ORIGINAL PAPER

An adaptive model order reduction technique for parameter-dependent modular structures

Stephan Ritzert[1] · Domen Macek[1] · Jaan-Willem Simon[2] · Stefanie Reese[1]

Abstract

This work is concerned with an adaptive reduced order model of modular structures assembled from parameter-dependent substructures. The substructures are reduced by proper orthogonal decomposition (POD) and connected by means of a tied contact formulation. We present a method to adapt the matrices of the substructures to parameter changes. We employ interpolation on Grassmann manifolds for the parametric adaption of the projection matrices. For the adaptation of the stiffness matrices, we use the direct empirical interpolation method (DEIM). Manifold interpolation of the reduced stiffness matrices, cannot be applied here since it would require semi-positive definiteness, which is here not fulfilled because of necessary rigid body motion modes. The novelty of this work is the application of these interpolation methods to the special problem class of POD-based tied contact model order reduction. Furthermore, we show a methodology to compute significant snapshots on the substructure level to compute a POD basis that can be used in different global structures.

Keywords POD · DEIM · Substructuring · Interpolation

1 Introduction

Many different structures such as industrial halls or multistory buildings in civil engineering can be assembled from a set of substructures, like columns, girders, and plates. In mechanical finite element simulations, these substructures can be meshed individually and connected by means of tied contact formulations. The computational cost to solve those systems is high. This cost is unnecessary, in particular when the many systems consist of the same recurring substructures. To increase efficiency one can take advantage of the modular nature of those systems, by reusing precomputed stiffness matrices and applying model order reduction techniques (MOR) to each substructure. In this work, we use the proper orthogonal decomposition (POD) to reduce the substructures. The POD bases of the substructures can be used for different tied contact and boundary conditions. Often the mechanical behavior of substructures depends on parame-

ters. Such parameters can be used to change the geometry of a module to use it in a more flexible way, or to change material parameters like the fiber orientation in transversely isotropic materials. The POD-bases, that were computed for one specific set of parameters lack accuracy when the parameters are varied (cf. [3]). Therefore, parametric MOR techniques are needed to ensure good accuracy over the whole parameter range. In this work, we use interpolation techniques to adapt the precomputed matrices of all substructures to the desired parameter.

In structural dynamics, substructuring with a reduction of the degrees of freedom of the subdomains has a long history. Early works were done by [15, 24], which is still a widely used technique in structural dynamics. In the Craig-Bampton method and other component mode synthesis (CMS) methods, the nodes of each subdomain are split into interface nodes and internal nodes. The degrees of freedom of the system are reduced by a combination of eigenmodes of the internal nodes of a substructure and static boundary modes of the interfaces. An overview of CMS methods is given in [1, 16].

In contrast to the Craig-Bampton method, the present reduced order model is obtained as a combination of tied contact (cf. [19, 30]), and MOR based on proper orthogonal decomposition (POD). In POD, the projection matrices,

✉ Stephan Ritzert
  stephan.ritzert@rwth-aachen.de

1   Institute of Applied Mechanics, RWTH Aachen University, Mies-van-der-Rohe-Str. 1, 52074 Aachen, Germany

2   Computational Applied Mechanics, University of Wuppertal, Pauluskirchstr. 7, 42285 Wuppertal, Germany

that project the high-dimensional system onto the low-dimensional subspace, are computed from data by collecting snapshots of solutions of the system and taking the left modes of their singular value decomposition, see the survey [26] and references therein. The POD method is an established method in fluid mechanics, large-scale dynamics, and control [4, 8]. In recent years its popularity increased also in nonlinear solid mechanics [22, 25, 32]. For linear systems there exist also other methods to compute reduced order models (ROM), like e.g. balanced truncation. We chose POD because it can be extended to nonlinear problems. In this work, each substructure and its contact stiffness matrix contributions are projected onto its reduced space separately, see e.g. [36] who was one of the first to suggest this type of model order reduction. The novelty here lies in the new methodology to compute snapshots on the independent substructures and the extension to parameter-dependent substructures. These parameter dependencies increase the number of structures, that can be assembled from a small number of modules, but require an parametric reduced order model (pROM). Two types of parameter dependencies are considered here. Material parameters, like the fiber direction for transversely isotropic material, and geometric parameters that adapt the geometry of the substructure.

In this work interpolation techniques are applied for the adaption of the ROM to parameter changes of the substructures. The projection matrices for a desired parameter are computed by interpolation on Grassmann manifolds. This technique was chosen to keep the projection matrix as small as possible because the substructuring method already requires relatively large POD bases for each substructure. Other parametric MOR methods, that use a common basis over the whole parameter range like e.g. the reduced-basis method would result in larger projection matrices. An overview of parametric reduced order models is given in [7]. The method to interpolate projection matrices on Grassmann manifolds was proposed by [3]. It is a generalization of the subspace angle method [27]. There, the riemannian framework for interpolation on manifolds by [29] is applied to Grassmann manifolds. In this framework, the interpolants are projected onto a tangent space, where the interpolation is performed. Then, the interpolated matrix is mapped back to the manifold. In solid mechanics, these interpolation techniques were applied to hyperelastic transverse isotropic materials by [20], who discussed the stability of interpolations on Grassmann manifolds.

The stiffness matrices of each substructure also need to be adapted to parameter changes. In [2] a method was presented where the reduced mass, damping, and stiffness matrices were also interpolated on a manifold. This method is restricted to semi-positive definite matrices. Therefore, it cannot be applied in the present case, because the here presented MOR technique requires modes for the rigid body

motions of every substructure, which leads to indefinite reduced stiffness matrices, see Sect. 3.1. Furthermore, the whole stiffness matrices have to be known for every parameter to apply Dirichlet boundary conditions, that are not known beforehand. Therefore, the full stiffness matrices are interpolated. In [12, 23] a Taylor series expansion was used to approximate the mass, damping, and stiffness matrices. The accuracy of this approach decreases for multi-dimensional parameter spaces. Therefore, we use here the direct empirical interpolation method (DEIM) by [14] to approximate the stiffness matrices. DEIM belongs to the class of gappy POD methods, where a high-dimensional vector is approximated by computing a small number of entries and reconstructing the whole vector by POD modes.

The DEIM was developed on the basis of the gappy POD method of [18]. The first discrete form of the empirical interpolation method (EIM) of [6] was presented in [21]. The name discrete empirical interpolation method (DEIM) was given to the method in [14], where it was used to approximate nonlinear terms. A further development of DEIM is QDEIM proposed by [17], which provides a more efficient selection operator in most cases. In the literature there exist also matrix discrete empirical interpolation methods (MDEIM) to interpolate operator matrices of partial differential equations (see e. g. [9, 13, 28]).

The outline of the paper is as follows. First of all, in Sect. 2 the methods used for the adaptive parameter-dependent ROM of modular structures are explained. This includes the tied contact formulation, its reduced order model, and the interpolation techniques used to adapt the matrices of the substructures to parameter changes. In Sect. 3 the novel simulation technique for structures assembled from a set of parameter-dependent modules is explained, which utilizes the methods shown in Sect. 2. Furthermore, the methodology to compute significant snapshots for the POD of modules is explained. In Sect. 4 numerical examples to show the capabilities of the method are presented. The paper ends with a conclusion and outlook.

2 Methods

2.1 Substructuring with tied contact

The domain of a substructure s in the reference configuration is denoted by Ω_0^s. The total domain

$$\Omega_0 = \bigcup_{s=1}^{n_s} \Omega_0^s \tag{1}$$

is the union of all domains of the substructures of the structure, where n_s is the number of substructures. The Dirichlet
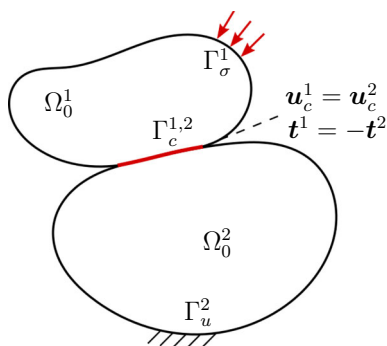
Fig. 1 Illustration of a domain, that is composed of two subdomains Ω_0^1 and Ω_0^2, with the Neumann boundary Γ_σ^1 and Dirichlet boundary Γ_u^2. At the contact interface $\Gamma_c^{1,2}$, the two tied contact conditions are displayed

and Neumann boundaries of each substructure are denoted by Γ_u^s and Γ_σ^s, respectively. The contact interface of two neighboring substructures is denoted as $\Gamma_c^{s,r}$, with $s \neq r$. The domains and boundaries of two neighboring structures are illustrated in Fig. 1. In theory, in order to connect two substructures, two conditions have to be fulfilled on every contact interface. The first condition states, that on every contact interface the displacements of the two connected bodies have to be equal:

$$\boldsymbol{u}_c^s = \boldsymbol{u}_c^r \quad \text{on} \quad \Gamma_c^{s,r}, \quad \text{with } s \neq r. \tag{2}$$

Secondly, the tractions at each contact interface have to be in equilibrium:

$$\boldsymbol{t}^s = -\boldsymbol{t}^r \quad \text{on} \quad \Gamma_c^{s,r}, \quad \text{with } s \neq r. \tag{3}$$

Additionally, the balance of linear momentum

$$\text{div}(\boldsymbol{F}^s \boldsymbol{S}^s) + \rho_0^s(\boldsymbol{b}^s - \ddot{\boldsymbol{u}}^s) = 0 \quad \text{in} \quad \Omega_0^s, \tag{4}$$

must hold for every substructure s, where $\boldsymbol{S}$ is the second Piola-Kirchhoff stress tensor, $\boldsymbol{F}$ the deformation gradient, ρ_0 the density in the reference configuration, $\boldsymbol{b}$ the body force, and $\ddot{\boldsymbol{u}}$ the acceleration vector. Every substructure can have Dirichlet boundary conditions $\boldsymbol{u}^s = \bar{\boldsymbol{u}}^s$ and Neumann boundary conditions $\boldsymbol{F}^s \boldsymbol{S}^s \cdot \boldsymbol{N}^s = \bar{\boldsymbol{t}}_0^s$ acting on their respective boundaries Γ_u^s and Γ_σ^s.

For the finite element discretization, the above equations have to be transferred into the weak form

$$g = \sum_{s=1}^{n_s}(g_{standard}^s - g_c^s) = 0, \tag{5}$$

where a contact term g_c^s has to be considered in addition to the standard weak form of the balance of linear momentum

$g_{standard}^s$. For quasi-static processes the acceleration vector $\ddot{\boldsymbol{u}}$ can be neglected. The weak forms are

$$g_{standard}^s = \int_{\Omega_0^s}(\boldsymbol{S}^s : \delta\boldsymbol{E}^s - \rho_0^s\boldsymbol{b}^s \cdot \delta\boldsymbol{u}^s)\,\text{d}V$$
$$- \int_{\Gamma_t^s} \delta\boldsymbol{u}^s \cdot \bar{\boldsymbol{t}}_0^s\,\text{d}A, \tag{6}$$

$$g_c^s = \int_{\Gamma_c^s} \delta\boldsymbol{u}^s \cdot \boldsymbol{t}^s\,\text{d}A, \tag{7}$$

where the contact term can be interpreted as the virtual work of the tractions acting on the contact surfaces. The virtual work contribution on each contact interface $\Gamma_c^{s,r}$ is

$$g_c^{s,r} = g_c^s + g_c^r = \int_{\Gamma_c^{s,r}} \delta\boldsymbol{u}^s \cdot \boldsymbol{t}^s\,\text{d}A + \int_{\Gamma_c^{s,r}} \delta\boldsymbol{u}^r \cdot \boldsymbol{t}^r\,\text{d}A, \tag{8}$$

which is the sum of the contact terms of the substructures s and r. To enforce the tied contact condition Eq. (2), the penalty method is used, where a penalty parameter penalizes the difference in the displacements. It relates the traction on the contact surface to the displacement difference

$$\boldsymbol{t}^s = \varepsilon\boldsymbol{w}^s. \tag{9}$$

The displacement and virtual displacement differences are defined as

$$\boldsymbol{w}^s := \boldsymbol{u}_c^s - \boldsymbol{u}_c^r, \qquad \delta\boldsymbol{w}^s := \delta\boldsymbol{u}^s - \delta\boldsymbol{u}^r. \tag{10}$$

Inserting Eqs. (9) and (10) into Eq. (8) and exploiting the equilibrium condition Eq. (3) leads to the virtual work contribution of a contact interface $\Gamma_c^{s,r}$

$$g_c^{s,r} = \int_{\Gamma_c^{s,r}} \varepsilon\delta\boldsymbol{w}^s \cdot \boldsymbol{w}^s\,\text{d}A. \tag{11}$$

This equation can be discretized by finite elements, which leads to

$$g_c^{s,r} \overset{FEM}{=} \left[(\delta\boldsymbol{U}_c^s)^T \ (\delta\boldsymbol{U}_c^r)^T\right] \underbrace{\begin{bmatrix} \boldsymbol{K}_{c,ss}^{s,r} & \boldsymbol{K}_{c,sr}^{s,r} \\ \boldsymbol{K}_{c,rs}^{s,r} & \boldsymbol{K}_{c,rr}^{s,r} \end{bmatrix}}_{:=\boldsymbol{K}_c^{s,r}} \begin{bmatrix} \boldsymbol{U}_c^s \\ \boldsymbol{U}_c^r \end{bmatrix}. \tag{12}$$

Here, $\boldsymbol{K}_c^{s,r}$ is the contact stiffness matrix of the interface $\Gamma_c^{s,r}$, and $\boldsymbol{U}_c^s$, $\boldsymbol{U}_c^r$ are the displacement vectors corresponding to the DOF on the contact interface of the substructures s and r. The contact stiffness matrix couples the two substructures by ensuring that the displacement vectors $\boldsymbol{U}_c^s$ and $\boldsymbol{U}_c^r$ are almost equal. For the discretization, we use linear shape functions and a node-to-node approach. Other contact discretizations like the mortar method are also possible. For further information, the reader is referred to e.g. [30]. Discretizing also

the balance of linear momentum of every substructure leads to the global nonlinear system of equations

$$R + K_c U = P, \tag{13}$$

where the internal force vector $R = R(U)$, the displacement vector U, the contact stiffness matrix K_c, and the external force vector P are assembled from the vectors and matrices of the substructure. This nonlinear system of equations is typically solved by the Newton–Raphson method. In the special case of linear mechanics, the internal force vector of each substructure depends linearly on the displacement vector $R^s = K^s U^s$, where K^s is the stiffness matrix of a substructure. This results in the linear system of equations

$$(K + K_c)U = P, \tag{14}$$

where the global stiffness matrix K is assembled from the stiffness matrices of the substructures K^s. The vectors of the substructures are written into the global residual, displacement, and external force vectors

$$R = \begin{bmatrix} \vdots \\ R^s \\ R^r \\ \vdots \end{bmatrix}, \quad U = \begin{bmatrix} \vdots \\ U^s \\ U^r \\ \vdots \end{bmatrix}, \text{ and } \quad P = \begin{bmatrix} \vdots \\ P^s \\ P^r \\ \vdots \end{bmatrix}, \tag{15}$$

which is shown here in an exemplary way for the substructures s and r. The global matrices are

$$K = \begin{bmatrix} \ddots & & & \\ & K^s & & \\ & & K^r & \\ & & & \ddots \end{bmatrix}, \text{ and}$$

$$K_c = \begin{bmatrix} \ddots & & & \\ & K^{s,r}_{c,ss} & K^{s,r}_{c,sr} & \\ & K^{s,r}_{c,rs} & K^{s,r}_{c,rr} & \\ & & & \ddots \end{bmatrix}, \tag{16}$$

where the stiffness matrix K^s takes a block-diagonal form, and K_c contains the contact stiffness matrices of all contact interfaces. It has entries outside of the block-diagonal form of K and therefore couples the substructures.

2.2 Reduced-order model

Different structures can be assembled and solved from a set of substructures by the method described in Sect. 2.1. The total number of degrees of freedom is typically a large number, which results in a high computational effort to solve

such a system of equations. Therefore, model order reduction (MOR) techniques are applied to reduce the computational effort. In the technique used here, every substructure is projected onto its own reduced space, by a substructure projection matrix Ψ^s. The m^s columns of the projection matrix are modes that span an m^s-dimensional basis for the displacement vector of the substructure s. Multiplication of the projection matrix Ψ^s with a reduced displacement vector U^s_{red} gives an approximation for the substructure displacement vector

$$U^s \approx \Psi^s U^s_{\text{red}}, \tag{17}$$

Using Eq. (17), the n^s-dimensional displacement vector can be expressed by the m^s-dimensional reduced displacement vector, where $m^s \ll n^s$ holds. The projection matrices of each substructure Ψ^s are computed by the method of snapshots (cf. [10, 32]), where displacement vectors of a substructure are collected as columns of a snapshot-matrix and decomposed into three matrices by a singular value decomposition

$$S_{\text{Snap}} = [u^s_1, u^s_2, \ldots, u^s_k] = \Phi^s \Lambda^s \Upsilon^s \quad \text{(SVD)}. \tag{18}$$

The projection matrix is obtained by taking only the first m columns of the matrix Φ into account

$$\Psi^s = [\Phi^s_1, \Phi^s_2, \ldots, \Phi^s_m]. \tag{19}$$

The computation of POD bases for substructures, that can be used in different systems is an open challenge. The method to compute snapshots we use in this work is explained in Sect. 3.1. It should be noted that POD is typically used in MOR of nonlinear problems. We chose the method here because our goal is to extend the method to nonlinear problems. A method to compute such a POD basis for linear problems is a necessary first step toward such POD bases for nonlinear problems.

To approximate the global displacement vectors the projection matrices of the k substructures are assembled into a global projection matrix

$$\Psi = \begin{bmatrix} \Psi^1 & & & \\ & \Psi^2 & & \\ & & \ddots & \\ & & & \Psi^k \end{bmatrix}. \tag{20}$$

The dimensions of Ψ are $n \times m$, with $m = \sum_{s=1}^{k} m^s$ and $n = \sum_{s=1}^{k} n^s$. Using Eq. (20), the global displacement vector is approximated by

$$U \approx \Psi U_{\text{red}}, \tag{21}$$

where U_{red} contains all reduced displacement vectors of the substructures U_{red}^s. Inserting Eq. (21) into Eqs. (13) and (14) and applying a Galerkin projection with the global projection matrix $\boldsymbol{\Psi}$ leads to the general nonlinear system of equations

$$\boldsymbol{\Psi}^T \boldsymbol{R} + \boldsymbol{\Psi}^T \boldsymbol{K}_c \boldsymbol{\Psi} U_{\text{red}} = \boldsymbol{\Psi}^T \boldsymbol{P}, \tag{22}$$

which can be solved by the Newton–Raphson method. The specialization for linear elasticity is given by the relation

$$\boldsymbol{\Psi}^T \boldsymbol{K} \boldsymbol{\Psi} U_{\text{red}} + \boldsymbol{\Psi}^T \boldsymbol{K}_c \boldsymbol{\Psi} U_{\text{red}} = \boldsymbol{\Psi}^T \boldsymbol{P}, \tag{23}$$

which will be used in the remainder of this work. For the linear case, the projection matrices $\boldsymbol{\Psi}^s$ and the stiffness matrices $\boldsymbol{K}^s$ of all substructures can be precomputed, only the contact stiffness matrix $\boldsymbol{K}_c$ has to be computed for every new structure. The reduced stiffness matrix of the system is defined as

$$\boldsymbol{K}_{\text{red}} := \boldsymbol{\Psi}^T \boldsymbol{K} \boldsymbol{\Psi} = \begin{bmatrix} \boldsymbol{\Psi}^{1T} \boldsymbol{K}^1 \boldsymbol{\Psi}^1 & & & \\ & \boldsymbol{\Psi}^{2T} \boldsymbol{K}^2 \boldsymbol{\Psi}^2 & & \\ & & \ddots & \\ & & & \boldsymbol{\Psi}^{kT} \boldsymbol{K}^k \boldsymbol{\Psi}^k \end{bmatrix}$$

$$= \begin{bmatrix} \boldsymbol{K}_{\text{red}}^1 & & & \\ & \boldsymbol{K}_{\text{red}}^2 & & \\ & & \ddots & \\ & & & \boldsymbol{K}_{\text{red}}^k \end{bmatrix}. \tag{24}$$

Here $\boldsymbol{K}_{\text{red}}^s$ are the reduced stiffness matrices of the substructures. The special situation here is that each substructure is reduced by its associated projection matrix. Consequently, the contact stiffness matrices are reduced by the projection matrices of the two connected substructures. This is shown here in an exemplary way for the contact interface $\Gamma_c^{s,r}$:

$$\boldsymbol{K}_{\text{c,red}} := \boldsymbol{\Psi}^T \boldsymbol{K}_c \boldsymbol{\Psi}$$

$$= \begin{bmatrix} \ddots & & \\ & \boldsymbol{\Psi}_c^{sT} & \\ & & \boldsymbol{\Psi}_c^{rT} \end{bmatrix} \begin{bmatrix} \ddots & & \\ & \boldsymbol{K}_{c,ss}^{s,r} & \boldsymbol{K}_{c,sr}^{s,r} \\ & \boldsymbol{K}_{c,rs}^{s,r} & \boldsymbol{K}_{c,rr}^{s,r} \end{bmatrix} \begin{bmatrix} \ddots & & \\ & \boldsymbol{\Psi}_c^s & \\ & & \boldsymbol{\Psi}_c^r \end{bmatrix}$$

$$= \begin{bmatrix} \ddots & & \\ & \boldsymbol{\Psi}_c^{sT} \boldsymbol{K}_{c,ss}^{s,r} \boldsymbol{\Psi}_c^s & \boldsymbol{\Psi}_c^{sT} \boldsymbol{K}_{c,sr}^{s,r} \boldsymbol{\Psi}_c^r \\ & \boldsymbol{\Psi}_c^{rT} \boldsymbol{K}_{c,rs}^{s,r} \boldsymbol{\Psi}_c^s & \boldsymbol{\Psi}_c^{rT} \boldsymbol{K}_{c,rr}^{s,r} \boldsymbol{\Psi}_c^r \end{bmatrix} \tag{25}$$

The matrix $\boldsymbol{\Psi}_c^s$ contains all rows of the projection matrix that correspond to the contact interface DOF. We want to emphasize the mixed projection of the coupling terms of the contact stiffness matrix $\boldsymbol{K}_{c,sr}^{s,r}$ and $\boldsymbol{K}_{c,rs}^{s,r}$. They are projected by means of the projection matrices $\boldsymbol{\Psi}_c^s$ and $\boldsymbol{\Psi}_c^r$ of the two substructures.

With $\boldsymbol{P}_{\text{red}} = \boldsymbol{\Psi}^T \boldsymbol{P}$ and the definitions in Eqs. (24) and (25) the reduced linear system is written as

$$(\boldsymbol{K}_{\text{red}} + \boldsymbol{K}_{c,\text{red}}) U_{\text{red}} = \boldsymbol{P}_{\text{red}}. \tag{26}$$

These reduced stiffness matrices of the modules are invariant against rotations. Because any Dirichlet boundary conditions should be possible, the matrices $\boldsymbol{K}_{\text{red}}^s$ cannot be precomputed.

2.3 Parametric reduced-order model

The challenge of the here-described ROM is that every projection matrix is linked to the material parameters and the geometry for which it was computed. It is not possible to use it for systems that significantly deviate in material and geometry. The fixed geometry of the substructures is a tremendous restriction on the applicability of the method. The ability to adapt the parameterized geometries of the substructures increases the number of different possible structures, that can be assembled. Considering transverse isotropic materials, we want to be able to choose any fiber direction. In this work, we parameterize the geometry of the modules and the fiber direction in transversely isotropic materials. These two different cases require an adaption of the projection and stiffness matrices to allow for the efficient computation of many different structures.

Every substructure depends on a set of parameters that strongly influence its mechanical behavior. The model order reduction (MOR) technique explained in Sect. 2.2 will now be extended to allow for a free choice of these parameters, while preserving the computational efficiency by approximating all parameter-dependent matrices. In the case of linear mechanics, the projection matrix $\boldsymbol{\Psi}^s = \boldsymbol{\Psi}^s(\boldsymbol{p})$ and the stiffness matrix $\boldsymbol{K}^s = \boldsymbol{K}^s(\boldsymbol{p})$ of all substructures depend on the parameter vector $\boldsymbol{p}$. Theoretically, both matrices have to be computed for all new values of the parameter vector $\boldsymbol{p}$. To compute the projection matrices for new parameters we have to perform many precomputations on the substructure level for different load cases, which is described in Sect. 3.1. The computational effort to compute the stiffness matrices depends on the number n^s of degrees-of-freedom of the substructure. For an efficient online simulation, this computational effort is too high. Therefore, we approximate the matrices by interpolation techniques, to make an efficient online simulation of different parameter-dependent systems possible.

In our model, the projection matrices and the stiffness matrices are interpolated separately. The stiffness matrices are obtained using the direct empirical interpolation method (DEIM), (cf. [11, 14]). The projection matrices are interpolated by the interpolation on Grassmann manifolds as suggested by [3].

The interpolation of stiffness matrices is based on an assumed affine parameter dependence

$$K^s(p) = \sum_{i=1}^{k} f_i(p) W_i^s, \tag{27}$$

where W_i are parameter-independent basis matrices and $f_i(p)$ are unknown functions of the parameter vector p. The stiffness matrix $K^(\bar{p})$ for a target parameter vector $\bar{p}$ is obtained by computing the function values $f_i(\bar{p})$ and multiplying with the basis matrices W_i^s. In DEIM, these sought-after function values are approximated by a sparse sampling of distinct values of the stiffness matrix $K^s(p)$. DEIM cannot be used to approximate the projection matrix because projection matrices can only be computed as a whole. Therefore, for the interpolation of the projection matrices, we choose interpolation on Grassmann manifolds. In manifold interpolation schemes, the matrices are interpolated on the geodesic line connecting them (cf. [29]). The geodesic line is the shortest path between two points on a manifold. The Grassmann manifold interpolation ensures, that the interpolated projection matrix is a basis with the same rank as the precomputed bases.

In the literature, also reduced system matrices like the stiffness matrix K_{red} were interpolated using manifold interpolation techniques, cf. [2]. This interpolation uses the matrix manifold and its tangent space. It cannot be used in our case because it requires a mapping from the matrix manifold to its tangent space, which is only defined for semi-positive definite (SPD) matrices. The rigid body motions contained in our projection matrices lead to indefinite matrices, for which the mappings are not defined. Furthermore, the unreduced stiffness matrix is needed to apply arbitrary Dirichlet boundary conditions.

2.3.1 Direct empirical interpolation of stiffness matrices

The direct empirical interpolation method (DEIM) is a method to sparsely reconstruct vectors, where the sampling points are chosen with a greedy algorithm. It is typically used to approximate nonlinear terms in POD-based model order reduction, cf. [33]. In contrast to other matrix DEIM versions (e.g. [9, 13, 28]) we apply the standard DEIM algorithm to data vectors of stiffness matrices consisting of its nonzero entries.

For the application to the approximation of parameter-dependent stiffness matrices, these stiffness matrices have to be transferred to a vector format. Here we make use of the sparsity of stiffness matrices and use a format where only non-zero entries are saved. In the coordinate format, every

stiffness matrix can be expressed by three vectors

$$K^s(p) = \{k^s(p), r^s, c^s\}, \tag{28}$$

where $k^s(p)$ is referred to as the stiffness data vector and contains all nonzero entries of the stiffness matrix, the position of each value is stored in the vectors r^s and c^s for the row and column positions respectively. Here, only the stiffness data vector $k^s(p)$ is parameter-dependent, the row and column position vectors are constant for all parameters.

With DEIM, the n_k-dimensional stiffness data vector $k^s(p)$ is approximated by

$$k^s(p) \approx \Omega\, a(p), \tag{29}$$

where the k columns of Ω span a basis for the stiffness data vector $k^s(p)$, and $a(p)$ is a k-dimensional coefficient vector. The basis Ω is computed analogously to the POD-basis Ψ^s of the displacement vector U^s. Stiffness data vectors for different parameters are collected in a matrix which is multiplicatively decomposed by an SVD:

$$A = \begin{bmatrix} k^s(p_1)\ k^s(p_2)\ \dots\ k^s(p_k) \end{bmatrix} = W\Lambda V. \tag{30}$$

The basis Ω is obtained by taking the first k columns of the left mode matrix W into account

$$\Omega = [W_1, W_2, \dots, W_k]. \tag{31}$$

To compute the stiffness matrix for a target parameter vector $\bar{p}$, only k coefficients in $a(\bar{p})$ have to be determined. By selecting k rows of the linear system in Eq. (29) the coefficient vector $\bar{a}$ for the target parameter vector $\bar{p}$ can be determined with

$$\bar{a} = a(\bar{p}) \approx (Z^T \Omega)^{-1} Z^T k^s(\bar{p}), \tag{32}$$

where the $n_k \times k$-dimensional selection matrix $Z = [e_{\gamma_1}, e_{\gamma_2}, \dots, e_{\gamma_k}]$ contains the unit vectors of the selected sampling points. Then, the whole stiffness data vector for the target parameter vector $\bar{p}$ can be reconstructed with

$$k^s(\bar{p}) \approx \Omega\,\bar{a} = \Omega(Z^T \Omega)^{-1} Z^T k^s(\bar{p}). \tag{33}$$

Which rows are evaluated is determined by the DEIM algorithm, where it has to be ensured that $Z^T \Omega$ is non-singular. The DEIM algorithm is a greedy algorithm to successively choose the sampling points, at the position of the maximum of a residual. The algorithm developed by [14] is shown in Algorithm 1. The elements that need to be evaluated to compute the selected rows of $k^s(\bar{p})$ can be determined from the DOFs in the corresponding entries in the row and column

position vectors r and c. With this method, the computational effort to compute stiffness matrices is reduced from $\mathcal{O}(n)$ to $\mathcal{O}(k)$.

Algorithm 1: DEIM

Input: mode matrix: $\boldsymbol{\Omega} = [\boldsymbol{W}_1, \boldsymbol{W}_2, \ldots \boldsymbol{W}_k]$
Output: selection matrix: $\boldsymbol{Z}$
Initialize: $\gamma_1 = \text{maxloc}(\boldsymbol{W}_1)$
$\qquad\qquad \boldsymbol{Z} = [\boldsymbol{e}_{\gamma_1}], \quad \boldsymbol{\Omega} = [\boldsymbol{W}_1]$
for $s \in \{2, \ldots, k\}$ **do**
$\quad | \quad \boldsymbol{a}_s = (\boldsymbol{Z}^T \boldsymbol{\Omega})^{-1} \boldsymbol{Z}^T \boldsymbol{W}_s$
$\quad | \quad \boldsymbol{R}_s = \boldsymbol{W}_s - \boldsymbol{\Omega}$
$\quad | \quad \gamma_s = \text{maxloc}(\boldsymbol{R}_s)$
$\quad | \quad \boldsymbol{Z} \leftarrow [\boldsymbol{Z}, \boldsymbol{e}_{\gamma_s}], \quad \boldsymbol{\Omega} \leftarrow [\boldsymbol{\Omega}, \boldsymbol{W}_s]$

2.3.2 Interpolation of projection matrices on Grassmann manifolds

A projection matrix $\bar{\boldsymbol{\Psi}}$ for a target parameter $\bar{p}$ is computed by interpolating on the geodesic line on a Grassmann manifold between projection matrices in the neighborhood. A Grassmann manifold $\mathcal{G}(m, n)$ is defined as the set of m-dimensional subspaces of the $\mathbb{R}^n$. Therefore a m-dimensional projection matrix $\boldsymbol{\Psi}$ computed with POD non-uniquely defines a point $\mathcal{Y}$ on the Grassmann manifold $\mathcal{G}(m, n)$. The interpolation is done on the geodesic line, which is defined as the shortest path between two points $\mathcal{Y}_0$ and $\mathcal{Y}_1$ on a differentiable manifold. By interpolating on the geodesic line, the underlying subspaces are interpolated, which ensures that the interpolated matrix is a basis.

For simplicity, the method is here explained for a one-dimensional parameter space with the parameter p. To extend it to a multi-dimensional parameter space the interpolation on the tangent space would need to be multi-dimensional. A bilinear two-dimensional interpolation scheme is explained in Appendix 1. To compute the projection matrix $\bar{\boldsymbol{\Psi}}$ for a target parameter $\bar{p}$, the POD-basis $\{\boldsymbol{\Psi}_1, \ldots, \boldsymbol{\Psi}_k\}$ with their parameters $\{p_1, \ldots, p_k\}$ in the neighborhood of the target parameter $\bar{p}$ are considered. The first step is to choose an origin point, here $\boldsymbol{\Psi}_o = \boldsymbol{\Psi}_1$. For the interpolation, all other points are projected onto the tangent space of the origin $\mathcal{T}_{\mathcal{Y}_o}$ by a logarithmic mapping $\mathcal{X}_j = \text{Log}_{\mathcal{Y}_o}(\mathcal{Y}_j)$. For Grassmann manifolds, this logarithmic mapping is defined as

$$\boldsymbol{\Psi}_j(\boldsymbol{\Psi}_o^T \boldsymbol{\Psi}_j)^{-1} - \boldsymbol{\Psi}_o = \boldsymbol{W}_j \boldsymbol{\Lambda}_j \boldsymbol{V}_j \quad \text{(thin SVD)}$$
$$\boldsymbol{\Gamma}_j = \boldsymbol{W}_j \arctan(\boldsymbol{\Lambda}_j) \boldsymbol{V}_j, \tag{34}$$

where $\boldsymbol{\Gamma}_j$ spans the point on the tangent space $\mathcal{X}_j$. The term thin SVD refers to an SVD of an $n \times m$ matrix where only the first $k = \min(n, m)$ singular vectors are computed (cf. [5]). The logarithmic mappings can be done in a pre-computation step. In the next step, the points on the tangent space are
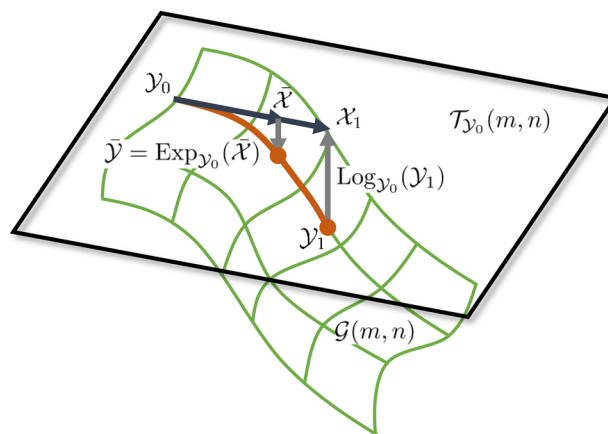
Fig. 2 Illustration of the interpolation on the Grassmann manifold for two points

interpolated with

$$\bar{\boldsymbol{\Gamma}} = \sum_{j=1}^{k} f(\bar{p}, p_j) \boldsymbol{\Gamma}_j, \tag{35}$$

where $f(\bar{p}, p_j)$ can be any interpolation function. For the interpolation between two matrices, it would be $f(\bar{p}, p_j) = (\bar{p} - p_o)/(p_j - p_o)$, where p_o is the parameter corresponding to the origin. To obtain the interpolated point on the Grassmann manifold, the interpolated point on the tangent space $\bar{\mathcal{X}}$ that is spanned by $\bar{\boldsymbol{\Gamma}}$ is mapped back to it by an exponential mapping $\bar{\mathcal{Y}} = \text{Exp}_{\mathcal{Y}_o}(\bar{\mathcal{X}})$. For Grassmann manifolds, this mapping can be computed with

$$\bar{\boldsymbol{\Gamma}} = \bar{\boldsymbol{W}} \bar{\boldsymbol{\Lambda}} \bar{\boldsymbol{V}} \quad \text{(thin SVD)}$$
$$\bar{\boldsymbol{\Psi}} = \boldsymbol{\Psi}_o \bar{\boldsymbol{V}} \cos(\bar{\boldsymbol{\Lambda}}) + \bar{\boldsymbol{U}} \sin(\bar{\boldsymbol{\Lambda}}), \tag{36}$$

where $\bar{\boldsymbol{\Psi}}$ is the interpolated projection matrix that spans the point $\bar{\mathcal{Y}}$. The interpolation between two points is illustrated in Fig. 2 With this method, the projection matrix for any parameter inside the parameter range can be approximated.

3 The modular method

In this section, it is explained how the methods explained in Sect. 2 can be applied to efficiently compute structures that are assembled from a set of parameter-dependent modules. In this paper, we consider two different kinds of parameters. The first group are material parameters like the primary direction of transversely isotropic materials. The second type are geometrical parameters like the length or the curvature of a substructure. First, a new method to compute snapshots on the substructure level is explained. Then the procedure to solve a parametric reduced modular structure is described.
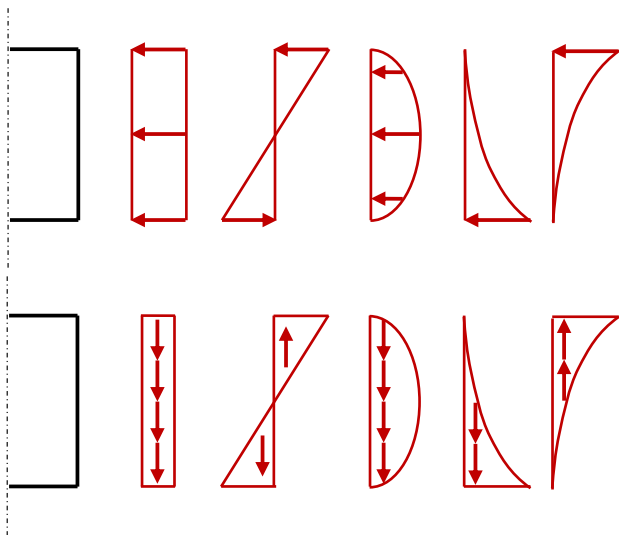
Fig. 3 Characteristic load cases for all contact surfaces

3.1 Computation of snapshots

The projection matrices of the modules strongly influence the quality of the solution. The unique situation in this work is, that the projection matrices are computed on the substructure level and are used in arbitrary global structures. Therefore, the snapshots should represent the mechanical behavior of the modules in a general sense. The snapshots of all substructures are computed by the same method. In the following, the method we developed to compute snapshots on the substructure level is presented.

In this method, we combine three different types of snapshots. The first type of snapshots are computed under Neumann boundary conditions. The purpose of these snapshots is to obtain a set of basic deformation modes like stretching, bending, or shearing. The second type of snapshots are computed from Dirichlet boundary conditions on single nodes on possible contact interfaces. These snapshots enhance the modes to better approximate the displacements on possible contact interfaces. The third type of snapshots are rigid body motions. In systems with substructuring, some substructures have rigid body displacement because they are connected to deformed substructures. The projection matrices have to include rigid body deformation modes because every substructure is reduced by a separate projection matrix.

For the first type of snapshots the first step is to define all possible contact surfaces. Then, one contact surface is selected as a Neumann boundary and one as a Dirichlet boundary. On the Neumann boundary surface, the load cases depicted in Fig. 3 are applied, while all DOFs on the Dirichlet boundary are fixed. The shapes of the load cases consist of three quadratic, one linear, and one constant function. These functions are chosen to represent different traction distribu-

tions on contact interfaces. The snapshots are computed for all possible combinations of Neumann and Dirichlet surfaces.

For the second type of snapshots, all DOFs of a single contact surface or all contact surfaces altogether are fixed. On each DOF a unit displacement is applied and the resulting displacement vector is saved into a separate snapshot matrix $S_{BC} = [u_1, \ldots, u_{n_c}]$. From the left mode matrix of the SVD of this snapshot matrices S_{BC}, we choose to use five dominant modes as snapshots for the projection matrix. This way we take only the most dominant effects from this type of snapshots into account. Taking more than five modes would increase the size of the POD basis without increasing accuracy in the simulations. In the cases where snapshots for a single contact surface are computed, two different cases are considered. In the first case, the unit displacement is applied to every DOF separately. In the second case, the unit displacement is kept, which means that the number of DOFs with a unit displacement increases until the displacement is applied to all DOFs.

With these three kinds of snapshots, a relatively large number of physically reasonable POD modes can be found, which is needed to get accurate results for modules used in different ways.

3.2 Solution procedure

With the method presented in this paper, a large number of different structures that are assembled from parameter-dependent substructures can be computed efficiently. An advantage of this method is, that the substructures are used many times and their matrices are obtained from precomputations. Substructures that are used often are referred to as modules. For every module s, the DEIM modes $\mathbf{\Omega}^s$ and the sampling points γ_i^s, as well as projection matrices $\mathbf{\Psi}_j^s$ for different parameters, are known. Moreover, the points in the tangent space $\mathbf{\Gamma}_j^s$ for the interpolation on Grassmann manifolds can be precomputed (cf. Sect. 2.3.2). All of these matrices are defined in a reference coordinate system.

To solve the whole reduced system, the global reduced stiffness matrix K_{red} (Eq. (24)), the global projection matrix $\mathbf{\Psi}$ (Eq. (20)), and the reduced contact stiffness matrix $K_{c,\text{red}}$ (Eq. (25)) have to be assembled. For every structure, the contact stiffness matrix has to be computed, the other matrices are approximated for the desired parameter vector $\bar{p}$. The global projection matrix $\mathbf{\Psi}$ and global reduced stiffness matrix K_{red} are assembled from the corresponding matrices of the substructures. For every substructure, the reduced stiffness matrix K_{red}^s and the projection matrix $\mathbf{\Psi}^s$ are computed in three steps. In the first step, the unreduced stiffness matrix K^s and the projection matrix $\mathbf{\Psi}^s$ are computed for the desired parameter in the reference coordinate system. The matrices are interpolated from sampled data with the methods explained in Sects. 2.3.1 and 2.3.2. In the second step,
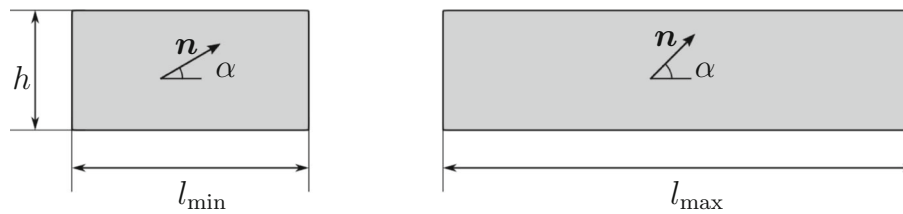
Fig. 4 Module (a): Rectangle with transversely isotropic material. The parameters that can be changed are the length l and the fiber direction angle α. The length can vary between $l_{\min} = 600\,\text{mm}$ and $l_{\max} = 1200\,\text{mm}$. The fiber direction angle α can vary between 0° and 90°

the Dirichlet boundary conditions are applied to the stiffness matrix and the reduced stiffness matrix is computed. In the third step, the projection matrix is rotated if the module is rotated in the global structure. The reduced stiffness matrices are invariant against rotations. Finally, these matrices are assembled into the global system Eq. (26) and solved. The total solution is obtained by projecting the reduced vector back. The stresses can be computed in postprocessing.

4 Numerical examples

4.1 Example 1: DEIM of stiffness matrices

In the first numerical example, the quality of the DEIM approximation is assessed. For that, the approximated stiffness matrices of two different modules are compared to the standard stiffness matrix. The error norm for the comparison is defined as

$$e_k := \frac{\| k - k_{\text{ref}} \|}{\| k_{\text{ref}} \|}, \tag{37}$$

where k and k_{ref} are the approximated and the standard stiffness data vectors respectively. The first parameter-dependent module (a) is a rectangle with a transversely isotropic material, where the length l and the fiber direction α are variable. It is illustrated in Fig. 4. The transversely isotropic material law is the linear version of the material law from [34]. For the computation of the basis $\boldsymbol{\Omega}$ the stiffness matrices are computed at the lengths $l = \{600, 700, \dots, 1200\}$ and the fiber angles $\alpha = \{0°, 10°, \dots, 90°\}$. The second module (b) is a circular arc with variable angle ϕ and constant arc length $l_{\text{arc}} = 800$, which is shown in Fig. 5. Here, the parameters for the basis computation are $\phi = \{10°, 15°, \dots, 90°\}$.

In Fig. 6 the error norms of the modules (a) and (b) are plotted over their parameter ranges. For module (a) the error is plotted over the length of the module for three different fixed fiber directions controlled by the parameter α. For all three values of α the error is in the magnitude of 10^{-14}, which leads to the conclusion that the approximation is accurate. For module (b) the error is shown over the angle ϕ. With a magnitude of 10^{-9}, the error is significantly higher compared

to module (a). Moreover, the error norm has peaks at the two ends of the parameter range. Even though the error is higher, the approximation is still accurate.

A possible explanation for the higher error of the curved module can be seen when analyzing the decay of the normalized singular values of the basis computation for the DEIM approximation. A sign of a good basis is a drop in the singular values because it means that all snapshots can be represented with a small number of modes. This sudden drop can be clearly seen in Fig. 7 for module (a). In contrast to module (a), the decay of the normalized singular values of module (b) does not show this characteristic.

However, it can be concluded, that DEIM gives an accurate approximation of stiffness matrices. The accuracy depends on the quality of the basis, which can be assessed by the decay of the singular values.

4.2 Example 2: Two dimensional parameter space

In the second numerical example, the quality of the interpolated ROM over a two-dimensional parameter space is analyzed, with a small system shown in Fig. 8. The system consists of three substructures from two different modules. The rectangular module is discretized by 6000 elements and the square module by 3600 elements. At the contact interfaces, 60 line elements are used. All substructures have a transversely isotropic material (cf. [34]) with the same fiber orientation α. The material parameters are shown in Table 1. The parameter space is two-dimensional because the top right rectangular module can vary in length.

The MOR solution of the system is compared to a reference solution over the parameter ranges of the length and the fiber direction. For the MOR solution, the projection matrices are precomputed for the parameter values $l = \{600, 700, \dots, 1200\}$ and $\alpha = \{0°, 5°, \dots, 90°\}$. The stiffness matrices are approximated by the DEIM basis from Sect. 4.1. The rectangular module has 40 POD modes and 12 DEIM modes. The square module has 52 POD modes and 5 DEIM modes.

For the reference solution, the tied contact formulation is used without model order reduction. The displacement and
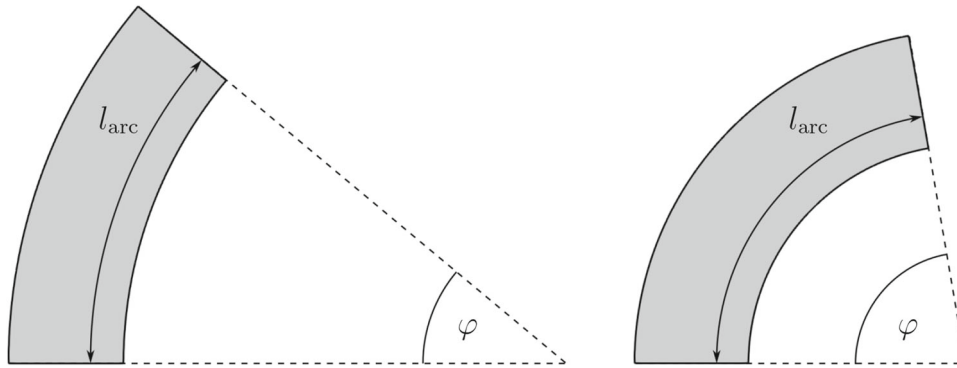
Fig. 5 Module (b): Parameter-dependent circular arc module, where the arc length $l_{arc} = 800$ is constant and the angle ϕ can be chosen between $10°$ and $90°$ to change the curvature of the module
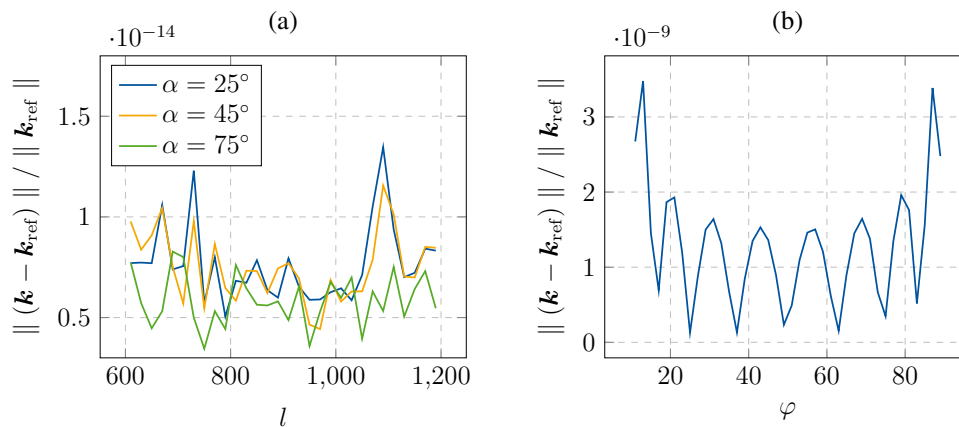

Fig. 6 Error of the stiffness data vector over a parameter range. **a** Rectangle with transversely isotropic material dependent on the length of the rectangle and the fiber orientation. **b** Circular arc module with variable arc angle

stress errors are defined analogously to Eq. (37) as

$$e_u := \frac{\|\, \boldsymbol{u} - \boldsymbol{u}_{ref}\,\|}{\|\, \boldsymbol{u}_{ref}\,\|}, \quad \text{and} \quad e_\sigma := \frac{\|\, \boldsymbol{\sigma} - \boldsymbol{\sigma}_{ref}\,\|}{\|\, \boldsymbol{\sigma}_{ref}\,\|}. \tag{38}$$

In Fig. 9 the displacement and stress error norms are plotted over the length of the top right module for different fiber directions.

The displacement error is for most parameter combinations below 0.2%. Only for some parameter values peaks occur, where the displacement error is still below 1 %. The largest peak is at $l = 650$ and $\alpha = 47.5°$. The stress errors are in general several magnitudes larger. They are mostly below 10% but the largest peak reaches 16%. The maximum here is at $l = 670$ and $\alpha = 47.5°$. To assess where these large stress errors come from the von Mises stress distributions are analyzed.

In Fig. 10 the von Mises stresses of the MOR solution, the reference solution, as well as the difference between those solutions are shown for the parameter combination with the largest errors. It can be seen that the stress distributions are very similar. Only at the substructure interfaces differences

are visible. The stress difference plot shows, that the large differences are concentrated at the boundaries of the modules and at the singularity. In large parts with large stresses, the differences are small. This suggests that locally high stress differences result in large stress error norms. The big stress differences are in regions with low stresses. To demonstrate this, we compute the stress error norm only for stresses greater than a threshold σ_{min}. In Fig. 11 the stress error is plotted over the thresholds σ_{min}. The error norms decrease with increasing threshold σ_{min}. This shows, that the large contributions to the stress error come from regions with low stresses.

It can be concluded that the MOR technique with the new methodology to compute significant snapshots for the projection matrices of the modules accurately represents the high-dimensional system. The interpolated ROM gives reasonable solutions over the whole parameter range. The peak of the stresses at $\alpha = 47.5°$ and $l = 670$ mm can be treated as an outlier. The displacement field, which is the independent variable of the system is approximated accurately over the whole parameter range.
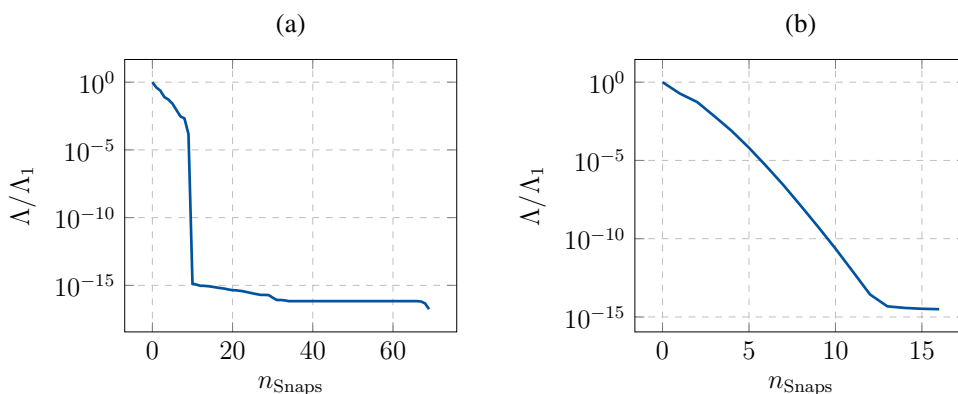
Fig. 7 Decay of normalized singular values of the modes for the DEIM approximation of stiffness matrices. **a** Rectangle with transversely isotropic material dependent on the length of the rectangle and the fiber orientation. **b** Circular arc module with variable arc angle

Fig. 8 Illustration of the geometry and mesh of the parametric system. The length l of the top right module is variable. The fiber orientation is the same in all modules. The system is loaded by the distributed load $q = 8\,\mathrm{N/mm}$
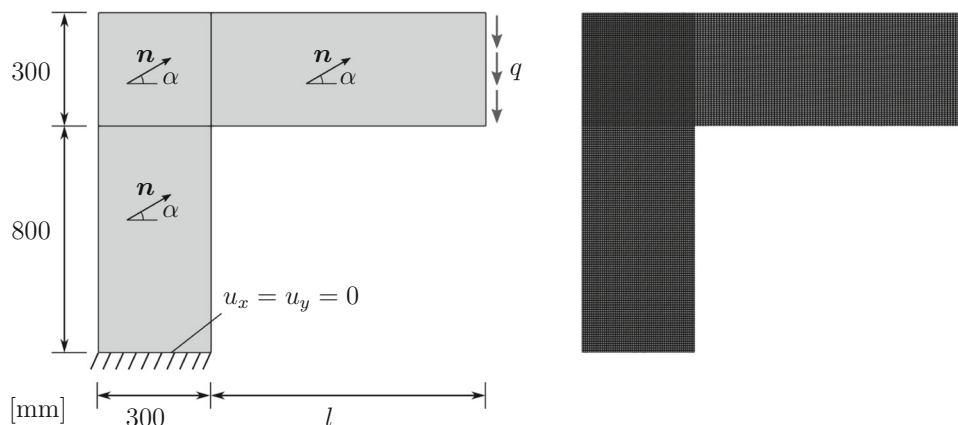
This example shows the speed up due to the MOR and that the modules can be used in different ways with the same projection matrices. The geometry of the system is shown in Fig. 12. It consists of 24 substructures with originally 266 448 DOFs. Every module is made of transversely isotropic material and has a randomly chosen fiber direction. The material parameters are shown in Table 1. The rectangular module is used 18 times, and the square module is used 6 times. The rectangular module is discretized by 6000 elements and the square module by 3600 elements. At the contact interfaces, 60 line elements are used. For each substructure, the projection matrix is interpolated from the same precomputed POD bases of the rectangular and the square module, respectively. For both modules, the projection matrices are precomputed for the parameter values $\alpha = \{0°, 10°, \dots, 90°\}$. The rectangular module has 40 POD modes and 5 DEIM modes. The square module has 52 POD modes and 5 DEIM modes.

Table 1 Material parameters for the transversely isotropic material model in 10^5 [MPa]

k_{iso}^1	k_{iso}^2	k_{ani}^1	k_{ani}^2	k_{coup}
1.39	−2.62	15.42	0.53	0.37

4.3 Example 3: Large modular system

The quadratic module is used with two, three, and four adjacent elements. In the reduced order model, the DOFs are reduced to 1032, which leads to a reduction factor of $n_{\mathrm{DOF}}/m_{\mathrm{DOF}} = 258$. The speed-up of wall-clock time is $t_{\mathrm{ref}}/t_{\mathrm{MOR}} = 15.06$. In Fig. 13 we show the relative times of the four major simulation steps related to the total simulation time of the full-order model (FOM). The computational steps are explained in Sect. 3.2. The MOR method achieved major time savings in the first step of the computation. For the ROM this step includes the reading of the mesh, computation of the stiffness-, projection- and reduced stiffness matrix by the interpolation methods, and translation and rotation of the nodal coordinates for every substructure. For the ROM,

Fig. 9 Displacement and stress error over the length of the rectangular module for different fiber orientations. The fiber orientation is the same in every module
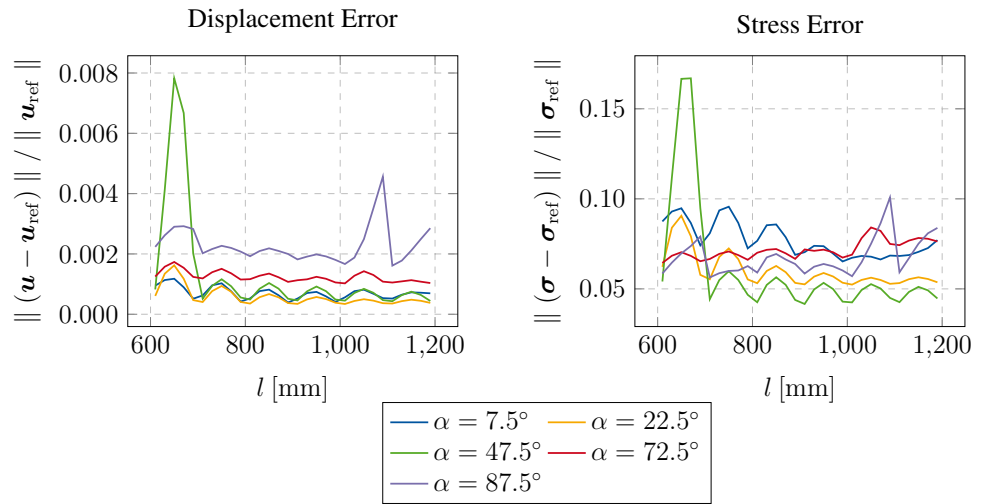
Fig. 10 Contour plots of von Mises stresses for the parameters with the largest error norms ($l = 670$, $\alpha = 47.5°$). **a** Model order reduction (MOR) solution. **b** Reference solution. **c** Difference between MOR and reference solution
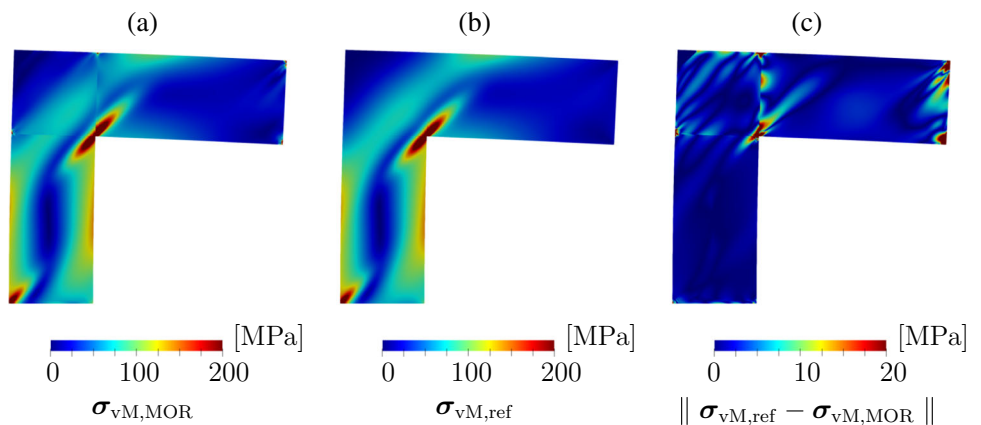
reading of the mesh and transformation of the nodal coordinates takes roughly 65% of the time of the first step. For the FOM only the stiffness matrix is computed, the other steps remain the same. The computation of the contact stiffness matrix is the same for the ROM and the FOM and takes the same amount of time. The amount of time to assemble and solve the system of equations is reduced because the number of DOF of the ROM is 258 times smaller. It has to be emphasized, that the times of the different steps depend strongly on the implementation.

The displacement and stress error norms are

$$e_u = \frac{\| \boldsymbol{u} - \boldsymbol{u}_{\text{ref}} \|}{\| \boldsymbol{u}_{\text{ref}} \|} = 0.0013, \quad \text{and}$$

$$e_\sigma = \frac{\| \boldsymbol{\sigma} - \boldsymbol{\sigma}_{\text{ref}} \|}{\| \boldsymbol{\sigma}_{\text{ref}} \|} = 0.088, \tag{39}$$

which are in the same magnitude as the errors in Sect. 4.2. Figure 14 shows the von Mises stress contour plots of the MOR solution, the reference solution, and their difference. Analogously to Sect. 4.2 the contour plots look very similar and the errors are concentrated at the interfaces between
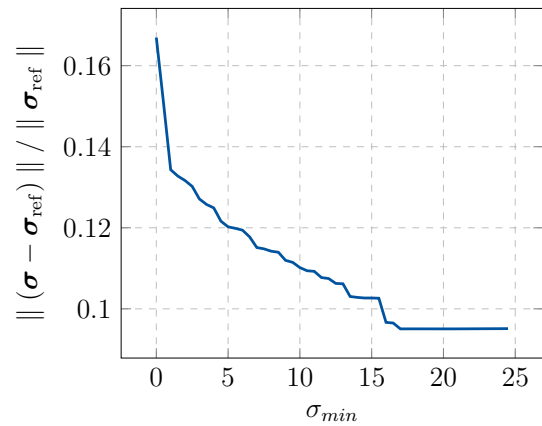
Fig. 11 Stress error plot over a minimum stress σ_{min}. For the stress error computation, only the stresses are used where $|\sigma| > \sigma_{min}$ holds. This shows that the large stress errors come from big stress differences in regions where the stresses are very low, compared to the maximum stress of $\sigma_{max} = 1517.4$ [MPa]. Removing the low-stress regions from the error norm reduces it

modules. In the regions with the highest stresses, the stresses are very accurate. This result shows, that the modules can be
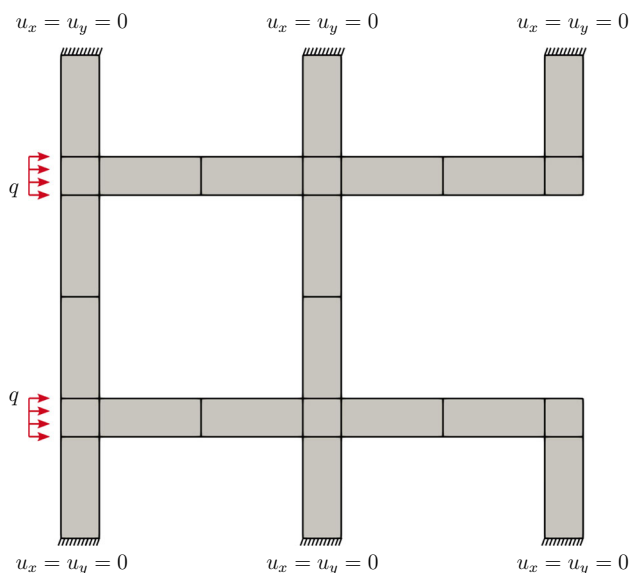
Fig. 12 Illustration of the geometry of the parametric system, where the fiber orientation in every substructure is randomly selected. The edges of the square module are 300 mm and the rectangular module has a length of 800 mm. The system is loaded by the distributed load $q = 33.3$ N/mm

used in different parts of the structure, where the boundary or contact conditions differ.

4.4 Example 4: Parameter dependent geometry

The fourth example is a frame with variable height, which is shown in Fig. 15. The system consists of seven substructures, from two different parameter-dependent modules. The rectangular module is discretized by 6000 elements and the circular arc module by 12,000 elements. At the contact interfaces, 60 line elements are used. For the parameter-dependent rectangle module, the projection matrices are precomputed at $l = \{600, 800, \ldots, 1200\}$. The module has 44 POD modes and 3 DEIM modes for the stiffness matrix. The circular arc module is precomputed at $\phi = \{10°, 20°, \ldots, 90°\}$. The arc module has 36 POD modes and 9 DEIM modes for the stiffness matrix. The stiffness matrices for the DEIM

basis are computed at $l = \{600, 700, \ldots, 1200\}$ and $\phi = \{10°, 15°, \ldots, 90°\}$, for the rectangle and the circular arc respectively.

The angle ϕ_1 controls the height of the structure because for a fixed width of $b = 3000$ mm and a constant arc length of all arc modules of $l_{arc} = 800$ mm all other parameters depend on it. The height of the vertical rectangular substructures is constant at $l_1 = 800$ mm. The material is linear elastic with Young's modulus of $E = 210{,}000$ MPa and Poissons ratio of $\nu = 0.3$. The system is loaded with $q = 50$ N/mm and $w = 20$ N/mm.

The purpose of this example is to illustrate how the method can be used. The parameter ϕ_1 is varied to find the optimal structure, which is computationally cheap because all systems can be computed with MOR. Once the optimal parameter is found, this system can be solved exactly. In our case, we search for the stiffest frame. Therefore, in Fig. 16 the maximum displacement in x- and y-direction and the sum of these two values are plotted over the parameter ϕ_1. With increasing angle ϕ_1, the maximum displacement in x-direction decreases, while the maximum y-displacement increases. As a criterion for the stiffest frame, we use the minimum of the sum of the two maximum displacements, which yields $\phi_1 = 72°$ for the given loading conditions.

For the stiffest version of this frame, the unreduced solution was also computed. Compared to the unreduced solution the reduced solution had 432 times fewer degrees of freedom and could be solved 19.6 times faster. The reason for this higher speed up compared to Sect. 4.3 is that the contact stiffness matrix was already pre-computed and was not included in the timed simulations. The displacement and stress errors are

$$e_u = \frac{\| \boldsymbol{u} - \boldsymbol{u}_{\text{ref}} \|}{\| \boldsymbol{u}_{\text{ref}} \|} = 0.0013, \quad \text{and}$$

$$e_\sigma = \frac{\| \boldsymbol{\sigma} - \boldsymbol{\sigma}_{\text{ref}} \|}{\| \boldsymbol{\sigma}_{\text{ref}} \|} = 0.032, \tag{40}$$

where the stress errors are low compared to the examples in Sects. 4.2 and 4.3, because the system does not show any stress singularity. Figure 17 shows the von Mises stresses of

Fig. 13 Bar chart of the relative times of the different simulation steps related to the simulation time of the Full Order Model (FOM)
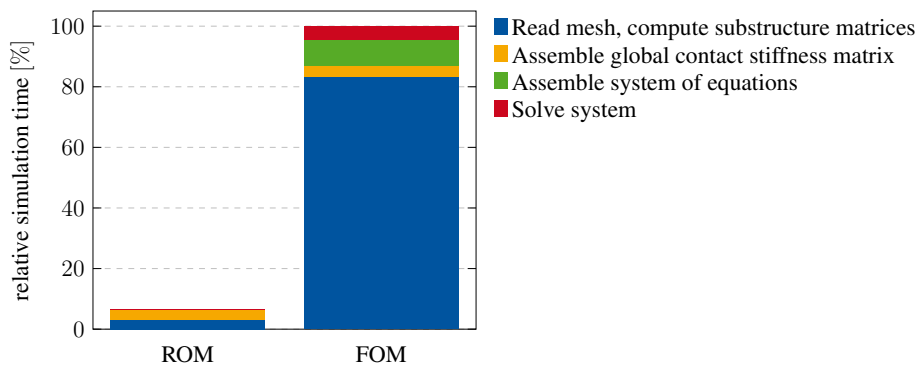
Fig. 14 Contour plots of von Mises stresses. **a** Model order reduction (MOR) solution. **b** Reference solution. **c** Difference between MOR and reference solution
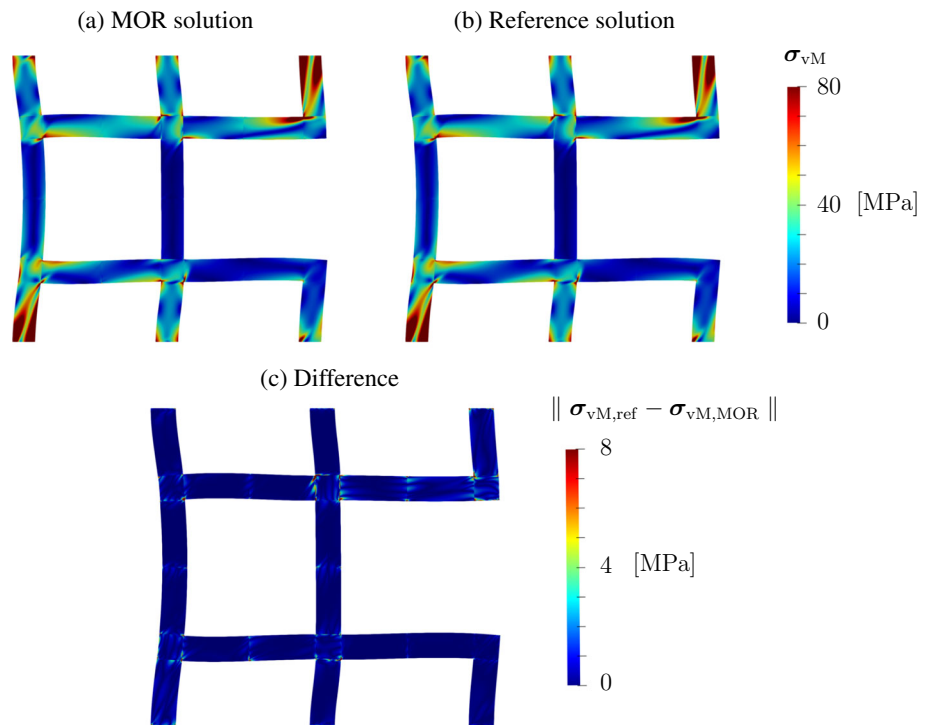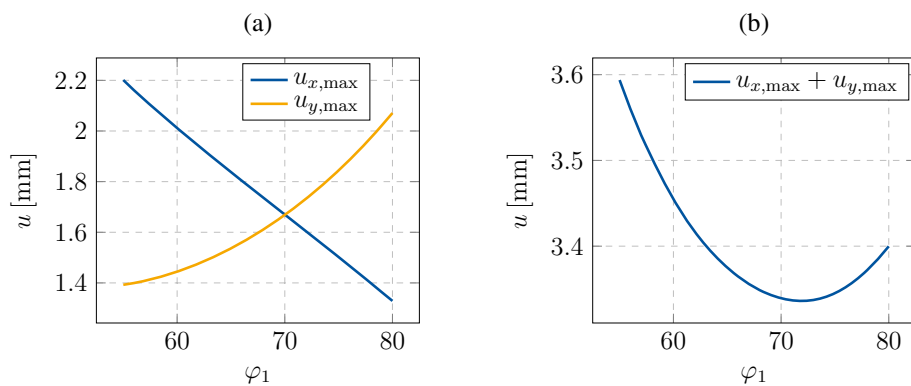
(a) MOR solution

(b) Reference solution

σ_{vM}

80

40 [MPa]

0

(c) Difference

$\| \sigma_{\mathrm{vM,ref}} - \sigma_{\mathrm{vM,MOR}} \|$

8

4 [MPa]

0

Fig. 15 Illustration of the geometry of the parametric system. The height of the system depends on the angle ϕ_1. The thickness $t = 300$mm, the width $b = 3000$ mm, and the height $l_1 = 800$mm are constant. All other parameters are derived from the angle ϕ_1. The system is loaded by the distributed loads $w = 20$ N/mm and $q = 50$ N/mm
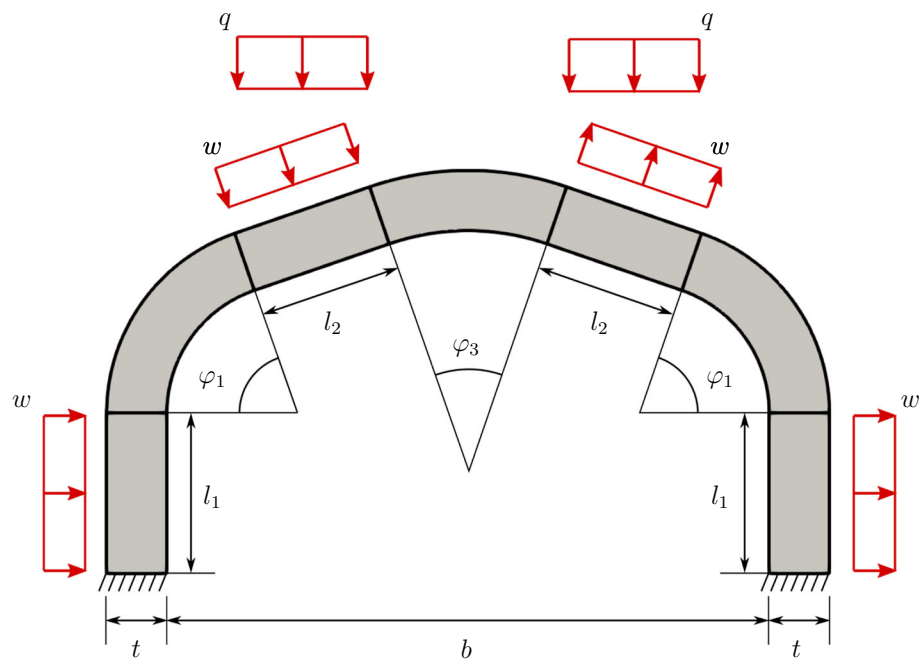
the reduced solution, the unreduced solution, and their difference. Analogously to Sect. 4.3 the stresses are accurate in most regions, while at the interfaces in the rectangular modules with variable lengths the stress differences are higher.

5 Conclusions and outlook

We developed a method to efficiently solve structures, that are assembled from parameter-dependent substructures, by connecting the substructures with a tied contact approach and applying POD-based MOR on the substructure level. It was shown that the ROM could be adapted to the parameters chosen for each substructure. To adapt the POD projec-

Fig. 16 **a** Maximum x- and y-displacement over angle ϕ_1. **b** Sum of the maximum x- and y-displacement over angle ϕ_1

Fig. 17 Contour plots of von Mises stresses. **a** Model order reduction (MOR) solution. **b** Reference solution. **c** Difference between MOR and reference solution
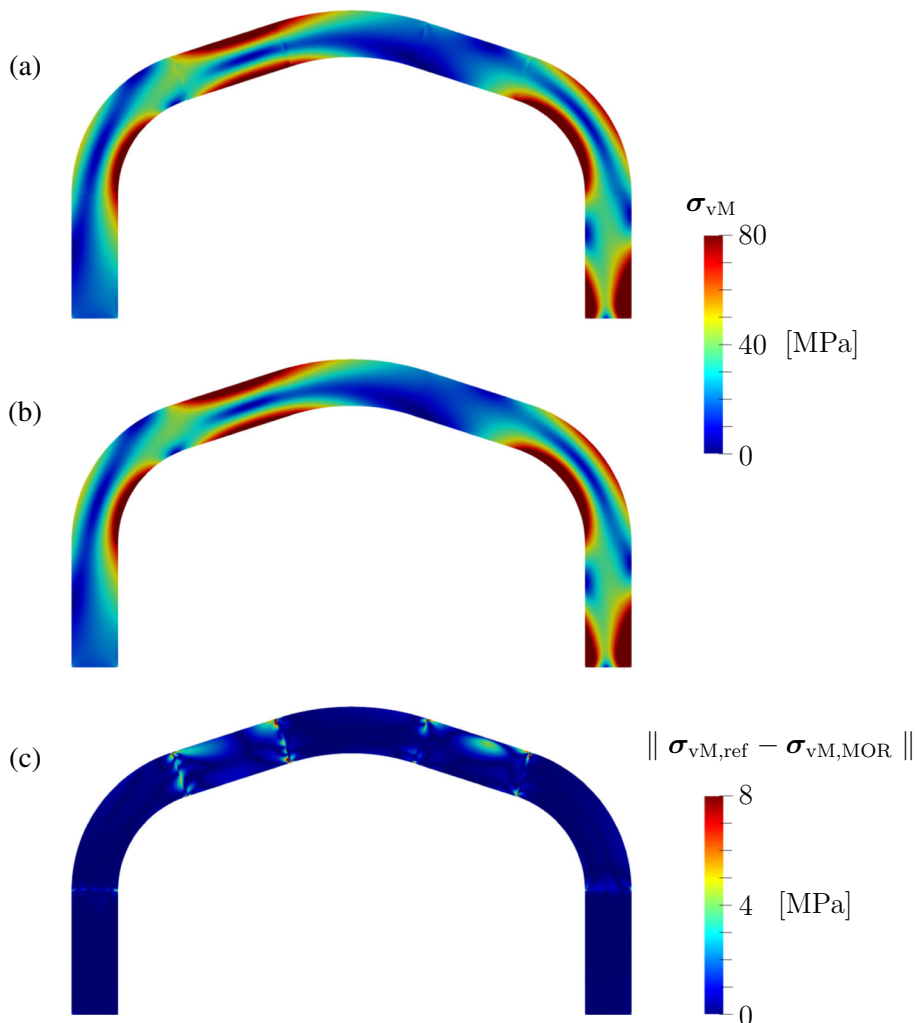
tion matrices, we used an interpolation method on geodesics on Grassmann manifolds. For the adaption of the stiffness matrix, we presented a way to apply DEIM to parameter-dependent stiffness matrices. This was necessary because our method requires rigid body displacement modes in the projection matrices, which lead to indefinite reduced stiffness matrices. Therefore, matrix manifold interpolation methods could not be used because they require semi-positive definite

matrices. Moreover, we developed a method to compute significant snapshots for POD modes of the modules that allow for flexible use of these modules in global structures.

The numerical examples show, that the new method to compute snapshots leads to an accurate representation of the displacement field. The stress field is also approximated quite well, but at interfaces and singularities, the differences to the reference solution can get relatively large locally,

which drives up the error norms. Moreover, it was shown, how the ROM of modular structures is adapted to parameter changes of the modules by interpolation. The stiffness matrices of each module were approximated accurately with DEIM. These interpolations lead to a significant reduction of the computation time (cf. Sects. 4.3, 4.4). The slowest parts of the reduced simulation are the computation of the contact stiffness matrix and the computation of the reduced stiffness matrices of the substructures. The projection of the substructure stiffness matrices into the reduced space still scales with the original problem. This drawback has to be addressed in future works. Finally, it was shown, that the proposed method is well-suited to find optimal parameters for parameter-dependent structures. The efficient adaption of the matrices makes simulations over whole parameter ranges computationally cheap. These solutions are then compared to find optimal parameters. Optimization techniques like the gradient descent algorithm could also be applied. The necessary gradients are cheap to compute because of the adaptive model order reduction.

In the future, the method will be extended to geometrically and materially nonlinear problems. The main difference would be that the system is solved by a Newton–Raphson scheme and the tangential stiffness matrix as well as the internal force vector would have to be approximated by DEIM. Furthermore, it will be applied to modular carbon-fiber reinforced concrete shell structures.

Code and Data Availability statement All meshes and the projection matrices for the numerical examples are stored in the repository [35].

A Appendix

A.1 Bilinear interpolation on Grassmann Manifolds for two-dimensional parameter spaces

For the numerical example in Sect. 4.2 we need to interpolate the projection matrices in two parameter dimensions. In the following, we will briefly explain the bilinear interpolation scheme we used to compute the projection matrix for a target parameter point $(\bar{p}, \bar{q})$. In a pre-computation step, we compute POD-bases Ψ for every grid point (p_i, q_j) from the sampling parameter sets $P = \{p_1, p_2, \ldots, p_k\}$ and $Q = \{q_1, q_2, \ldots, q_k\}$. We will now explain the interpolation scheme to compute the projection matrix The first step is to find the two closest sampling parameters to each target parameter, such that $p_i < \bar{p} < p_j$ and $q_k < \bar{q} < q_l$. The interpolation is then performed between the projection matrices of the four points: (p_i, q_k), (p_i, q_l), (p_j, q_k), and (p_j, q_l). We choose the projection matrix of (p_i, q_k) as our origin point $\Psi_o := \Psi_{ik}$. Then we compute the points in the tangent space Γ_{il}, Γ_{jl}, and Γ_{jkl} for all three other parameter points (p_i, q_l), (p_j, q_k), and (p_j, q_l), by using Eq. (34). In the next step, we interpolate the in this tangent space by a bilinear interpolation [31]:

$$\bar{\Gamma} = \frac{1}{(p_j - p_i)(q_l - q_k)} \begin{bmatrix} p_j - \bar{p} & \bar{p} - p_i \end{bmatrix} \begin{bmatrix} \mathbf{0} & \Gamma_{ik} \\ \Gamma_{il} & \Gamma_{jk} \end{bmatrix}$$
$$\begin{bmatrix} q_l - \bar{q} \\ \bar{q} - q_k \end{bmatrix}. \tag{41}$$

Finally, we compute the projection matrix $\bar{\Psi}$ for the point $(\bar{p}, \bar{q})$ by the exponential map from Eq. (36).

References

1. Allen MS, Rixen D, Van der Seijs M, Tiso P, Abrahamsson T, Mayes RL (2020) Substructuring in engineering dynamics. Springer, New York
2. Amsallem D, Cortial J, Carlberg K, Farhat C (2009) A method for interpolating on manifolds structural dynamics reduced-order models. Int J Numer Meth Eng 80(9):1241–1258
3. Amsallem D, Farhat C (2008) Interpolation method for adapting reduced-order models and application to aeroelasticity. AIAA J 46(7):1803–1813
4. Antoulas AC (2005) An overview of approximation methods for large-scale dynamical systems. Annu Rev Control 29(2):181–190
5. Bai Z, Demmel J, Dongarra J, Ruhe A, van der Vorst H (2000) Templates for the solution of algebraic eigenvalue problems: a practical guide. SIAM, Philadelphia
6. Barrault M, Maday Y, Nguyen NC, Patera AT (2004) An 'empirical interpolation' method: application to efficient reduced-basis dis-

cretization of partial differential equations. CR Math 339(9):667–672
7. Benner P, Gugercin S, Willcox K (2015) A survey of projection-based model reduction methods for parametric dynamical systems. SIAM Rev 57(4):483–531
8. Besselink B, Tabak U, Lutowska A, Van de Wouw N, Nijmeijer H, Rixen DJ, Hochstenbach M, Schilders W (2013) A comparison of model reduction techniques from structural dynamics, numerical mathematics and systems and control. J Sound Vib 332(19):4403–4422
9. Bonomi D, Manzoni A, Quarteroni A (2017) A matrix deim technique for model reduction of nonlinear parametrized problems in cardiac mechanics. Comput Methods Appl Mech Eng 324:300–326
10. Breuer KS, Sirovich L (1991) The use of the Karhunen–Loeve procedure for the calculation of linear eigenfunctions. J Comput Phys 96(2):277–296
11. Brunton SL, Kutz JN (2022) Data-driven science and engineering: machine learning, dynamical systems, and control. Cambridge University Press, Cambridge
12. Bui-Thanh T, Willcox K, Ghattas O (2008) Parametric reduced-order models for probabilistic analysis of unsteady aerodynamic applications. AIAA J 46(10):2520–2529
13. Carlberg K, Tuminaro R, Boggs P (2015) Preserving Lagrangian structure in nonlinear model reduction with application to structural dynamics. SIAM J Sci Comput 37(2):B153–B184
14. Chaturantabut S, Sorensen DC (2010) Nonlinear model reduction via discrete empirical interpolation. SIAM J Sci Comput 32(5):2737–2764
15. Craig RR Jr, Bampton MC (1968) Coupling of substructures for dynamic analyses. AIAA J 6(7):1313–1319
16. de Klerk D, Rixen DJ, Voormeeren S (2008) General framework for dynamic substructuring: history, review and classification of techniques. AIAA J 46(5):1169–1181
17. Drmac Z, Gugercin S (2016) A new selection operator for the discrete empirical interpolation method—improved a priori error bound and extensions. SIAM J Sci Comput 38(2):A631–A648
18. Everson R, Sirovich L (1995) Karhunen–Loeve procedure for gappy data. JOSA A 12(8):1657–1664
19. Farah P, Popp A, Wall WA (2015) Segment-based vs. element-based integration for mortar methods in computational contact mechanics. Comput Mech 55(1):209–228
20. Friderikos O, Baranger E, Olive M, Neron D (2022) On the stability of POD basis interpolation on Grassmann manifolds for parametric model order reduction. Comput Mech 70:1–24
21. Grepl MA, Maday Y, Nguyen NC, Patera AT (2007) Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. ESAIM Math Model Numer Anal 41(3):575–605
22. Hernández J, Oliver J, Huespe AE, Caicedo M, Cante J (2014) High-performance model reduction techniques in computational multiscale homogenization. Comput Methods Appl Mech Eng 276:149–189
23. Hong S-K, Epureanu BI, Castanier MP, Gorsich DJ (2011) Parametric reduced-order models for predicting the vibration response of complex structures with component damage and uncertainties. J Sound Vib 330(6):1091–1110
24. Hurty WC (1965) Dynamic analysis of structural systems using component modes. AIAA J 3(4):678–685
25. Kastian S, Moser D, Grasedyck L, Reese S (2020) A two-stage surrogate model for neo-Hookean problems based on adaptive proper orthogonal decomposition and hierarchical tensor approximation. Comput Methods Appl Mech Eng 372:113368
26. Kerschen G, Golinval J-C, Vakakis AF, Bergman LA (2005) The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: an overview. Nonlinear Dyn 41(1):147–169
27. Lieu, T, Farhat C (2005) Adaptation of POD-based aeroelastic ROMs for varying Mach number and angle of attack: application to a complete F-16 configuration. In: 2005 US air force T&E Days', p 7666
28. Negri F, Manzoni A, Amsallem D (2015) Efficient model reduction of parametrized systems by matrix discrete empirical interpolation. J Comput Phys 303:431–454
29. Pennec X, Fillard P, Ayache N (2006) A Riemannian framework for tensor computing. Int J Comput Vision 66(1):41–66
30. Popp, A, Wriggers, P. (2018) Contact modeling for solids and particles, Vol. 585, Springer, New York
31. Press WH (2007) Numerical recipes. The art of scientific computing, 3rd edn. Cambridge University Press, Cambridge
32. Radermacher A, Reese S (2014) Model reduction in elastoplasticity: proper orthogonal decomposition combined with adaptive sub-structuring. Comput Mech 54(3):677–687
33. Radermacher A, Reese S (2016) Pod-based model reduction with empirical interpolation applied to nonlinear elasticity. Int J Numer Meth Eng 107(6):477–495
34. Reese S, Raible T, Wriggers P (2001) Finite element modelling of orthotropic material behaviour in pneumatic membranes. Int J Solids Struct 38(52):9525–9544
35. Ritzert, S., Macek, D., Simon, J.-W, Reese, S. (2023) Projection matrices for numerical examples of "An adaptive model order reduction technique for parameter-dependent modular structures. https://doi.org/10.5281/zenodo.8246556
36. Zhou L, Simon J-W, Reese S (2018) Proper orthogonal decomposition for substructures in nonlinear finite element analysis: coupling by means of tied contact. Arch Appl Mech 88(11):1975–2001