Available online at www.sciencedirect.com
ScienceDirect

Comput. Methods Appl. Mech. Engrg. 317 (2017) 458–479

Computer methods in applied mechanics and engineering

www.elsevier.com/locate/cma

POD–DEIM model order reduction for strain-softening viscoplasticity

F. Ghavamian[a,*], P. Tiso[b], A. Simone[a]

[a] Faculty of Civil Engineering and Geosciences, Delft University of Technology, P.O. Box 5048, 2600 GA Delft, The Netherlands
[b] Institute for Mechanical Systems, ETH Zürich, Leonhardstrasse 21, 8092 Zürich, Switzerland

Received 19 May 2016; received in revised form 14 October 2016; accepted 18 November 2016
Available online 10 December 2016

Abstract

We demonstrate a Model Order Reduction technique for a system of nonlinear equations arising from the Finite Element Method (FEM) discretization of the three-dimensional quasistatic equilibrium equation equipped with a Perzyna viscoplasticity constitutive model. The procedure employs the Proper Orthogonal Decomposition-Galerkin (POD-G) in conjunction with the Discrete Empirical Interpolation Method (DEIM). For this purpose, we collect samples from a standard full order FEM analysis in the offline phase and cluster them using a novel k-means clustering algorithm. The POD and the DEIM algorithms are then employed to construct a corresponding reduced order model. In the online phase, a sample from the current state of the system is passed, at each time step, to a nearest neighbor classifier in which the cluster that best describes it is identified. The force vector and its derivative with respect to the displacement vector are approximated using DEIM, and the system of nonlinear equations is projected onto a lower dimensional subspace using the POD-G. The constructed reduced order model is applied to two typical solid mechanics problems showing strain-localization (a tensile bar and a wall under compression) and a three-dimensional square-footing problem.

Keywords: Model order reduction; Proper orthogonal decomposition; Discrete empirical interpolation method; Perzyna viscoplasticity; Strain-softening; Machine learning; k-means clustering algorithm

1. Introduction

In spite of constant advances in theoretical development and computing power, there are regularly applied many-query procedures, such as parameter sensitivity analysis or parameter estimation for inverse problems, that require many computationally expensive simulations. Model Order Reduction (MOR) techniques aim to reduce the computational cost of these procedures by breaking them into a two-phase procedure commonly known as the offline–online decomposition. A reduced order model is first trained in the computationally expensive offline phase

[*] Corresponding author.
E-mail addresses: f.ghavamian@tudelft.nl, fari.ghavamian@gmail.com (F. Ghavamian), ptiso@ethz.ch (P. Tiso), a.simone@tudelft.nl (A. Simone).

http://dx.doi.org/10.1016/j.cma.2016.11.025

and then used in the online phase to efficiently carry out a many-query procedure. Successful examples can be found in structural dynamics [1], computational fluid dynamics [2,3] and control systems [4]. Here we employ a set of machine learning techniques that can significantly improve the computational efficiency of a reduced order model.

The main hypothesis of any MOR technique can be put forth as follows [5]:

> “For a particular system, the solution space is often attracted to a low-dimensional manifold”.

This hypothesis being true, it is in principle possible to build a set of basis vectors for a lower-dimensional subspace of the solution space, in which solution vectors can be approximated with acceptable accuracy, and therefore reduce the computational cost. The lower-dimensional subspace is built by means of samples collected from the solution space and the application of a Principal Component Analysis [6].

Many MOR techniques (Proper Orthogonal Decomposition-Galerkin (POD-G) [7], Balanced Truncation [8] and Moment Matching [9] to cite a few examples) have proven effective in reducing the computational cost of a linear model. However, they all fail to effectively reduce the computational cost of a nonlinear model [5], as the cost of the evaluation of the nonlinear contribution scales to the size of the full order system. To overcome this inefficiency, hyper-reduction methods, such as Missing Point Estimation [10], Gappy Proper Orthogonal Decomposition [11] and Discrete Empirical Interpolation Method (DEIM) [5], have been proposed to approximate the nonlinear contributions.

In the field of solid mechanics, MOR techniques such as POD-G [12,13], Karhunen Loeve Expansion [14,15], and Proper Generalized Decomposition (PGD) [16,17] are employed to reduce the dimension of the system of equations. Nevertheless, the complexity of the evaluation of the constitutive equation remains a computational bottleneck. At variance with these models, in the hyper-reduction technique proposed in Ref. [18] the integration of the constitutive equation is performed in a reduced integration domain. In Ref. [19], an FE efficient unassembled variant of DEIM (UDEIM) is proposed and applied to geometrically nonlinear structural dynamics problems. In [20], UDEIM is further enhanced to reduce the associated offline cost. Apart from PGD, which is an a-priori method, these contributions focus only on the MOR technique itself, presuming that the underlying samples are fully capable of constructing an efficient reduced order model. We show that this is not necessarily always a valid assumption for the class of problems we are tackling here.

In this contribution we demonstrate an application of a MOR technique – POD-G, Section 3.1 – in conjunction with a hyper-reduction method – DEIM, Section 3.2 – to build a reduced order model for a boundary value problem – quasistatic equilibrium equation – with a nonlinear and path-dependent constitutive law—Perzyna viscoplasticity [21], Section 2. We observe that the singular values of the collected samples, which we wish had a fast decay as a proof of the central hypothesis of the MOR techniques, decrease at a slow rate. As a result, the efficiency of the reduced order model is diminished. To overcome this issue, Peherstorfer et al. [22] and Haasdonk et al. [23] divide the time domain and the parameter space into several local regions and construct a reduced order model for each one of them individually. Following a similar logic, we employ a set of Machine Learning techniques, such as the k-means clustering algorithm, Section 3.3.1, and the nearest neighbor classifier, Section 3.3.2, as suggested in [22]. As novel contributions, (i) we modify the distance definition in the k-means clustering algorithm to increase the consistency of the clustering procedure and, consequently, the effectiveness of DEIM; (ii) we modify the distance metric in the k-means clustering algorithm and the nearest neighbor classifier, which we heuristically show to be vital for a consistent classification; and (iii) we propose a trivial initial clustering in k-means clustering algorithm that exploits the path-dependent characteristic of the problem. The merits of the proposed MOR technique are demonstrated by three typical solid mechanics problems in Section 4.

2. Problem statement

A strain-softening Perzyna viscoplasticity constitutive model is employed as an archetype of a class of constitutive models typically encountered in nonlinear solid mechanics. The three-dimensional quasistatic equilibrium equation

$$\boldsymbol{L}^{\mathrm{T}}\,\boldsymbol{\sigma} = \boldsymbol{q} \qquad \text{in} \qquad \Omega, \tag{1}$$

expressed using matrix notation, is defined in the body Ω that is bounded by the surface $\Gamma = \Gamma_{\mathrm{u}} \cup \Gamma_{\mathrm{t}}$. A set of appropriate Dirichlet and Neumann boundary conditions is applied on Γ_{u} and Γ_{t}, respectively. In (1), $\boldsymbol{\sigma}$ is an array containing the components of the stress tensor in engineering notation, $\boldsymbol{q}$ the body force vector, $\boldsymbol{u}$ the displacement vector, and the operator matrix $\boldsymbol{L}$ is defined as

$$\boldsymbol{L}^{\mathrm{T}} = \begin{bmatrix} \partial_x & 0 & 0 & \partial_y & 0 & \partial_z \\ 0 & \partial_y & 0 & \partial_x & \partial_z & 0 \\ 0 & 0 & \partial_z & 0 & \partial_y & \partial_x \end{bmatrix}, \tag{2}$$

in which $\partial_{\#} = \frac{\partial}{\partial \#}$ is the partial derivative operator with respect to #. The equilibrium equation is equipped with the stress–strain relationship

$$\dot{\boldsymbol{\sigma}} = \boldsymbol{D}^{\mathrm{e}} \, (\dot{\boldsymbol{\epsilon}} - \dot{\boldsymbol{\epsilon}}^{\mathrm{vp}}) \tag{3}$$

in rate form, where $\dot{\#} = \frac{\partial \#}{\partial t}$ is the time derivative of #, $\boldsymbol{D}^{\mathrm{e}}$ represents the elastic modulus tensor, $\boldsymbol{\epsilon}$ represents the strain tensor, and $\boldsymbol{\epsilon}^{\mathrm{vp}}$ the viscoplastic strain tensor. In a Perzyna viscoplasticity constitutive model [21], the viscoplastic strain evolves following the flow rule

$$\dot{\boldsymbol{\epsilon}}^{\mathrm{vp}} = \dot{\lambda} \, \frac{\partial \theta}{\partial \boldsymbol{\sigma}} \tag{4}$$

with the rate of the plastic multiplier

$$\dot{\lambda} = \eta \, \langle \phi_{(\theta)} \rangle^{\beta}, \tag{5}$$

the yield function

$$\theta = \sigma_{\mathrm{vm}} - \sigma_{\mathrm{Y}}, \tag{6}$$

and the overstress function

$$\phi_{(\theta)} = \frac{\theta}{\sigma_{\mathrm{Y}}}. \tag{7}$$

In the above equations, $\langle \# \rangle$ is the Macaulay bracket, η is the viscosity parameter, β is a model parameter, σ_{Y} is the current yield stress, and σ_{vm} is the von Mises stress. To introduce a strain-softening response, we employ the relation

$$\sigma_{\mathrm{Y}} = \sigma_{\mathrm{Y}_0} \, ((1 + a) \, e^{-b \, \kappa} - a \, e^{-2b \, \kappa}), \tag{8}$$

where a and b are model parameters, σ_{Y_0} is the initial yield stress, and the plastic strain

$$\kappa = \lambda. \tag{9}$$

The strong form of the quasistatic equilibrium equation (1) is cast into its corresponding weak form and the displacement field is discretized in space using the Finite Element Method (FEM). Standard procedures yield the nonlinear discrete set of equations

$$\boldsymbol{r}(\boldsymbol{a}) = \boldsymbol{f}_{\mathrm{int}}(\boldsymbol{a}) - \boldsymbol{f}_{\mathrm{ext}} = \boldsymbol{0}, \tag{10}$$

where

$$\begin{aligned} \boldsymbol{f}_{\mathrm{int}}(\boldsymbol{a}) &= \int_{\Omega} \boldsymbol{B}^{\mathrm{T}} \boldsymbol{\sigma}(\boldsymbol{a}) \, \mathrm{d}\Omega, \qquad \text{and} \\ \boldsymbol{f}_{\mathrm{ext}} &= \int_{\Gamma_{\mathrm{t}}} \boldsymbol{N}^{\mathrm{T}} \bar{\boldsymbol{t}} \, \mathrm{d}\Gamma \, + \, \int_{\Omega} \boldsymbol{N}^{\mathrm{T}} \boldsymbol{q} \, \mathrm{d}\Omega. \end{aligned} \tag{11}$$

In (11), $\boldsymbol{N} \in \mathbb{R}^{3 \times N}$ is a matrix that contains shape functions, $\boldsymbol{B} = \boldsymbol{L} \, \boldsymbol{N} \in \mathbb{R}^{6 \times N}$ contains derivatives of the shape functions, $\bar{\boldsymbol{t}}$ are the prescribed tractions on Γ_{t}, $\boldsymbol{a} \in \mathbb{R}^{N}$ is a vector of the nodal values of the displacement vector $\boldsymbol{u}$, and N is the total number of degrees of freedom (DOFs).

The system of equations (10) is then solved using a Newton–Raphson scheme according to which

$$\begin{aligned} \boldsymbol{K}^{j} \, \mathrm{d}\boldsymbol{a}^{j+1} &= -\boldsymbol{r}^{j}, \qquad \text{and} \\ \boldsymbol{a}^{j+1} &= \boldsymbol{a}^{j} \, + \, \mathrm{d}\boldsymbol{a}^{j+1}, \end{aligned} \tag{12}$$

where $\boldsymbol{r}^j \in \mathbb{R}^N$ is the residual vector at iteration j, $\boldsymbol{K}^j \in \mathbb{R}^{N\times N}$ is the stiffness matrix at iteration j, and $\mathrm{d}\boldsymbol{a}^{j+1} \in \mathbb{R}^N$ is the increment of the displacement vector at iteration $j+1$. From here onwards, the iteration number j is dropped for the sake of readability. The residual vector

$$\boldsymbol{r} = \boldsymbol{K}^{\mathrm{L}}\,\boldsymbol{a} \;+ \boldsymbol{f} - \boldsymbol{f}_{\mathrm{ext}} \tag{13}$$

consists of a nonlinear part, $\boldsymbol{f}$, and a linear part, $\boldsymbol{K}^{\mathrm{L}}\,\boldsymbol{a} - \boldsymbol{f}_{\mathrm{ext}}$. In the linear part, the matrix

$$\boldsymbol{K}^{\mathrm{L}} = \int_{\Omega} \boldsymbol{B}^{\mathrm{T}}\,\boldsymbol{D}^{\mathrm{e}}\,\boldsymbol{B}\,\mathrm{d}\Omega \;\in \mathbb{R}^{N\times N} \tag{14}$$

is the linear elastic stiffness matrix, equal to the stiffness matrix $\boldsymbol{K}$ at the first iteration of the first time step. In the nonlinear part, the vector $\boldsymbol{f} \;= \boldsymbol{f}(\boldsymbol{a}) \;\in\; \mathbb{R}^N$ is referred to as the force vector. The derivative of the force vector, $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{a}} \in \mathbb{R}^{N\times N}$, generates the nonlinear part of the stiffness matrix.

Strain-softening viscoplasticity typically induces strongly localized strain fields. As a result, a large number of degrees of freedom is required to resolve these gradients, thus causing two computational bottlenecks:

1. the solution of an $N \times N$ system (12.1), and
2. the evaluation of the force vector $\boldsymbol{f}$ and its derivative with respect to the displacement vector $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{a}}$.

3. Model Order Reduction

Essentially, a Model Order Reduction technique decomposes a many-query procedure into two parts: a computationally expensive offline phase, and a computationally efficient online phase. In the offline phase, a set of samples is collected from a standard analysis (in this context, this is done using FEM). This information is employed to construct a reduced order model. In the online phase, the reduced order model is used to carry out the many-query procedure. The MOR technique employed in this study is summarized in Section 3.5 while the corresponding online/offline procedures are detailed next.

The POD-G [5] is employed to project the nonlinear system of equations (10) onto a lower-dimensional subspace. DEIM [5] is then used to approximate the force vector and its derivative with respect to the displacement vector. In practice, the Proper Orthogonal Decomposition [24] is applied to a set of samples collected from a full order FEM analysis to compute a set of basis vectors. These basis vectors are later used in the POD-G and the DEIM procedures. Finally, the Localized Discrete Empirical Interpolation Method (LDEIM) [22] is employed to overcome a shortcoming of DEIM that occurs when the collected samples of the force vector are not attracted to a sufficiently lower-dimensional subspace. At variance with [22], we propose (i) a modified distance definition in the k-means clustering algorithm that is tailored for DEIM, (ii) a modified distance metric that is used in the k-means clustering algorithm and the nearest neighbor classifier, and (iii) a trivial initial clustering in the k-means clustering algorithm that exploits the path-dependent characteristic of the problem.

3.1. Proper Orthogonal Decomposition-Galerkin

3.1.1. Galerkin projection

To begin with, we attempt to project the nonlinear system of equations (10) onto a lower-dimensional subspace. Suppose that the solution of (10) is attracted to a lower-dimensional subspace $\mathscr{Y} \subset \mathbb{R}^N$. This solution can then be approximated by a weighted linear combination of the set of orthonormal basis vectors $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k\}$ $(k \ll N)$ spanning $\mathscr{Y}$ as in

$$\boldsymbol{a} \approx \boldsymbol{V}\,\tilde{\boldsymbol{a}}, \tag{15}$$

where $\tilde{\boldsymbol{a}} \in \mathbb{R}^k$ is a set of scalar weights associated to the set of basis vectors $\boldsymbol{V} \in \mathbb{R}^{N\times k}$. Inserting (15) into (10) yields an overdetermined nonlinear system of N equations in k unknowns. To compute an approximate value for $\tilde{\boldsymbol{a}}$, we construct an equivalent system of equations $\boldsymbol{g} = \boldsymbol{0}$ that is determined. To find such a system, we search for a set of k functions $\boldsymbol{g}$ in subspace $\mathscr{Y}$ such that, for any given value of $\tilde{\boldsymbol{a}}$, the L_2 norm of the difference between the residual $\boldsymbol{r}(\boldsymbol{V}\tilde{\boldsymbol{a}})$ and its counterpart in the lower-dimensional subspace $\boldsymbol{V}\boldsymbol{g}$ is minimized according to [25, Theorem 3.2]

$$\underset{\boldsymbol{g}}{\arg\min}\ \parallel \boldsymbol{r}(\boldsymbol{V}\tilde{\boldsymbol{a}}) - \boldsymbol{V}\boldsymbol{g} \parallel_2. \tag{16}$$

Owing to the choice of the L_2 norm, it is straightforward to solve (16) for $\boldsymbol{g}$. Consequently, the equivalent system of equations $\boldsymbol{g} = \boldsymbol{0}$ becomes

$$\boldsymbol{V}^{\mathrm{T}}\, \boldsymbol{r}(\boldsymbol{V}\tilde{\boldsymbol{a}}) = \boldsymbol{0}. \tag{17}$$

The system of equations (17) is then solved using a Newton–Raphson scheme according to which

$$\begin{aligned} \tilde{\boldsymbol{K}}^j\, \mathrm{d}\tilde{\boldsymbol{a}}^{j+1} &= -\tilde{\boldsymbol{r}}^j, \qquad \text{and} \\ \tilde{\boldsymbol{a}}^{j+1} &= \tilde{\boldsymbol{a}}^j + \mathrm{d}\tilde{\boldsymbol{a}}^{j+1}, \end{aligned} \tag{18}$$

where $\tilde{\boldsymbol{K}}^j = \boldsymbol{V}^{\mathrm{T}}\boldsymbol{K}^j\, \boldsymbol{V} \in \mathbb{R}^{k\times k}$ is the reduced stiffness matrix and $\tilde{\boldsymbol{r}}^j = \boldsymbol{V}^{\mathrm{T}}\boldsymbol{r}^j \in \mathbb{R}^k$ is the reduced residual vector. The set of basis vectors $\boldsymbol{V}$ is computed using the Proper Orthogonal Decomposition as explained in Section 3.1.2.

In spite of the system of equations (18) being of dimension $k \ll N$, the complexity of problem still depends on N when computing the nonlinear contributions. Efficient reduction is achieved later in Sections 3.2 and 3.3 by approximating the force vector and its derivative with respect to the displacement vector using DEIM.

3.1.2. Proper Orthogonal Decomposition

A set of orthonormal basis vectors $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k\}$ is used to project the system of equations (10) onto a lower-dimensional subspace as in (17). This set is computed by means of the Proper Orthogonal Decomposition (POD) method [24,26]. For this purpose, samples of the displacement and the force vectors are collected from a standard full order FEM analysis and gathered in the sample matrices $\boldsymbol{A}^{\mathrm{samp}} = [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_{n_s}]$ and $\boldsymbol{F}^{\mathrm{samp}} = [\boldsymbol{f}_1, \ldots, \boldsymbol{f}_{n_s}]$, respectively. For the displacement samples (and similarly for the force samples), we search for a subspace $\mathscr{Y} = \mathrm{span}\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k\} \subset \mathbb{R}^N$ in which the samples can be optimally described. In other words, it is preferable that the error between samples and their projection on $\mathscr{Y}$ is minimized in a certain sense. Formally, the set of orthonormal basis vectors spanning $\mathscr{Y}$ is computed by solving the optimization problem [5]

$$\begin{aligned} &\underset{\{\boldsymbol{v}_1,\ldots,\boldsymbol{v}_k\}}{\arg\min} \sum_{i=1}^{n_s} \left\| \boldsymbol{a}_i - \sum_{j=1}^{k} (\boldsymbol{v}_j^{\mathrm{T}}\boldsymbol{a}_i)\, \boldsymbol{v}_j \right\|_2^2, \\ &\text{subjected to} \quad \boldsymbol{v}_m^{\mathrm{T}}\, \boldsymbol{v}_n = \begin{cases} 1 & m = n \\ 0 & m \neq n \end{cases}, \quad m, n = 1, \ldots, k. \end{aligned} \tag{19}$$

In the equation above, the error between a sample $\boldsymbol{a}_i$ and its projection $\sum_{j=1}^{k}(\boldsymbol{v}_j^{\mathrm{T}}\boldsymbol{a}_i)\, \boldsymbol{v}_j$ onto the subspace $\mathscr{Y}$ is minimized in an L_2 sense. Note that the norms in (19) are squared to facilitate the derivation of (21). It can be shown [26] that the solution to (19) is a set of left singular vectors $\boldsymbol{V}$ obtained from the application of the Singular Value Decomposition (SVD) to the sample matrix $\boldsymbol{A}^{\mathrm{samp}}$ such that

$$\boldsymbol{V}\, \boldsymbol{\Sigma}\, \boldsymbol{W}^{\mathrm{T}} = \boldsymbol{A}^{\mathrm{samp}}, \tag{20}$$

where $\boldsymbol{V} \in \mathbb{R}^{N\times N}$ is a set of left orthonormal singular basis vectors, $\boldsymbol{W}^{n_s\times n_s}$ is a set of right orthonormal singular basis vectors, and $\boldsymbol{\Sigma}$ is a diagonal matrix that contains the singular values in a descending manner $\sigma_1 > \cdots > \sigma_r$, $r =$ rank $(\boldsymbol{A}^{\mathrm{samp}})$. Therefore, by using an SVD [26, Theorem 1.1.1], the residual of (19) becomes [5]

$$\sum_{i=1}^{n_s} \left\| \boldsymbol{a}_i - \sum_{j=1}^{k} (\boldsymbol{v}_j^{\mathrm{T}}\boldsymbol{a}_i)\, \boldsymbol{v}_j \right\|_2^2 = \sum_{i=k+1}^{r} \sigma_i^2. \tag{21}$$

From the equation above, it is possible to infer that the projection error scales to the sum of the squares of the neglected ($i = k+1$ to r) singular values (RHS of (21)). As a result of this observation, a heuristic criterion to select the number of basis vectors k would be to consider the ratio of the sum of the squares of the neglected singular values to the sum of all the singular values as in

$$\frac{\sum\limits_{i=k+1}^{r} \sigma_i^2}{\sum\limits_{i=1}^{r} \sigma_i^2} \leq \epsilon \tag{22}$$

where $\epsilon \in [0\ 1]$ is a user-defined tolerance. As the value of ϵ deviates from one and tends to zero, the number of neglected singular values decreases. Simultaneously, the number of basis vectors which are taken into account increases, yielding a more accurate reduced order model. The POD procedure is summarized in Algorithm 1.

```
Algorithm 1: POD procedure
  input  : A set of sample $\boldsymbol{A}^{\text{samp}}$, a tolerance $\epsilon$
  output: A set of truncated basis vectors $\boldsymbol{V} \in \mathbb{R}^{N \times k}$

    Apply Singular Value Decomposition: $\boldsymbol{A}^{\text{samp}} = \boldsymbol{V}\,\boldsymbol{\Sigma}\,\boldsymbol{W}^{\text{T}}$
    Truncate $\boldsymbol{V}$ based on $\boldsymbol{\Sigma}$ and criterion (22)
```

3.2. Discrete empirical interpolation method

We now tackle the computational bottleneck related to the complexity of the evaluation of the force vector and its derivative with respect to the displacement vector. This is done by employing DEIM [5]. Suppose that the force vector $\boldsymbol{f}$ in (13) is attracted to a lower-dimensional subspace $\mathscr{U} = \text{span}\{\boldsymbol{u}_1, \dots, \boldsymbol{u}_m\} \subset \mathbb{R}^N$, where $m \ll N$. The force vector can then be approximated using a weighted linear combination of the basis vectors spanning $\mathscr{U}$ according to

$$\boldsymbol{f} \approx \hat{\boldsymbol{f}} = \boldsymbol{U}\,\boldsymbol{c}, \tag{23}$$

where $\boldsymbol{c} \in \mathbb{R}^m$ is a set of scalar weights associated to the set of basis vectors $\boldsymbol{U}$. To find these scalar weights, m rows of both sides of (23) are selected by means of a matrix $\boldsymbol{P} \in \mathbb{R}^{N \times m}$:

$$\boldsymbol{P}^{\text{T}}\boldsymbol{f} \approx \boldsymbol{P}^{\text{T}}\boldsymbol{U}\,\boldsymbol{c}. \tag{24}$$

Solving (24) for $\boldsymbol{c}$ and inserting it in (23) yield the approximated force vector

$$\hat{\boldsymbol{f}} = \boldsymbol{U}\,(\,\boldsymbol{P}^{\text{T}}\boldsymbol{U}\,)^{-1}\boldsymbol{P}^{\text{T}}\boldsymbol{f} = \boldsymbol{U}\,(\,\boldsymbol{P}^{\text{T}}\boldsymbol{U}\,)^{-1}\,\boldsymbol{f}_{\text{DEIM}}. \tag{25}$$

Here, $\boldsymbol{f}_{\text{DEIM}} = \boldsymbol{P}^{\text{T}}\boldsymbol{f} \in \mathbb{R}^m$ contains m entries of the force vector, and the matrix $\boldsymbol{P}$ is defined with the aid of the DEIM algorithm listed in the next subsection. The complexity of the evaluation of $\boldsymbol{f}$ is now reduced to that of $\hat{\boldsymbol{f}}$; thus, this consists of the evaluation of $\boldsymbol{f}_{\text{DEIM}}$ and a matrix multiplication (note that the computation of $\boldsymbol{U}\,(\,\boldsymbol{P}^{\text{T}}\boldsymbol{U}\,)^{-1}$ is carried out in the offline phase).

To compute the derivative of the approximated force vector (25) with respect to the displacement vector, several methods are suggested in [27]. Here we choose to utilize the same set of basis vectors previously computed for the force vector, as originally suggested in [5]:

$$\frac{\partial \hat{\boldsymbol{f}}}{\partial \boldsymbol{V}\tilde{\boldsymbol{a}}} = \boldsymbol{U}\,(\,\boldsymbol{P}^{\text{T}}\boldsymbol{U}\,)^{-1}\boldsymbol{P}^{\text{T}}\,\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{V}\tilde{\boldsymbol{a}}} = \boldsymbol{U}\,(\,\boldsymbol{P}^{\text{T}}\boldsymbol{U}\,)^{-1}\,\boldsymbol{K}_{\text{DEIM}}, \tag{26}$$

where $\boldsymbol{K}_{\text{DEIM}} = \boldsymbol{P}^{\text{T}}\,\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{V}\tilde{\boldsymbol{a}}} \in \mathbb{R}^{m \times N}$ contains m rows of the nonlinear part of the stiffness matrix. The procedure to assemble $\boldsymbol{f}_{\text{DEIM}}$ and $\boldsymbol{K}_{\text{DEIM}}$ is detailed in Algorithm 2.

3.2.1. Discrete empirical interpolation method algorithm

The error bound for the DEIM approximation $\hat{\boldsymbol{f}}$ of the force vector $\boldsymbol{f}$ is given by [5]

$$\|\boldsymbol{f} - \hat{\boldsymbol{f}}\|_2 \ \leq\ \|(\boldsymbol{P}^{\text{T}}\boldsymbol{U})^{-1}\|_2\ \ \|(\boldsymbol{I} - \boldsymbol{U}\,\boldsymbol{U}^{\text{T}})\boldsymbol{f}\|_2. \tag{27}$$

To increase the accuracy of the DEIM approximation, the value of $\|(\boldsymbol{I} - \boldsymbol{U}\,\boldsymbol{U}^{\text{T}})\boldsymbol{f}\|_2$ should be decreased while the value of $\|(\boldsymbol{P}^{\text{T}}\boldsymbol{U})^{-1}\|_2$ is kept bounded [5, Lemma 3.2].

The quantity $\|(\boldsymbol{I} - \boldsymbol{U}\,\boldsymbol{U}^{\text{T}})\boldsymbol{f}\|_2$ is the projection error induced by POD (21). Therefore, by increasing the number of basis vectors in $\boldsymbol{U}$, according to (22), its value becomes smaller. However, by increasing the number of basis vectors, the value of $\|(\boldsymbol{P}^{\text{T}}\boldsymbol{U})^{-1}\|_2$ becomes larger, jeopardizing the accuracy of the DEIM approximation. According to Chaturantabut and Sorensen [5, Lemma 3.2], the DEIM Algorithm 3 computes the matrix $\boldsymbol{P}$ in a way that the growth

```
Algorithm 2: Assembly of $\boldsymbol{K}_{\mathrm{DEIM}}$ and $\boldsymbol{f}_{\mathrm{DEIM}}$
  input  : A set $\boldsymbol{IND} = [IND_1, \ldots, IND_m]$ of entries selected by the DEIM Algorithm 3
  output: $\boldsymbol{K}_{\mathrm{DEIM}}$ and $\boldsymbol{f}_{\mathrm{DEIM}}$

    Set $\boldsymbol{K}_{\mathrm{DEIM}}$ and $\boldsymbol{f}_{\mathrm{DEIM}}$ to zero
    for $i = 1, \ldots, m$ do
        Find elements $\boldsymbol{E} = \{E_1, \ldots, E_{n_e}\}$ sharing the $IND_i$ entry
        for $e = 1 \ \ldots \ n_e$ do
            Compute the nonlinear part of element stiffness matrix and the force vector: $\boldsymbol{K}_e^{\mathrm{E}}$ and $\boldsymbol{f}_e^{\mathrm{E}}$
            Find the local DOF $j$ associated to the entry $IND_i$
            $\boldsymbol{f}_{\mathrm{DEIM}}(i) = \boldsymbol{f}_{\mathrm{DEIM}}(i) + \boldsymbol{f}_e^{\mathrm{E}}(j)$
            Define $\boldsymbol{gIndex}$ as the vector of global DOFs associated with element $e$
            $\boldsymbol{K}_{\mathrm{DEIM}}(i,\ \boldsymbol{gIndex}) = \boldsymbol{K}_{\mathrm{DEIM}}(i,\ \boldsymbol{gIndex}) + \boldsymbol{K}_e^{\mathrm{E}}(j,\ \ :)$
        end
    end
```

of this term is bounded. Consequently, the accuracy of the approximation can be improved by taking more and more basis vectors into account.

```
Algorithm 3: DEIM algorithm
  input  :  A set of basis vectors $\boldsymbol{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m]$
  output:  Entries of the force vector selected by DEIM $\boldsymbol{IND} = [IND_1, \ldots, IND_m]$, and
           the matrix $\boldsymbol{P} = [\boldsymbol{e}_{IND_1}, \ldots, \boldsymbol{e}_{IND_m}]$

    Set $IND_1 =$ index of the maximum entry of $|\boldsymbol{u}_1|$
    Initiate matrices: $\boldsymbol{W} = [\boldsymbol{u}_1],\ \boldsymbol{P} = [\boldsymbol{e}_{IND_1}],\ \boldsymbol{IND} = [IND_1]$
    for $i = 2, \ldots, m$ do
        Solve $\boldsymbol{P}^{\mathrm{T}} \boldsymbol{W}\, \boldsymbol{c} = \boldsymbol{P}^{\mathrm{T}} \boldsymbol{u}_i$ for $\boldsymbol{c}$
        Compute residual $\boldsymbol{r} = \boldsymbol{u}_i - \boldsymbol{W}\, \boldsymbol{c}$
        Set $IND_i =$ index of the maximum component of $|\boldsymbol{r}|$
        Augment $\boldsymbol{W} \leftarrow [\boldsymbol{W}\ ,\ \boldsymbol{u}_i]$ , $\boldsymbol{P} \leftarrow [\boldsymbol{P}\ ,\ \boldsymbol{e}_{IND_i}]$ and $\boldsymbol{IND} \leftarrow [\boldsymbol{IND}\ ,\ IND_i]$
    end
```

In Algorithm 3, $\boldsymbol{e}_i$ is the ith column of an identity matrix. The notation $\boldsymbol{M} \leftarrow [\boldsymbol{M}\ ,\ \boldsymbol{v}_i]$ indicates that the matrix $\boldsymbol{M}$ is augmented by adding the column vector $\boldsymbol{v}_i$.

3.3. Localized Discrete Empirical Interpolation Method

DEIM succeeds in defining an efficient reduced order model, as in [5], only if the singular values of the force vector decay sufficiently fast—this effectively means that, when a relatively small number of basis vectors are considered, the sum of the squares of the singular values corresponding to the neglected basis vectors becomes negligibly small. However, in the context of a viscoplasticity constitutive model this assumption is, apparently, not valid. As it will be shown in Section 4.2, a considerably large number of basis vectors are required to sufficiently reduce the projection error. Hence, an efficient reduced order model cannot be constructed merely by applying the standard version of DEIM.

We employ a Localized DEIM [22] to remedy the above shortcoming. The general idea is to group samples, which are close to each other in a certain sense, into clusters. We show that the decay of the singular values becomes significantly faster for each cluster, implying that the samples are attracted to a considerably lower-dimensional subspace. In this work we only investigate the performance of DEIM in conjunction with the clustered force samples. Nonetheless, the displacement samples could also be clustered as suggested in [27]. For the purpose of clustering the samples, we make use of machine learning techniques such as the k-means clustering algorithm (for clustering samples in the offline phase), and the nearest neighbor classifier (for finding the correct cluster in the online phase).

3.3.1. Modified k-means clustering algorithm

We now describe a procedure to divide a set of samples $\boldsymbol{F}^{\text{samp}}$, schematically shown in Fig. 1(a), into n_c clusters with the intention that samples in each cluster are close to each other according to a certain distance definition. For this purpose, a k-means clustering algorithm, informally explained in [28], is used. In this contribution the k-means clustering algorithm is tailored for DEIM and the path-dependent nature of the problem. More specifically, we will improve on the cluster initiation, the distance definition, and the distance metric.

The strong path-dependency of the problem under consideration leads to a rather straightforward choice of the initial guess for the clustering. In other words, since force vectors from a certain part of the loading history are already "close" to each other, the samples are simply divided into n_c clusters without any permutation as an initial guess for the k-means clustering algorithm.

In contrast to the definition of the distance in the standard k-means clustering algorithm, which is the distance between a sample and the center of clusters, the distance is here defined between a sample and its DEIM approximation. For instance, consider a sample $\boldsymbol{f}_i$ and, for each cluster $c = 1 \ \ldots \ n_c$, a set of basis vectors ${}_c\boldsymbol{U}$ and a matrix ${}_c\boldsymbol{P}$ (from here onwards, the number of clusters is indicated by a subscript on the left-hand side of the corresponding quantity). The distance measure used in the k-means clustering algorithm is defined by the difference between $\boldsymbol{f}_i$ and its DEIM approximation ${}_c\hat{\boldsymbol{f}}_i = {}_c\boldsymbol{A} \ {}_c\boldsymbol{P}^{\text{T}}\boldsymbol{f}_i$ for each cluster $c = 1 \ \ldots \ n_c$, where ${}_c\boldsymbol{A} = {}_c\boldsymbol{U} \ ({}_c\boldsymbol{P}^{\text{T}} {}_c\boldsymbol{U})^{-1}$. In the standard k-means clustering algorithm, it is typical to consider a Euclidean distance metric. Shortcomings of the Euclidean distance are further discussed in [29]. Here, a stricter metric, based on the Euclidean distance and the cosine similarity, is defined:

$$d = \sqrt{d^2_{\text{Euclidean}} + d^2_{\text{cosine}}}, \tag{28}$$

where

$$d_{\text{Euclidean}} = \|\boldsymbol{f}_i - {}_c\hat{\boldsymbol{f}}_i\|_2 \ / \|\boldsymbol{f}_i\|_2 \quad \text{and} \quad d_{\text{cosine}} = 1 - |\boldsymbol{f}_i^T \ {}_c\hat{\boldsymbol{f}}_i| \ /(\|\boldsymbol{f}_i\|_2 \ \|{}_c\hat{\boldsymbol{f}}_i\|_2). \tag{29}$$

Since $\|\boldsymbol{f}_i - {}_c\hat{\boldsymbol{f}}_i\|_2 \ < \|\boldsymbol{f}_i\|_2$, these two measures are comparable in terms of magnitude; values of $d_{\text{Euclidean}}$ and d_{cosine} fall between zeros and one, and it is therefore safe to take their norm as in (28).

The proposed k-means clustering algorithm is detailed in Algorithm 4. The procedure initiates by dividing samples in $\boldsymbol{F}^{\text{samp}}$ into n_c clusters, as schematically shown in Fig. 1(b), without permuting them. Additionally, a mapping (***sampleToClus*** : sample → cluster number) is initiated to keep track of samples and their corresponding cluster number. In the k-means iterative scheme, the POD (Algorithm 1) and the DEIM (Algorithm 3) procedures are applied to each one of the clusters, and a reduced set of basis vectors ${}_c\boldsymbol{U}$, a matrix ${}_c\boldsymbol{P}$, and a set of DEIM selected entries ${}_c\boldsymbol{IND}$ are computed. The distance between each sample $\boldsymbol{f}_i$ and its DEIM approximation ${}_c\hat{\boldsymbol{f}}_i = {}_c\boldsymbol{A} \ {}_c\boldsymbol{P}^{\text{T}}\boldsymbol{f}_i$ (28) is saved in a matrix $\boldsymbol{G} \in \mathbb{N}^{n_s \times n_c}$ (n_s being the number of samples).

Next, each sample $\boldsymbol{f}_i$ is checked to make sure that it belongs to its cluster. To do so, ${}_c\hat{\boldsymbol{f}}_i = {}_c\boldsymbol{A} \ {}_c\boldsymbol{P}^{\text{T}}\boldsymbol{f}_i$ is computed for each one of the clusters $c = 1 \ \ldots \ n_c$. Then, the distance between $\boldsymbol{f}_i$ and each one of the ${}_c\hat{\boldsymbol{f}}_i$ is measured by means of (28). The cluster that helps generating the most accurate DEIM approximation of $\boldsymbol{f}_i$, such that ${}_c\hat{\boldsymbol{f}}_i$ is closest to $\boldsymbol{f}_i$ as shown in Fig. 1(c), is the cluster $\boldsymbol{f}_i$ actually belongs to. To keep track of samples in the clusters, when a sample is designated to another cluster, the mapping ***sampleToClus*** is updated. The procedure continues until a maximum number of iterations is reached or all the samples are (sub)optimally clustered.

When an insufficient number of basis vectors is considered, the rate of convergence of the k-means clustering algorithm becomes very slow and the quality of the clusters diminishes. From the implementation point of view, to prevent division by zero in d_{cosine} and $d_{\text{Euclidean}}$ (29), it is vital to eliminate zero vectors from the matrix of force samples.

In order to check the validity of the proposed approach we can plot the singular values associated with each cluster and observe their rate of decay. If their rate of decay is not substantially faster than that of the singular values associated with all the samples, the number of clusters needs to be increased at the cost of a more expensive k-means clustering algorithm.

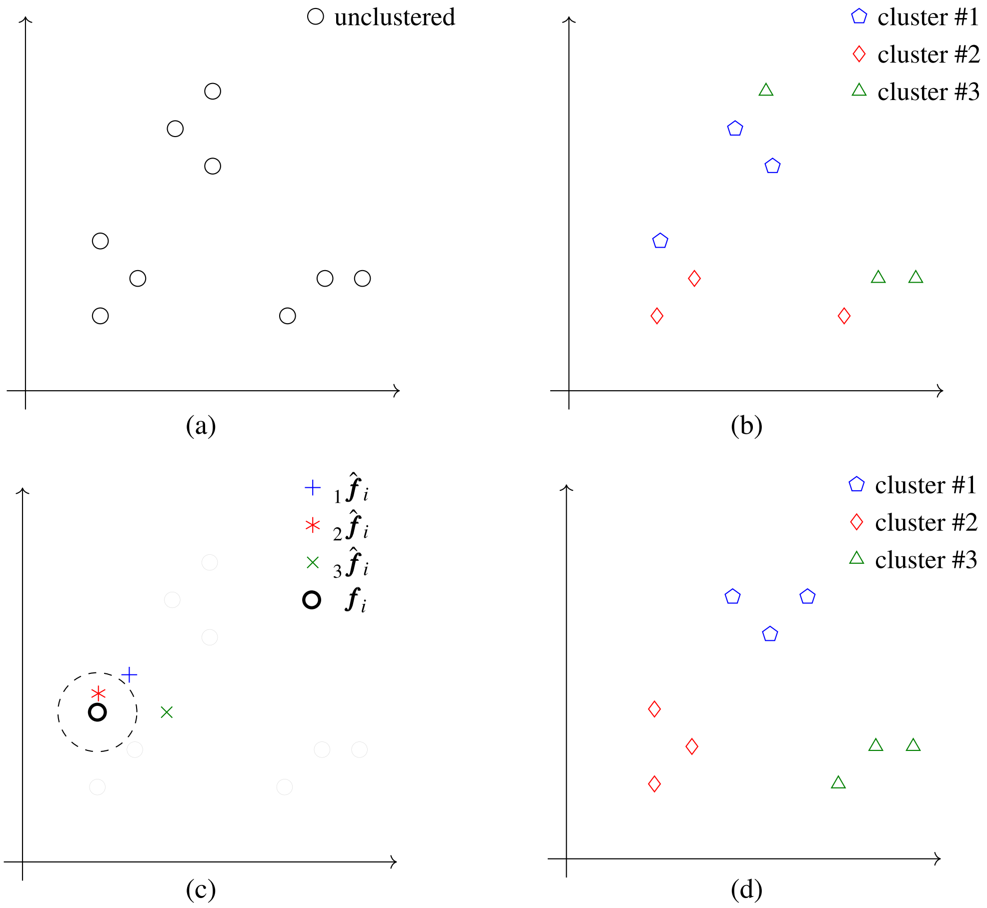


Fig. 1. Modified k-means clustering algorithm: (a) unclustered samples; (b) initial clustering of samples; (c) distance between a sample $\boldsymbol{f}_i$ and its DEIM approximation using $_c\boldsymbol{A}$ and $_c\boldsymbol{P}$ of all clusters, $c = 1 \ \ldots \ n_c$. The sample $\boldsymbol{f}_i$ belongs to the cluster that generates the closest $_c\hat{\boldsymbol{f}}_i = {}_c\boldsymbol{A}\ {}_c\boldsymbol{P}^{\mathrm{T}}\boldsymbol{f}_i$ in the sense of (28); (d) final form of clusters after convergence of the modified k-means clustering algorithm.

3.3.2. *Nearest neighbor classifier*

To find the cluster that best represents the current state of the system during the runtime, we employ a nearest neighbor classifier and modify its distance metric according to (28). At each time step, the nearest neighbor classifier computes distances (28) between the converged force vector $\hat{\boldsymbol{f}}$ and each one of the clustered samples $\boldsymbol{f}_i$, $i = 1 \ \ldots \ n_s$. The classifier then determines the sample with the least distance to the converged force vector. The cluster that contains this sample best describes the current state of the system.

To reduce the computational cost of the classification, the nearest neighbor classifier uses a low-dimensional representation of the force vector. In this context, we employ the DEIM-based feature extraction [22]

$$\hat{\boldsymbol{f}}_{\mathrm{FE}} = {}_{c_{\mathrm{old}}}\boldsymbol{P}^T\,\hat{\boldsymbol{f}} \quad \text{and} \quad \boldsymbol{f}_{\mathrm{FE},i} = {}_{c_{\mathrm{old}}}\boldsymbol{P}^T\,\boldsymbol{f}_i, \tag{30}$$

where $\hat{\boldsymbol{f}}_{\mathrm{FE}} \in \mathbb{R}^m$ and $\boldsymbol{f}_{\mathrm{FE},i} \in \mathbb{R}^m$ are a low-dimensional representation of $\hat{\boldsymbol{f}}$ and $\boldsymbol{f}_i$, respectively, and $_{c_{\mathrm{old}}}\boldsymbol{P} \in \mathbb{R}^{N\times m}$ is the DEIM matrix from the previous time step. Distance metric in (28) is computed as

$$d_{\mathrm{Euclidean}} = \|\boldsymbol{f}_{\mathrm{FE},i} - \hat{\boldsymbol{f}}_{\mathrm{FE}}\|_2 \,/\, \|\boldsymbol{f}_{\mathrm{FE},i}\|_2 \quad \text{and} \quad d_{\mathrm{cosine}} = 1 - |\boldsymbol{f}_{\mathrm{FE},i}^{\mathrm{T}}\ \hat{\boldsymbol{f}}_{\mathrm{FE}}| \,/\, (\|\boldsymbol{f}_{\mathrm{FE},i}\|_2\ \|\hat{\boldsymbol{f}}_{\mathrm{FE}}\|_2). \tag{31}$$

3.4. *Discussion on the computational cost*

The computational cost of DEIM and LDEIM is already discussed in [5,22]. For the sake of completeness, this section is dedicated to a brief review of the key points. Moreover, we also discuss some of our observations.

Algorithm 4: Modified k-means clustering algorithm

```
input  : A set of force vector samples $\boldsymbol{F}^{\mathrm{samp}} = [\boldsymbol{f}_1, \ldots, \boldsymbol{f}_{n_s}]$,
         total number of clusters $n_c$,
         maximum number of iterations $maxIter$
output: ${}_c\boldsymbol{A}$ , ${}_c\boldsymbol{P}$ , ${}_c\boldsymbol{IND}$ where $c = 1, \ldots, n_c$,
         the mapping $\boldsymbol{sampleToClus}$

  Cluster samples without sorting $\boldsymbol{F}^{\mathrm{samp}} = \{{}_1\boldsymbol{F}\ ,\ \ldots\ ,\ {}_{n_c}\boldsymbol{F}\}$
  Initiate $\boldsymbol{sampleToClus}$ (mapping from a sample to its corresponding cluster)
  Set $converged = 1$
  while $converged \neq 1$ and $maxIter$ not reached do
      for $c = 1 \ldots n_c$ do
          Compute a set of reduced basis vectors: ${}_c\boldsymbol{U} =$ POD $({}_c\boldsymbol{F})$        Algorithm 1
          Compute the DEIM entries: $[{}_c\boldsymbol{IND},\ {}_c\boldsymbol{P}] =$ DEIM $({}_c\boldsymbol{U})$        Algorithm 3
          Compute the DEIM coefficient matrix: ${}_c\boldsymbol{A} = {}_c\boldsymbol{U}\ ({}_c\boldsymbol{P}^T\ {}_c\boldsymbol{U})^{-1}$
          for $i = 1$ to $n_s$ do
              Compute the DEIM approximation: ${}_c\hat{\boldsymbol{f}}_i = {}_c\boldsymbol{A}\ {}_c\boldsymbol{P}^{\mathrm{T}}\ \boldsymbol{f}_i$
              Compute the distance $d$ and save it in $\boldsymbol{G}(i, c)$        Equation (28)
          end
      end
      for $i = 1$ to $n_s$ do
          Get the old cluster number: $c_{\mathrm{old}} = \boldsymbol{sampleToClus}(i)$
          Find the new cluster number: $c_{\mathrm{new}} =$ index of the minimum value of $\boldsymbol{G}(i, :)$
          if $c_{\mathrm{old}} \neq c_{\mathrm{new}}$ then
              $converged = 0$
              Update mapping and, consequently, clusters : $\boldsymbol{sampleToClus}(i) = c_{\mathrm{new}}$
          end
      end
      if $converged = 1$ then
          Save ${}_c\boldsymbol{A}$ , ${}_c\boldsymbol{P}$ , and ${}_c\boldsymbol{IND}$ for all $c = 1 \ldots n_c$
          Save the mapping $\boldsymbol{sampleToClus}$
      end
  end
```

Algorithm 5: Nearest neighbor classifier

```
input  : A set of force vector samples $\boldsymbol{F}^{\mathrm{samp}} = [\boldsymbol{f}_1, \ldots, \boldsymbol{f}_{n_s}]$,
         the $\boldsymbol{sampleToClus}$ (mapping from a sample to its corresponding cluster),
         the force vector $\hat{\boldsymbol{f}} \in \mathbb{R}^N$ at time step $t$,
         the current cluster $c_{\mathrm{old}}$
output: New cluster $c_{\mathrm{new}}$

  Extract feature of the force vector: $\hat{\boldsymbol{f}}_{\mathrm{FE}} = {}_{c_{\mathrm{old}}}\boldsymbol{P}^T\ \hat{\boldsymbol{f}}$
  for $i = 1 \ldots n_s$ do

      Extract feature of the sample: $\boldsymbol{f}_{\mathrm{FE},i} = {}_{c_{\mathrm{old}}}\boldsymbol{P}^T\ \boldsymbol{f}_i$
      Compute the distance $\boldsymbol{d}(i)$ between $\hat{\boldsymbol{f}}_{\mathrm{FE}}$ and $\boldsymbol{f}_{\mathrm{FE},i}$        Equations (28) and (31)
  end
  Find the index $i_{\mathrm{min}}$ of the minimum component of $\boldsymbol{d}$
  Find the new cluster $c_{\mathrm{new}} = \boldsymbol{sampleToClus}(i_{\mathrm{min}})$
```

- In spite of the additional computational cost of the clustering, the cost of the offline phase is still strongly dominated by the sample collection.
- For the online phase, the primary contributor to the computational cost is the nearest neighbor classifier, Algorithm 5. Note that the computational complexity of this procedure is independent of the number of dofs N, thanks to the feature extraction concept (30). However, it strongly depends on the total number of samples n_s. As n_s becomes larger, the nearest neighbor classification loses its efficiency. To tackle this problem, it is possible to either reduce the number of samples through Redundancy Reduction methods such as the QR-SVD technique [30,31], or perform the classification based on a set of surrogate quantities instead of samples. For instance, the centroid of the clusters computed in classical k-means clustering algorithm can be an adequate substitute. In this manner, the procedure becomes dependent on the number of clusters. However, hopefully, the number of clusters are always significantly smaller than the number of samples.
- If, as a result of the POD and the DEIM approximation, the consistency of the linearization of (17) is jeopardized, the number of iterations in the Newton–Raphson scheme in the reduced order model will increase. Nevertheless, the iteration counts could be kept the same by accepting a solution that is not fully converged. This claim could be conjectured as follows. Consider the error between the full order model and the outcome of the Newton–Raphson scheme in the reduced order model, $\boldsymbol{e} = \boldsymbol{a} - \boldsymbol{a}^{\text{NR}}$. By adding and subtracting the exact solution of (17), $\boldsymbol{a}^{\text{ROM}}$, to the right hand side of the error expression and taking the L_2 norm of both sides, an upper bound to the error becomes

$$\|\boldsymbol{e}\|_2 \leq \|\boldsymbol{a} - \boldsymbol{a}^{\text{ROM}}\|_2 + \|\boldsymbol{a}^{\text{ROM}} - \boldsymbol{a}^{\text{NR}}\|_2. \tag{32}$$

  From the equation above, it can be inferred that, regardless of how accurately (17) is solved, even at the limit where $\|\boldsymbol{a}^{\text{ROM}} - \boldsymbol{a}^{\text{NR}}\|_2$ becomes effectively zero, the error as a result of the MOR approximation $\|\boldsymbol{a} - \boldsymbol{a}^{\text{ROM}}\|_2$ remains untouched. As a result, the bound on the error is, at most, as tight as $\|\boldsymbol{a} - \boldsymbol{a}^{\text{ROM}}\|_2$ permits. Hence, a solution that is relatively accurate, considering the error as a result of MOR, but not fully converged in the Newton–Raphson scheme should be considered as acceptable. Nevertheless, in Section 4, we always consider enough basis vectors to ensure the consistency of the linearization and, consequently, quadratic convergence rate of the Newton–Raphson scheme.
- In theory, as the number of clusters becomes larger, the computer memory occupied by the basis vectors in the online phase grows dramatically. However, in practice, we experienced that by increasing the number of clusters, the number of basis vectors in each cluster becomes significantly smaller. As a result, the cumulative number of basis vectors does not increase dramatically, as shown in Section 4.3.

3.5. Summary of the proposed model order reduction technique

Next, a step-by-step guide for the proposed MOR technique is outlined. The offline phase (Algorithm 6) is run first and its output data are saved to a file. In the online phase (Algorithm 7), the data are read and the reduced order model is employed to run the simulation.

4. Applications

We demonstrate the proposed MOR technique by means of three typical solid mechanics problems. A specific set of parameters ($\eta = 10^{-5}\ \text{s}^{-1}$, $a = -1$, $b = 50$, $Y = 1\ \text{N/mm}^{-2}$, $E = 1000\ \text{N/mm}^{-2}$, $\nu = 0$) is considered unless stated otherwise.

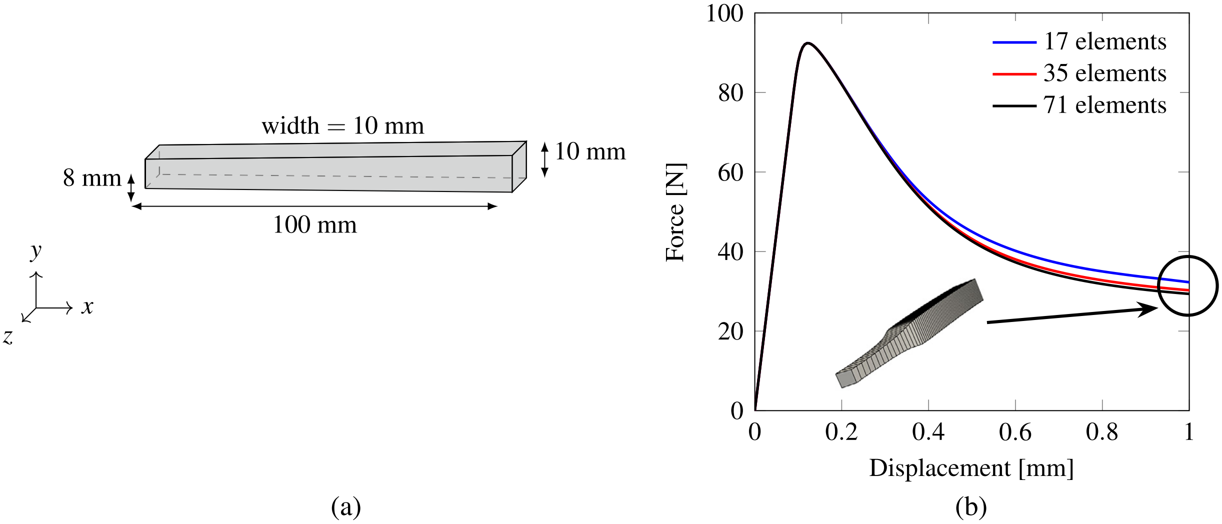
4.1. Tensile bar: One-dimensional strain-localization

The tapered bar in Fig. 2(a) is clamped at the left-hand side. The bar is subjected to a monotonic tensile loading. The final displacement of 1 mm is applied at the free end in 200 steps with a constant rate of 0.66×10^{-4} mm/s. Fig. 2(b) shows the results of a mesh refinement study performed with uniform subdivisions of the bar domain along the axis. The mesh refinement study identifies the solution related to the 71 eight-node regular hexahedral element mesh as the reference solution for the development of a reduced order model.

To test the MOR technique, we consider three different values of the b parameter ($b = [0, 20, 100]$). According to (8), large positive values of b result in a fast decay of σ_{Y} as κ grows and characterize the softening branch

Algorithm 6: Offline phase

```
1. Collect samples of the displacement and the force vectors in $\boldsymbol{A}^{\text{samp}}$ and $\boldsymbol{F}^{\text{samp}}$,
respectively, using a standard FEM analysis.
2. Compute a set of POD basis vectors for the displacement samples: $\boldsymbol{V} = \text{POD}(\boldsymbol{A}^{\text{samp}})$.    Section 3.1.2
3. Cluster force samples using the $k$-means clustering algorithm (Algorithm 4),    Section 3.3.1
and compute the converged values of the DEIM coefficient matrices ${}_c\boldsymbol{A}$,
the matrix ${}_c\boldsymbol{P}$, the DEIM selected entries ${}_c\boldsymbol{IND}$
for each cluster ($c = 1 \ \dots \ n_c$), and the mapping $\boldsymbol{sampleToClus}$.
In Algorithm 4 for $c = 1 \ \dots \ n_c$ :
    3.1. use the POD procedure (Algorithm 1): ${}_c\boldsymbol{U} = \text{POD}\ ({}_c\boldsymbol{F})$,    Section 3.1.2
    3.2. use the DEIM Algorithm 3: $[{}_c\boldsymbol{IND},\ {}_c\boldsymbol{P}] = \text{DEIM}({}_c\boldsymbol{U})$,    Section 3.2.1
    3.3. compute ${}_c\boldsymbol{A} = {}_c\boldsymbol{U}\ ({}_c\boldsymbol{P}^T\ {}_c\boldsymbol{U})^{-1}$, and
    3.4. update the mapping matrix $\boldsymbol{sampleToClus}$.
4. Save $\boldsymbol{V}$, $\boldsymbol{F}^{\text{samp}}$, $\boldsymbol{sampleToClus}$, and the converged values of ${}_c\boldsymbol{A}$, ${}_c\boldsymbol{P}$, ${}_c\boldsymbol{IND}$.
```



(a) (b)

Fig. 2. One-dimensional strain-localization problem: (a) the clamped end of the tapered bar has a smaller cross-section to trigger the initiation of plasticity and consequent strain-localization; (b) the regularizing properties of viscoplasticity are reflected in the mesh refinement study in terms of the resultant force at the clamped end plotted against the displacement at the free end.

of the load–displacement curve as shown in Fig. 2(b). A reference solution is computed for each value of b, and the corresponding displacement and force samples are collected in matrices $\boldsymbol{A}^{\text{samp}} = [\boldsymbol{a}_1 \ \dots \ \boldsymbol{a}_{200}]$ and $\boldsymbol{F}^{\text{samp}} = [\boldsymbol{f}_1 \ \dots \boldsymbol{f}_{200}]$.

A reduced order model is constructed using these samples for each value of the b parameter. The result of the online computation, depicted in Fig. 3, evidently shows the complexity reduction achieved by employing our MOR technique. For instance, for the steepest strain-softening curve, obtained with $b = 100$, the size of the system of equations is reduced, without apparent loss of accuracy, to 93 × 93 from 864 × 864 and, in the light of DEIM, evaluation of the force vector is required to be carried out at 57 entries rather than 864. A subdivision of the samples into three clusters allows to further reduce this number to 63-17-15 (the notation 63-17-15 indicates that we have

Algorithm 7: Online phase

```
1. Initiate standard FEM code and load:
    1.1. the set $\boldsymbol{V}$ of POD basis vectors associated with the displacement field,    Algorithm 6
    1.2. the sample matrix of the force vector $\boldsymbol{F}^{\mathrm{samp}}$ and    Algorithm 6
    the mapping $\boldsymbol{sampleToClus}$ between samples and clusters,
    1.3. the DEIM coefficient matrices ${}_{c}\boldsymbol{A}$, the matrix ${}_{c}\boldsymbol{P}$, and    Algorithm 6
    the DEIM selected entries ${}_{c}\boldsymbol{IND}$ for all clusters ($c = 1 \ \ldots \ n_c$).
2. Set the cluster $c_{\mathrm{new}} = 1$.
3. Break the time domain into steps $\rightarrow$ For each time step $t$ :
    3.1. execute the Newton–Raphson scheme $\rightarrow$ For each iteration $j$ :
        3.1.0. compute and save the linear part of the stiffness matrix $\boldsymbol{K}^{\mathrm{L}}$    Equation (14)
        in the first time step and the first iteration,
        3.1.1. compute $\hat{\boldsymbol{f}} = {}_{c_{\mathrm{new}}}\boldsymbol{A}\ \boldsymbol{f}_{\mathrm{DEIM}}({}_{c_{\mathrm{new}}}\boldsymbol{IND})$ and $\hat{\boldsymbol{K}} = {}_{c_{\mathrm{new}}}\boldsymbol{A}\ \boldsymbol{K}_{\mathrm{DEIM}}({}_{c_{\mathrm{new}}}\boldsymbol{IND})$    Section 3.2
        3.1.2. compute the reduced residual vector $\tilde{\boldsymbol{r}}^{j} = \boldsymbol{V}^{\mathrm{T}}\ (\boldsymbol{K}^{\mathrm{L}}\ \boldsymbol{a}^{j} + \hat{\boldsymbol{f}})$ and    Section 3.1.1
        the reduced stiffness matrix $\tilde{\boldsymbol{K}}^{j} = \boldsymbol{V}^{\mathrm{T}}\ (\boldsymbol{K}^{\mathrm{L}} + \hat{\boldsymbol{K}})\ \boldsymbol{V}$,    Section 3.1.1
        3.1.3. solve the reduced system of equations $\tilde{\boldsymbol{K}}^{j}\ \mathrm{d}\tilde{\boldsymbol{a}}^{j+1} = -\tilde{\boldsymbol{r}}^{j}$,    Equation (18.1)
        3.1.4. update the reduced solution: $\tilde{\boldsymbol{a}}^{j} = \tilde{\boldsymbol{a}}^{j} + \mathrm{d}\tilde{\boldsymbol{a}}^{j+1}$,    Equation (18.2)
        3.1.5. check convergence: $\tilde{\boldsymbol{r}}^{j} < tol$.
    3.2. Set the cluster $c_{\mathrm{old}} = c_{\mathrm{new}}$.
    3.3. Pass $c_{\mathrm{old}}, \hat{\boldsymbol{f}}, \boldsymbol{F}^{\mathrm{samp}}$, and $\boldsymbol{sampleToClus}$ to the nearest neighbor classifier    Section 3.3.2
    (Algorithm 5) and find the cluster $c_{\mathrm{new}}$ that best represents the current
    state of the problem.
    3.4. Go to the next time step.
```

considered three clusters and that the first cluster requires the evaluation of 63 entries, the second requires 17 entries, and the last only 15). For the other two curves, the size of the system of equations is reduced to 115 × 115 ($b = 20$) and 94 × 94 ($b = 0$). The evaluation of the force vector is carried out at 30 and 53 entries, respectively, when employing DEIM. With LDEIM and three clusters, it is possible to further reduce the number of entries to 32-11-8 and 51-22-13, respectively.

4.2. Wall under compression: Shear band formation

This example is dedicated to a wall under compression that shows a localization zone in the form of a shear band. The lower and the left-hand side edges of the wall depicted in Fig. 4(a) are constrained in the x and y directions. The wall is subjected to a monotonic compressive loading. The final displacement of 2 mm is applied at the upper edge in 400 steps with a constant rate of 1.33×10^{-4} mm/s. The parameter b in (8) is taken equal to 100.

A mesh refinement study has been carried out considering uniform grids with one element across the thickness. Fig. 4(b) shows the results of the mesh refinement study. The solution obtained with 196 (14 × 14) eight-node regular hexahedral elements is taken as the reference solution.

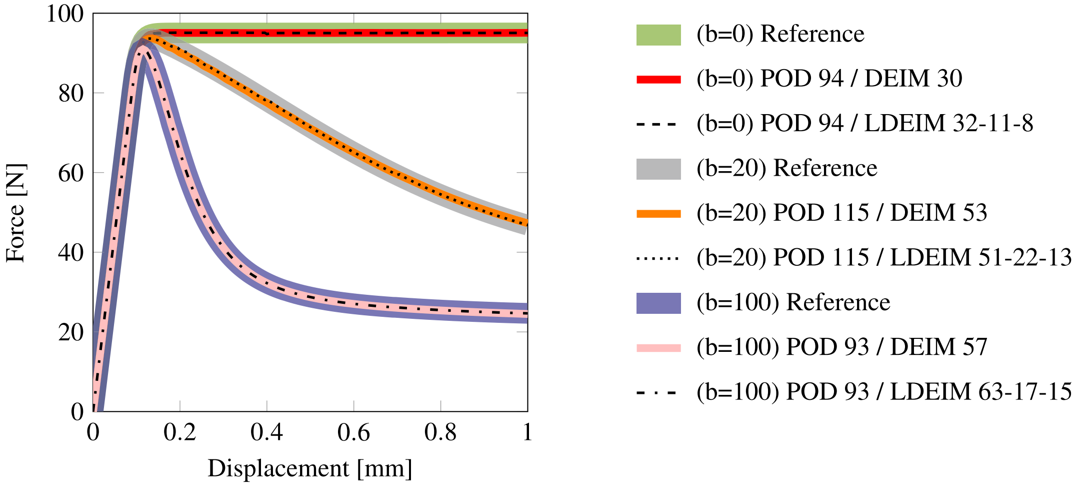


Fig. 3. The MOR technique applied to the one-dimensional strain-localization problem. The reference curves are obtained with the 71 eight-node regular hexahedral element mesh (864 DOFs) using different values of the b parameter (8).

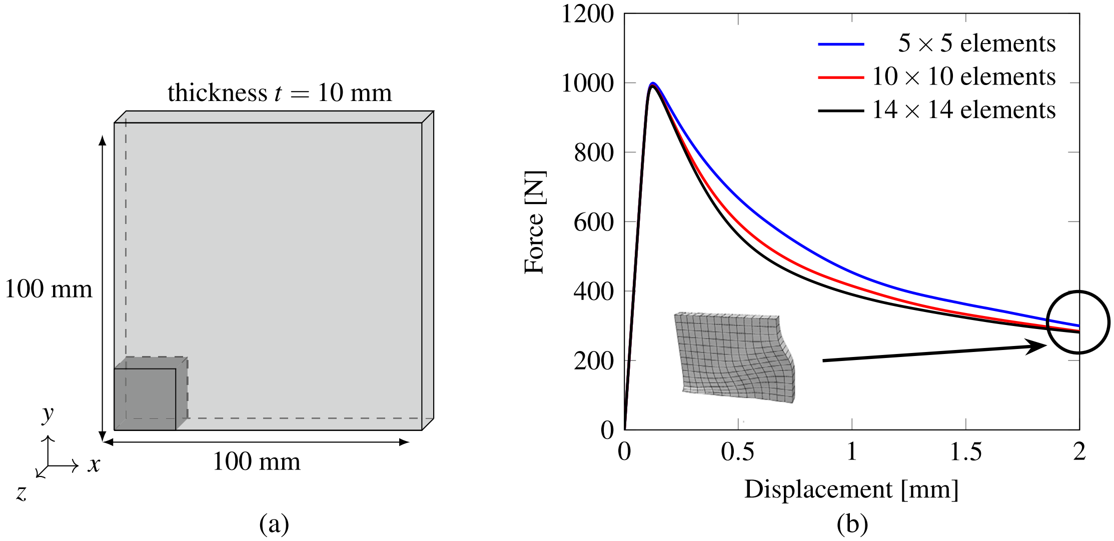


Fig. 4. Two-dimensional strain-localization problem: (a) geometry of the wall under compression with the dark shaded part indicating the weak region that localizes the initiation of plasticity and triggers a shear band; (b) the regularizing properties of viscoplasticity are reflected in the mesh refinement study in terms of the resultant of the force in the vertical direction on the lower edge against the value of the displacement of the upper edge; the inset shows the deformed wall with the shear band due to strain-localization.

Displacement and force samples are collected at each time step in matrices $\boldsymbol{A}^{\text{samp}} = [\boldsymbol{a}_1 \ \dots \ \boldsymbol{a}_{400}]$ and $\boldsymbol{F}^{\text{samp}} = [\boldsymbol{f}_1 \ \dots \ \boldsymbol{f}_{400}]$. Fig. 5(a) shows the convergence of DEIM to the reference solution by a basis addition study. The number of the basis vectors is chosen based on the sum of the square of the neglected singular values as explained in Section 3.1.2. It is possible to make an analogy between adding the basis vectors associated with the singular values shown in Fig. 5(b) for a basis addition study and increasing the number of elements for a mesh refinement study. By adding each subsequent basis vector, the squared value of its associated singular value (say σ_m) is deducted from the error defined in (21)—the error then becomes $\sum_{i=m+1}^{r} \sigma_i^2$ in which the contribution of σ_m^2 is missing thus yielding a smaller value.

Fig. 5(b) evidently shows the slow rate of decay of the singular values. As a result, a large number of basis vectors are required to decrease the projection error, that is the sum of the square of the neglected singular values, to an acceptable value and render the reduced order model sufficiently accurate. Fig. 5(a) shows that at least 88 DEIM

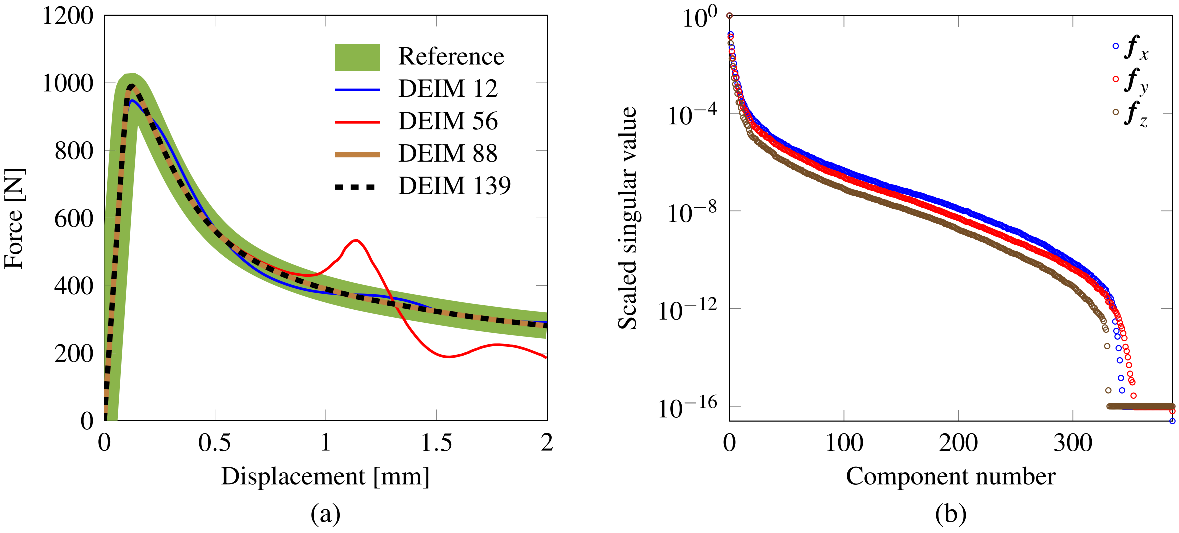


Fig. 5. Performance of the DEIM approximation in the two-dimensional strain-localization problem: (a) basis addition study (reference solution: 1350 DOFs, 196 elements; reduced order model: 227 POD basis vectors associated with the displacement field); (b) singular values of the force vector samples. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

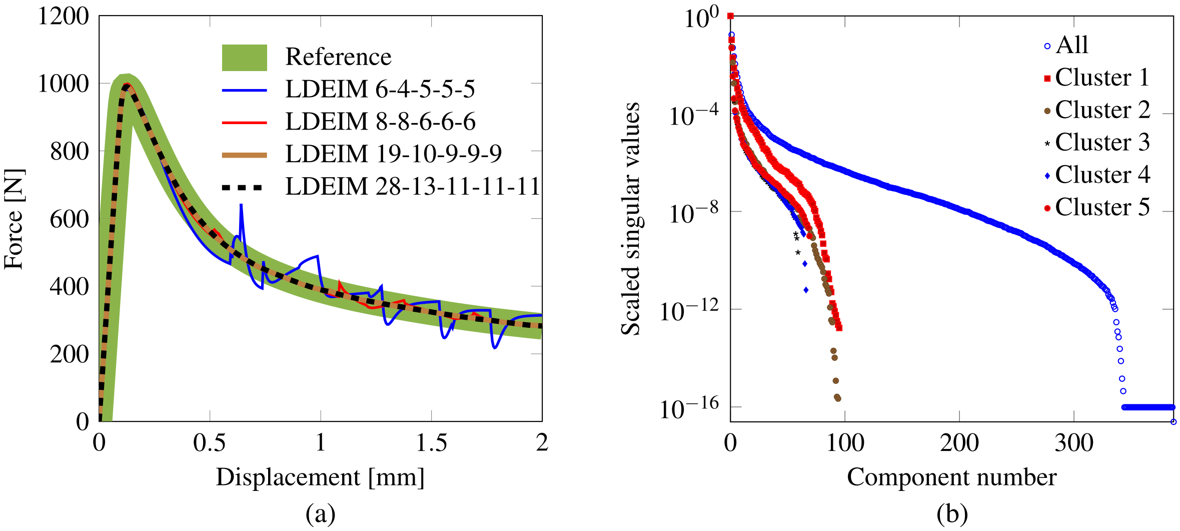


Fig. 6. Performance of the LDEIM approximation in the two-dimensional strain-localization problem: (a) basis addition study (reference solution: 1350 DOFs, 196 elements; reduced order model: 227 POD basis vectors associated with the displacement field); (b) singular values of the x component of the force vector samples using unclustered and clustered samples (those corresponding to the y and the z components follow the same trend). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

entries (brown curve) are required to construct a reduced order model that is capable of reproducing the reference solution (green curve). The black dashed line is to confirm that we have reached the converged solution. Checking the difference between results of two consecutive bases addition cases can be an adequate criterion to determine the convergence of a reduced order model. It is also noteworthy to report a case where the DEIM approximation results in an unstable reduced order model as shown by the red curve in Fig. 5(a). The source of this instability is, to the best of our knowledge, still unknown. The same type of behavior is reported in [31] and [32].

The force vector and the stiffness matrix associated with these 88 DOFs are computed by evaluating all the elements shown in Fig. 7(a), clearly a major subset of the elements in the mesh. As a result, significant reduction in computation cost cannot be achieved. To mitigate this issue, we cluster the samples of the force vector that are ‘close’ to each other

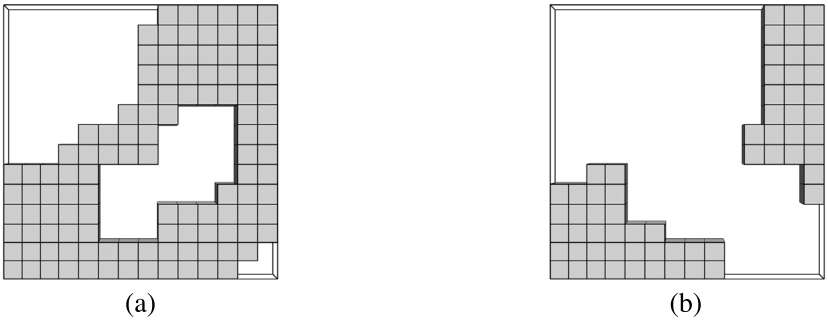


Fig. 7. Elements used in the online phase in the two-dimensional strain-localization problem: (a) DEIM–brown line in Fig. 5(a); (b) LDEIM–brown line in Fig. 6(a).

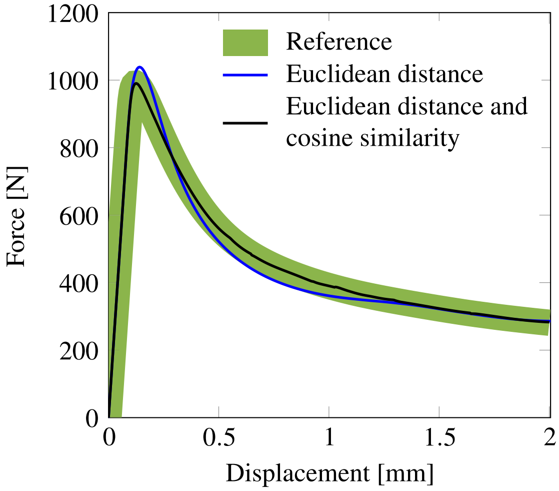


Fig. 8. The effect of the distance metric on the load–displacement curves of the two-dimensional strain-localization problem (reference solution: 1350 DOFs, 196 element; reduced order model: 227 POD basis vectors associated with the displacement field and LDEIM with 5 clusters and 19-10-9-9-9 DEIM entries in each cluster).

and compute, for each cluster, a set of reduced basis vectors and a set of DEIM selected entries as discussed in Section 3.3. As shown in Fig. 6(a), a basis addition study is carried out with 5 clusters. The reduced order model with 19-10-9-9-9 DEIM entries (brown curve) accurately approximates the reference solution (green curve). The singular values corresponding to the x component of the force vector samples are depicted in Fig. 6(b) for each cluster of the reduced order model (those corresponding to the y and the z components follow the same trend). It is clear that the singular values associated with each cluster decay significantly faster than those of all samples.

The elements used in this case are depicted in Fig. 7(b). Due to the path-dependent nature of the model, all elements depicted in this figure need to be updated for their history parameters. However, only a subset of them is used to compute the force vector and the stiffness matrix at each Newton–Raphson iteration. Fig. 7(a) and (b) show that the online phase of LDEIM is computationally cheaper than that of DEIM.

As discussed in Section 3.3.2, a nearest neighbor classifier is executed at each time step to find the cluster the current force vector belongs to. The classification is carried out based on the $m \ll N$ DEIM-selected entries of the force vector. Due to the non-periodic nature of the mechanical response in this class of problems, clusters do not get repeated during the run time. For instance, if the first cluster is selected at a time step close to the peak of the load–displacement curve, it will not be selected when the load–displacement curve reaches a plateau at the end. Cluster selection being automatic, this observation can be used to monitor the consistency of cluster switching. In other words, repetition of a cluster during the simulation is an indication that either the clustering procedure in the offline phase or the classification procedure in the online phase is inconsistent with the nature of the path-dependent problem. Additionally, this observation can be used to increase the efficiency of the nearest neighbor classifier, Algorithm 5.

Indeed, by neglecting samples associated with previously selected clusters, the classifier can compare the current state of the system to fewer number of samples.

To achieve consistency as explained in the previous paragraph, we modify the distance metric in the k-means clustering algorithm and the nearest neighbor classifier algorithms. Fig. 8 shows that the reduced order model based on the Euclidean distance and the cosine similarity metric (28) (black curve) properly reproduces the reference solution, whereas the reduced order model based on the standard Euclidean distance definition alone fails to do so. For the Euclidean distance case, the cluster number switches from 1 to 5 right before the peak load (note that cluster 5 is associated with the plateau part of the load–displacement curve). This is not consistent and, as a result, the blue curve deviates from the reference equilibrium path.

4.2.1. Parameter sensitivity analysis

We now demonstrate an application of the reduced order model to a simple parameter sensitivity analysis study. The goal is to construct a reduced order model response related to a set of parameters using samples obtained from two different parameter sets. To this end, we run the standard FEM model for $b = 80$ and $b = 120$ (the other parameters are defined at the beginning of Section 4). Samples are uniformly collected in matrices $\boldsymbol{A}_{b80}^{\text{samp}}$, $\boldsymbol{F}_{b80}^{\text{samp}}$, $\boldsymbol{A}_{b120}^{\text{samp}}$ and $\boldsymbol{F}_{b120}^{\text{samp}}$ at each time step. These samples are combined into two matrices $\boldsymbol{A}^{\text{samp}}$ and $\boldsymbol{F}^{\text{samp}}$. Since the k-means initial clustering uniformly divides samples into clusters, it is a good practice to combine $\boldsymbol{F}_{b80}^{\text{samp}}$ and $\boldsymbol{F}_{b120}^{\text{samp}}$ in a way that samples from the peak load region are collected in the initial columns of $\boldsymbol{F}^{\text{samp}}$ and samples from plateau region are collected in the last columns as described in Algorithm 8.

```
Algorithm 8: Construct $\boldsymbol{F}^{\text{samp}}$
    input  : Two sets of force vector samples for $b = 80$ and $b = 120$: $\boldsymbol{F}_{b80}^{\text{samp}} \in \mathbb{R}^{N\times n_s}$ and $\boldsymbol{F}_{b120}^{\text{samp}} \in \mathbb{R}^{N\times n_s}$
    output: A mixed sample matrix $\boldsymbol{F}^{\text{samp}} \in \mathbb{R}^{N\times 2n_s}$
    Initiate the mixed sample matrix: $\boldsymbol{F}^{\text{samp}} = [\boldsymbol{F}_{b80}(:,1)\ ,\ \boldsymbol{F}_{b120}(:,1)]$
    for $i = 2 \ldots n_s$ do
        Append $i$th columns of $\boldsymbol{F}_{b80}$ and $\boldsymbol{F}_{b120}$ to $\boldsymbol{F}^{\text{samp}}$
    end
```

A reduced order model is constructed using these samples. In Fig. 9(a), the reduced order model is employed to simulate the case with $b = 100$. To approximate the standard FEM solution, 85 DEIM entries are required when the samples are not clustered. However, when the samples are divided into 5 clusters, only 31-19-19-19-17 DEIM entries are required. It is also shown that fewer DEIM entries are needed to reach an acceptably accurate solution (black curve) when the number of clusters increases.

In Fig. 9(a) we observe fluctuations in the load–displacement curves of the reduced order models, starting from a displacement equal to 0.4 mm. A possible explanation could be related to the set of samples that underlies the construction of the basis vectors for the cases with $b = 80$ and $b = 120$. According to (8), the nonlinear parts of the stress related to these two cases, and the corresponding force vectors, differ slightly at the beginning when the plastic strain κ is small; hence, the reduced subspaces corresponding to the two force vectors are close to one another. However, as κ grows, the nonlinear part of the stress for $b = 80$ case decays slower than that for $b = 120$ and, as a result, their corresponding force vectors tend to deviate from one another. Consequently, the corresponding reduced subspaces these two cases belong to are no longer as close as they used to be. At this point, let us consider the case $b = 100$ with a reduced order model constructed using samples collected from the cases with $b = 80$ and $b = 120$. Fig. 9(a) supports the hypothesis just advanced. At the beginning, the load–displacement curves of the reduced order models are fairly close to that of the reference curve (green curve), suggesting that the force vectors for the $b = 100$ case are adequately approximated. Later on, however, small fluctuations in the load–displacement curves of the reduced order models become visible as κ grows.

To reduce these fluctuations, either the number of the basis vectors (and consequently the DEIM entries) in each cluster should be increased or more clusters should be taken into account. The first approach decreases the efficiency of the reduced order model and is not considered. The second approach increases the accuracy of the reduced order model and permits the use of the same or even smaller number of DEIM entries as shown in Fig. 9(a).

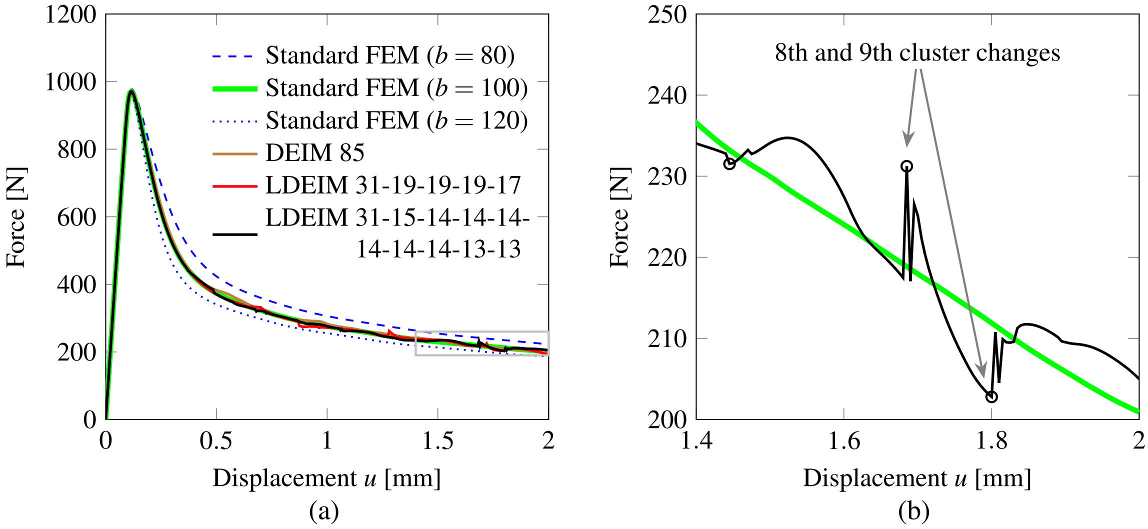


Fig. 9. Simple parameter sensitivity analysis for the two-dimensional strain-localization problem: (a) evaluation of the case with $b = 100$ using the reduced order model constructed from samples collected from reference cases with $b = 80$ and $b = 120$ (reduced order model: 325 POD basis vectors associated with the displacement field for DEIM and LDEIM); (b) enlargement of the boxed region in (a) with locations of cluster change marked.

To quantify the impact of the fluctuations in the load–displacement curve of the reduced order model with 10 clusters (black line), we consider the points that coincide with the 8th and the 9th cluster change as shown in Fig. 9(b). These two points correspond to the largest deviation from the reference solution (green curve) and are used to compare their corresponding displacement fields with that of the reference solution employing the relative error measurement

$$\boldsymbol{e} = \frac{|\boldsymbol{u}_{\text{FEM}} - \boldsymbol{u}_{\text{LDEIM}}|}{\|\boldsymbol{u}_{\text{FEM}}\|_2}, \tag{33}$$

where $|\boldsymbol{\Theta}|$ is the component-wise absolute value of $\boldsymbol{\Theta}$. Fig. 10 shows the deformed specimen and the relative error corresponding to these two points. It can be observed that the reduced order model is sufficiently accurate in the entire domain with maximum relative error of 3×10^{-4}.

It is noteworthy that we do not enforce clusters to change consecutively. However, we would expect them to change in this manner due to the special initial k-means clustering discussed in Section 3.3.1.

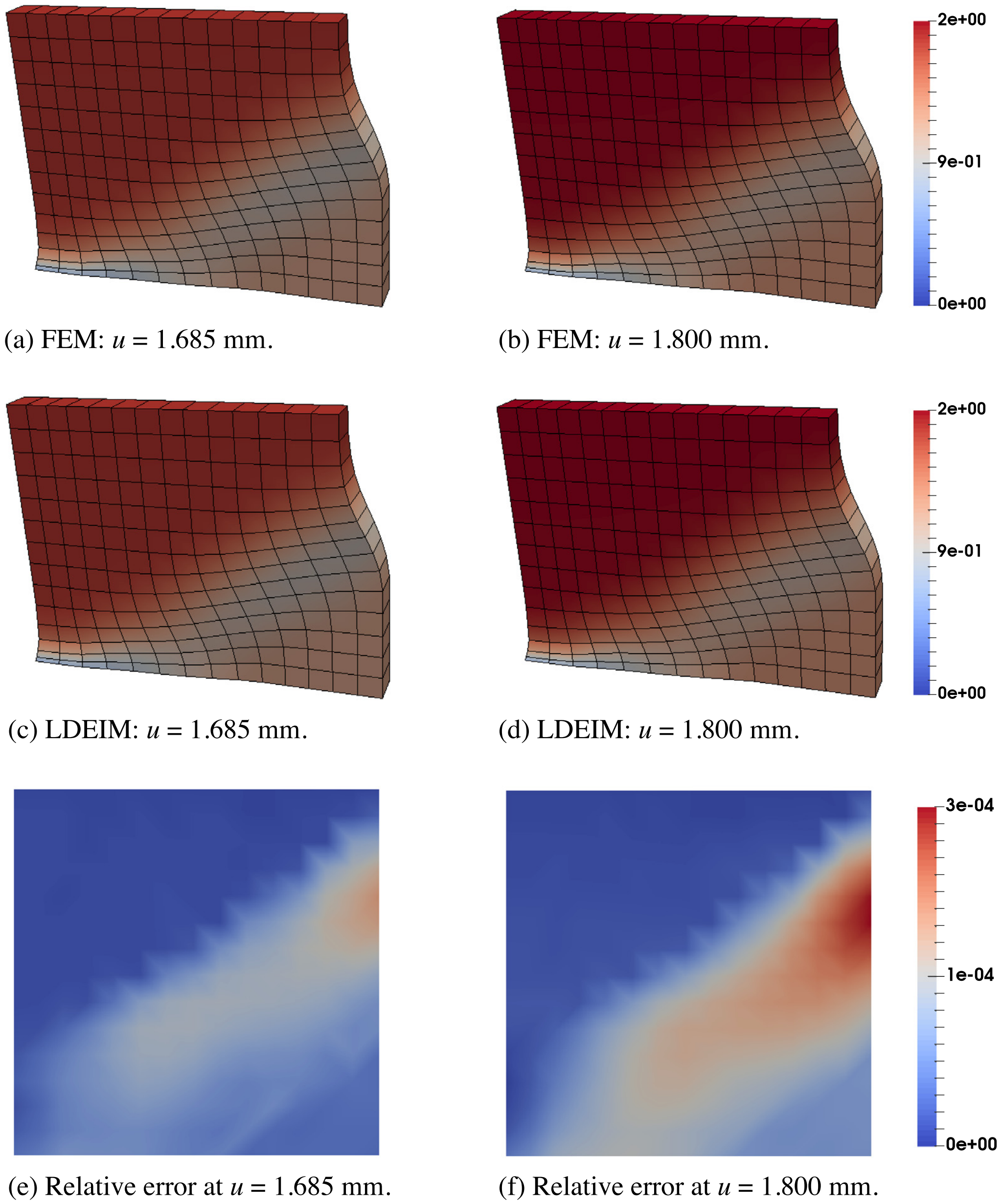
4.3. Three-dimensional square footing

The footing depicted in Fig. 11(a) is subjected to a monotonic compressive loading. The final compressive displacement of 1 mm is applied at the dark shaded plate on the upper surface in 1200 steps with a constant rate of 0.66×10^{-4} mm/s. The surfaces on the left-hand side, bottom, and the front are constrained from moving in the x, the z and the y direction, respectively.

A mesh refinement study is carried out and the result is depicted in Fig. 11(b). Load–displacement curves are associated with the resultant of the forces on the y plane at the bottom and the vertical displacement of the loading plate. Here, we choose the load–displacement curve obtained with 1331 eight-node hexahedral elements as the reference solution. The numerical solution, although not yet converged, is sufficiently accurate for the purpose of this study.

In total, 1200 samples were uniformly collected from the reference solution to construct a reduced order model. A large number of POD basis vectors (609), associated with the displacement field, is considered to sufficiently reduce the POD-G projection error. In this manner, it is possible to objectively study the performance of the DEIM approximation.

As shown in Fig. 12, at least 335 DEIM entries are required to almost reconstruct the reference solution. Note that the load–displacement curve of the reduced order model starts to deviate from the reference solution before reaching the final displacement value. Nevertheless, it is possible to considerably reduce the number of the DEIM entries by clustering samples. For instance, considering 20, 30, and 40 clusters at most, as depicted in Fig. 13, 117, 101, and



(a) FEM: u = 1.685 mm.

(b) FEM: u = 1.800 mm.

(c) LDEIM: u = 1.685 mm.

(d) LDEIM: u = 1.800 mm.

(e) Relative error at u = 1.685 mm.

(f) Relative error at u = 1.800 mm.

Fig. 10. Two-dimensional strain-localization problem: deformed configuration corresponding to the black curve depicted in Fig. 9(a) at displacement levels corresponding to the 8th and 9th cluster changes indicated in Fig. 9(b). The reference FEM solution and reduced order model are depicted in the upper rows and the error between them is depicted in the bottom row.

43 DEIM entries are required, respectively. This observation implies that by increasing the number of clusters it is possible to construct a significantly more efficient reduced order model. In practice, the amount of memory usage in the online phase is not necessarily proportional to the number of clusters. As depicted in Fig. 13, when considering only one cluster, 335 basis vectors are required. Evidently, this number grows to 827 by increasing the number of clusters to 30. However, when the number of clusters becomes 40, the cumulative number of basis vectors shrinks down to 564. The computer memory usage not being necessarily proportional to the number of clusters, considering a large number of clusters is a feasible option.

It is also noteworthy to realize the importance of the distance metric in achieving a consistent clustering. Load–displacement curves for the 40-cluster case (black dashed line in Fig. 12) associated with two distance metrics

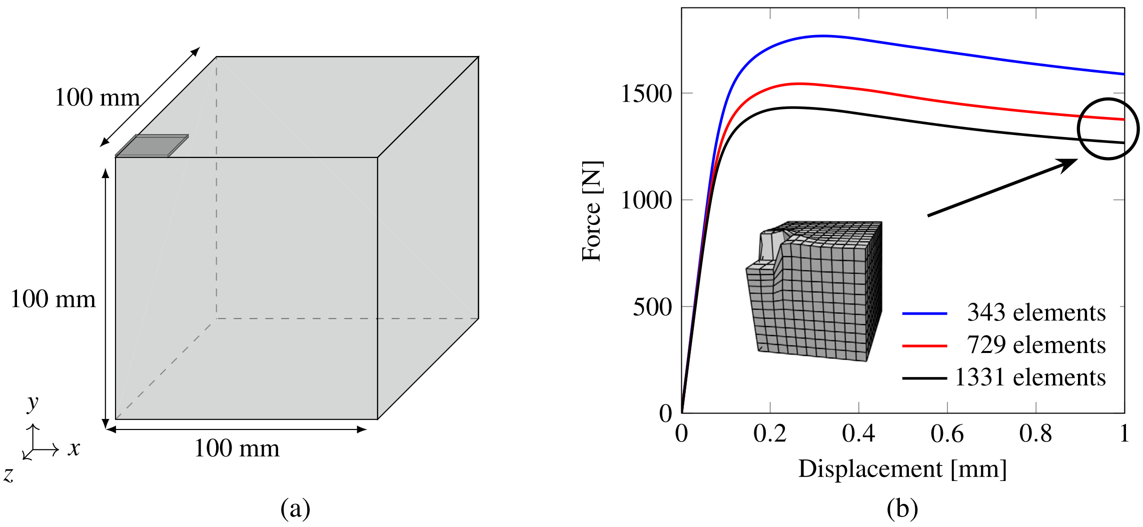


Fig. 11. Footing: (a) geometry; (b) mesh refinement study.

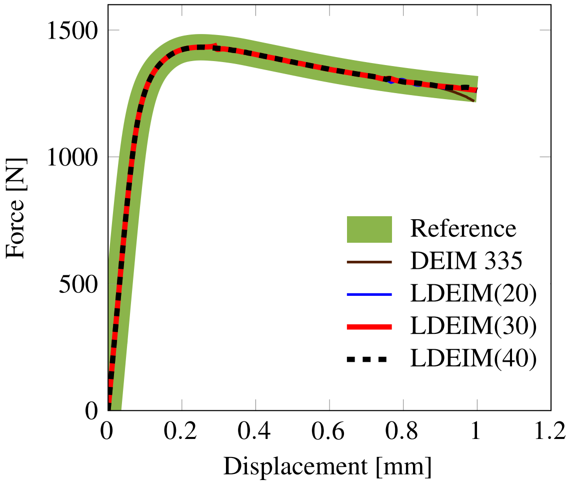


Fig. 12. Footing: load–displacement curves associated with different number of clusters (reference solution: 5184 DOFs, 1331 elements; reduced order model: 609 POD basis vectors associated with the displacement field for DEIM and LDEIM).

are depicted in Fig. 14. It is clear that the metric based on both the Euclidean distance and the cosine similarity definition gives a proper result, whereas the metric based on the Euclidean distance alone fails to do so.

5. Conclusions

Considering the slow decay of the singular values of the samples collected in the offline phase, it is possible to infer that the hypothesis central to all the MOR techniques, reported in Section 1, is not valid for the entire loading history of the class of problems studied in this paper. To mitigate this issue, samples are divided into clusters such that for each cluster the central hypothesis becomes valid. For this purpose, a potentially computationally costly k-means clustering algorithm is added to the offline phase and, employing feature extraction, an efficient nearest neighbor classifier is added to the online phase.

We now identify a possible computational bottleneck pertinent to the reduced stiffness matrix (step 3.1.2 in Algorithm 7)—note that this is merely a technical problem and not a fundamental one as those described at the

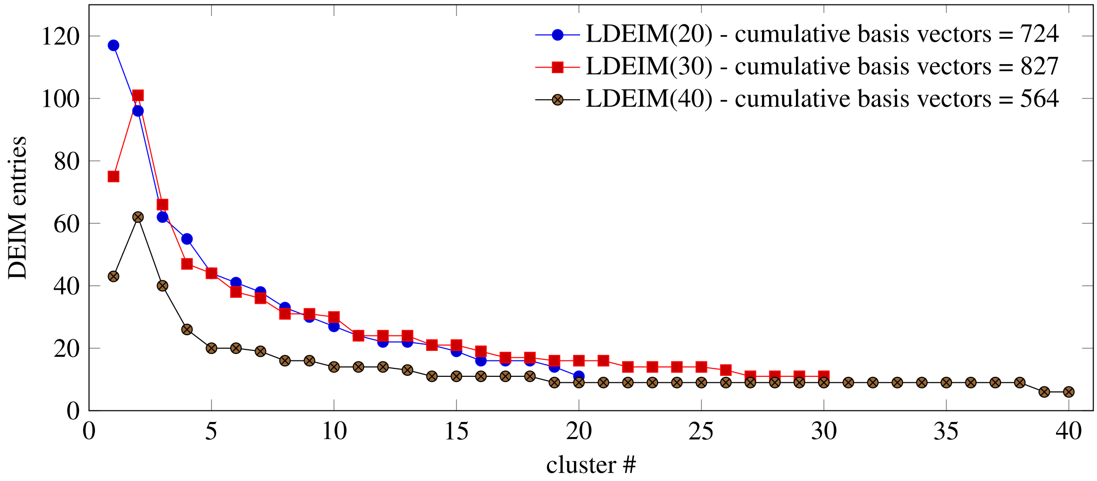


Fig. 13. Footing: number of the DEIM entries in each cluster (reduced order model: 609 POD basis vectors associated with the displacement field).

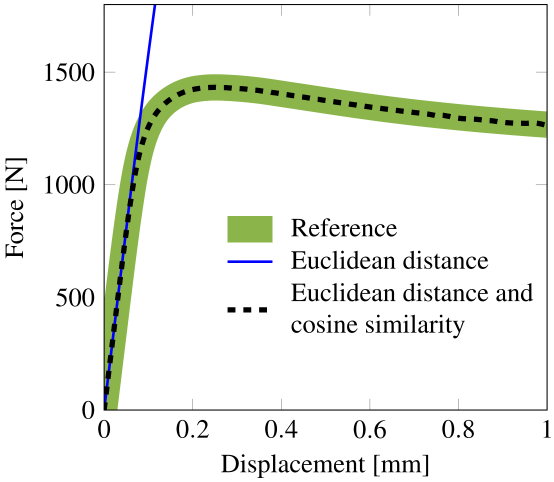


Fig. 14. Footing: the effect of the distance definition on the load–displacement diagram (reference solution: 5184 DOFs, 1331 elements; reduced order model: 609 POD basis vectors associated with the displacement field, LDEIM with 40 clusters).

end of Section 2. The storage of $\boldsymbol{K} = \boldsymbol{K}^{\mathrm{L}} + \hat{\boldsymbol{K}}$ could be quite memory consuming due to the structure of the DEIM approximated nonlinear part $\hat{\boldsymbol{K}}$. Unlike the linear part of the stiffness matrix $\boldsymbol{K}^{\mathrm{L}}$, its sparsity could be considerably less than what is expected in a classical FEM analysis. Additionally, standard procedures that take advantage of the sparsity of the stiffness matrix, such as matrix assembly, might not be able to perform as efficiently as anticipated in a FEM analysis. As a result, we suggest to avoid explicitly forming $\hat{\boldsymbol{K}}$ when computing the reduced stiffness matrix by employing an appropriate matrix chain multiplication.

Acknowledgments

The insightful comments of one of the anonymous reviewers is highly appreciated. The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007–2013)/ERC Grant Agreement No. 617972.

References

[1] B. Besselink, U. Tabak, A. Lutowska, N. van de Wouw, H. Nijmeijer, D.J. Rixen, M.E. Hochstenbach, W.H.A. Schilders, A comparison of model reduction techniques from structural dynamics, numerical mathematics and systems and control, J. Sound Vib. 332 (19) (2013) 4403–4422.

[2] C.W. Rowley, Model reduction for fluids, using balanced proper orthogonal decomposition, Int. J. Bifurcation Chaos 15 (3) (2005) 997–1013.

[3] K. Willcox, J. Peraire, Balanced model reduction via the proper orthogonal decomposition, AIAA J. 40 (11) (2002) 2323–2330.

[4] S. Lall, J.E. Marsden, S. Glavaški, A subspace approach to balanced truncation for model reduction of nonlinear control systems, Internat. J. Robust Nonlinear Control 12 (6) (2002) 519–535.

[5] S. Chaturantabut, D.C. Sorensen, Nonlinear model reduction via discrete empirical interpolation, SIAM J. Sci. Comput. 32 (5) (2010) 2737–2764.

[6] I.T. Jolliffe, Principal Component Analysis, second ed., Springer, 2002.

[7] Y.C. Liang, H.P. Lee, S.P. Lim, W.Z. Lin, K.H. Lee, C.G. Wu, Proper orthogonal decomposition and its applications—Part I: Theory, J. Sound Vib. 252 (3) (2002) 527–544.

[8] S. Gugercin, A.C. Antoulas, A survey of model reduction by balanced truncation and some new results, Internat. J. Control 77 (8) (2004) 748–766.

[9] A. Astolfi, Model reduction by moment matching for linear and nonlinear systems, IEEE Trans. Automat. Control 55 (10) (2010) 2321–2336.

[10] P. Astrid, S. Weiland, K. Willcox, T. Backx, Missing point estimation in models described by proper orthogonal decomposition, IEEE Trans. Automat. Control 53 (10) (2008) 2237–2251.

[11] K. Willcox, Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition, Comput. & Fluids 35 (2) (2006) 208–226.

[12] S. Herkt, M. Hinze, R. Pinnau, Convergence analysis of Galerkin POD for linear second order evolution equations, Electron. Trans. Numer. Anal. 40 (2013) 321–337.

[13] P. Kerfriden, J.C. Passieux, S.P.A. Bordas, Local/global model order reduction strategy for the simulation of quasi-brittle fracture, Internat. J. Numer. Methods Engrg. 89 (2) (2012) 154–179.

[14] P. Krysl, S. Lall, J.E. Marsden, Dimensional model reduction in non-linear finite element dynamics of solids and structures, Internat. J. Numer. Methods Engrg. 51 (4) (2001) 479–504.

[15] A. Doostan, R.G. Ghanem, J. Red-Horse, Stochastic model reduction for chaos representations, Comput. Methods Appl. Mech. Engrg. 196 (37–40) (2007) 3951–3966.

[16] 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 (2) (2012) 241–268.

[17] A. Ammar, B. Mokdad, F. Chinesta, R. Keunings, A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids, J. Non-Newton. Fluid Mech. 139 (3) (2005) 153–176.

[18] D. Ryckelynck, Hyper-reduction of mechanical models involving internal variables, Internat. J. Numer. Methods Engrg. 77 (1) (2009) 75–89.

[19] P. Tiso, D.J. Rixen, Discrete empirical interpolation method for finite element structural dynamics, in: Topics in Nonlinear Dynamics, vol. 1, Springer, 2013, pp. 203–212.

[20] P. Tiso, R. Dedden, D.J. Rixen, A modified discrete empirical interpolation method for reducing non-linear structural finite element models, in: ASME 2013 international Design Engineering Technical Conferences and Computers and information in Engineering Conference, American Society of Mechanical Engineers, 2013.

[21] P. Perzyna, Fundamental problems in viscoplasticity, Adv. Appl. Mech. 9 (1966) 243–377.

[22] B. Peherstorfer, D. Butnaru, K. Willcox, H.J. Bungartz, Localized discrete empirical interpolation method, SIAM J. Sci. Comput. 36 (1) (2014) A168–A192.

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

[24] A. Chatterjee, An introduction to the proper orthogonal decomposition, Current Sci. 78 (7) (2000) 808–817.

[25] K. Carlberg, M. Barone, H. Antil, Galerkin v. least-squares Petrov–Galerkin projection in nonlinear model reduction, J. Comput. Phys. 330 (2017).

[26] S. Volkwein, Proper Orthogonal Decomposition: Theory and Reduced-Order Modeling, University of Konstanz, 2013, Lecture Notes.

[27] K. Carlberg, C. Farhat, J. Cortial, D. Amsallem, The GNAT method for nonlinear model reduction: Effective implementation and application to computational fluid dynamics and turbulent flows, J. Comput. Phys. 242 (2013) 623–647.

[28] A. Naik, Data Clustering Algorithms: K-Means Clustering Algorithm, 2010. https://sites.google.com/site/dataclusteringalgorithms/k-means-clustering-algorithm (Accessed 15 May 2016).

[29] D. Amsallem, B. Haasdonk, PEBL-ROM: Projection-error based local reduced-order models, Adv. Modeling Simul. Eng. Sci. 3 (1) (2016) 1–25.

[30] Q. Liang, L. Wang, Redundancy reduction in wireless sensor networks using SVD-QR, in: Military Communications Conference, MILCOM, IEEE, 2005, pp. 1857–1861.

[31] S. Chaturantabut, D.C. Sorensen, Application of POD and DEIM on dimension reduction of nonlinear miscible viscous fingering in porous media, Math. Comput. Model. Dyn. Syst. 17 (4) (2009) 337–353.

[32] C. Farhat, P. Avery, T. Chapman, J. Cortial, Dimensional reduction of nonlinear finite element dynamic models with finite rotations and energy-based mesh sampling and weighting for computational efficiency, Internat. J. Numer. Methods Engrg. 98 (9) (2014) 625–662.