag{14}$$

and the contribution of each point is given by:

$$\mathbf{f}_j^{i} = \mathbf{\Phi}_i^{\mathrm{T}}\mathbf{B}_i^{\mathrm{T}}\mathbf{S}_j\psi_j \tag{15}$$

where $\mathbf{\Phi}_i$ is a submatrix of the POD basis corresponding to the DOFs of the finite element that contains the point i, $\mathbf{B}_i$ is the shape function gradient matrix of Eq. (3) at point i and S_j and $\mathbf{\Psi}_j$ are the jth singular value and mode of ψ, respectively.

With these definitions, the least-squares minimization of the force approximation error can be posed as:

$$(\boldsymbol{\beta}, \boldsymbol{\mathcal{Z}}) = \underset{\overline{\beta} \geq \mathbf{0}, \overline{\boldsymbol{\mathcal{Z}}}}{\arg\min} \left\| \mathbf{J}_{(\overline{\boldsymbol{\mathcal{Z}}})} \overline{\boldsymbol{\beta}} - \mathbf{b} \right\|^2 \tag{16}$$

where the terms $\mathbf{J}$ and $\mathbf{b}$ are given by:

$$\mathbf{J} = \begin{bmatrix} \boldsymbol{\Lambda} & \sqrt{\mathbf{w}} \end{bmatrix}^{\mathrm{T}} \qquad \mathbf{b} = \begin{bmatrix} \mathbf{0} & \Omega \end{bmatrix}^{\mathrm{T}} \tag{17}$$

This non-negative least-squares problem is solved with a greedy selection algorithm that gradually inserts points in $\mathcal{Z}$ until the approximation error is sufficiently small or $\mathbf{J}_{(\mathcal{Z})}$ attains full rank (*i.e.* when size$(\mathcal{Z}) = qn + 1$) [12]. Finally, the modified weights $\boldsymbol{\varpi}$ are computed as:

$$\varpi_i = \sqrt{w_i}\beta_i \tag{18}$$

2.4. History recovery

Performing the integration of $\mathbf{f}^{\Omega}$ only on the reduced set of points $\mathcal{Z}$ means that material history ($\boldsymbol{\mu}$ in Eq. (3)) is not computed at the remaining integration points. In the context of an adaptive framework that periodically switches back to the full-order solution space, this outdated history on most of the mesh may lead to convergence issues when the next full step is executed.

Material history can be recovered through a *Gappy Data* procedure by using the history at $\mathcal{Z}$ to interpolate the values of the remaining points [22–24]. In order to provide a basis for such interpolation, history snapshots are gathered during ECM training and stored as columns in a matrix $\mathbf{X}_{\mu} \in \mathbb{R}^{rM \times P}$ and a basis matrix $\boldsymbol{\Upsilon} \in \mathbb{R}^{rM \times p}$ (with $\boldsymbol{\mu} \in \mathbb{R}^{rM}$ and $p \leq P$) is extracted through SVD:

$$\mathbf{X}_{\mu} \approx \boldsymbol{\Upsilon}\mathbf{S}\mathbf{V}^{\mathrm{T}} \tag{19}$$

Defining $\mathcal{Y}$ as the set of points with outdated history, the basis matrix $\boldsymbol{\Upsilon}$ can be used to recover $\boldsymbol{\mu}_{(\mathcal{Y})}$ through a least-squares procedure:

$$\boldsymbol{\mu}_{(\mathcal{Y})} = \boldsymbol{\Upsilon}_{(\mathcal{Y})} \left(\boldsymbol{\Upsilon}^{\mathrm{T}}_{(\mathcal{Z})} \boldsymbol{\Upsilon}_{(\mathcal{Z})} \right)^{-1} \boldsymbol{\Upsilon}^{\mathrm{T}}_{(\mathcal{Z})} \boldsymbol{\mu}_{(\mathcal{Z})} \tag{20}$$

2.5. k-means clustering

Partitioning data into clusters through a k-means clustering procedure is a popular technique in machine learning and data mining and consists in grouping data points with similar values together in such a way as to minimize the difference between each individual observation and the average value of the cluster it belongs to. If each data point j can be represented by a scalar value x_j, this minimization problem can be written as:

$$(\boldsymbol{\mathcal{C}}, \overline{\boldsymbol{x}}) = \underset{\overline{\boldsymbol{\mathcal{C}}}}{\arg\min} \sum_{i=1}^{k} \sum_{j \in \overline{\mathcal{C}}_i} \left\| x_j - \overline{x}_i \right\|^2 \tag{21}$$

where $\overline{\boldsymbol{x}}$ is a vector of cluster averages and $\boldsymbol{\mathcal{C}}$ is a set of data clusters:

$$\boldsymbol{\mathcal{C}} = \begin{bmatrix} \mathcal{C}_1 & \mathcal{C}_2 & \cdots & \mathcal{C}_k \end{bmatrix} \tag{22}$$

In the context of model-order reduction, clustering integration points or elements together allows for significant complexity reduction if one assumes that response fields (*e.g.* strains [17] or stresses and material history [24]) are uniform within a given cluster. Computational effort is therefore reduced to computing the response of whole clusters instead of that of each individual point or element. Clustering can also be used to divide displacement snapshots into groups and compute separate POD bases for each cluster in order to keep the size of the reduced equilibrium problem (n) small [13].

In this work, clustering is used to divide elements into domains, similar to the approach adopted in [17]. No uniformity assumption within clusters is made but each domain has an independently trained set of reduced integration points $\mathcal{Z}$, as will be discussed in the following sections.

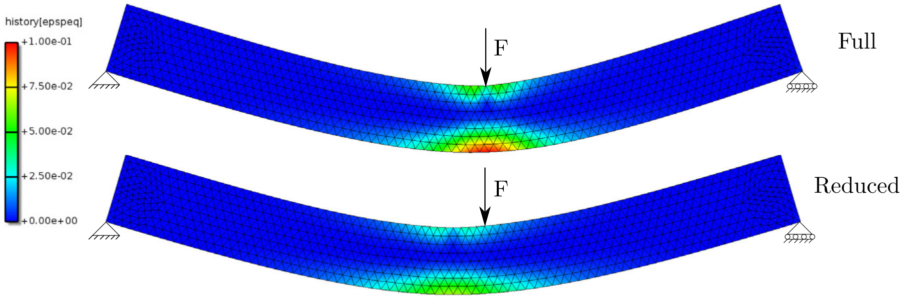

Fig. 1. Erroneous equivalent plastic strain response of a reduced model trained with a slightly different load application point.

3. Adaptive reduction

In this research, the techniques of Section 2 are combined in an adaptive hyper-reduced modeling framework in order to overcome the hurdles associated with the *offline* training of a model with highly nonlinear response. Fig. 1 illustrates the issue of using an insufficiently trained reduced model in *online* predictions of a similar problem (also called a *nearby problem* [19]). The beam model is trained with a load F at midspan and used to predict the response when the load is moved off-center by a distance of 5 % of the total span. In the linear regime, the error between full and reduced solutions (measured as the vertical displacement at the load application point) is limited to a reasonably low level of around 2.5 %. At later steps, however, the deficient reduced basis leads to 50 % lower plastic strain development, leading to an error of more than 12 % in displacements by the end of the analysis.

The framework presented in the following attempts to solve this problem by altogether eliminating the *offline* training process. To this end, the volume Ω is adaptively divided into three distinct regions: the elements responsible for most of the nonlinearity are solved in the full space, the region nearby the fully-solved area is solved with POD with full integration and the remaining regions are further reduced with ECM.

3.1. Partitioned POD

In Fig. 1, plastic strain localizes under the load application point while most of the beam remains elastic. Since the incorrect reproduction of these plastic strains is the cause behind the erroneous reduced model predictions, one potential solution to the issue is to solve regions where strain localizes in the full-order space while still using POD to solve for DOFs in the remaining regions. This local/global approach, first proposed by Kerfriden et al. [19] is briefly summarized in the following.

The complete set of degrees of freedom ($\mathcal{U}$) is divided into full ($\mathcal{F}$) and reduced ($\mathcal{R}$) parts with sizes N_f and N_r, respectively (with $\mathcal{R} \cup \mathcal{F} = \mathcal{U}$ and $N_\mathrm{f} + N_\mathrm{r} = N$). With such partitioning, the solution vector becomes:

$$\mathbf{x} = \begin{pmatrix} \boldsymbol{\alpha} \\ \mathbf{u}_\mathrm{f} \end{pmatrix} \tag{23}$$

where DOFs in $\mathcal{F}$ ($\mathbf{u}_\mathrm{f}$) are directly solved for and the reduced DOFs $\boldsymbol{\alpha}$ now only affect the reconstructed displacement fields for the displacements in $\mathcal{R}$. The reconstruction of $\mathbf{u}_\mathrm{h}$, previously given by Eq. (6), now becomes:

$$\mathbf{u}_{\mathrm{h}(\mathcal{R})} = \boldsymbol{\Phi}_{(\mathcal{R})}\boldsymbol{\alpha} \quad \mathbf{u}_{\mathrm{h}(\mathcal{F})} = \mathbf{u}_\mathrm{f} \tag{24}$$

where it is recalled from Section 1.1 that the selection operator ($\mathcal{R}$) applied to $\boldsymbol{\Phi}$ extracts from it a submatrix of size $N_r \times n$ with rows corresponding to DOFs in $\mathcal{R}$.

Following such partial reduction of the equilibrium equations, the reduced versions of the force vectors can be written as [19]:

$$\mathbf{f}^{\Omega} = \begin{pmatrix} \mathbf{f}_\mathrm{r}^{\Omega} \\ \mathbf{f}_\mathrm{f}^{\Omega} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\Phi}_{(\mathcal{R})}^\mathrm{T}\mathbf{f}_{\mathrm{h}(\mathcal{R})}^{\Omega} \\ \mathbf{f}_{\mathrm{h}(\mathcal{F})}^{\Omega} \end{pmatrix} \quad \mathbf{f}^{\Gamma} = \begin{pmatrix} \mathbf{f}_\mathrm{r}^{\Gamma} \\ \mathbf{f}_\mathrm{f}^{\Gamma} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\Phi}_{(\mathcal{R})}^\mathrm{T}\mathbf{f}_{\mathrm{h}(\mathcal{R})}^{\Gamma} \\ \mathbf{f}_{\mathrm{h}(\mathcal{F})}^{\Gamma} \end{pmatrix} \tag{25}$$

and the partitioned stiffness matrix is given by:

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{\mathrm{rr}} & \mathbf{K}_{\mathrm{rf}} \\ \mathbf{K}_{\mathrm{fr}} & \mathbf{K}_{\mathrm{ff}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}^{\mathrm{T}}_{(\mathcal{R})} \mathbf{K}_{\mathrm{h}(\mathcal{R},\mathcal{R})} \boldsymbol{\Phi}_{(\mathcal{R})} & \boldsymbol{\Phi}^{\mathrm{T}}_{(\mathcal{R})} \mathbf{K}_{\mathrm{h}(\mathcal{R},\mathcal{F})} \\ \mathbf{K}_{\mathrm{h}(\mathcal{F},\mathcal{R})} \boldsymbol{\Phi}_{(\mathcal{R})} & \mathbf{K}_{\mathrm{h}(\mathcal{F},\mathcal{F})} \end{bmatrix} \tag{26}$$

After partitioning, the size of the equilibrium system $\delta \mathbf{x} = -\mathbf{K}^{-1}\mathbf{r}$ increases from n to $n + N_{\mathrm{f}}$. This partially reduced problem remains efficient as long as $N_{\mathrm{f}} \ll N$. It is therefore important to judiciously choose the DOFs in $\mathcal{F}$ in order to cover most of the sources of nonlinearity while maintaining the efficiency of the reduced model.

3.1.1. Choice of fully-solved DOFs

With the definition of the partitioned equilibrium system, it is necessary to devise strategies to define the fully-solved DOF set $\mathcal{F}$. Similar to the approach adopted in [19], $\mathcal{F}$ is updated at the beginning of every time step based on the converged material history of the previous step and remains unchanged until the next step. The first strategy explored here is a simple one: whenever an integration point enters the inelastic regime — *i.e.* when the material starts to plastically deform — the DOFs of the finite element containing the point are added to $\mathcal{F}$. In the examples of Section 4, this strategy is termed *All inelastic*.

Naturally, including every inelastic element in $\mathcal{F}$ leads to an inefficient reduced model if the structure is undergoing plasticity over a large domain. The second strategy for populating $\mathcal{F}$ consists in ranking the integration points by gradient of energy dissipation. For the plasticity model used in this work, this dissipation gradient is computed as:

$$\dot{\varXi} = \boldsymbol{\sigma}\dot{\boldsymbol{\varepsilon}}^{\mathrm{p}} \tag{27}$$

where $\dot{\boldsymbol{\varepsilon}}^{\mathrm{p}}$ is the plastic strain increment over the previous load step. After the sorting process, the $p_{\varXi}N$ points with the highest dissipation rates are included in $\mathcal{F}$. This strategy, termed *Highest dissipation* in the examples of Section 4, is more suitable for distributed plasticity scenarios as it tends to exclude points away from the strain localization zones. The size of $\mathcal{F}$ can be tweaked by adjusting the ratio $0 \leq p_{\varXi} \leq 1$. In contrast to the first strategy, this approach allows DOFs to drop out of $\mathcal{F}$ once regions with a higher gradient arise — *e.g.* when a band with localized plasticity forms.

One last strategy exploits the idea of clustering in order to choose DOFs for $\mathcal{F}$. Instead of sorting integration points from highest to lowest $\dot{\varXi}$, a k-means clustering algorithm can be used to group the points into k clusters by solving a minimization problem similar to Eq. (21):

$$\left(\mathcal{C}, \bar{\dot{\boldsymbol{\Xi}}}\right) = \underset{\overline{\mathcal{C}}}{\arg\min} \sum_{i=1}^{k} \sum_{j \in \overline{\mathcal{C}}_i} \left\| \dot{\varXi}_j - \bar{\dot{\varXi}}_i \right\|^2 \tag{28}$$

and including in $\mathcal{F}$ the points belonging to the cluster with the highest average dissipation rate. The clustering procedure makes the size and composition of $\mathcal{F}$ sensitive to the dispersion in dissipation among the integration points, therefore yielding different results than the *highest dissipation* strategy. This approach is referred to as *Clustered dissipation* in the examples of Section 4.

3.1.2. Dirichlet-type constraints

Dealing with force boundary conditions (Neumann-type) after POD reduction is straightforward: $\mathbf{f}^{\varGamma}_{\mathrm{h}}$ is assembled and its reduced version is obtained by pre-multiplying the POD basis $\boldsymbol{\Phi}$. For displacement boundary conditions (Dirichlet-type) enforced through master/slave constraint equations, a number of additional steps are necessary when transitioning to a POD-reduced model.

The goal is to convert to the reduced space a constraint between a slave DOF u_i and an arbitrary number of master DOFs u_j of the following general form:

$$u_i = r + \sum_j c_j u_j \tag{29}$$

with r being a scalar and $\mathbf{c}$ a coefficient vector. The straightforward approach is to move both u_i and u_j to $\mathcal{F}$ and enforce the constraint as it is. However, the solution can be accelerated by recognizing that some constraints can also be posed directly in the reduced space and others become redundant once reduction takes place. Here, the following checks are performed for each constraint equation before moving any of the involved DOFs to the full space:

- If $r = 0$ and $\mathbf{c} = \mathbf{0}$, the row of $\mathbf{\Phi}$ corresponding to DOF i is checked: if $\mathbf{\Phi}_{(i)} = \mathbf{0}$, the constraint is discarded;
- If $r = 0$ but $\mathbf{c} \neq \mathbf{0}$, check if every basis vector $\boldsymbol{\phi}$ satisfies the constraint: if $\mathbf{\Phi}_l^{\mathrm{T}}\mathbf{c} - \Phi_{(i,l)} = 0$ for every $l \in [1, \ldots, n]$, the constraint is discarded;
- If $\mathbf{c} = \mathbf{0}$ but $r \neq 0$, the constraint is recast in the reduced space involving all reduced DOFs: $\alpha_k = \frac{1}{\Phi_{(i,k)}}\left(r - \sum_{l \neq k}^{n-1} \Phi_{(i,l)}\alpha_l\right)$, where $k = \arg\max_{\overline{k}} \left\| \Phi_{(i,\overline{k})} \right\|$.

If the constraint does not meet any of these requirements, the involved DOFs are moved to $\mathcal{F}$.

3.1.3. Error estimation

Due to nonlinear phenomena occurring in the fully-solved region which gradually change the global structural behavior, the basis $\mathbf{\Phi}$ gradually becomes unable to describe the behavior of the reduced DOFs $\mathcal{R}$. It is important to estimate this loss of accuracy in order to trigger a switch back to a full analysis (*i.e.* move all DOFs to $\mathcal{F}$ for a single step) and include the resulting field $\mathbf{u}$ in $\mathbf{\Phi}$.

One straightforward way to estimate the error is to compute the deviation from global full-order equilibrium with Eq. (1) [9]. As the quality of $\mathbf{\Phi}$ degrades, the equivalence between reduced equilibrium ($\mathbf{f}^{\Omega} = \mathbf{f}^{\Gamma}$) and full-order equilibrium ($\mathbf{f}_{\mathrm{h}}^{\Omega} = \mathbf{f}_{\mathrm{h}}^{\Gamma}$) is gradually lost. By introducing a tolerance parameter, a full step can be triggered whenever this deviation crosses a certain threshold. This requires virtually no additional computational effort to the POD-reduced model, as computing $\mathbf{f}_{\mathrm{h}}^{\Omega}$ and $\mathbf{f}_{\mathrm{h}}^{\Gamma}$ is in any case necessary in order to perform the reductions of Eq. (25).

3.1.4. Augmented conjugate gradient solver

Solving the partitioned system of equations (25) and (26) does not require a specific choice of solver. Most of the examples of Section 4 employ a direct solver based on the LU decomposition of $\mathbf{K}$ with a Skyline-type matrix storage, but an alternative approach that exploits features of classic domain decomposition techniques and was first proposed by Kerfriden et al. [9,19,20] is also explored.

The scheme is based on the idea of master and slave DOFs (also termed border and interior DOFs) used in domain decomposition. The reduced part of the solution ($\boldsymbol{\alpha}$) is removed from the system through condensation and only the DOFs in $\mathcal{F}$ are solved for:

$$\delta\mathbf{u}_{\mathrm{f}} = \mathbf{K}_{\mathrm{s}}^{-1}\mathrm{r}_{\mathrm{s}} \tag{30}$$

where $\mathbf{K}_{\mathrm{s}}$ is the Schur complement of $\mathbf{K}_{\mathrm{rr}}$ and $\mathbf{r}_{\mathrm{s}}$ the residual associated with the condensed problem:

$$\mathbf{K}_{\mathrm{s}} = \mathbf{K}_{\mathrm{ff}} - \mathbf{K}_{\mathrm{fr}}\left(\mathbf{K}_{\mathrm{rr}}\right)^{-1}\mathbf{K}_{\mathrm{rf}} \quad \mathbf{r}_{\mathrm{s}} = \mathbf{r}_{\mathrm{r}} - \mathbf{K}_{\mathrm{fr}}\left(\mathbf{K}_{\mathrm{rr}}\right)^{-1}\mathbf{r}_{\mathrm{r}} \tag{31}$$

with $\mathbf{r}_{\mathrm{r}} = \mathbf{f}_{\mathrm{r}}^{\Omega} - \mathbf{f}_{\mathrm{r}}^{\Gamma}$. The original equilibrium system is therefore converted into the problem of solving only for $\mathbf{u}_{\mathrm{f}}$, after which the reduced DOFs are recovered:

$$\delta\boldsymbol{\alpha} = -\left(\mathbf{K}_{\mathrm{rr}}\right)^{-1}\left(\mathbf{r}_{\mathrm{r}} + \mathbf{K}_{\mathrm{rf}}\delta\mathbf{u}_{\mathrm{f}}\right) \tag{32}$$

The idea put forward in [19] consists in solving Eq. (30) through an augmented conjugate gradient (CG) solver that exploits two of the consequences of partitioning the original DOFs into $\mathcal{F}$ and $\mathcal{R}$. Firstly, supposing that the basis $\mathbf{\Phi}$ correctly describes a problem close to the one being solved (*i.e.* $\mathbf{\Phi}$ can describe at least part of the behavior in $\mathcal{F}$), the reduced solution for $\mathbf{u}_{\mathrm{f}}$ can be used as a first guess $\delta\mathbf{u}_{\mathrm{f}}^{\mathrm{init}}$ to accelerate convergence (*initialization step*):

$$\delta\mathbf{u}_{\mathrm{f}}^{\mathrm{init}} = \mathbf{\Phi}_{(\mathcal{F})}\left(\mathbf{\Phi}_{(\mathcal{F})}^{\mathrm{T}}\mathbf{K}_{\mathrm{s}}\mathbf{\Phi}_{(\mathcal{F})}\right)^{-1}\mathbf{\Phi}_{(\mathcal{F})}^{\mathrm{T}}\mathbf{r}_{\mathrm{s}} \tag{33}$$

Secondly, if $\mathbf{\Phi}$ is not enough to fully describe $\mathbf{u}_{\mathrm{f}}$, whatever information is missing should be sought on a solution space orthogonal to $\mathbf{\Phi}$. This can be achieved by computing a projector $\mathbf{P}$ given by:

$$\mathbf{P} = \mathbf{I} - \mathbf{\Phi}_{(\mathcal{F})}\left(\mathbf{\Phi}_{(\mathcal{F})}^{\mathrm{T}}\mathbf{K}_{\mathrm{s}}\mathbf{\Phi}_{(\mathcal{F})}\right)^{-1}\mathbf{\Phi}_{(\mathcal{F})}^{\mathrm{T}}\mathbf{K}_{\mathrm{s}} \tag{34}$$

and using it to augment the search direction of the conjugate gradient algorithm (*augmentation step*):

$$\mathbf{Z}_{\mathrm{aug}} = \mathbf{P}\mathbf{Z} \tag{35}$$

where $\mathbf{Z}$ is the original search direction obtained by preconditioning the CG residual (using for instance the diagonal or an incomplete LU decomposition of $\mathbf{K}_{\mathrm{s}}$). In this work, these additional solution steps are implemented in a classic CG algorithm whose details are omitted for compactness. The interested reader is referred to [19] for a complete description of the algorithm.

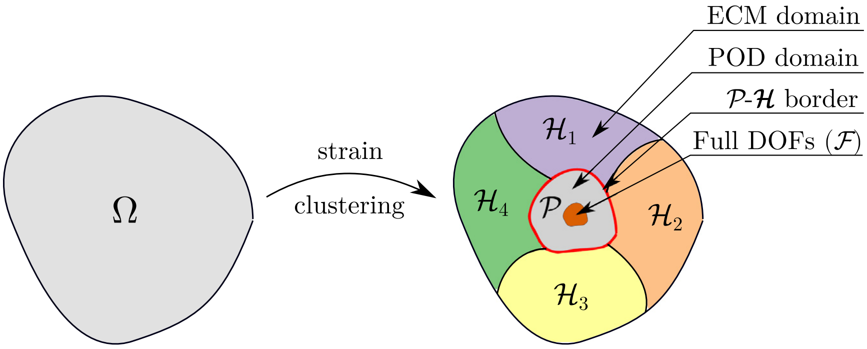

Fig. 2. Domain-based reduction scheme. The mesh is divided into POD and ECM subdomains through a clustering process. A subdomain $\mathcal{F}$ of full DOFs is allowed to exist within $\mathcal{P}$.

3.2. Domain-based hybrid full/POD/ECM reduction

In order to improve acceleration, it is desirable to further reduce the cost of the reduced model by employing a hyper-reduction method such as ECM (Section 2.3). However, the lack of a fully-integrated $\mathbf{f}^{\Omega}$, a consequence of hyper-reduction, leads to a number of incompatibilities with the current adaptive POD framework:

- Populating the fully-solved DOF set $\mathcal{F}$ requires material history to be known at least at a number of candidate points,
- Computing the force term $\mathbf{f}_{\mathrm{f}}^{\Omega}$ associated with DOFs in $\mathcal{F}$ requires all elements around a fully-solved node to be fully integrated,
- The decision to switch to a full step is based on the full-order equilibrium (Section 3.1.3) and thus requires full integration of $\mathbf{f}^{\Omega}$,
- The convergence behavior of full steps may benefit from up-to-date information on material history at all points.

An obvious way to bypass these incompatibilities is to forgo the partitioning strategy of Section 3.1 (*i.e.* have all DOFs in $\mathcal{R}$ at all times) and periodically run a full step for retraining at a predefined interval (*e.g.* every 10 steps). However, this strategy would be inefficient and potentially unstable if the rate of decay in the quality of $\boldsymbol{\Phi}$ varies significantly throughout the analysis — *e.g.* if a cyclic load is applied and plasticity only evolves during reloading branches [24] or if a sharp transition to a steep softening branch is triggered [19]. Such a model would not be truly adaptive.

The solution proposed here is illustrated in Fig. 2 and involves dividing the volume Ω into subdomains in order to allow the use of different levels of reduction on different parts of the mesh. The goal is to use ECM on most of the mesh (a set of subdomains $\boldsymbol{\mathcal{H}}$) while still employing adaptive components on a mesh region $\mathcal{P}$ where $\mathbf{f}^{\Omega}$ is fully integrated. The resultant internal force vector is the combination of a fully-integrated part with multiple hyper-reduced parts:

$$\mathbf{f}^{\Omega} = \boldsymbol{\Phi}^{\mathrm{T}} \left(\sum_{i \in \mathcal{P}}^{M_{\mathcal{P}}} \mathbf{f}_{\mathrm{h}}\left(\mathbf{x}_i\right) w_i + \sum_{j \in \mathcal{Z}_{\mathcal{H}_1}}^{m_1} \mathbf{f}_{\mathrm{h}}\left(\mathbf{x}_j\right) \varpi_j + \cdots + \sum_{k \in \mathcal{Z}_{\mathcal{H}_n}}^{m_n} \mathbf{f}_{\mathrm{h}}\left(\mathbf{x}_k\right) \varpi_k \right) \tag{36}$$

where allowing for an arbitrary number of independently trained ECM domains affords increased control over the total number of ECM points in the mesh and therefore over the integration error and the efficiency of history recovery. It is important to note that the present strategy differs from classic domain decomposition approaches that partition the equilibrium system and separately solve for border and interior DOFs (*cf.* [20]). The decomposition made here is only related to the integration of $\mathbf{f}^{\Omega}$ and $\mathbf{K}$ and maintains a single equilibrium system $\delta\mathbf{x} = -\mathbf{K}^{-1}\mathbf{r}$ and a single POD basis $\boldsymbol{\Phi}$.

The natural choice for $\mathcal{P}$ is the region where most of the sources of nonlinearity are located. Although *a priori* knowledge of the mechanical behavior of the structure being modeled can be used to manually define

$\mathcal{P}$, a fully adaptive approach is preferred because sources of nonlinearity that should be contained in $\mathcal{P}$ might move across the domain during the analysis, *e.g.* a propagating strain localization band. Here, the domains $\mathcal{P}$ and $\boldsymbol{\mathcal{H}}$ are updated after every full step by grouping elements together based on their strain response using k-means clustering:

$$(\boldsymbol{\mathcal{C}}, \bar{\boldsymbol{\varepsilon}}) = \arg\min_{\overline{\boldsymbol{\mathcal{C}}}} \sum_{i=1}^{k} \sum_{j\in\overline{\mathcal{C}}_i} \left\| \boldsymbol{\varepsilon}_j - \bar{\boldsymbol{\varepsilon}}_i \right\|^2 \tag{37}$$

The most straightforward clustering strategy is to use $k = 2$ and assign the cluster with the highest average strain to $\mathcal{P}$. For the extension to multiple ECM domains, a stepwise approach is taken by using the original clustering algorithm of [25] two consecutive times: elements are first divided into two clusters and elements in the cluster with lower average strain are further clustered into k subclusters that define the k ECM domains $\boldsymbol{\mathcal{H}}$. This effectively guarantees that the size of $\mathcal{P}$ is objective with respect to the number of ECM domains. The steps of the clustering process are also shown in Algorithm 2. It is important to mention that since constant strain triangles with a single integration point are used in this work, clustering elements is equivalent to clustering integration points. A different strategy of converting point clusters to element clusters might be required if higher-order integration is employed.

Upon defining $\mathcal{P}$ and $\boldsymbol{\mathcal{H}}$, the node set that defines the border between the POD domain and any of the ECM domains (the $\mathcal{P}$-$\boldsymbol{\mathcal{H}}$ border depicted in Fig. 2) is also stored. Since nodes at this border might not be fully integrated, care is taken as to not include them in $\mathcal{F}$ or use them to estimate the reduced solution error (Section 3.1.3).

3.2.1. Analysis flow and error control

The analysis starts with one or more fully-solved steps and no *a priori* information on structural behavior — *i.e.* no *offline* training is required. The main steps of this phase are shown in Algorithm 1. In the examples of Section 4, only one full step is run before POD reduction ($n_{\text{full}} = 1$), but the framework is flexible in allowing for more full steps and consequently to a larger set of displacement and strain snapshots.

```
update DOF sets: $\mathcal{F} \leftarrow \mathcal{U}$, $\mathcal{R} \leftarrow \varnothing$;
clear snapshot matrices: $\mathbf{X}_\varepsilon = \mathbf{0}$, $\mathbf{X}_\sigma = \mathbf{0}$, $\mathbf{X}_\mathrm{h} = \mathbf{0}$;
while $\frac{\left\|\mathbf{f}_\mathrm{h}^\Gamma - \mathbf{f}_\mathrm{h}^\Omega\right\|}{\lambda} < \epsilon_\text{solver}$ :
    compute $\mathbf{f}_\mathrm{h}^\Omega(\mathbf{u}_\mathrm{h})$ and $\mathbf{K}_\mathrm{h}(\mathbf{u}_\mathrm{h})$;
    use a solver to compute $\delta\mathbf{u}_\mathrm{h} = \mathbf{K}_\mathrm{h}^{-1}\left(\mathbf{f}_\mathrm{h}^\Gamma - \mathbf{f}_\mathrm{h}^\Omega\right)$;
    update $\mathbf{u}_\mathrm{h} \leftarrow \mathbf{u}_\mathrm{h} + \delta\mathbf{u}_\mathrm{h}$;
set $\mathbf{X}_\mathrm{u}^\mathrm{f} \leftarrow \left[\mathbf{X}_\mathrm{u}^\mathrm{f} \quad \mathbf{u}_\mathrm{h}\right]$, $\mathbf{X}_\varepsilon \leftarrow [\mathbf{X}_\varepsilon \quad \boldsymbol{\varepsilon}]$;
set $i_\text{full} \leftarrow i_\text{full} + 1$;
if $i_\text{full} = n_\text{full}$ :
    compute the SVD of $\mathbf{X}_\mathrm{u}$ and update $\boldsymbol{\Phi}$;
    update strain clusters and domains (Algorithm 2);
    switch to reduced steps (Algorithm 3);
commit material history;
```

Algorithm 1: Analysis flow of a fully-solved step.

Snapshots for POD training are taken both after a full step ($\mathbf{X}_\mathrm{u}^\mathrm{f}$) as well as after a POD step with $\mathcal{F} \neq \varnothing$ ($\mathbf{X}_\mathrm{u}^\mathrm{a}$) and together form the complete snapshot matrix:

$$\mathbf{X}_\mathrm{u} = \left[\mathbf{X}_\mathrm{u}^\mathrm{f} \quad \mathbf{X}_\mathrm{u}^\mathrm{a}\right] \tag{38}$$

from which $\boldsymbol{\Phi}$ is extracted by truncating the SVD operation after the singular values become lower than an input tolerance ϵ_SV. As the analysis progresses and the number of snapshots reaches preset values n_f and n_a, the snapshot matrices are gradually renewed by discarding the oldest values whenever new snapshots are added. In order to extract as much information as possible from a single retraining step, both $\mathbf{u}_\mathrm{h}$ and its change from the previous time step $\Delta\mathbf{u}_\mathrm{h}$ are included in $\mathbf{X}_\mathrm{u}$ if $n_\text{full} = 1$.

```
Input: Matrix with integration point strains $\mathbf{X}_\varepsilon$, number of ECM domains $k$
Output: One POD domain $\mathcal{P}$ and a set of ECM domains $\boldsymbol{\mathcal{H}}$
if $\mathcal{P} \neq \varnothing$ and $\boldsymbol{\mathcal{H}} \neq \varnothing$ :
    initialize centroids: $\overline{\boldsymbol{\varepsilon}}_1 \leftarrow \frac{1}{n_{\mathcal{P}}} \sum_p \mathbf{X}^p_{\varepsilon(\mathcal{P})}$, $\overline{\boldsymbol{\varepsilon}}_2 \leftarrow \frac{1}{M - n_{\mathcal{P}}} \sum_{c=1}^{k} \sum_p \mathbf{X}^p_{\varepsilon(\mathcal{H}_c)}$;
else
    choose two points at random to initialize $\overline{\boldsymbol{\varepsilon}}$;
use clustering [25] to divide points into two clusters: $(\boldsymbol{\mathcal{C}}, \overline{\boldsymbol{\varepsilon}}) \leftarrow \mathrm{kmc}(\mathbf{X}_\varepsilon, 2)$;
set $\mathcal{C}_1 \leftarrow \mathcal{C}_{\overline{c}}$, with $\overline{c} = \arg\max_{c} \|\overline{\boldsymbol{\varepsilon}}_c\|$ and set $\mathcal{P} \leftarrow \mathcal{C}_1$;
if $\mathcal{P} \neq \varnothing$ and $\boldsymbol{\mathcal{H}} \neq \varnothing$ :
    initialize centroids: $\overline{\boldsymbol{\varepsilon}}_c \leftarrow \frac{1}{n_{\mathcal{H}_c}} \sum_p \mathbf{X}^p_{\varepsilon(\mathcal{H}_c)}$;
else
    choose $k$ points in $\mathcal{C}_2$ at random to initialize $\overline{\boldsymbol{\varepsilon}}$;
divide points in $\mathcal{C}_2$ into $k$ subclusters: $(\boldsymbol{\mathcal{H}}, \overline{\boldsymbol{\varepsilon}}) \leftarrow \mathrm{kmc}(\mathbf{X}_{\varepsilon(\mathcal{C}_2)}, k)$;
```

Algorithm 2: Mesh domain update procedure with k-means clustering (kmc).

Strain snapshots are used to define the domains $\mathcal{P}$ and $\boldsymbol{\mathcal{H}}$ through the procedure shown in Algorithm 2. If the current full analysis phase is not the first one (if the model was previously reduced but switched back to a full step), the clustering algorithm is pre-initialized with the latest cluster topology in order to accelerate retraining.

After updating cluster topology and training POD, the analysis switches to a reduced step (Algorithm 3). At the beginning of each step, $\mathcal{F}$ is populated with DOFs from $\mathcal{P}$, redundant constraints are removed and the reduced DOFs $\boldsymbol{\alpha}$ are reconstructed through a least-squares procedure in case $\boldsymbol{\Phi}$ is different from the one from the previous step. At first, all domains are fully integrated and stresses and material history at every integration point are collected (*POD step*). After a certain number n_{pod} of POD steps ($n_{\text{pod}} = 1$ in the examples of Section 4), these snapshots are used to train ECM and update the history reconstruction basis $\boldsymbol{\Upsilon}$ for each domain in $\boldsymbol{\mathcal{H}}$. If a previous set of ECM points is available, a fast retraining that keeps the set $\mathcal{Z}$ intact and only updates the weights $\boldsymbol{\varpi}$ is attempted (Algorithm 4). After this training, the model switches to a hybrid scheme in which $\mathcal{P}$ is fully integrated and the domains $\boldsymbol{\mathcal{H}}$ are integrated with ECM.

At the end of each time step, the quality of the reduced solution is measured by computing the deviation from full-order equilibrium. Since most of the mesh is not fully integrated, only DOFs in the POD domain can be used for this purpose. Defining $\mathbf{f}_{\text{p}}^{\Omega}$ and $\mathbf{f}_{\text{p}}^{\Gamma}$ as the force vectors associated with the DOFs in $\mathcal{P}$ (excluding the $\mathcal{P}$-$\boldsymbol{\mathcal{H}}$ border), the following condition is checked:

$$\frac{\left\|\mathbf{f}_{\text{p}}^{\Gamma} - \mathbf{f}_{\text{p}}^{\Omega}\right\|}{\lambda} < \epsilon_{\text{force}} \tag{39}$$

where λ is the load scale factor of Eq. (5) and ϵ_{force} is an input tolerance. If the condition is not satisfied, the current displacement increment is discarded, material history is recovered and the analysis switches back to a fully-solved phase (Algorithm 1).

An additional check becomes necessary if the size of domain $\mathcal{F}$ approaches that of $\mathcal{P}$. Whenever a DOF is included in $\mathcal{F}$, its full-order equilibrium is satisfied exactly since it is explicitly included in the equilibrium problem. If $\mathcal{F}$ occupies most or all of $\mathcal{P}$, Eq. (39) is always satisfied even if the quality of $\boldsymbol{\Phi}$ is decaying. It is therefore complemented by a second condition that compares the values of $\mathbf{u}_{\text{f}}$ obtained directly from the equilibrium problem with their POD approximation:

$$\frac{\left\|\mathbf{u}_{\text{f}} - \boldsymbol{\Phi}_{(\mathcal{F})}\boldsymbol{\alpha}\right\|}{\|\mathbf{u}_{\text{f}}\|} < \epsilon_{\text{disp}} \tag{40}$$

where ϵ_{disp} is an additional tolerance factor. A full step is triggered when either of the conditions is violated.

```
update fully-solved DOF set $\mathcal{F}$ (Section 3.1.1) and set $\mathcal{R} \leftarrow \mathcal{U} \setminus \mathcal{F}$;
convert Dirichlet-type constraints (Section 3.1.2);
if $\mathbf{\Phi} \neq \mathbf{\Phi}^{\text{o}}$ :
    recompute $\boldsymbol{\alpha} \leftarrow \left(\mathbf{\Phi}^{\text{T}}\mathbf{\Phi}\right)^{-1} \mathbf{\Phi}^{\text{T}}\mathbf{u}_{\text{h}}^{\text{o}}$
while $\frac{\left\|\mathbf{f}^{\Gamma}-\mathbf{f}^{\Omega}\right\|}{\lambda} < \epsilon_{\text{solver}}$ :
    reconstruct $\mathbf{u}_{\text{h}}$ from $\mathbf{x}$: $\mathbf{u}_{\text{h}(\mathcal{R})} \leftarrow \mathbf{\Phi}_{(\mathcal{R})}\boldsymbol{\alpha} \quad \mathbf{u}_{\text{h}(\mathcal{F})} \leftarrow \mathbf{u}_{\text{f}}$;
    for every domain $d$ :
        compute $\mathbf{f}_d^{\Omega}(\mathbf{u}_{\text{h}})$ and $\mathbf{K}_d(\mathbf{u}_{\text{h}})$;
        add domain contributions: $\mathbf{f}^{\Omega} \leftarrow \mathbf{f}^{\Omega} + \mathbf{f}_d^{\Omega}$, $\mathbf{K} \leftarrow \mathbf{K} + \mathbf{K}_d$;
    use a solver to compute $\delta\mathbf{x} = \mathbf{K}^{-1}\left(\mathbf{f}^{\Gamma} - \mathbf{f}^{\Omega}\right)$;
    update $\mathbf{x} \leftarrow \mathbf{x} + \delta\mathbf{x}$;
if $\frac{\left\|\mathbf{f}_{\text{p}}^{\Gamma}-\mathbf{f}_{\text{p}}^{\Omega}\right\|}{\lambda} > \epsilon_{\text{force}}$ or $\frac{\left\|\mathbf{u}_{\text{f}}-\mathbf{\Phi}_{(\mathcal{F})}\boldsymbol{\alpha}\right\|}{\left\|\mathbf{u}_{\text{f}}\right\|} > \epsilon_{\text{disp}}$ :
    set $\mathbf{x} \leftarrow \mathbf{x}^{\text{o}}$, $i_{\text{full}} \leftarrow 0$, $i_{\text{pod}} \leftarrow 0$;
    if step is hyper-reduced :
        reconstruct history: $\boldsymbol{\mu}_{(\mathcal{Y})} \leftarrow \mathbf{\Upsilon}_{(\mathcal{Y})}\left(\mathbf{\Upsilon}_{(\mathcal{Z})}^{\text{T}}\mathbf{\Upsilon}_{(\mathcal{Z})}\right)^{-1}\mathbf{\Upsilon}_{(\mathcal{Z})}^{\text{T}}\boldsymbol{\mu}_{(\mathcal{Z})}^{\text{o}}$;
    restart with a full step (Algorithm 1);
if step is not hyper-reduced :
    set $\mathbf{X}_{\sigma} \leftarrow [\mathbf{X}_{\sigma} \quad \boldsymbol{\sigma}]$, $\mathbf{X}_{\mu} \leftarrow \left[\mathbf{X}_{\mu} \quad \boldsymbol{\mu}\right]$;
    if $\mathcal{F} \neq \varnothing$ :
        set $\mathbf{X}_{\text{u}}^{\text{a}} \leftarrow \left[\mathbf{X}_{\text{u}}^{\text{a}} \quad \mathbf{u}_{\text{h}}\right]$;
        compute the SVD of $\mathbf{X}_{\text{u}}$ and update $\mathbf{\Phi}$;
    set $i_{\text{pod}} \leftarrow i_{\text{pod}} + 1$;
    if $i_{\text{pod}} = n_{\text{pod}}$ :
        for every ECM domain :
            update ECM integration (Algorithm 4);
            compute the SVD of $\mathbf{X}_{\mu}$ and update $\mathbf{\Upsilon}$;
        switch to hyper-reduced steps;
commit material history;
```

Algorithm 3: Analysis flow of a (hyper-)reduced step. The superscript "o" indicates converged values from the previous time step.

4. Numerical examples

The framework of Section 3 has been implemented in an in-house Finite Element software using the open-source Jem/Jive C++ numerical analysis library [26]. The examples of this section were executed on a single core of a Core i7-7500U processor on a laptop with 8 GB RAM running Ubuntu 16.04.3.

The simple problem of a bar with a circular cutout loaded in plane strain tension is considered (Fig. 3). As the bar is loaded, stress concentrations around the cutout trigger the development of plastic strain. By varying the cutout radius r, the extent of the zone undergoing plastic deformations can be adjusted. This allows for assessing the performance of the reduction strategy for both localized and distributed material nonlinearity.

An inviscid pressure-dependent plasticity model with non-associative plastic flow is employed [21,27]. The material properties are $E = 2500\,\text{MPa}$, $\nu = 0.37$ and $\nu_{\text{p}} = 0.32$ (plastic Poisson ratio) and the yield surfaces in tension and compression are given by:

$$\sigma_{\text{t}} = 64.80 - 33.6\text{e}^{-\epsilon_{\text{eq}}^{\text{p}}/0.003407} - 10.21\text{e}^{-\epsilon_{\text{eq}}^{\text{p}}/0.06493} \tag{41}$$

$$\sigma_{\text{c}} = 81.0 - 42.0\text{e}^{-\epsilon_{\text{eq}}^{\text{p}}/0.003407} - 12.77\text{e}^{-\epsilon_{\text{eq}}^{\text{p}}/0.06493} \tag{42}$$

```
Input: Set of domain int. points $\mathcal{H}$ with weights $\mathbf{w}$ and stress snapshots $\mathbf{X}_\sigma$
Output: Reduced integration point set $\mathcal{Z}$ with modified weights $\boldsymbol{\varpi}$
compute the SVD of the snapshot matrix: $\mathbf{X}_\sigma = \boldsymbol{\Psi}\mathbf{S}\mathbf{V}^{\mathrm{T}}$;
set $\boldsymbol{\Lambda}^i_j \leftarrow \sqrt{w_i}\left(\mathbf{f}^i_j - \frac{1}{\Omega}\mathbf{f}^\Omega_j\right)$ with $\mathbf{f}^i_j = \boldsymbol{\Phi}^{\mathrm{T}}_i\mathbf{B}^{\mathrm{T}}_i S_j \boldsymbol{\Psi}_j$ for $i \in \mathcal{H}$, $j \in [1, ..., n]$;
set $\mathbf{J} \leftarrow \begin{bmatrix}\boldsymbol{\Lambda} & \sqrt{\mathbf{w}}\end{bmatrix}^{\mathrm{T}}$, $\mathbf{b} \leftarrow \begin{bmatrix}\mathbf{0} & \Omega\end{bmatrix}^{\mathrm{T}}$;
if $\mathcal{H} = \mathcal{H}^{\mathrm{o}}$ :
    adjust weights without changing the points: $\boldsymbol{\beta}_{(\mathcal{Z})} \leftarrow \left(\mathbf{J}^{\mathrm{T}}_{(\mathcal{Z})}\mathbf{J}_{(\mathcal{Z})}\right)^{-1}\mathbf{J}^{\mathrm{T}}_{(\mathcal{Z})}\mathbf{b}$;
    update residual: $\mathbf{r} \leftarrow \mathbf{b} - \mathbf{J}_{(\mathcal{Z})}\boldsymbol{\beta}_{(\mathcal{Z})}$;
    if $\frac{\|\mathbf{r}\|}{\|\mathbf{b}\|} < \epsilon_{\mathrm{greedy}}$ :
        compute weights $\varpi_i \leftarrow \sqrt{w_i}\,\beta_{(\mathcal{Z})i}$;
        return (skip the rest of this box);
set $\mathcal{Z} \leftarrow \varnothing$;
use the greedy algorithm of [23,24]: $(\mathcal{Z}, \boldsymbol{\varpi}) \leftarrow \mathrm{greedy}\,(\mathbf{J}, \mathbf{b})$;
```

Algorithm 4: ECM update procedure. The superscript "o" indicates entities from the most recent previous ECM retraining.

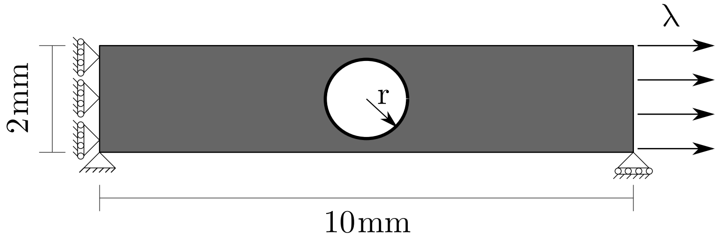

Fig. 3. Numerical example of a bar with a circular cutout loaded in tension.

where $\varepsilon^{\mathrm{p}}_{\mathrm{eq}}$ is the equivalent plastic strain. These yield surfaces promote an initial plastic hardening followed by a plateau with an approximately perfectly plastic response.

In all of the examples that follow, $n_{\mathrm{f}} = 4$ and $n_{\mathrm{a}} = 2$. Further increasing the size of the snapshot matrix has not been found to affect the performance of the framework for this specific problem. Solver precision and SVD truncation tolerance are both set as $\epsilon_{\mathrm{solver}} = \epsilon_{\mathrm{SV}} = 1 \times 10^{-6}$. The bar is meshed using constant strain triangles with one integration point per element. Unless otherwise specified, an element length of $l_{\mathrm{e}} = 0.05\,\mathrm{mm}$ is used ($\sqrt{n_{\mathrm{DOF}}} \approx 150$), which was found through a mesh convergence study to be an adequate level of discretization for the problem at hand.

The model is loaded at its right edge (Fig. 3). An arc-length constraint is used to adjust the load factor λ in order to make the right edge of the model displace by a prescribed value $\overline{u}$. A point load is applied to the horizontal DOF i of the node at the bottom-right corner of the structure and the remaining horizontal DOFs of the edge are tied to this corner node. The enforced arc-length constraint can be written in the full and reduced-order spaces as:

$$a_{\mathrm{h}} = u_{(\mathrm{i})} - \overline{u} = 0 \quad \text{or} \quad a = \boldsymbol{\Phi}_{(\mathrm{i})}\boldsymbol{\alpha} - \overline{u} = 0 \tag{43}$$

and its derivatives are given by:

$$\frac{\partial a_{\mathrm{h}}}{\partial \mathbf{u}} = \mathbf{I}_{(\mathrm{i})} \quad \frac{\partial a_{\mathrm{h}}}{\lambda} = 0 \quad \text{or} \quad \frac{\partial a}{\partial \boldsymbol{\alpha}} = \boldsymbol{\Phi}_{(\mathrm{i})} \quad \frac{\partial a}{\partial \lambda} = 0 \tag{44}$$

where $\mathbf{I} \in \mathbb{R}^N$ is the identity matrix. This approach is equivalent to loading the structure in displacement control but yields a value for λ at every time step that is not computed from $\mathbf{f}^\Omega_{\mathrm{h}}$. This becomes necessary in order to obtain the force–displacement response of a hyper-reduced model for which internal force contributions are generally undefined due to the reduced cubature scheme.

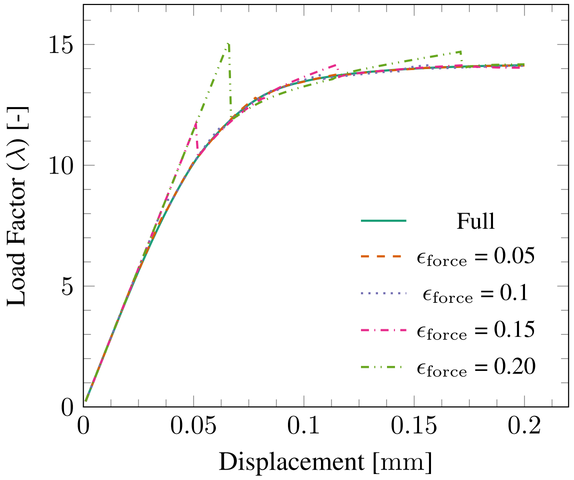

Fig. 4. Adaptive POD response without DOF partitioning for multiple values of ϵ_{force} ($r = 0.9\,\text{mm}$).

4.1. Adaptive POD

In this section, the adaptive POD strategy of Section 3.1 is demonstrated with a series of numerical examples. In order to investigate the POD reduction scheme in isolation, ECM is deactivated and the domain $\mathcal{P}$ is extended to cover the whole structure. Without ECM, the retraining condition related to ϵ_{disp} is kept deactivated and retraining can only be triggered by the full-order equilibrium condition of Eq. (39).

The first two examples aim to demonstrate the ability of POD to sense when its reduced basis becomes inadequate and trigger a retraining step. For the first example, the cutout radius is fixed at $r = 0.9\,\text{mm}$ and the model is executed for multiple values of ϵ_{force}. Results are shown in Fig. 4. Using the basis obtained during the first step, the POD model shows an approximately linear response until the force error exceeds ϵ_{force}. Upon triggering a new full step, the reduced curve immediately snaps back to the reference full-order response. As expected, reducing the threshold ϵ_{force} leads to a smoother response that agrees very well with the full curve, but at the cost of an increased number of full steps (up to 10 correction steps out of a total of 200 steps).

A similar behavior is observed for a bar with $r = 0.5\,\text{mm}$, as shown in Fig. 5. The adaptive strategy is therefore shown to also be effective in correcting the POD basis for models with distributed sources of nonlinearity. However, it is important to note that even though the error control condition of Eq. (39) is adimensionalized by λ, the choice of ϵ_{force} is still problem dependent: the present model requires lower values of ϵ_{force} to trigger a given number of correction steps when compared to the model with $r = 0.9\,\text{mm}$.

It should also be mentioned that even though the plasticity model employed here is robust enough to withstand the sharp correction jumps observed for cases with high ϵ_{force}, that might not be the case for constitutive models already notorious for suffering from convergence issues (*e.g.* continuum damage). Employing less robust models may therefore require a different strategy for transitioning from an aborted reduced step to a full one, for instance a combination of a higher number of full steps (n_{full}) and a path-following method that carefully leads the structure back to the full curve, such as an arc-length that controls energy dissipation [28,29]. Depending on the material model and on the structure being modeled, too high values for ϵ_{force} may also lead to spurious damage activation or erroneous plastic strain distributions even after a converged retraining step. Although this is not observed in the examples treated here, lowering ϵ_{force} would alleviate the issue.

4.1.1. Choice of $\mathcal{F}$

The preceding examples did not take advantage of the system partitioning of Eq. (23). By directly solving for DOFs in highly nonlinear regions, the reduced solution should, in theory, become less sensitive to a decay in the quality of $\boldsymbol{\Phi}$, since in this case the basis would only be used to approximate the solution at regions away from the nonlinear zones. This would in turn be reflected in the number of full-order correction steps. Four strategies for defining $\mathcal{F}$ are explored here: all DOFs on elements deforming inelastically, elements with the highest average

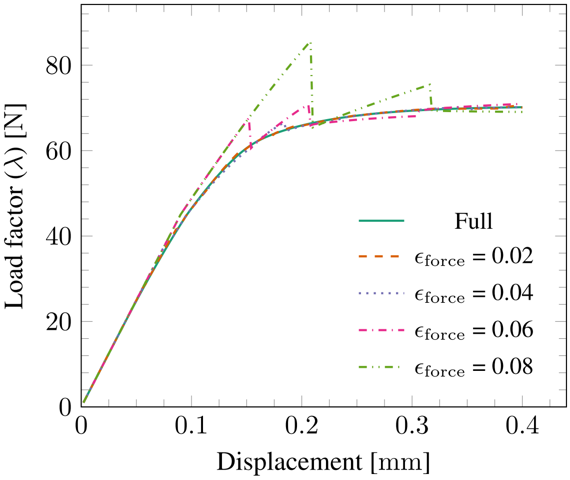

Fig. 5. Adaptive POD response without DOF partitioning for multiple values of ϵ_{force} ($r = 0.5\,\text{mm}$).

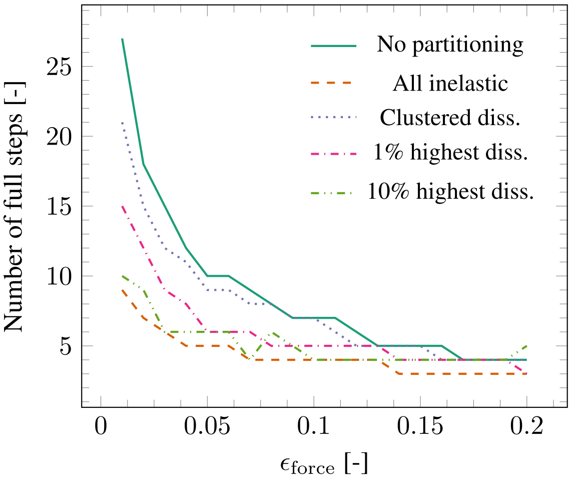

Fig. 6. Number of correction steps versus ϵ_{force} for different partitioning strategies ($r = 0.9\,\text{mm}$).

dissipation rate after clustering with $k = 2$ and the 1 % and 10 % of elements with the highest energy dissipation rate. Models with $r = 0.9\,\text{mm}$ are executed for multiple values of ϵ_{force} for each of the strategies and changes in the number of full steps are observed. Results are shown in Fig. 6.

As expected, the number of correction steps is higher if no partitioning strategy is adopted, with the highest differences observed for lower values of ϵ_{force}. For a given level of tolerance, the number of full steps tends to be inversely proportional to the size of $\mathcal{F}$: the more DOFs are fully solved, the smaller the number of correction steps. For dissipation-based strategies, since $\mathcal{F}$ is allowed to change at every time step, an additional layer of complexity is added since the resulting behavior not only depends on the number of fully-solved DOFs but also on the way with which nodes periodically enter and leave $\mathcal{F}$. Fig. 7 shows which DOFs are included in $\mathcal{F}$ during the last time step for three of the strategies.

Fig. 8 shows the average load factor error between full and reduced responses obtained during the 200 analysis steps for different tolerances and partitioning strategies. Although changes in error with ϵ_{force} are slightly different depending on how $\mathcal{F}$ is defined, all methods show comparable accuracy. It is therefore concluded that even though the frequency of fully-solved steps is significantly lower when a partitioning strategy is applied (Fig. 6) and the quality of $\mathbf{\Phi}$ is allowed to degrade for several time steps, solving the most critical mesh regions in the full space effectively keeps the accuracy of the reduced model at acceptable levels.

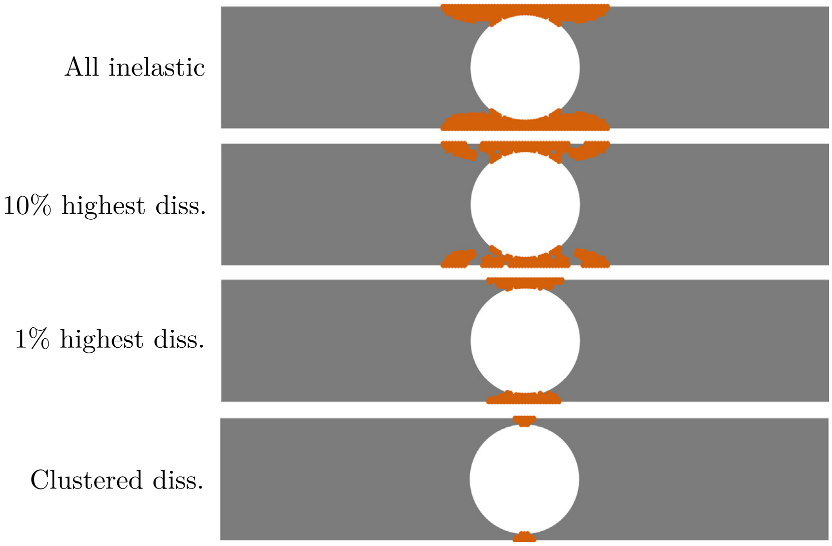

Fig. 7. Fully-solved DOF set $\mathcal{F}$ at the last analysis step ($r = 0.9$ mm).

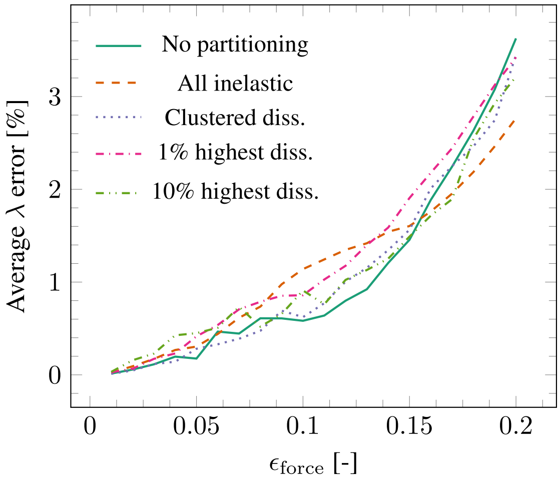

Fig. 8. Average reduced model error for different values of ϵ_{force} and partitioning strategies ($r = 0.9$ mm).

Changes in the composition of $\mathcal{F}$ with time are also investigated. By plotting the number of nodes in $\mathcal{F}$ at each load step for the different partitioning approaches (Fig. 9), it can be seen that each strategy tends to form domains of different sizes, as suggested by Fig. 7. Even though the clustered dissipation strategy does not explicitly impose a maximum size for $\mathcal{F}$, the clustering process opts for small cluster sizes throughout the analysis. Sudden changes in $\mathcal{F}$ observed for the 10 % *highest diss.* strategy occur when the model switches to a full step towards the end of the curve, when strain localizes in a small patch of elements. Since the global behavior is constrained by an increasingly outdated $\boldsymbol{\Phi}$, the ability of the model to form this plastic mechanism is hindered, causing elements away from the localization zone to dissipate energy. When a full step is triggered, $\boldsymbol{\Phi}$ is updated and these elements stop dissipating energy for a number of time steps, causing them to leave $\mathcal{F}$.

In order to assess the performance of the partitioning strategies for the case of distributed plasticity, the same tests are repeated for a bar with $r = 0.5$ mm. The number of correction steps for different values of ϵ_{force} and partitioning strategies is shown in Fig. 10. Although a reduction in the number of full steps is still observed in this case, it is much less pronounced when dissipation-based techniques are used. Since a smaller cutout leads to plastic

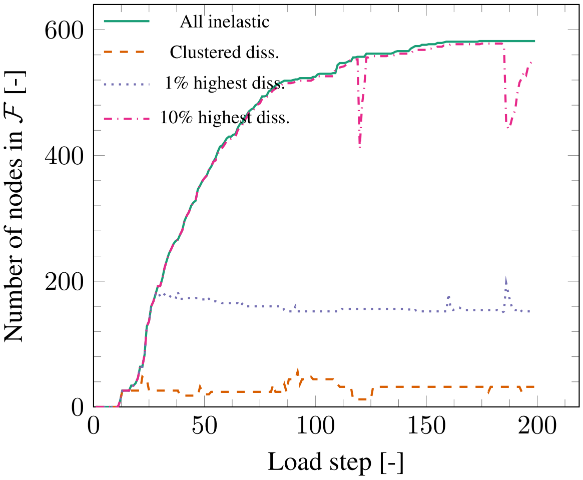

Fig. 9. Evolution of $\mathcal{F}$ with load steps for different partitioning techniques ($r = 0.9$ mm, $\epsilon_{\text{force}} = 0.01$).

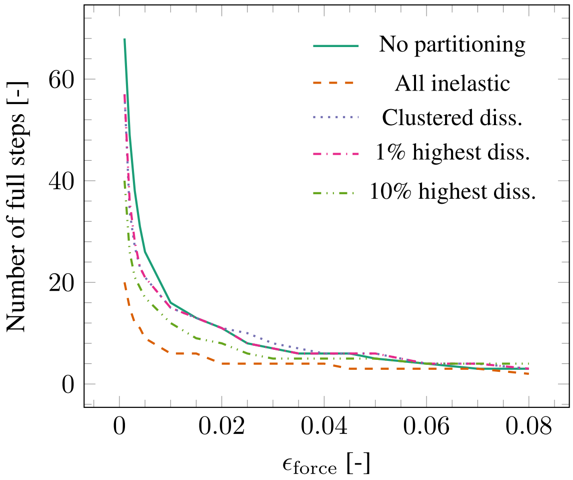

Fig. 10. Number of correction steps versus ϵ_{force} for different partitioning strategies ($r = 0.5$ mm).

strain development everywhere in the model, including a limited number of DOFs in $\mathcal{F}$ is not as beneficial as it is for models with a localized nonlinearity source.

In contrast with dissipation-based strategies, including all inelastic elements in $\mathcal{F}$ remains effective in reducing the number of full steps, but at the cost of efficiency: as can be seen in Fig. 11, almost every DOF in the mesh is included in $\mathcal{F}$ by the end of the analysis. Fig. 12 shows that while partitioning does not help to reduce the number of full steps in this case, it does improve the overall accuracy of the solution.

In such cases, it would be more advantageous to opt for a fully-reduced solution if $0 \leq \epsilon_{\text{force}} \leq 0.05$ or use a dissipation-based approach if $\epsilon_{\text{force}} > 0.05$, yielding a smaller average error (Fig. 12) while keeping the number of correction steps unchanged.

4.1.2. Performance of the augmented CG solver

The preceding tests were solved with a direct LU solver. It is also interesting to investigate the possibility of employing the augmented CG solver of Section 3.1.4 in solving the partitioned POD equilibrium problem. The tensioned bar problem with $r = 0.9$ mm and $\epsilon_{\text{force}} = 0.04$ is solved both with a conventional CG solver and with a modified version that includes the initialization and augmentation steps of Eqs. (33) and (35). The *all inelastic* partitioning strategy is chosen for this part of the investigation. Up to 2000 CG iterations are allowed per global

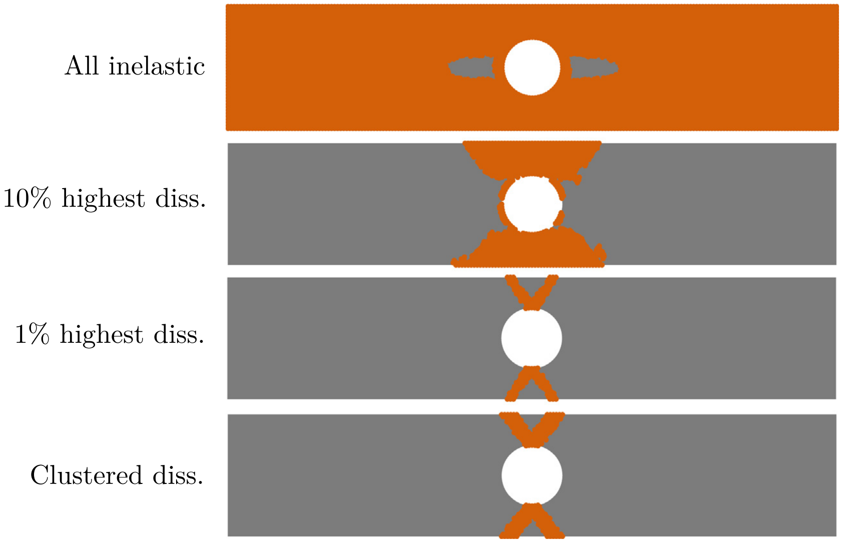

Fig. 11. Fully-solved DOF set $\mathcal{F}$ at the last analysis step ($r = 0.5\,\text{mm}$).

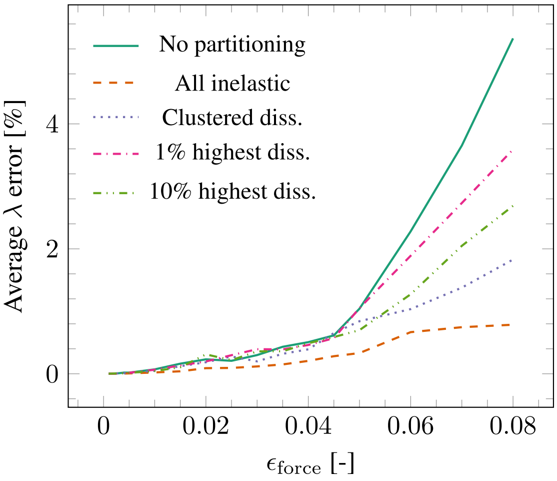

Fig. 12. Average reduced model error for different values of ϵ_{force} and partitioning strategies ($r = 0.5\,\text{mm}$).

Newton–Raphson iteration and a precision $\epsilon_{\text{CG}} = 5 \times 10^{-4}$ is adopted. A direct LU solver is used to recover the reduced DOFs as per Eq. (32).

The total number of CG iterations necessary for convergence (including all global Newton–Raphson iterations) is recorded for each load step. In the first example, the diagonal of $\mathbf{K}_{\text{s}}$ is used as preconditioner, following the tests of Kerfriden et al. [19] for lattice structures with damage. Results are shown in Fig. 13. The modifications incorporated in the solver are effective in accelerating convergence, with a dramatic reduction in the number of iterations. Furthermore, the increase in the number of iterations as the analysis progresses and the plastic localization mechanism is formed are much less pronounced for the modified solver.

The same test is repeated using the incomplete LU decomposition (*iLUd*) of $\mathbf{K}_{\text{s}}$ as preconditioner. The number of iterations for convergence at each time step can be seen in Fig. 14. Similar to the previous case, the initialization and $\mathbf{\Phi}$-augmentation have a beneficial effect on the convergence behavior, with a reduction of up to 23 times in the number of CG iterations for convergence. Although these results are encouraging, the additional computations associated with using an iterative solver — assembling $\mathbf{K}_{\text{s}}$ and computing $\delta\mathbf{u}_{\text{f}}^{\text{init}}$ and $\mathbf{P}$ of Eqs. (33) and (34) every

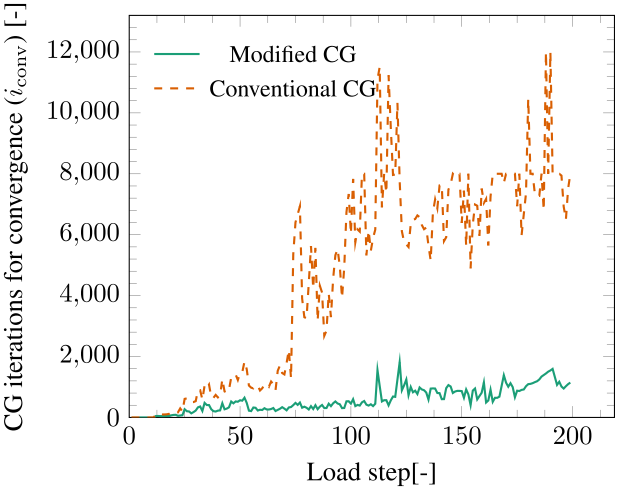

Fig. 13. Number of CG iterations necessary for global convergence of each load step (diagonal preconditioner).

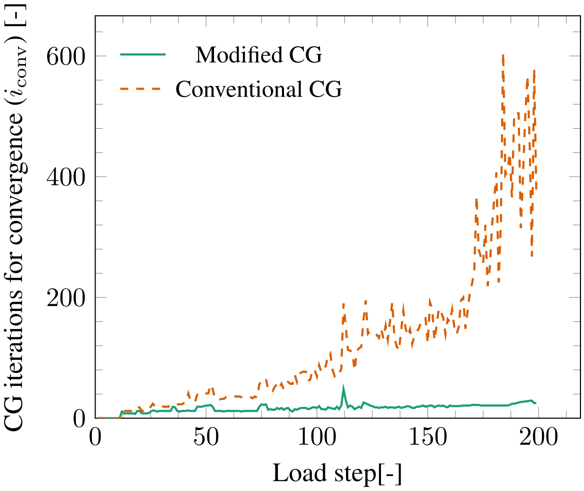

Fig. 14. Number of CG iterations necessary for global convergence of each load step (iLUd preconditioner).

time the stiffness matrix changes — might make direct solvers the more efficient alternative depending on solver implementation (*e.g.* sequential versus parallel) and on the problem being solved.

4.1.3. Acceleration

This section focuses on the level of acceleration provided by the adaptive POD reduction scheme. The bar example with $r = 0.9$ mm is executed with different levels of mesh discretization and using different partitioning techniques. The acceleration associated with the modified CG solver is also investigated. A tolerance $\epsilon_{\text{force}} = 0.04$, which guarantees an average error lower than 0.4 % with respect to the full solution (Fig. 8), is used in all cases. The speed-ups obtained from the average of 3 executions are shown in Fig. 15. For every mesh, the full-order solution computed with the direct solver is used as reference.

Similar levels of acceleration are obtained for all of the partitioning strategies, with a speed-up as high as 2.8. Interestingly, running the model without partitioning yields higher speed-ups across all mesh densities. The resulting acceleration of the scheme is determined by a number of factors. When the ϵ_{force} threshold is crossed, the model must discard its current solution and run an expensive full step. Partitioning the equilibrium system alleviates this issue, but at the cost of solving for a significantly higher number of DOFs at every reduced time step. The balance

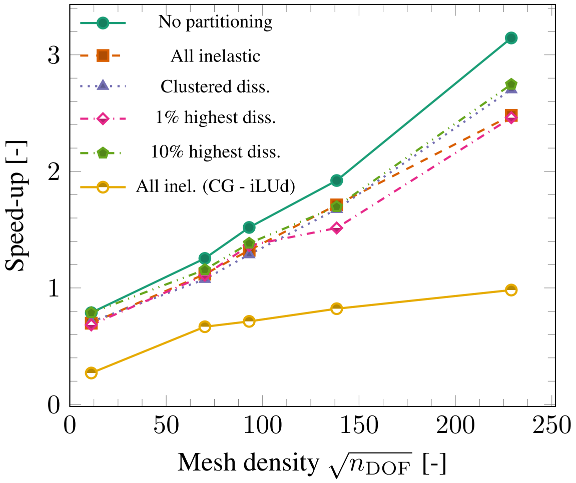

Fig. 15. Acceleration provided by the adaptive POD model for different partitioning strategies and solvers ($r = 0.9\,\mathrm{mm}$).

between these competing factors in the search for an optimized level of acceleration is a complex and problem-dependent issue. It is also important to recall that partitioning promotes a smoother transition from a reduced phase with outdated $\mathbf{\Phi}$ to a full step. In this particular example, the plasticity model used here is robust enough to handle these transitions without partitioning, but that might not be the case for different structures or material models.

Finally, the speed-ups obtained with the modified CG solver are significantly lower than the ones obtained with the direct solver. It should be noted, however, that since solving the equilibrium problem with the arc-length constraint of Eq. (43) requires two solving operations for each iteration, the direct solver, which performs the factorization of $\mathbf{K}_{\mathrm{h}}$ only once and reuses it for the second solve operation, has a clear advantage over the iterative solver. Therefore, computing the speed-ups of the modified CG solver with respect to the direct solver does not yield a fair performance comparison.

4.2. Domain-based hyper-reduction

The next tests investigate the performance of the domain-based hyper-reduced model of Section 3.2. For the first set of examples, the bar with $r = 0.9\,\mathrm{mm}$ is executed with one POD domain $\mathcal{P}$ and one ECM domain $\mathcal{H}_1$, *i.e.* only the first phase of the clustering procedure of Algorithm 2 is performed. These domains are updated every time a full step is executed.

First, the ability of the model to trigger full steps based only on information in $\mathcal{P}$ is assessed. The model is executed without DOF partitioning for multiple values of $\epsilon_{\mathrm{force}}$, with the resulting load–displacement curves shown in Fig. 16. The model performs in a similar way as the one with only POD reduction (Fig. 4), with an increase in precision as $\epsilon_{\mathrm{force}}$ decreases.

The shape of the $\mathcal{P}$ and $\mathcal{H}_1$ domains at the first and last steps of the analysis can be seen in Fig. 17. As the plastic mechanism at the edges of the cutout is formed and the model is retrained, the strain clustering process tends to lead to progressively smaller $\mathcal{P}$ domains. Since most of the domain is integrated with ECM, the cost of computing the constitutive response is significantly reduced while keeping relevant information about the region where plasticity is most active in order to compute the error control condition of Eq. (39).

The preceding test did not make use of any DOF partitioning strategy ($\mathcal{F} = \varnothing$). When $\mathcal{F}$ if adaptively populated, the displacement-based error control condition of Eq. (40) becomes necessary in order to enforce the quality control of $\mathbf{\Phi}$. The same model is executed for multiple values of ϵ_{disp} while keeping $\epsilon_{\mathrm{force}} = 0.05$. The average error obtained with the different partitioning strategies is shown in Fig. 18. The addition of this second error control criterion effectively restores the ability of the model to trigger full steps for strategies that tend to make $\mathcal{F}$ occupy most of the domain $\mathcal{P}$. It is interesting to note that if a low value for p_{Ξ} is chosen, the *highest dissipation* strategy tends to limit the size of $\mathcal{F}$ and maintain solution accuracy for any value of ϵ_{disp}.

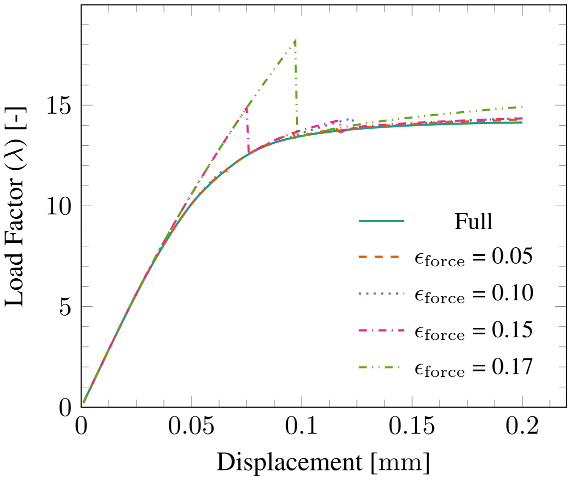

Fig. 16. Hybrid POD/ECM response without DOF partitioning for multiple values of ϵ_{force} ($r = 0.9\,\text{mm}$).

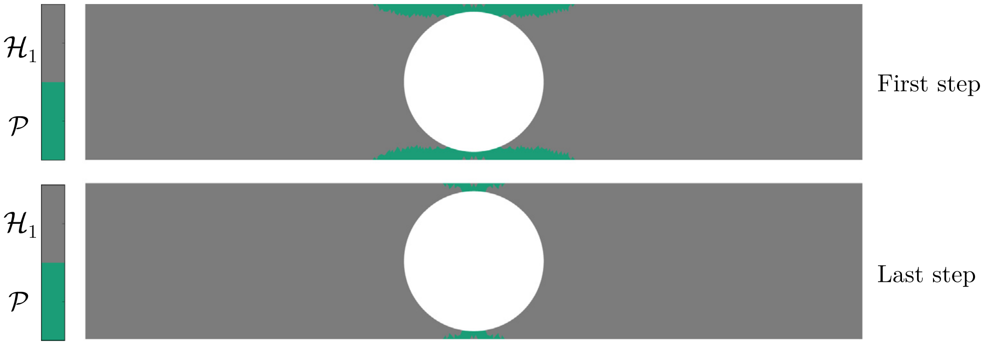

Fig. 17. Adaptive reduction domains at the first and last load steps ($r = 0.9\,\text{mm}$).

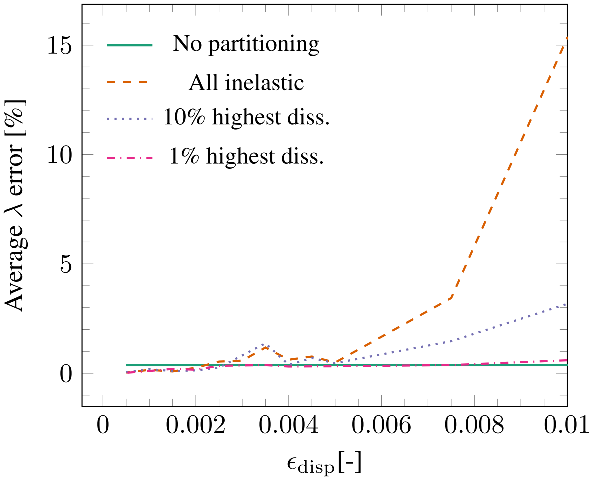

Fig. 18. Average reduced model error for different values of ϵ_{disp} ($r = 0.9\,\text{mm}$, $\epsilon_{\text{force}} = 0.05$).

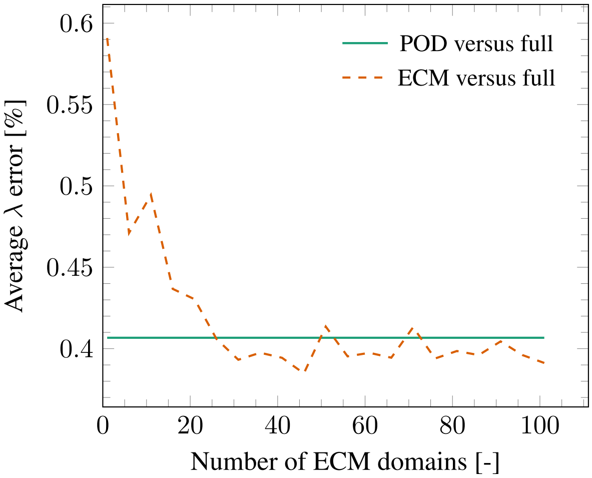

Fig. 19. Average error obtained by models with different numbers of ECM domains ($r = 0.5$).

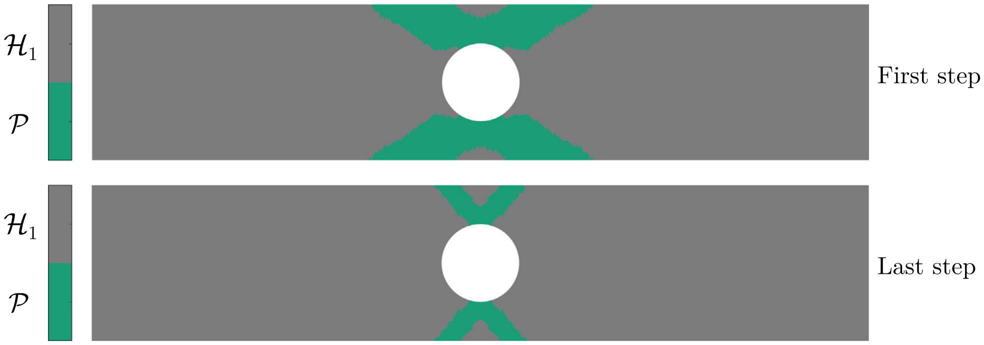

Fig. 20. Adaptive reduction domains at the first and last load steps ($r = 0.5$ mm).

4.2.1. Performance with number of ECM clusters

Previous tests only performed the first phase of the strain clustering of Algorithm 2, which yields only a single ECM domain $\mathcal{H}_1$. This single ECM domain can be further divided through a second clustering phase, yielding a set $\boldsymbol{\mathcal{H}}$ of ECM domains which are independently trained. This results in an increased number of integration points which can be beneficial in reducing the integration error inherent to ECM. In the upper-bound case where there are as many ECM domains as finite elements, integration error vanishes and the response of the domain-based reduced model converges to the one obtained with full integration.

For this example, a bar with $r = 0.5$ mm is executed with a number of ECM domains ranging from 0 ($\mathcal{P}$ covers the whole domain Ω) to 100 and the average load factor error is computed. In order to consistently compare the results obtained with and without ECM, partitioning is deactivated and a full step is manually triggered after every 30 load steps, forcing all models to have the same number of correction steps. Results can be seen in Fig. 19. Even though integration error is already low for this particular problem, it is further decreased with an increase in the number of ECM domains, as expected. After 20 domains, the reduced response reaches the error level of a fully-integrated model. The slight fluctuations after this point can be attributed to numerical error.

The reduced domain topologies obtained after the first clustering procedure and the one at the last load step can be seen in Figs. 20 and 21. Similar to the test with $r = 0.9$ mm (Fig. 17), the size of $\mathcal{P}$ tends to decrease as strain localizes on a narrow band around the cutout. Similar to the strain clusters generated by Liu et al. [17], the ECM domains are not necessary contiguous. It is also interesting to note that cluster topology is symmetric with respect to the cutout.

The advantage of using a clustering procedure with two distinct phases (Algorithm 2) can be seen when comparing Fig. 20 with Fig. 21. Although the k-means clustering procedure tends to form increasingly smaller

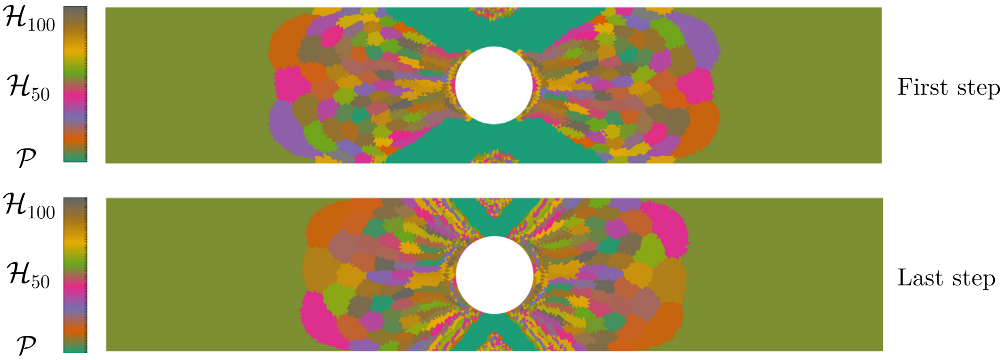

Fig. 21. Adaptive reduction domains at the first and last load steps for 100 ECM clusters ($r = 0.5$ mm).

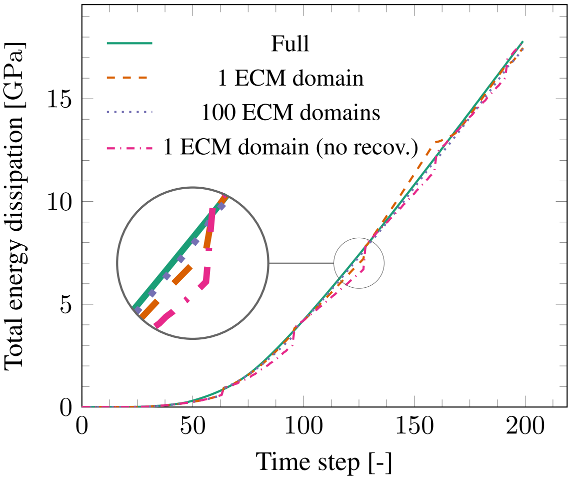

Fig. 22. Energy dissipation predicted with different numbers of ECM domains.

ECM domains as k increases, performing a first clustering with $k = 2$ guarantees that the size of $\mathcal{P}$ is independent of the number of ECM domains used on the rest of the mesh.

4.2.2. History recovery

Next, the performance of the history recovery procedure of Section 2.4 is discussed when used in combination with the present domain-based reduction framework. Given the robustness of the plasticity model used in the present investigation, the history recovery step has been skipped in the previous hyper-reduced numerical tests. Although the constitutive model was able to cope with this outdated history upon switching to a full step, history recovery would be mandatory for certain constitutive models — *e.g.* a viscoelastic model with hereditary stresses.

The example explored here consists in predicting the total energy dissipation of the model at each time step. For the sake of illustration, history recovery is performed after every hyper-reduced time step — thus not only when a full step is triggered (Algorithm 3). The full-order prediction and the responses predicted by three different models — 1 ECM domain with and without recovery and 100 ECM domains — are shown in Fig. 22.

Since recovery is performed by approximating the history values based on an interpolation of history at ECM points, it follows that increasing the number of ECM points should lead to better predictions. Indeed, the dissipation response with 100 ECM domains is more accurate than the one obtained with only 1 ECM domain, as can be seen on the zoomed-in portion of the dissipation curve of Fig. 22 showing the reduced steps immediately prior to a correction step. This is an interesting consequence of the proposed domain-based approach and can be seen as another possible motivation for using a larger number of ECM domains. Nevertheless, the reduced response with a single domain and no history recovery is already remarkably accurate. This highlights another advantage of keeping

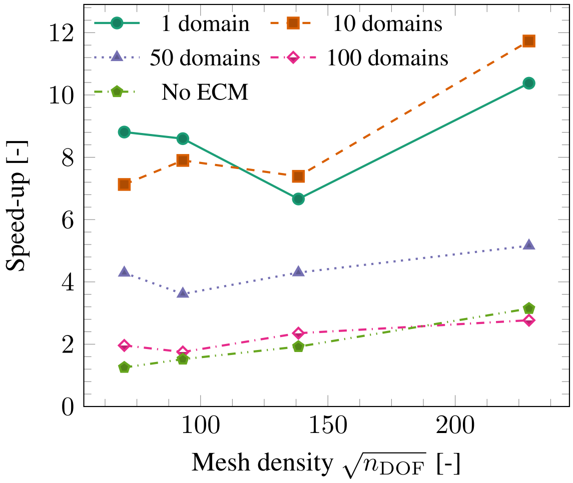

Fig. 23. Acceleration provided by the domain-based reduction framework with different numbers of ECM domains ($r = 0.9\,\text{mm}$).

part of the mesh fully integrated: as most of the energy is dissipated by elements in $\mathcal{P}$, the loss of history information due to ECM is greatly alleviated. This can also be seen as one of the reasons why the model with no recovery shows virtually no convergence issues when switching to full steps.

4.2.3. *Acceleration*

The next set of numerical tests investigates the additional acceleration promoted by integrating most of the finite element mesh with ECM. In order to compare the speed-ups with those of Fig. 15, the bar example with $r = 0.9\,\text{mm}$ and $\epsilon_{\text{force}} = 0.04$ is considered. In order to keep the discussion as focused as possible, no DOF partitioning is performed and only the effect of adding more ECM domains is investigated. Results for multiple numbers of ECM domains are shown in Fig. 23.

Similar acceleration levels are obtained for 1 and 10 ECM domains, with values up to 12 with respect to the full analysis. Introducing hyper-reduction leads to speed-ups approximately 4 times higher than the ones obtained for the POD-reduced model with full integration, included in Fig. 23 for comparison (*No ECM*). The computational overhead associated with defining $\mathcal{P}$ and $\mathcal{H}$ increases with the number of ECM domains, as the k-means clustering operation becomes more complex. This is reflected by the significant reduction in speed-up for 50 and 100 clusters.

In order to further investigate this effect, the same tests are executed while keeping the cluster topology obtained after the first time step intact throughout the analysis. The obtained speed-ups are shown in Fig. 24. In this case, the speed-ups for different levels of domain decomposition are more similar. A speed-up decay is still observed as more ECM domains are added and can be attributed to the cost of the first (and only) clustering process and to increases in the number of ECM points. It is also interesting to note the reduction in speed-up for 1 and 10 domains with respect to the results in Fig. 23, which is related to the fact that $\mathcal{P}$ is not allowed to shrink if cluster topology is fixed.

Finally, it is interesting to investigate the alternative approach of refraining from using a domain-based approach and opting for extending $\mathcal{H}$ to the whole domain Ω. Since all of the adaptive components of the framework are lost, reduction error is controlled by manually triggering a fully-solved step at predefined intervals. Speed-ups are shown in Fig. 25.

Retraining the model after every 5 steps yields a level of speed-up that is lower than the one obtained with the adaptive approach. This retraining frequency is therefore higher than it could be while still maintaining good accuracy. Increasing the interval to one retraining every 10 steps still yields lower speed-ups than those of Fig. 23 but the results suggest that the frequency should not be further increased: the model with mesh density $\sqrt{n_{\text{DOF}}} = 93$ fails to converge and is therefore missing from the curve. As expected, further decreasing the frequency to one retraining every 30 steps does bring a higher level of acceleration but convergence cannot be obtained for most of the models. Even with a robust constitutive model, convergence is difficult or impossible due to the fact that retraining is not triggered at the correct moments and due to a lack of accuracy in material history at the most critical mesh regions upon retraining.

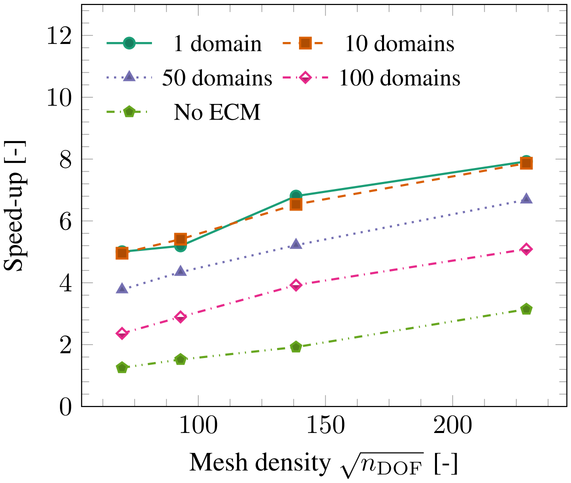

Fig. 24. Acceleration levels for models with different numbers of ECM domains and fixed cluster topology ($r = 0.9$ mm).

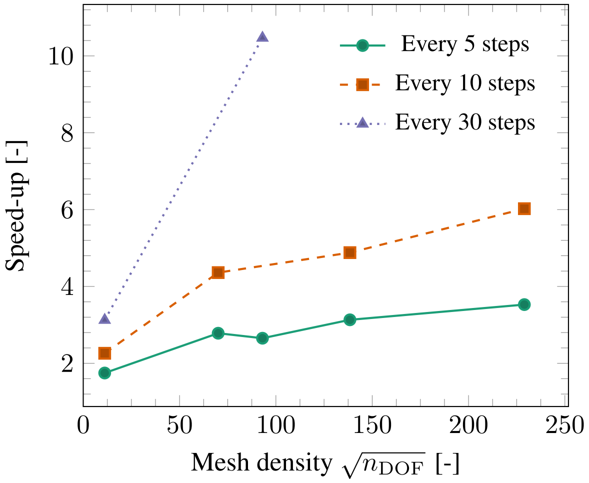

Fig. 25. Acceleration levels for purely hyper-reduced models retrained at predefined intervals ($r = 0.9$ mm).

4.3. Multiple cutouts

A second example is shown in order to demonstrate the ability of the adaptive reduction framework to deal with more complex distributions of nonlinearity. A square plate with a number of randomly distributed cutouts is modeled in plane strain and loaded horizontally in tension, as shown in Fig. 26. As the plate is loaded, plastic strain initially arises between closely packed cutouts and later expands into the surrounding material, eventually forming a strain localization band that leads to an almost perfectly plastic structural behavior. The plate is meshed with 22 854 constant strain triangles with one integration point each ($N = 23856$, $M = 22854$). The reduction parameters are the same as the ones used for the bar example but $\epsilon_{\mathrm{force}}$ is fixed at 0.015. The *clustered dissipation* strategy is used to populate $\mathcal{F}$. For the strain clustering that defines $\mathcal{P}$ and $\boldsymbol{\mathcal{H}}$, five ECM domains are used and domain configuration is fixed throughout the analysis.

The domain topology resulting from the clustering procedure can be seen in Fig. 26. Similar to the bar example, the domains are composed of spatially disconnected volumes, but here they are highly asymmetrical and reflect the complex strain field that arises from the presence of the cutouts. Nevertheless, the framework performs well, resulting in nearly identical plastic strain distribution (Fig. 27) and load–displacement responses (Fig. 28).

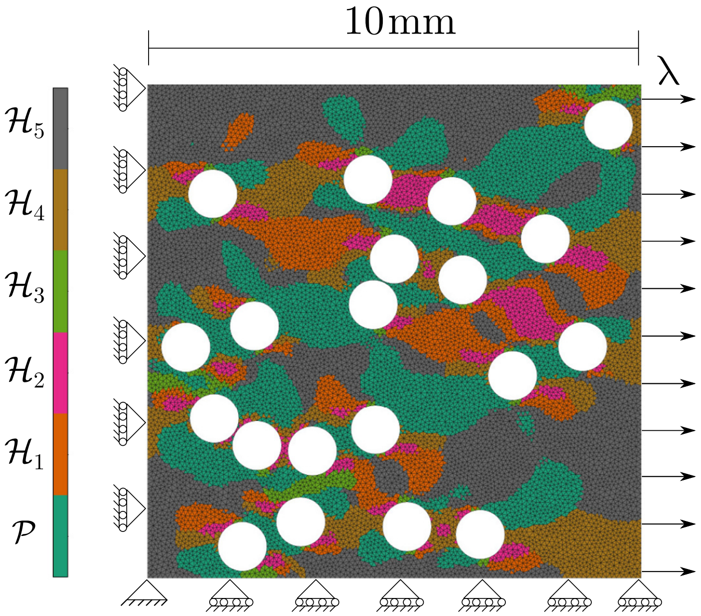

Fig. 26. Square plate with multiple cutouts. Loads, boundary conditions and ECM domain topology.

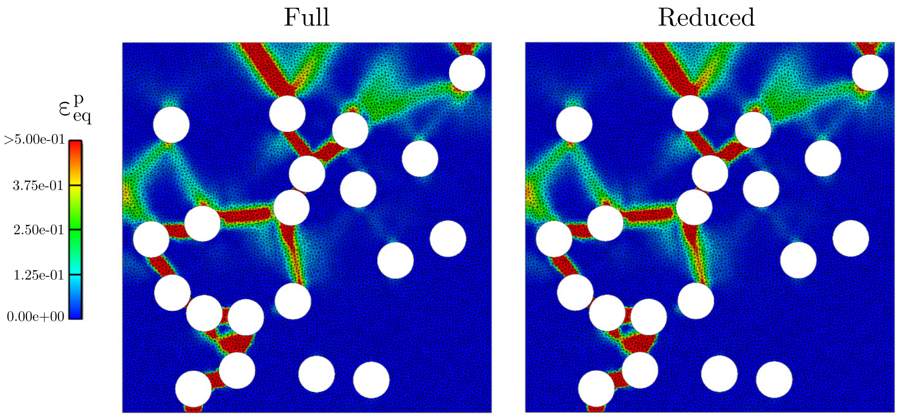

Fig. 27. Plastic strain distribution for the example with multiple cutouts: equivalent plastic strain fields of the fully-solved and hyper-reduced models.

The hyper-reduced model runs approximately 9 times faster than the fully-solved one. Of the total execution time of the reduced model (224.3s), a total of 7.13 s (3.2 %) is spent on the adaptive components of the framework — computing $\mathcal{P}$, $\boldsymbol{\mathcal{H}}$ and the $\mathcal{P}-\boldsymbol{\mathcal{H}}$ border at the beginning of the analysis (0.015 s), updating $\mathcal{F}$ (6.17 s) and checking the error of Eq. (39) (0.58 s) at every time step and recomputing $\boldsymbol{\Phi}$, $\mathcal{Z}$ and $\boldsymbol{\varpi}$ after each of the 20 retraining steps (0.37 s). The reason why tasks associated with building the reduced model — which are conventionally performed *offline* — are relatively cheap here is that the POD basis remains very small throughout the analysis.

5. Conclusions

An adaptive reduced-order modeling framework combining POD, ECM and equilibrium system partitioning has been proposed. By starting with a fully-solved step and progressively building a hyper-reduced approximation of the problem being solved, the need for an *offline* training phase is precluded. The framework uses the strain distribution obtained during full steps to compute an adaptive domain topology that allows for multiple levels of reduction to

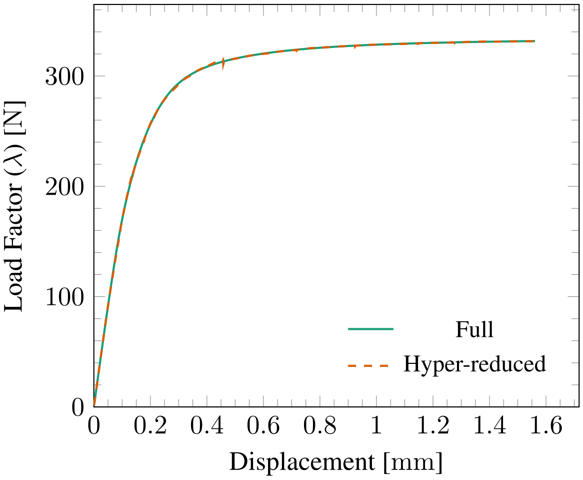

Fig. 28. Plate with multiple cutouts: load–displacement curves of the fully-solved and hyper-reduced models.

coexist within the same finite element mesh. By applying hyper-reduction only on mesh regions away from the main sources of nonlinearity, most of the relevant material history is preserved and error control conditions can be used to adaptively trigger a retraining process.

The framework has been applied in a series of numerical tests in order to assess its accuracy and levels of acceleration. First, models with only POD reduction were investigated. The retraining procedure was able to successfully monitor when the reduced basis becomes outdated and move the solution back to the correct path after a fully-solved correction step. Adaptively moving DOFs from elements responsible for most sources of nonlinearity to the full solution space has been found to significantly decrease the number of retraining steps necessary to maintain a given level of accuracy, although this strategy was better suited for models with localized nonlinearity. A modified CG solver that takes advantage of the DOF partitioning to solve a smaller problem condensed to only the full DOFs was found to effectively decrease the number of iterations necessary for convergence. However, at least for the present investigation, a direct solver was found to yield higher speed-ups than the modified iterative solver.

Including the domain-based ECM hyper-reduction strategy in the POD-reduced models increased the obtained acceleration levels (up to 12 times faster than the full-order solution) while maintaining the same level of accuracy. However, using the hybrid POD/ECM model in combination with the equilibrium system partitioning strategy required the introduction of an additional error control criterion in order to maintain the ability of the model to trigger full-order correction steps. Increasing the number of ECM domains was found to improve both the integration error caused by hyper-reduction and the error inherent to the *Gappy Data* history reconstruction procedure, although the added complexity introduced by increasing the number of clusters led to significant reductions in speed-up.

Acknowledgment

FM acknowledges financial support from the Netherlands Organization for Scientific Research (NWO) under Vidi Grant No. 16464.

References

[1] C. Miehe, J. Schotte, J. Schröder, Computational micro-macro transitions and overall moduli in the analysis of polycrystals at large strains, Comput. Mater. Sci. 16 (1999) 372–382.

[2] V. Kouznetsova, W.A.M. Brekelmans, F.P.T. Baaijens, An approach to micro-macro modeling of heterogeneous materials, Comput. Mech. 27 (2001) 37–48.

[3] I.B.C.M. Rocha, F.P. van der Meer, S. Raijmaekers, F. Lahuerta, R.P.L. Nijssen, L.P. Mikkelsen, L.J. Sluys, A combined experimental/numerical investigation on hygrothermal aging of fiber-reinforced composites, Eur. J. Mech. A-Solids 73 (2019) 407–419.

[4] A. Krairi, I. Doghri, A thermodynamically-based constitutive model for thermoplastic polymers coupling viscoelasticity, viscoplasticity and ductile damage, Int. J. Plast. 60 (2014) 163–181.

[5] S. Haouala, I. Doghri, Modeling and algorithms for two-scale time homogenization of viscoelastic-viscoplastic solids under large numbers of cycles, Int. J. Plast. 70 (2015) 98–125.

[6] K. Terada, M. Kurumatani, Two-scale diffusion-deformation coupling model for material deterioration involving micro-crack propagation, Internat. J. Numer. Methods Engrg. 83 (2010) 426–451.

[7] E.S. Barroso, E. Parente Jr, A.M.C. Melo, A hybrid PSO-GA algorithm for optimization of laminated composites, Struct. Multidiscip. Optim. 55 (2017) 2111–2130.

[8] S. Joglekar, K. von Hagel, M. Pankow, S. Ferguson, Exploring how optimal composite design is influenced by model fidelity and multiple objectives, Compos. Struct. 160 (2017) 964–975.

[9] P. Kerfriden, P. Gosselet, S. Adhikari, S.P.A. Bordas, Bridging proper orthogonal decomposition methods and augmented newton-krylov algorithms: An adaptive model order reduction for highly nonlinear mechanical problems, Comput Method Appl M 200 (2011) 850–866.

[10] M. Chevreuil, A. Nouy, Model order reduction based on proper generalized decomposition for the propagation of uncertainties in structural dynamics, Internat. J. Numer. Methods Engrg. 89 (2012) 241–268.

[11] S. Chaturantabut, D.C. Sorensen, Nonlinear model reduction via discrete empirical interpolation, SIAM J. Sci. Comput. 32 (2010) 2737–2764.

[12] J.A. Hernández, M.A. Caicedo, A. Ferrer, Dimensional hyper-reduction of nonlinear finite element models via empirical cubature, Comput. Methods Appl. Mech. 313 (2017) 687–722.

[13] F. Ghavamian, P. Tiso, A. Simone, POD-DEIM model order reduction for strain-softening viscoplasticity, Comput. Methods Appl. Mech. 317 (2017) 458–479.

[14] B. Peherstorfer, D. Butnaru, K. Willcox, H.-J. Bungartz, Localized discrete empirical interpolation method, SIAM J. Sci. Comput. 36 (2014) A168–A192.

[15] B. Haasdonk, M. Dihlmann, M. Ohlberger, A training set and multiple bases generation approach for parameterized model reduction based on adaptive grids in parameter space, Math. Comput. Model. Dyn. Syst. 17 (2011) 423–442.

[16] O. Goury, D. Amsallem, S.P.A. Bordas, W.K. Liu, P. Kerfriden, Automatised selection of load paths to construct reduced-order models in computational damage micromechanics: from dissipation-driven random selection to Bayesian optimization, Comput. Mech. 58 (2016) 213–234.

[17] Z. Liu, M. Bessa, W.K. Liu, Self-consistent clustering analysis: An efficient multi-scale scheme for inelastic heterogeneous materials, Comput. Methods Appl. Mech. 306 (2016) 319–341.

[18] R.A. van Tuijl, C. Harnish, K. Matouš, J.J.C. Remmers, M.G.D. Geers, Wavelet based reduced order models for microstructural analyses, Comput. Mech. 63 (2019) 535–554.

[19] P. Kerfriden, J.C. Passieux, S.P.A. Bordas, Local/global model order reduction strategy for the simulation of quasi-brittle failure, Internat. J. Numer. Methods Engrg. 89 (2012) 154–179.

[20] P. Kerfriden, O. Goury, T. Rabczuk, S.P.A. Bordas, A partitioned model order reduction approach to rationalise computational expenses in nonlinear fracture mechanics, Comput. Methods Appl. Mech. 256 (2013) 169–188.

[21] F.P. van der Meer, Micromechanical validation of a mesomodel for plasticity in composites, Eur. J. Mech. A-Solids 60 (2016) 58–69.

[22] R. Everson, L. Sirovich, Karhunen-Loeve procedure for gappy data, J. Opt. Soc. Amer. A 12 (1996) 1567–1664.

[23] J.A. Hernández, J. Oliver, A. Huespe, M. Caicedo, J. Cante, High-performance model reduction techniques in computational multiscale homogenization, Comput. Methods Appl. Mech. 276 (2014) 149–189.

[24] I.B.C.M. Rocha, F.P. van der Meer, L.J. Sluys, Efficient micromechanical analysis of fiber-reinforced composites subjected to cyclic loading through time homogenization and reduced-order modeling, Comput. Methods Appl. Mech. 345 (2019) 644–670.

[25] S.P. Lloyd, Least squares quantization in pcm, IEEE Trans. Inform. Theory 28 (1982) 129–137.

[26] Jive - Software development kit for advanced numerical simulations, http://jive.dynaflow.com, accessed: 04-03-2018.

[27] A.R. Melro, P.P. Camanho, F.M. Andrade Pires, S.T. Pinho, Micromechanical analysis of polymer composites reinforced by unidirectional fibres: Part I - Constitutive modelling, Int. J. Solids Struct. 50 (2013) 1897–1905.

[28] M.A. Gutiérrez, Energy release control for numerical simulations of failure in quasi-brittle solids, Commun. Numer. Methods Eng. 20 (2004) 19–29.

[29] F.P. van der Meer, C. Oliver, L.J. Sluys, Computational analysis of progressive failure in a notched laminate including shear nonlinearity and fiber failure, Compos. Sci. Technol. 70 (2010) 692–700.