



Roman Sartorti · Wadhah Garhuom · Alexander Düster

# Remeshing and eigenvalue stabilization in the finite cell method for structures undergoing large elastoplastic deformations

Received: 23 November 2023 / Accepted: 4 July 2024 / Published online: 20 July 2024  
© The Author(s) 2024

**Abstract** Large strain analysis is a challenging task, especially in fictitious or immersed boundary domain methods, since badly broken elements/cells can lead to an ill-conditioned global tangent stiffness matrix, resulting in convergence problems of the incremental/iterative solution approach. In this work, the finite cell method is employed as a fictitious domain approach, in conjunction with an eigenvalue stabilization technique, to ensure the stability of the solution procedure. Additionally, a remeshing strategy is applied to accommodate highly deformed configurations of the geometry. Radial basis functions and inverse distance weighting interpolation schemes are utilized to map the displacement gradient and internal variables between the old and new meshes during the remeshing process. For the first time, we demonstrate the effectiveness of the remeshing approach using various numerical examples in the context of finite strain elastoplasticity.

**Keywords** Elastoplasticity · Finite strains · Finite cell method · Remeshing · Data transfer · Stabilization

## 1 Introduction

The motivation of fictitious domain or immersed boundary methods, such as finite cell method (FCM) [1,2], cutFEM [3–5] or cutIGA [6], lies in their ability to handle arbitrary structures. This aim is usually achieved by creating simple meshes based on Cartesian grids while the geometry is described using an indicator function. The FCM has already been further developed to various fields of application. Thermoelastic [7], geometrically nonlinear [8,9] and elastoplastic [10–12] analyses have been performed on different numerical examples. Moreover, the FCM has successfully been applied to biomechanics [13–16], vibroacoustics [17] as well as in implicit [18] and explicit dynamics [19].

Since the mesh does not conform to the physical boundaries, some elements/cells can be badly broken, i.e., containing only a small amount of material. This can lead to ill-conditioning of the global stiffness matrix. To this end, different stabilization techniques are available. One straightforward approach is the  $\alpha$ -method [1,2], where a soft material is assumed in the fictitious domain. This is usually obtained by multiplying the material

---

All authors have contributed equally to this work.

---

R. Sartorti (✉) · W. Garhuom · A. Düster  
Numerical Structural Analysis with Application in Ship Technology, Institute for Ship Structural Design and Analysis (M-10),  
Hamburg University of Technology, Am Schwarzenberg Campus 4 (C), 21073 Hamburg, Germany  
E-mail: roman.sartorti@tuhh.de

W. Garhuom  
E-mail: wadhah.garhuom@tuhh.de

A. Düster  
E-mail: alexander.duester@tuhh.de

parameters such as the Young's modulus by a small positive number  $\alpha_0$ . This works well for linear as well as nonlinear material models as long as the strains remain small. However, it must be noted that the original problem is modified if  $\alpha_0$  is different from zero—leading to inaccuracies in the final solution.

Another way to stabilize the solution procedure of the large strain analysis is the eigenvalue stabilization (EVS) presented in [20] based on [21]. Thereby, the smallest nonzero eigenvalues and corresponding eigenvectors of the broken cell stiffness matrices are determined, because they are the main reason for the ill-conditioning of the tangent stiffness matrix. Then the eigenvectors are used to compute a stabilization term for the element stiffness matrix. The appealing property of this method is that the error induced in the stiffness matrix due to the stabilization part can be corrected by modifying the right-hand side (RHS) as well. Therefore, the physical system under consideration does not change.

Additionally, with regard to stabilizing the solution procedure, remeshing has successfully been applied to reach large deformation states [9, 22, 23]. This is achieved by creating a new mesh, once a critical deformation state is reached, to avoid the problem of heavily distorted and badly broken cells. In this context, there arises the additional task to transfer the deformation history to the new mesh. To this end, local radial basis functions (RBFs) are applied. This, however, requires the solution of many linear systems of equations. Therefore, it is of further interest whether other interpolation schemes such as the inverse distance weighting (IDW) could be applied to accelerate the data transfer.

In [20], the EVS has successfully been applied to linear and nonlinear FCM analyses including hyperelastic material. In the present work, we extend this framework to elastoplastic materials in connection with a remeshing procedure, along with different approaches for transferring the history variables between the different meshes during the remeshing procedure.

The rest of the paper is organized as follows: In Sect. 2, the basic formulation of the FCM for large strain analysis is provided as well as a brief summary of the elastoplastic material model. Section 3 details the remeshing procedure and explores the different methods used for data transfer. Section 4 focuses on preliminary investigations regarding the influence of the volume fraction on the conditioning of the stiffness matrix. Then, we discuss different numerical examples in Sect. 5 before ending with concluding remarks in Sect. 6.

## 2 Finite cell method

### 2.1 Basic formulation for nonlinear problems

The basic concept of the FCM is depicted in Fig. 1. The idea is that an arbitrary complex physical domain  $\Omega$  is embedded into a bigger domain of simple shape. This extended domain  $\Omega_e$  is discretized using simple Cartesian meshes. An indicator function  $\alpha(\mathbf{x})$  is introduced to distinguish the physical from the fictitious domain. This is defined by

$$\alpha(\mathbf{x}) = \begin{cases} 1, & \forall \mathbf{x} \in \Omega \\ \alpha_0, & \forall \mathbf{x} \in \Omega_e \setminus \Omega \end{cases} \quad (1)$$

where  $\alpha_0$  is typically a small positive number. Please note, that, when choosing  $\alpha_0 = 0$ , the exact geometry is recovered. The weak form reads

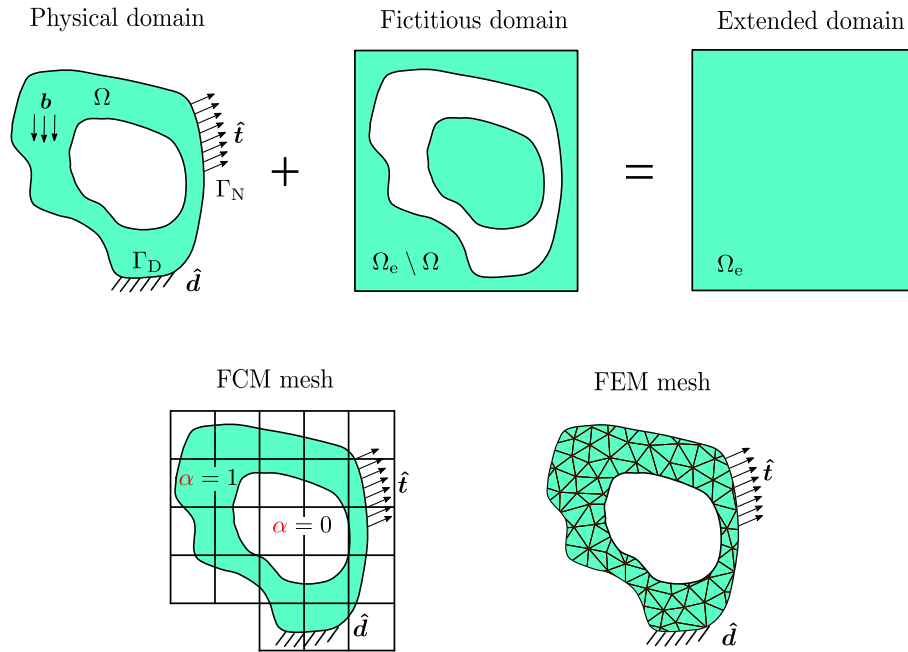
$$G_e^\alpha = \int_{\Omega_e} \alpha \mathbf{P} \cdot \text{Grad} \boldsymbol{\eta} \, d\Omega - \int_{\Gamma_N} \hat{\mathbf{t}} \cdot \boldsymbol{\eta} \, d\Gamma = 0 \quad (2)$$

where  $\mathbf{P}$  is the first Piola–Kirchhoff stress tensor,  $\hat{\mathbf{t}}$  are the tractions applied on the boundary  $\Gamma_N$  and  $\boldsymbol{\eta}$  represents the test function. From Eq. (2), it can be seen that if  $\alpha_0 = 0$ , the equation simplifies to

$$G_e^\alpha = \int_{\Omega} \mathbf{P} \cdot \text{Grad} \boldsymbol{\eta} \, d\Omega - \int_{\Gamma_N} \hat{\mathbf{t}} \cdot \boldsymbol{\eta} \, d\Gamma = 0 \quad (3)$$

which indicates that the governing weak form is defined on the physical domain of the problem. For the purpose of stabilization, we assign a small positive value  $\alpha_0$ , see also Section 4.1. In order to prepare the application of the Newton–Raphson method, we first linearize the weak form (2). Therefore, we compute the directional derivative w.r.t. a displacement increment  $\Delta \mathbf{d}$  around displacement state  $\bar{\mathbf{d}}$

$$DG_e^\alpha(\bar{\mathbf{d}}, \boldsymbol{\eta}) \cdot \Delta \mathbf{d} = \int_{\Omega_e} \alpha A(\bar{\mathbf{d}}) \text{Grad} \Delta \mathbf{d} \cdot \text{Grad} \boldsymbol{\eta} \, d\Omega \quad (4)$$



**Fig. 1** Visualization of the principle idea of the finite cell method. The physical domain  $\Omega$  is embedded into a fictitious domain, such that the extended domain  $\Omega_e$  can simply be meshed using a Cartesian grid. The indicator function is used to distinguish the void domain ( $\alpha = 0$ ) from the physical domain ( $\alpha = 1$ ). In contrast to the FEM, the mesh is not boundary conforming

where

$$\mathbf{A}(\bar{\mathbf{d}}) = \left. \frac{\partial \mathbf{P}}{\partial \mathbf{F}} \right|_{\mathbf{d}=\bar{\mathbf{d}}}. \quad (5)$$

This leads to the linearized weak form

$$\int_{\Omega_e} \alpha \mathbf{A}(\bar{\mathbf{d}}) \text{Grad} \Delta \mathbf{d} \cdot \text{Grad} \boldsymbol{\eta} \, d\Omega = \int_{\Gamma_N} \hat{\mathbf{t}} \cdot \boldsymbol{\eta} \, d\Gamma - \int_{\Omega_e} \alpha \mathbf{P} \cdot \text{Grad} \boldsymbol{\eta} \, d\Omega. \quad (6)$$

Next, the weak form is discretized using finite cells, as illustrated in Fig. 1. Thereby, the displacement field  $\mathbf{d}$  and the test function  $\boldsymbol{\eta}$  are discretized with the same set of hierarchic shape functions, see [2, 24]. Within the incremental/iterative solution procedure, the Newton–Raphson method is applied in every load step  $k$ , resulting in

$$\mathbf{K}_T^i(\mathbf{d}_k^i) \Delta \mathbf{d}^{i+1} = -\mathbf{G}^i(\mathbf{d}_k^i), \quad (7)$$

where  $\mathbf{K}_T^i(\mathbf{d}_k^i)$  represents the tangent stiffness matrix and  $\mathbf{G}^i(\mathbf{d}_k^i)$  defines the out-of-balance vector at iteration  $i$ . Having solved the system of equations, the total displacement is updated by

$$\mathbf{d}_k^{i+1} = \mathbf{d}_k^i + \Delta \mathbf{d}^{i+1} \quad (8)$$

until the solution converges within a specified tolerance. During this procedure each cell contributes to the global stiffness matrix and the out-of-balance vector. Therefore, they are assembled from the individual contributions of each cell, i.e.,

$$\mathbf{K}_T^i = \mathbf{A} \sum_{c=1}^{N_c} \mathbf{k}^{c,i}, \quad \mathbf{G}^i = \mathbf{A} \sum_{c=1}^{N_c} \mathbf{g}^{c,i}. \quad (9)$$

where  $N_c$  is the total number of cells and  $\mathbf{k}^c$  and  $\mathbf{g}^i$  define the cell stiffness and out-of-balance vector, respectively. A penalty method is applied to enforce Dirichlet boundary conditions on the immersed interface, see [7].

Due to the immersed interface within the broken cells, a discontinuity is introduced. Thus, standard Gaussian quadrature cannot be used for accurate numerical integration of those cells. Therefore, special integration schemes are required. One approach is the spacetree method [1]. In order to address the discontinuity, this

method subdivides every broken cell until a certain level  $k_{ST}$  is reached. However, this method introduces a lot of additional integration points. If the stiffness matrix has to be integrated multiple times, this method is not efficient. Therefore, in the present work, we apply the non-negative moment fitting quadrature (NNMF) for the integration of the cut cells, as proposed in [20,25]. This method computes the weights of the integration points individually for each broken cell. This reduces the number of integration points compared to spacetree methods. Therefore, this method is especially appealing for applications, where integrals have to be evaluated multiple times, such as required during elastoplastic simulations.

## 2.2 Finite $J_2$ elastoplasticity

In the present work, we apply the finite  $J_2$  plasticity model, which is based on the von Mises  $J_2$  flow theory of plasticity. This model assumes only the deviatoric part of the stress tensor is related to the plastic deformation. This is a widely used model, so the interested reader may be referred to [26,27].

We use a multiplicative split of the deformation gradient  $\mathbf{F}$  into a plastic part  $\mathbf{F}_p$  and an elastic part  $\mathbf{F}_e$ . Based on this split, the elastic part can be recovered by

$$\mathbf{F}_e = \mathbf{F}\mathbf{F}_p^{-1}, \quad (10)$$

and the elastic right and left Cauchy–Green tensors  $\mathbf{b}_e$  and  $\mathbf{C}_e$ , respectively, are defined by

$$\mathbf{C}_e = \mathbf{F}_p^{-\top} \mathbf{C} \mathbf{F}_p \quad (11)$$

$$\mathbf{b}_e = \mathbf{F} \mathbf{C}_p^{-1} \mathbf{F}^\top, \quad \text{where } \mathbf{C}_p = \mathbf{F}_p^\top \mathbf{F}_p. \quad (12)$$

For the elastic part, a hyperelastic compressible Neo–Hookean material model is assumed. The material is described in terms of the strain energy function

$$W_e = \frac{\lambda}{4} (I_3 - 1 - \ln I_3) + \frac{\mu}{2} (I_1 - 3 - \ln I_3), \quad (13)$$

where Cauchy–Green tensor invariants are defined as  $I_1 = \text{tr}(\mathbf{C}_e)$  and  $I_3 = \det \mathbf{C}_e$ . The Kirchhoff stress tensor can be computed by

$$\boldsymbol{\tau} = \frac{\partial W_e}{\partial \mathbf{F}_e} \mathbf{F}_e^\top. \quad (14)$$

With this, we can compute the deviatoric part of the stress tensor

$$\mathbf{s} = \boldsymbol{\tau} - \frac{1}{3} \text{tr}(\boldsymbol{\tau}) \mathbf{I}, \quad (15)$$

where  $\mathbf{I}$  is the second-order identity tensor. The von Mises yield criterion of the plasticity model reads

$$\Phi(\boldsymbol{\tau}, \bar{\alpha}) = \sqrt{\frac{3}{2} \mathbf{s} : \mathbf{s}} - K(\bar{\alpha}), \quad (16)$$

which is not only dependent on the Kirchhoff stress, but also on the equivalent plastic strain  $\bar{\alpha}$ . The function  $K(\bar{\alpha})$  describes the nonlinear isotropic hardening

$$K(\bar{\alpha}) = Y_0 + H\bar{\alpha} + (Y_\infty - Y_0)(1 - e^{-\delta\bar{\alpha}}). \quad (17)$$

The constants  $Y_0$  and  $Y_\infty$  describe the initial yield stress and the saturation stress, respectively. Additionally,  $H$  defines the linear hardening parameter and  $\delta$  is the hardening exponent. According to [28], the evolution of the equivalent plastic strain  $\bar{\alpha}$  and the internal variables can be described by

$$-\frac{1}{2} \mathcal{L}_v \mathbf{b}_e = \gamma \frac{\partial \Phi(\boldsymbol{\tau}, \bar{\alpha})}{\partial \boldsymbol{\tau}} \mathbf{b}_e, \quad (18)$$

$$\dot{\bar{\alpha}} = -\gamma \frac{\partial \Phi(\boldsymbol{\tau}, \bar{\alpha})}{\partial K(\bar{\alpha})}, \quad (19)$$

where  $\gamma \geq 0$  is the non-negative plastic multiplier and  $\mathcal{L}_v \mathbf{b}_e$  is the Lie-derivative of the elastic left Cauchy–Green tensor  $\mathbf{b}_e$ . Furthermore, the plastic flow rule can also be expressed for the left Cauchy–Green tensor defined as

$$\dot{\mathbf{C}}_p^{-1} = -2\gamma \mathbf{F}^{-1} \frac{\partial \Phi(\boldsymbol{\tau}, \bar{\alpha})}{\partial \boldsymbol{\tau}} \mathbf{F} \mathbf{C}_p^{-1}. \quad (20)$$

Please note that the material model must satisfy the Karush–Kuhn–Tucker conditions for both loading and unloading, i.e.,

$$\Phi \leq 0, \quad \gamma \geq 0, \quad \Phi \gamma = 0. \quad (21)$$

The integration of the internal variable is carried out with an implicit Euler scheme. Thus, the equivalent plastic strain is updated by

$$\bar{\alpha}_{n+1} = \bar{\alpha}_n + \Delta_t \gamma \quad (22)$$

where  $\Delta_t \gamma$  is the increment of the plastic strain multiplier. In addition, an exponential map algorithm is applied within an implicit time integration scheme to integrate the plastic variable  $\dot{\mathbf{C}}_p^{-1}$  as introduced by [29]. The plastic variable is then updated by evaluating

$$\mathbf{C}_{p,n+1}^{-1} = \mathbf{F}_{n+1}^{-1} \exp \left[ -2\Delta_t \gamma \frac{\partial \Phi_{n+1}}{\partial \boldsymbol{\tau}_{n+1}} \right] \mathbf{F}_{n+1} \mathbf{C}_{p,n}^{-1}. \quad (23)$$

For the linearized weak form the first Piola–Kirchhoff stress tensor  $\mathbf{P}$  is required. This can be computed by exploiting the relation

$$\mathbf{P} = \boldsymbol{\tau} \mathbf{F}_e^{-\top}. \quad (24)$$

For further details, the reader is referred to the references [30,31].

### 3 Remeshing and data transfer

#### 3.1 Kinematic relations

The remeshing procedure presented in this section is based on the total Lagrangian formulation, relating the current configuration  $\Omega_n$  to the initial configuration  $\Omega_0$ . The total displacement can, thus, be computed as the difference between the current position vector  $\mathbf{x}_{n+1}$  and the initial position vector  $\mathbf{X}$  by

$$\mathbf{d} = \mathbf{x}_{n+1} - \mathbf{X}. \quad (25)$$

If intermediate configurations are involved, the total displacement can also be rewritten in terms of the partial displacements between the different deformation states, i.e.,

$$\mathbf{d} = \tilde{\mathbf{d}}_1 + \tilde{\mathbf{d}}_2 + \cdots + \tilde{\mathbf{d}}_n + \tilde{\mathbf{d}}_{n+1} = \mathbf{d}_n + \tilde{\mathbf{d}}_{n+1}, \quad (26)$$

where  $\mathbf{d}_n$  is referred to as pre-displacement, summing the displacement that the structure has experienced until state  $\Omega_n$ . During the remeshing process, different deformation states and relations between the meshes are considered, as shown in Fig. 2. The initial configuration  $\Omega_0$  is deformed up to certain state  $\Omega_n$ . To this end, the deformation gradient  $\mathbf{F}_n$  can be computed directly from

$$\mathbf{F}_n = \frac{\partial \mathbf{x}_n}{\partial \mathbf{X}} = \frac{\partial (\mathbf{X} + \mathbf{d}_n)}{\partial \mathbf{X}} = \mathbf{I} + \frac{\partial \mathbf{d}_n}{\partial \mathbf{X}} = \mathbf{I} + \mathbf{H}_n, \quad (27)$$

where  $\mathbf{H}_n$  is the pre-displacement gradient at state  $\Omega_n$ . To compute the total deformation gradient  $\mathbf{F}$  we can employ the chain rule leading to

$$\mathbf{F} = \frac{\partial \mathbf{x}_{n+1}}{\partial \mathbf{x}_n} \frac{\partial \mathbf{x}_n}{\partial \mathbf{X}} = \tilde{\mathbf{F}}_{n+1} \mathbf{F}_n. \quad (28)$$

The total displacement gradient  $\mathbf{H}$  can also be constructed using the partial displacement gradients of the intermediate deformation states:

$$\mathbf{H} = \frac{\partial \mathbf{d}_n}{\partial \mathbf{X}} + \frac{\partial \tilde{\mathbf{d}}_{n+1}}{\partial \mathbf{x}_n} \frac{\partial \mathbf{x}_n}{\partial \mathbf{X}} = \mathbf{H}_n + \tilde{\mathbf{H}}_{n+1} (\mathbf{I} + \mathbf{H}_n) \quad (29)$$

In any case the pre-displacement or pre-deformation gradient are required. In the context of the FCM, this means that if a new mesh is created at deformation state  $\Omega_n$  the pre-deformation information has to be transferred to the new mesh. This data transfer will be addressed in Sect. 3.4

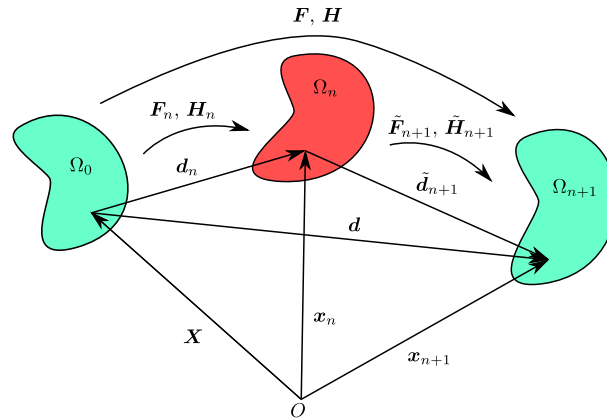


Fig. 2 Kinematic relations involved in the deformation of an arbitrary structure

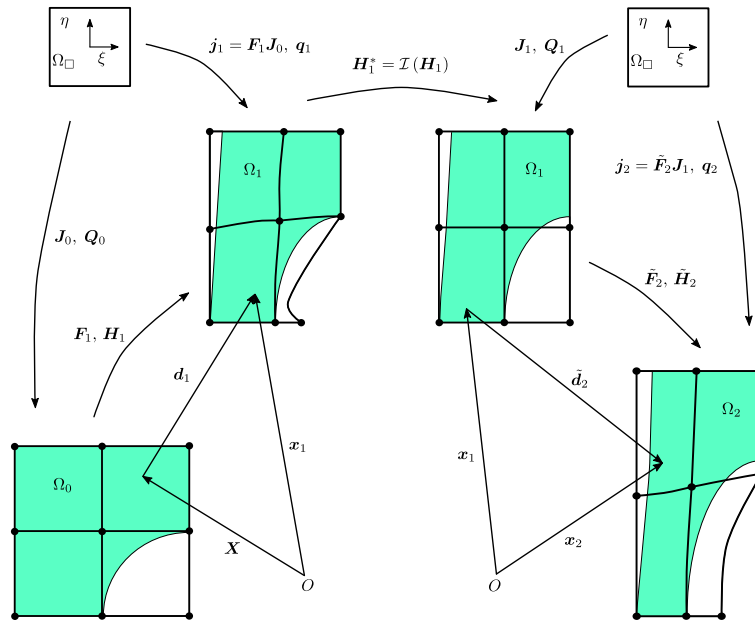


Fig. 3 Sketch of the different deformation states along with the quantities involved. Thereby  $\Omega_\square$  is a unit reference cell. The initial configuration  $\Omega_0$  is deformed until state  $\Omega_1$ , where remeshing is applied. During the remeshing the old displacement gradients  $H_1$  of deformation state  $\Omega_1$  are transferred to the new mesh via the interpolation  $\mathcal{I}$ . Then the mesh is deformed until final state  $\Omega_2$  is reached

### 3.2 Remeshing procedure

The procedure within the FCM context for one remeshing step is shown in Fig. 3 for a simple plate with a circular hole. The original structure is deformed until state  $\Omega_1$ . At this state, remeshing is initialized and a new, undeformed mesh is created based on the deformed geometry. To continue with the simulation, the displacement gradient is transferred to the integration points of the new mesh. Once the interpolation is done, the simulation can be continued until state  $\Omega_2$ , where either the desired deformation is reached or another remeshing is initiated.

The remeshing procedure is summarized in Algorithm 1. Note that there are three possibilities to trigger the remeshing: first, if the remeshing criteria limit (see Sect. 3.3) is reached. The second possibility is that the iterative solution does not converge and, lastly, the remeshing is triggered if a prescribed displacement is reached.

**Algorithm 1** Remeshing procedure scheme [22]

---

```

1: set mesh number  $m = 1$ 
2: set remeshing criteria threshold  $\rho$ 
3: Create mesh and apply boundary conditions
4: if ( $m > 1$ ) then
5:   Transfer data from the old to the new mesh ▷ see Section 3.4
6: end if
7: for ( $k = 1$  to  $k_{\text{end}}$ ) do ▷ Start load increment loop
8:   if ( $k > 2$ ) then
9:     if ( $\min\{R, O, A\} < \rho$ ) then ▷ see Section 3.3
10:      Restore the last converged solution  $\mathbf{d}$ 
11:      Modify the nodes of the geometry to obtain the deformed one
12:      Update  $m = m + 1$ 
13:      goto to line 3 ▷ A remeshing step is initiated
14:    end if
15:  end if
16:  for ( $i = 1$  to  $i_{\text{max}}$ ) do ▷ Start Newton–Raphson iteration
17:    Solve the global system of equations (53)
18:    Update the solution
19:    Check convergence of Newton–Raphson iteration
20:    if (converged) then
21:      Update the remeshing history data at each integration point
22:      break ▷ Proceed with next load step
23:    end if
24:    if ((not converged) and ( $i = i_{\text{max}}$ )) then
25:      Restore the last converged solution  $\mathbf{d}$ 
26:      Modify the nodes of the geometry to obtain the deformed one
27:      Update  $m = m + 1$ 
28:      goto to line 3 ▷ A remeshing step is initiated
29:    end if
30:  end for
31: end for

```

---

## 3.3 Remeshing criteria

To evaluate the mesh quality, several remeshing criteria are presented in [9]. In the following, the applied criteria are briefly summarized. All criteria presented in the following take the Jacobi matrix into account. The computation of the reference Jacobi matrix  $\mathbf{J}$  of the current reference state  $\Omega_n$  is defined as follows:

$$\mathbf{J}_n = \frac{\partial \mathbf{x}_n}{\partial \boldsymbol{\xi}} = [\mathbf{G}_1 \ \mathbf{G}_2 \ \mathbf{G}_3] \quad (30)$$

and the spatial Jacobi matrix is defined by

$$\mathbf{j}_{n+1} = \frac{\partial \mathbf{x}_{n+1}}{\partial \boldsymbol{\xi}} = \tilde{\mathbf{F}}_{n+1} \mathbf{J}_n = [\mathbf{g}_1 \ \mathbf{g}_2 \ \mathbf{g}_3]. \quad (31)$$

In the following, the Jacobi matrices will be evaluated at given points  $\mathbf{x}^k$  and are therefore denoted by  $\mathbf{J}_n^k$  and  $\mathbf{j}_{n+1}^k$ , respectively.

3.3.1 Ratio of Jacobians ( $R$ )

The first criterion measures the volumetric change by computing the ratio between the minimum and maximum determinant of the spatial Jacobi matrix  $\mathbf{j}$  for each cell. It is defined by

$$R = \frac{\min_k \det(\mathbf{j}_{n+1}^k)}{\max_\ell \det(\mathbf{j}_{n+1}^\ell)}. \quad (32)$$

For uniform deformation the ratio will be close to one, whereas the ratio will decrease for nonuniform deformations.

### 3.3.2 Orthogonality ( $O$ )

The second criterion measures the orthogonality of the Jacobi matrix columns at each point  $\mathbf{x}^k$ . Therefore, we first compute the product

$$b_{ij}^k = (\mathbf{g}_i^k \cdot \mathbf{g}_i^k) (\mathbf{g}_j^k \cdot \mathbf{g}_j^k). \quad (33)$$

The criterion is then evaluated by finding the minimum of the following expression

$$O = \min_{i,j,k} \frac{\sqrt{b_{ij}^k - (\mathbf{g}_i^k \cdot \mathbf{g}_j^k)^2}}{\sqrt{b_{ij}^k}}. \quad (34)$$

As long as the cell remains orthogonal, this criterion returns a value of one because the scalar product between two distinct column vectors is zero in this case.

### 3.3.3 Aspect ratio ( $A$ )

In this criterion, the aspect ratio of the cell is measured both, in undeformed and deformed configuration, i.e.,

$$A_n = \min_{i,j,k} \frac{\|\mathbf{G}_i^k\|}{\|\mathbf{G}_j^k\|} \quad (35)$$

$$A_{n+1} = \min_{i,j,k} \frac{\|\mathbf{g}_i^k\|}{\|\mathbf{g}_j^k\|}. \quad (36)$$

Finally, the aspect ratio is defined as the ratio

$$A = \frac{A_{n+1}}{A_n}. \quad (37)$$

In this case, the value is close to one if the cell is deformed in all dimensions equally, while the value decreases if the aspect ratio of the element changes.

## 3.4 Data transfer

The data transfer of pre-deformation and other history data can be carried out using different methods. In the present work, we apply point-based methods, namely radial basis functions (RBFs) and inverse distance weighting (IDW). Each method will be explained briefly in the subsequent sections.

### 3.4.1 Radial basis function interpolation

In the following, we present the local version of the RBF used within every remeshing step, see [32,33]. Thereby, for every target point  $\mathbf{x}^t$  in the new mesh, we search the  $N^s$  nearest source points  $\mathbf{x}_i^s$  from the old mesh. This is done with the help of a fast  $k$ - $d$  tree algorithm [34]. The  $k$ - $d$  tree is a space partitioning method to efficiently treat a set of data points in a  $k$ -dimensional space where in this case  $k = 3$  holds. Based on the  $N^s$  nearest source points to the target point, the RBF interpolation is constructed from the weighted sum of radial basis functions  $\Theta_i(r)$  that depend only on the radial distance to the source point  $\mathbf{x}_i^s$  as follows:

$$\sum_{i=1}^{N^s} w_i \Theta_i(\hat{r}_{ij}) = v^s(\mathbf{x}_j^s). \quad (38)$$

In the above equation  $\hat{r}_{ij} = \|\mathbf{x}_i^s - \mathbf{x}_j^s\|$  is the radial distance between two source points and  $v^s$  is the source value at source point  $\mathbf{x}_j^s$ , which is a known quantity from the old mesh. By doing this for every source point, it is possible to set up a linear system of equations, which can be written as

$$\begin{bmatrix} \Theta_1(\|\mathbf{x}_1^s - \mathbf{x}_1^s\|) & \dots & \Theta_{N^s}(\|\mathbf{x}_{N^s}^s - \mathbf{x}_1^s\|) \\ \vdots & \ddots & \vdots \\ \Theta_1(\|\mathbf{x}_1^s - \mathbf{x}_{N^s}^s\|) & \dots & \Theta_{N^s}(\|\mathbf{x}_{N^s}^s - \mathbf{x}_{N^s}^s\|) \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_{N^s} \end{bmatrix} = \begin{bmatrix} v^s(\mathbf{x}_1^s) \\ \vdots \\ v^s(\mathbf{x}_{N^s}^s) \end{bmatrix}, \quad (39)$$

or in a compact form as

$$\mathbf{A} \mathbf{w} = \mathbf{b}. \quad (40)$$

Since we use the integration points from the old mesh as source points, the values of the right-hand side are known. The matrix  $\mathbf{A}$  is symmetric with a size of  $(N^s \times N^s)$ , and the entries are obtained by evaluating the modified radial basis function

$$A_{ij} = \Theta_i(\hat{r}_{ij}) = \Phi\left(\beta \frac{\hat{r}_{ij}}{\tilde{r}_i}\right), \quad (41)$$

where  $\hat{r}_{ij}$  is the radial distance between the source points  $i$  and  $j$ . The additional parameters  $\tilde{r}_i$  and  $\beta$  can be used to tune the interpolation. Thereby,  $\beta$  is a scaling factor, and  $\tilde{r}_i$  is the mean distance between the source point  $\mathbf{x}_i^s$  and a specified number of  $N^t$  source points that are close to source point  $i$ .  $\Phi$  refers to the chosen radial basis function. Possible RBFs are given in Table 1. Having computed the matrix  $\mathbf{A}$  and the vector  $\mathbf{b}$ , the system (40) is solved to obtain the weights  $\mathbf{w}$ .

Once the weights are available, the target value  $v^t$  can be computed directly from the weighted sum as

$$v^t(\mathbf{x}^t) = \sum_{i=1}^{N^s} w_i \Theta_i(\|\mathbf{x}_i^s - \mathbf{x}^t\|), \quad (42)$$

or in matrix–vector notation

$$v^t(\mathbf{x}^t) = [\Theta_1(\|\mathbf{x}_1^s - \mathbf{x}^t\|) \dots \Theta_{N^s}(\|\mathbf{x}_{N^s}^s - \mathbf{x}^t\|)] \begin{bmatrix} w_1 \\ \vdots \\ w_{N^s} \end{bmatrix}. \quad (43)$$

Due to the local support of only the  $N^s$  nearest neighbors, the local RBF interpolation can easily be parallelized to efficiently compute the target values. An additional advantage is that the size of  $\mathbf{A}$  can be controlled by specifying the number of source points that should to be taken into consideration.

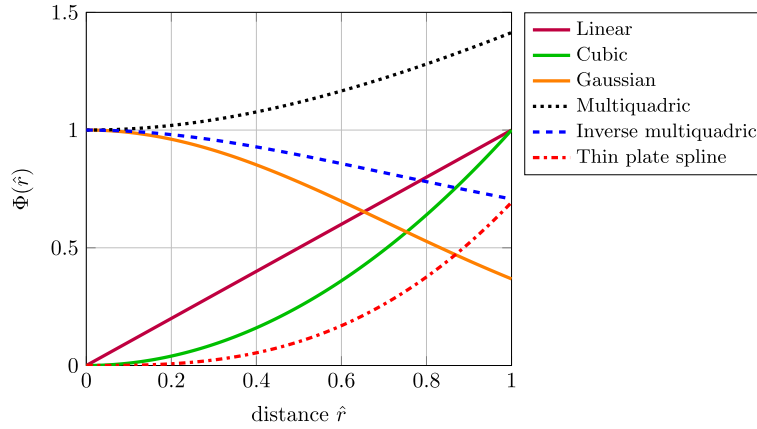
Conversely, the global RBF version uses all available source points to construct the interpolation (39), leading to a large system of equations. This has then to be solved once to evaluate any target point. The drawback of the global approach is that it can be computationally expensive.

Different RBFs are plotted in Fig. 4 as a function of the radial distance  $\hat{r}$ . The linear, cubic, and the thin plate spline function all approach zero if the radius is close to zero. Conversely, the other RBFs are close to one for small radii. However, the multiquadric RBF increases for larger distances. In this work, we consider the Gaussian and the inverse multiquadric RBF as possible candidates. From our experience, setting  $N^s = 50$ ,  $N^t = 3$ , and  $\beta = 1$  yields satisfactory interpolation results together with the inverse multiquadric RBF.

Note that only the integration points in the physical domain of the old mesh are considered as source points. This is done because the points in the fictitious domain are prone to large deformations and, therefore, may negatively impact the interpolation in the physical domain. On the other hand, all integration points in the new mesh are used as target points. Thereby, the displacement gradients gradually approach zero in the fictitious domain as the distance to the physical domain increases.

**Table 1** Some of the well-known radial basis functions  $\Phi(\hat{r})$  [35]

Name	$\Phi(\hat{r})$
Linear	$\hat{r}$
Cubic	$\hat{r}^3$
Gaussian	$\exp(-\hat{r}^2)$
Multiquadric	$\sqrt{1 + \hat{r}^2}$
Inverse multiquadric	$1/\sqrt{1 + \hat{r}^2}$
Thin plate spline	$\hat{r}^2 \log(1 + \hat{r})$



**Fig. 4** Plot of the different radial basis functions  $\Phi$  (compare Table 1) as a function of distance  $\hat{r}$

3.4.2 Inverse distance weighting

Similar to the RBF the inverse distance weighting (IDW) is a point-based interpolation method, also known as Shepard’s interpolation method [36]. However, in contrast to RBF interpolation, it is not necessary to solve a system of equations. The IDW is based on the weighted average of the values of existing source values  $v^s(\mathbf{x}_i^s)$ . For each target point  $\mathbf{x}^t$ , the value  $v^t$  can be computed as follows:

$$v^t(\mathbf{x}^t) = \begin{cases} \frac{\sum_{i=1}^{N^s} w_i(\mathbf{x}^t) v^s(\mathbf{x}_i^s)}{\sum_{i=1}^{N^s} w_i(\mathbf{x}^t)}, & \text{if } \hat{r}(\mathbf{x}_i^s, \mathbf{x}^t) \neq 0 \text{ for all } i = 1, \dots, N^s \\ v^s(\mathbf{x}_i^s), & \text{if } \hat{r}(\mathbf{x}_i^s, \mathbf{x}^t) = 0 \text{ for any } i = 1, \dots, N^s \end{cases} \quad (44)$$

where the weights  $w_i$  are computed from the inverse distance by

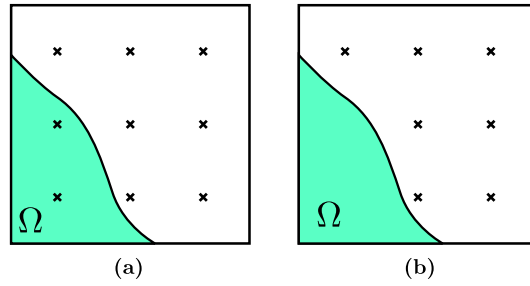
$$w_i(\mathbf{x}^t) = \frac{1}{(\hat{r}(\mathbf{x}_i^s, \mathbf{x}^t)^\beta)}. \quad (45)$$

The parameter  $\beta$  can be modified to control the influence of the neighboring points. Small values leverage the weights of points in a larger region around the target point, while large values tend to suppress the influence of source points with increasing radial distance  $\hat{r}$ .

Similarly to the local RBF approach, only the  $N^s$  nearest source points to the target point are considered for the interpolation to increase the efficiency of the method.

4 Stabilization techniques

As previously mentioned, fictitious domain approaches have the advantage that the mesh generation is simplified since the elements/cells do not have to conform to the geometry of the physical domain. Therefore, the boundary of the physical domain intersects the elements/cells. As shown in [9], the condition of the system is strongly related to the volume fraction of the broken cells. Because the condition number significantly impacts



**Fig. 5** Distribution of stabilization integration points for a broken cell of shape function ansatz order  $p = 2$ . First,  $n = 9$  standard Gauss-points are distributed (a). Then, the points in the physical domain are removed (b)

the robustness of the FCM when applied to nonlinear problems, we will explore two different stabilization approaches in the following sections.

#### 4.1 Material stabilization ( $\alpha$ -stabilization)

The idea of this stabilization technique is to fill the fictitious domain with a soft version of the material from the physical domain. Therefore, additional integration points are distributed in the fictitious domain where the material parameters are modified by the stabilization factor  $\alpha$  defined by

$$\alpha = \begin{cases} 1, & \text{for } x \text{ in } \Omega \\ \alpha_0 = 10^{-q}, & \text{for } x \text{ in } \Omega_e \setminus \Omega \end{cases} \quad (46)$$

where  $q = \{5, \dots, 9\}$ , for example. Similarly to [37], we first distribute  $n = (p + 1)^d$  points in the cell and then remove the integration points that are in the physical domain where  $d$  denotes the spatial dimension. This process is visualized in Fig. 5.

An advantage of the  $\alpha$ -method stabilization scheme is its straightforward implementation, since only an additional set of integration points in the fictitious domain is needed to apply the stabilization. However, the material stabilization technique can lead to a loss of accuracy if large  $\alpha_0$  values are applied. If  $\alpha_0$  is too large, the fictitious domain may modify the solution. This is of course not the desired behavior and, hence,  $\alpha_0$  has to be chosen carefully.

#### 4.2 Eigenvalue stabilization

The eigenvalue stabilization (EVS) was first introduced by [21] in the context of the extended FEM (XFEM). In [20], the method was extended to the FCM context for small as well as large strain elastic analysis. The principal idea is to perform an eigenvalue decomposition of the stiffness matrix  $\mathbf{k}^c$  of every broken cell:

$$\mathbf{k}^c = \mathbf{V} \mathbf{\Lambda} \mathbf{V}^\top = \begin{bmatrix} \bar{\mathbf{V}} & \tilde{\mathbf{V}} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{\Lambda}} & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{\Lambda}} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{V}}^\top \\ \tilde{\mathbf{V}}^\top \end{bmatrix}, \quad (47)$$

where  $\mathbf{V}$  contains the eigenvectors of the cell stiffness matrix with the corresponding eigenvalues collected in  $\mathbf{\Lambda}$ . Furthermore, we split the eigenvalues into a non-problematic part  $\bar{\mathbf{\Lambda}}$  and the problematic small values  $\tilde{\mathbf{\Lambda}}$ . The latter group contains small and zero eigenvalues. In order to avoid unphysical simulation results, the zero eigenvalues and related eigenvectors belonging to the rigid body modes are excluded from the stabilization. The other small eigenvalues and corresponding modes are stabilized by creating the stabilization matrix

$$\mathbf{k}^\epsilon = \sum_{j=1}^{\tilde{N}} \gamma_s^j \tilde{\mathbf{v}}^j \otimes \tilde{\mathbf{v}}^j, \quad (48)$$

where  $\tilde{N}$  corresponds to the number of eigenmodes that are considered for stabilization. Additionally, the factor  $\gamma_s^j$  has to be defined. There exist different approaches to determine this value. In the present work, we use the following definition:

$$\gamma_s^j = \epsilon \lambda (1 - v^c)^\beta \quad (49)$$

where  $\epsilon$  is a user defined threshold parameter to identify the critical modes. Additionally, this method depends on the first Lamé parameter  $\lambda$ , the volume fraction of the cell  $v^c$  and the tuning parameter  $\beta$ . The latter is defined in the present work to be  $\beta = 1$ .

Once the stabilization stiffness matrix is constructed, the original stiffness matrix is modified-yielding

$$\tilde{\mathbf{k}}^c = \mathbf{k}^c + \mathbf{k}^\epsilon \quad (50)$$

This, however, changes the original system. Therefore, a compensation force can be defined to modify the right-hand side

$$\mathbf{f}^\epsilon = \mathbf{k}^\epsilon \mathbf{d}^c \quad (51)$$

based on the stabilization matrix  $\mathbf{k}^\epsilon$  and the solution vector  $\mathbf{d}^c$ . Consequently, the out-of-balance vector is modified to result in

$$\tilde{\mathbf{g}}^c = \mathbf{g}^c + \mathbf{f}^\epsilon. \quad (52)$$

In order to apply the load gradually, we introduce a scaling parameter  $\bar{\lambda}_k$  that serves to linearly scale the applied boundary conditions. Once this is done for all cell stiffness matrices, the solution increment for the Newton–Raphson schemes is computed by

$$\tilde{\mathbf{K}}_T^i(\mathbf{d}_k^i) \Delta \mathbf{d}_k^{i+1} = -\tilde{\mathbf{G}}^i(\mathbf{d}_k^i, \bar{\lambda}_k). \quad (53)$$

The procedure of the stabilization is summarized in Algorithm 2. In the present work, we apply a combination of the  $\alpha$ -method and the EVS to stabilize the solution.

---

**Algorithm 2** Eigenvalue stabilization strategy according to [20]

---

```

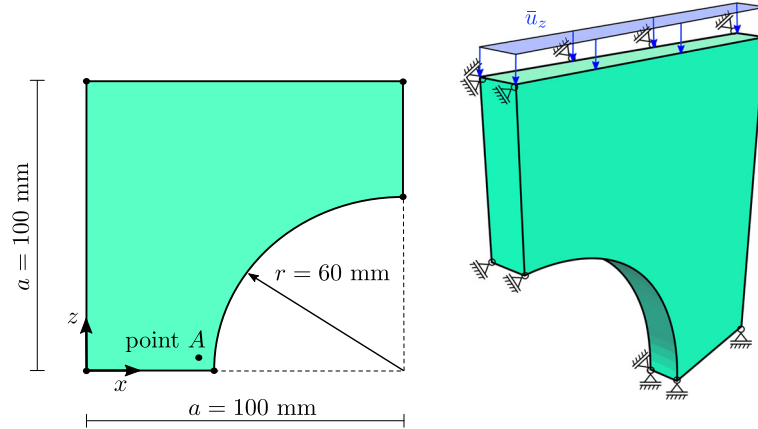
1: for ( $k = 1$  to  $k_{\text{end}}$ ) do ▷ Load increment loop
2:   for ( $i = 1$  to  $i_{\text{max}}$ ) do ▷ Newton iteration
3:     Compute  $\mathbf{k}_k^{c,i}$ , and  $\mathbf{g}_k^{c,i}$  for each cell  $c$ 
4:     if (cell  $c$  is cut) then ▷ Stabilize the cut cells
5:       if ( $i = 1$ ) then
6:         Factorize the cell stiffness matrix  $\mathbf{k}_k^{c,i} = \mathbf{V} \mathbf{\Lambda} \mathbf{V}^T$  (47)
7:         Identify the modes with small eigenvalues  $\tilde{\mathbf{V}}$ 
8:         Ignore the modes related to RBMs
9:         Build the stabilization matrix  $\mathbf{k}_k^\epsilon$  (48)
10:        end if
11:        Modify the stiffness matrix  $\tilde{\mathbf{k}}_k^{c,i} = \mathbf{k}_k^{c,i} + \mathbf{k}_k^\epsilon$  (50)
12:        Correct the right-hand side  $\tilde{\mathbf{g}}_k^{c,i} = \mathbf{g}_k^{c,i} + \mathbf{f}_k^\epsilon$  (52)
13:        end if
14:        Assemble  $\tilde{\mathbf{K}}_T^i(\mathbf{d}_k^i)$ ,  $\tilde{\mathbf{G}}^i(\mathbf{d}_k^i, \bar{\lambda}_k)$ 
15:        Solve the system  $\tilde{\mathbf{K}}_T^i(\mathbf{d}_k^i) \Delta \mathbf{d}_k^{i+1} = -\tilde{\mathbf{G}}^i(\mathbf{d}_k^i, \bar{\lambda}_k)$  (53)
16:        Update the solution  $\mathbf{d}_k^{i+1} = \mathbf{d}_k^i + \Delta \mathbf{d}_k^{i+1}$ 
17:        Check convergence of Newton–Raphson iteration
18:        if (converged) then
19:          break ▷ Proceed with next load step
20:        end if
21:      end for
22:    end for

```

---

**Table 2** Material parameters for the finite  $J_2$  elastoplasticity material model [38]

Parameter	Value
First Lamé parameter ( $\lambda$ )	110,743 MPa
Shear modulus ( $\mu$ )	80,194 MPa
Initial yield stress ( $Y_0$ )	450.0 MPa
Saturation stress ( $Y_\infty$ )	715 MPa
Linear hardening parameter ( $H$ )	129.24 MPa
Hardening exponent ( $\delta$ )	16.93

**Fig. 6** Geometry and boundary conditions for the plate with a cylindrical hole

## 5 Numerical examples

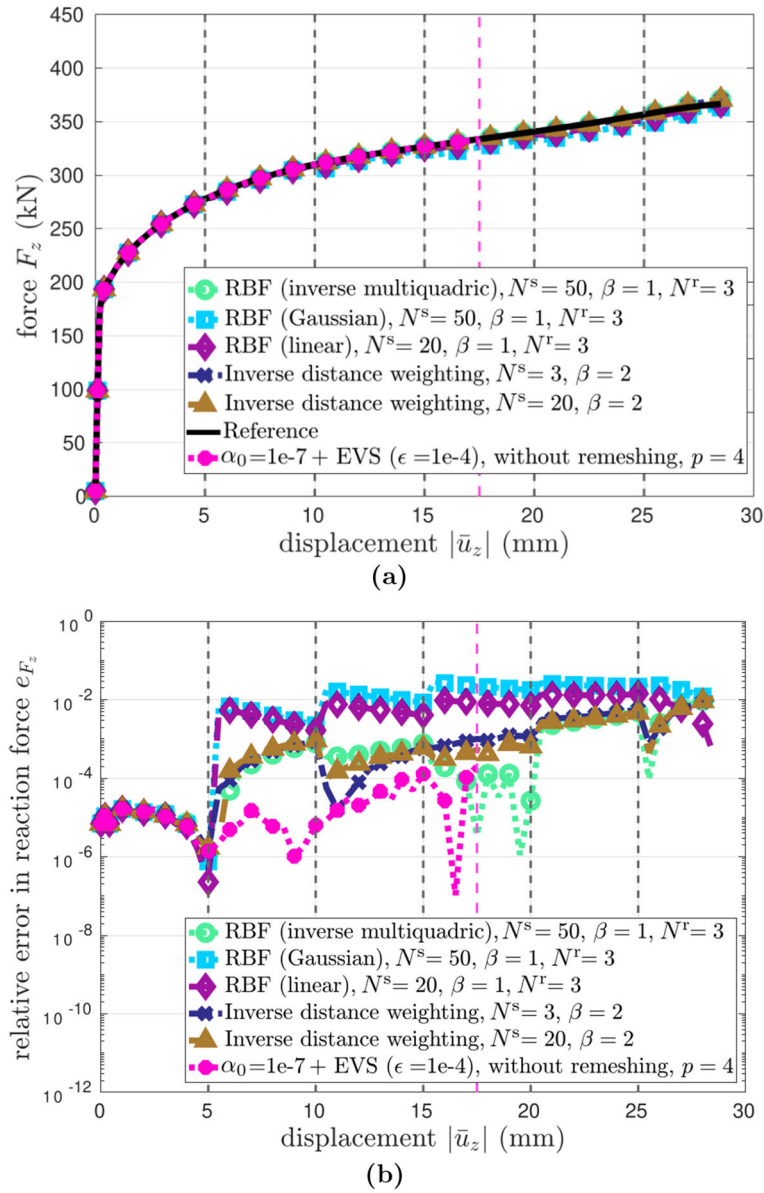
In the present work, we focus on the simulation of structures including finite  $J_2$  elastoplasticity in combination with the  $\alpha$ - and the eigenvalue stabilization as well as remeshing. The first example considers a benchmark of a plate with a cylindrical hole. This example examines the remeshing approach using various interpolation strategies for mapping the internal variables from the old to the new meshes. In the other three examples, the remeshing is combined with the  $\alpha$ - and eigenvalue stabilization to improve the robustness of the FCM simulations. In all cases, the remeshing criteria limits are set to  $\rho = 0$ .

For the integration of the broken cells, we apply NNMF with a spacetree depth of  $k_{ST} = 3$  to compute the moments. Moreover, we apply RBF interpolation with the inverse multiquadric RBF to transfer the displacement gradients between the meshes. Additionally, different approaches are applied to transfer the left elastic Cauchy–Green tensor and the equivalent plastic strain to the new mesh. The parameters for the elastoplastic material model are defined in Table 2.

### 5.1 Plate with a cylindrical hole

This section examines the different interpolation methods and their influence on the results. The model is shown in Fig. 6. The compressive displacement is applied to the upper surface while symmetry boundary conditions are applied to the other corresponding faces. The maximum prescribed displacement amounts to  $\bar{u}_z = 29.5$  mm. The cells have an initial edge length of 5 mm. Initially, they are distributed according to the maximum dimensions of the physical domain. After filtering the cells that are completely void, 604 cells remain. Then, the ansatz space is given by hierarchic shape functions based on integrated Legendre polynomials of order  $p = 4$ . To resolve the nonlinear behavior caused by the elastoplastic deformation, the first increments are set to  $\Delta\bar{u}_z = \{0.005, 0.025, 0.05, 0.1, 0.2, 0.3, 0.4\}$  mm. Afterward, equally sized increments of  $\Delta\bar{u}_z = 0.5$  mm are applied until the final displacement is reached.

For the sake of simplicity, we apply the remeshing every 5 mm. This leads to five remeshing steps in total. For the data transfer of the displacement gradients, we employ the inverse multiquadric RBF with the parameters  $N^s = 50$ ,  $\beta = 1$ , and  $N^t = 3$ . For the equivalent plastic strain different interpolation schemes are applied. First, different RBFs are tested followed by the IDW with different parameter sets.



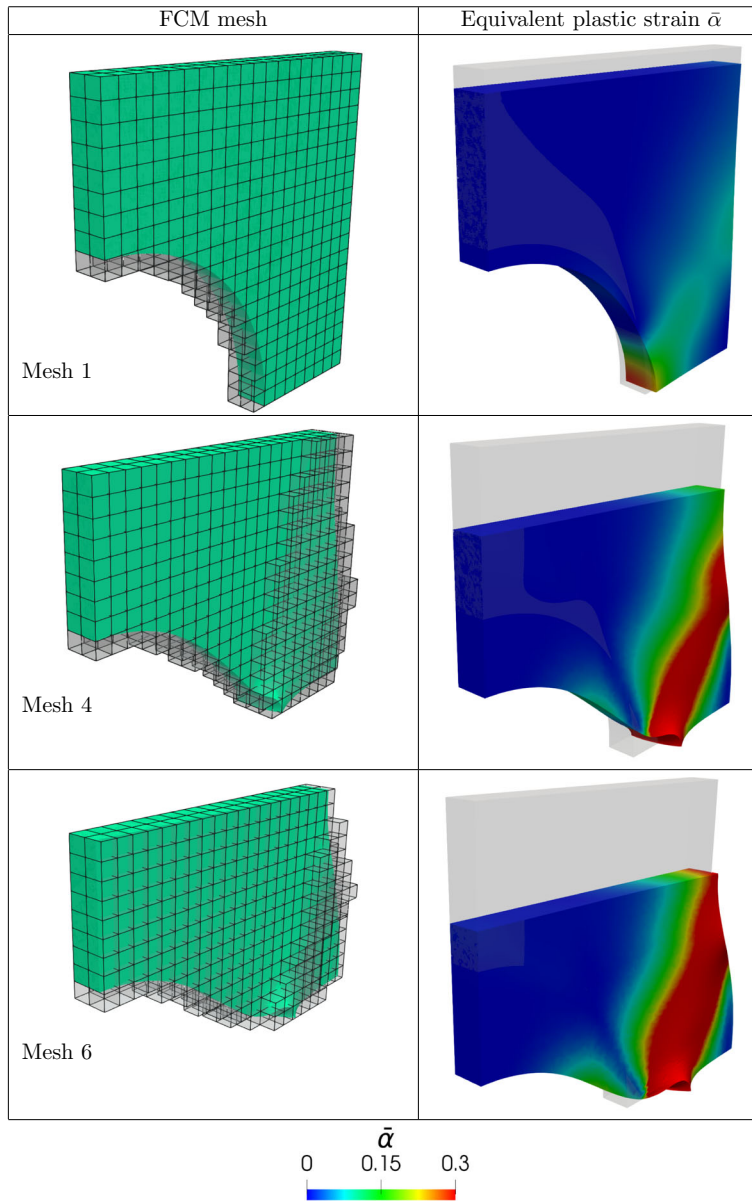
**Fig. 7** The plate with a cylindrical hole using ansatz order  $p = 4$ . **a** Load–displacement curves. **b** Relative error in reaction force compared to a reference solution

The load–displacement curves are shown in Fig. 7a. The reference solution is obtained from the simulation using a boundary conforming mesh with 3200 finite elements of ansatz order  $p = 4$ . As can be seen, all interpolation methods allow to reach the final deformation state of  $\bar{u}_z = 29.5$  mm. Without remeshing, but including EVS with a factor of  $\epsilon = 10^{-4}$  a final deformation state of  $\bar{u}_z = 17.5$  mm is obtained.

In Fig. 7b, the relative error in reaction force is plotted for the deformation states. Clearly, linear and Gaussian RBF introduce the largest error, while inverse multiquadric RBF and IDW interpolation lead to a similar error in the reaction forces. The IDW performs especially well for the parameter set  $N^s = 3$  and  $\beta = 2$ . Additionally, the error in the reaction forces for the simulation including EVS is shown. Note that the error is smaller compared to the simulation where only remeshing is applied.

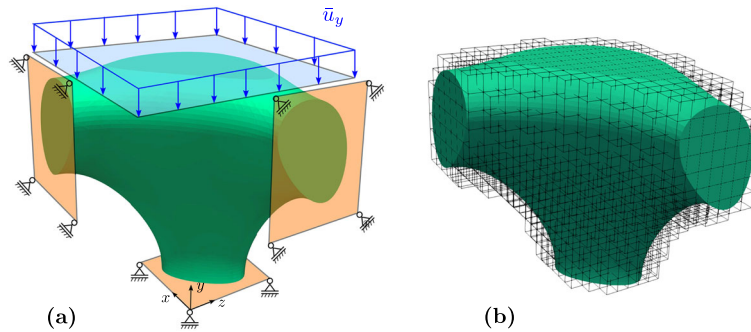
Lastly, Table 3 shows the equivalent plastic strain  $\bar{\alpha}$  for different deformation states where the IDW is applied. Additionally, the underlying meshes are shown.

**Table 3** The equivalent plastic strain  $\bar{\alpha}$  of the plate at different load steps using the inverse distance weighting interpolation scheme with  $N^s = 3$  and  $\beta = 2$



## 5.2 Single cube connector

The following example investigates the influence of the remeshing on the result for different polynomial degrees of the shape functions. The model as well as the finite cell mesh are shown in Fig. 8. A compressive displacement  $\bar{u}_y$  is applied to the upper surface, while symmetry boundary conditions are applied as indicated. The cells have an average edge length of 0.08 mm, leading to 2827 cells for the initial mesh to simulate the structure. For this model, we restrict the maximum deformation of each mesh to  $\bar{u}_y = 1.25$  mm. In all cases, we apply the  $\alpha$ -stabilization with a value of  $\alpha_0 = 10^{-7}$ . Additionally we apply EVS with  $\epsilon = 10^{-5}$ . The results from the FCM are thereby compared to a reference solution computed using a mesh of 50,646 s-order tetrahedral elements with the commercial FEA software Abaqus [39]. Verification of the FCM is the main objective of this comparison. The FEM applies second-order tetrahedral elements. Three small load steps



**Fig. 8** **a** The physical domain with the boundary conditions and **b** the initial finite cell mesh

are applied—at first with  $\Delta\bar{u}_y = \{0.0015625, 0.00625, 0.0125\}$  mm, followed by equal-sized steps of size  $\Delta\bar{u}_y = 0.025$  mm. Thus, 240 load increments are required to accomplish the desired deformation state.

The load–displacement curves for different shape function ansatz orders  $p$  are shown in Fig. 9a. First, it should be noted that, without any remeshing, the simulation terminates at  $\bar{u}_y = 3.4$  mm for ansatz order  $p = 4$ . However, if remeshing is applied considering the remeshing criteria, the final deformation of  $\bar{u}_y = 6$  mm can be reached. Additionally, the accuracy is increased by leveraging the shape function ansatz order.

Figure 9b shows the evolution of the remeshing criteria throughout the simulation for shape function ansatz order  $p = 4$ . As can be seen, all criteria start at 1 and decrease throughout the deformation. The first remeshing is initiated even though none of the criteria reached the limit. This remeshing is triggered by convergence problems of the Newton–Raphson method. The same applies also for later remeshing steps. The values of the remeshing criteria are heavily improved and start at values close to one after the remeshing. This indicates that the remeshing has a positive impact on the discretization.

Table 4 shows the different meshes in the first column, and the evolution of the equivalent plastic strain  $\bar{\alpha}$  in the second and third column. It can clearly be seen that the newly generated meshes are again composed of well shaped hexahedral cells. In addition, the last column in Table 4 shows the evolution of the equivalent plastic strain for the same state from the FEM solution computed with Abaqus. Both approaches show similar plastic strain distributions for this example—qualitatively and quantitatively.

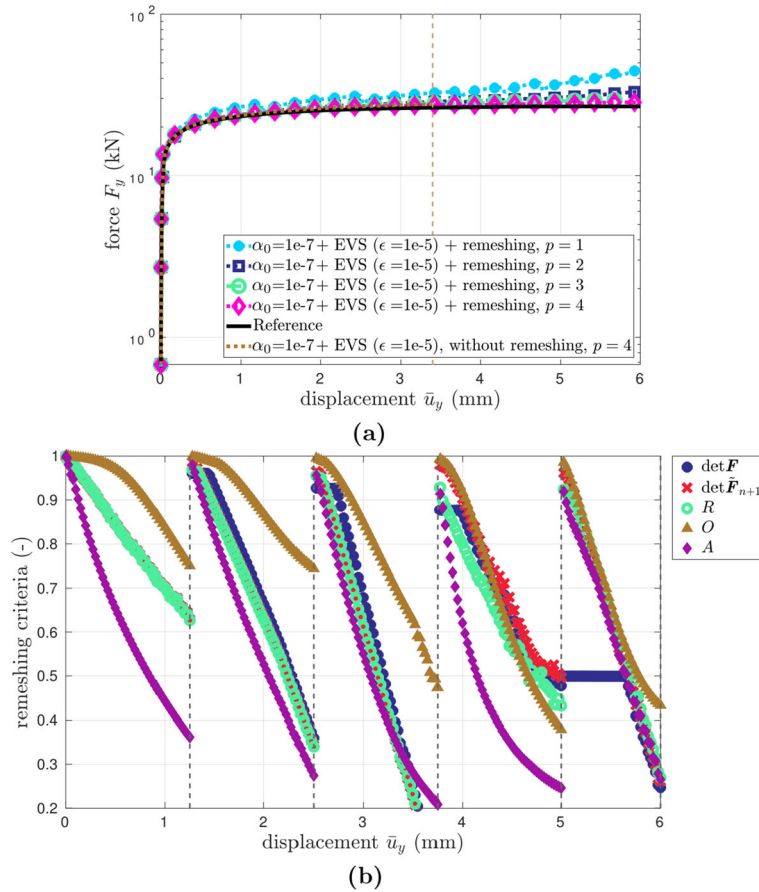
### 5.3 Complex cube connector

To increase the model complexity, the following example is an agglomerate of the single cube connector shown previously. The model is shown in Fig. 10. In this case, the displacement of the upper flat surface is fixed in  $x$  and  $y$  direction, while a compressive displacement  $\bar{u}_z$  is applied. The bottom, on the other hand, is fixed in all directions. The domain is discretized using 2968 cells of size 3.75 mm with shape function order  $p = 2$ . The first load increments are set to  $\Delta\bar{u}_z = \{0.01, 0.02, 0.03, 0.05, 0.1\}$  mm, followed by equal-sized steps of size  $\bar{u}_z = 0.2$  mm. To reach the final deformation of  $\Delta\bar{u}_z = 35$  mm, 175 load increments are required in total.

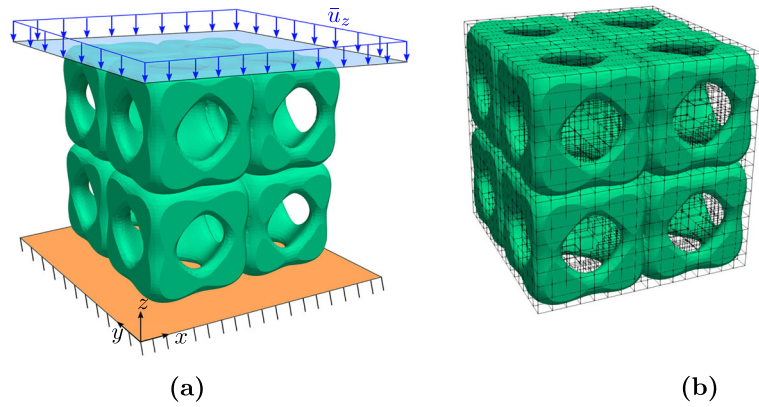
The load–displacement curve is shown in Fig. 11a. Without remeshing, the incremental/iterative solution process terminates after  $\bar{u}_z = 6.2$  mm. In contrast the desired deformation can be achieved by incorporating the remeshing procedure multiple times. However, it should be noted that using IDW for mapping of plastic variables requires only five remeshing steps—whereas six remeshings are required if RBF interpolation is used. Nevertheless, both methods reach the desired deformation of  $\bar{u}_z = 35$  mm.

Next, we examine the different remeshing criteria together with the IDW interpolation. Once again, the remeshing clearly improves the mesh quality as the values are close to one after each remeshing step. Initially, the Aspect ( $A$ ) has the lowest value among the different criteria. However, after the third remeshing, the ratio of the Jacobians ( $R$ ) tends to have the lowest value. This indicates that it is beneficial to use different remeshing criteria.

Lastly, the equivalent plastic strain  $\bar{\alpha}$  in the last converged state is shown in Table 5, along with the mesh used for the shown deformation states.



**Fig. 9** **a** Load–displacement curves for the single cube connector using different ansatz orders of the shape functions. **b** The remeshing criteria at different load steps for the single cube connector using  $p = 4$

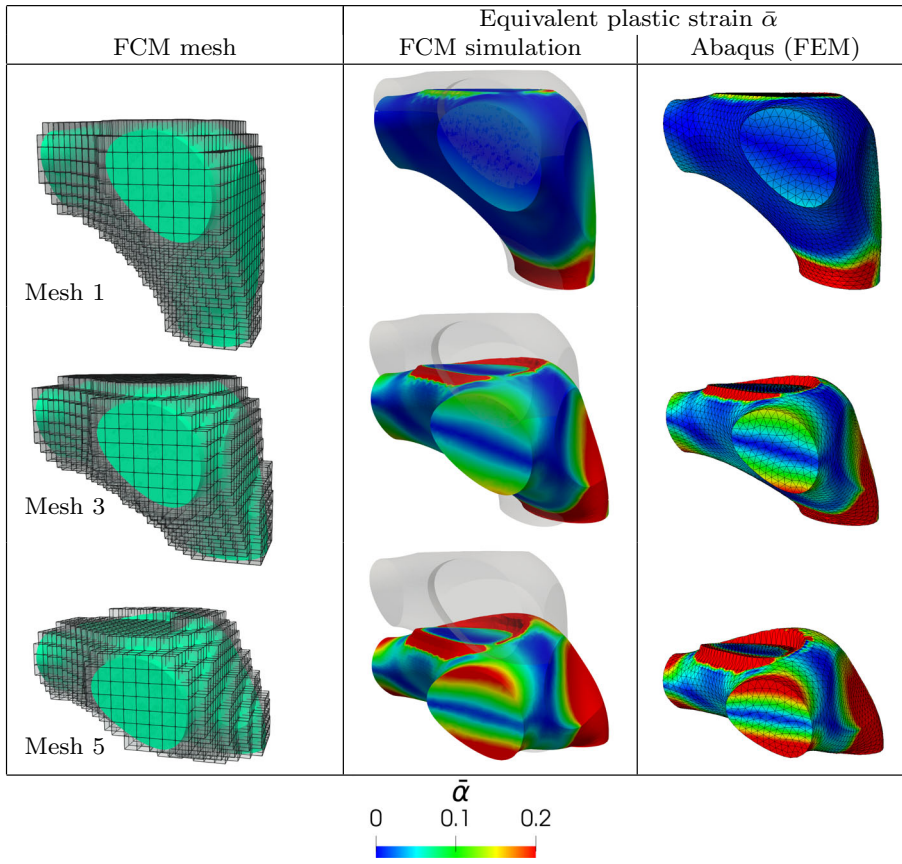


**Fig. 10** **a** Complex cube connector model and the applied boundary conditions and **b** the initial finite cell mesh

5.4 Single pore of a foam

The last example serves to show that the present combination of EVS and remeshing can also be applied to more realistic elastoplastic examples. The model as well as the initial mesh are shown in Fig. 12. The upper surface is fixed in  $x$  and  $z$ -direction, while a compressive displacement is applied in  $y$ -direction. The foundation of the pore is completely fixed. The initial mesh has 7007 cells of size 0.18mm, and a shape function ansatz order of  $p = 3$  is used for the simulation. The displacement is again subdivided into small

**Table 4** Comparison of the FEM reference solution with the FCM solution for different deformation states



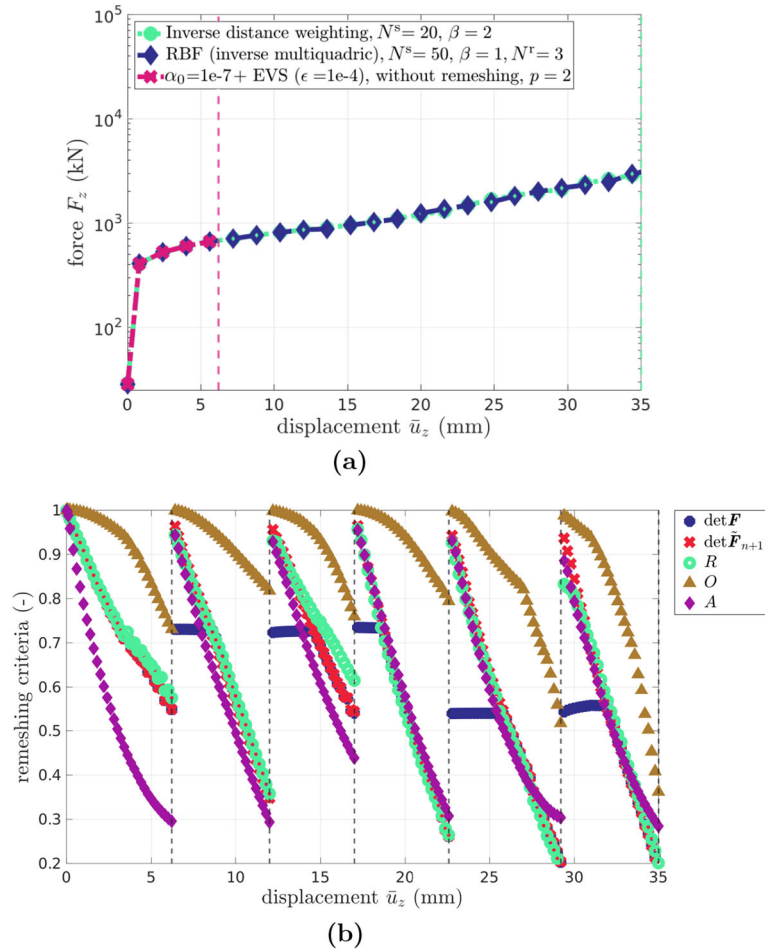
The underlying FCM mesh for each state is given in the first column

steps of  $\Delta \bar{u}_y = \{0.0001, 0.001, 0.003, 0.0058, 0.007\}$  mm before the displacement is incrementally increased by  $\Delta \bar{u}_y = 0.01$  mm. The  $\alpha$ -method is applied with  $\alpha_0 = 10^{-7}$  together with the EVS with  $\epsilon = 10^{-6}$ . The elastic left Cauchy–Green tensor and the equivalent plastic strain are interpolated using the IDW interpolation with  $N^s = 20$  and  $\beta = 2$ . For a reference solution, the same model is simulated in Abaqus [39] using 64,506 second-order tetrahedral elements.

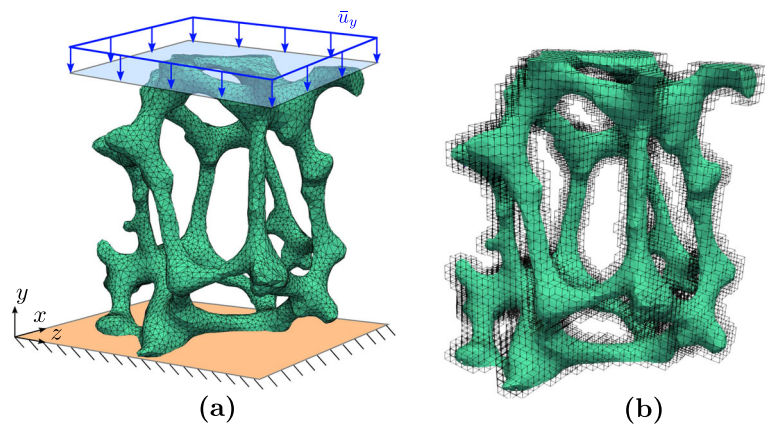
The force displacement curve is found in Fig. 13a. As indicated, the structure can only be compressed up to  $\bar{u}_y = 0.64$  mm without remeshing. However, the desired deformation state can be reached if remeshing is applied. Note that the reaction forces are in good agreement with the reference solution until  $\bar{u}_y = 1.8$  mm. Afterward, the solution deviates slightly from the reference. A possible reason for this is that cells that are broken more than once, mimic the contact formulation similar to the third medium approach, see also [40]. Additionally, if the structure penetrates itself and remeshing is performed, the current implementation treats the overlapping regions as a solid and connected region.

The remeshing criteria are shown in Fig. 13b. As indicated, 10 remeshings are required in total. Once again, the remeshing improves the mesh quality. However, in some cases remeshing is performed because the Newton–Raphson scheme does not converge, even though the remeshing criteria are not close to zero. This indicates, that the currently chosen remeshing criteria may not be the best choice for elastoplastic analysis. This will be further investigated in future research.

Finally, the equivalent plastic strains for the FCM simulation as well as for the Abaqus simulation are shown in Table 6. Both results show good agreement until self-contact occurs. As previously mentioned, the FEM model does not consider self-contact, leading to deviations in the load–displacement behavior. The regions where self-contact is expected are highlighted by dashed circles. Differences are evident in FCM mesh 11. The FEM simulation cannot capture self-contact, resulting in mesh overlap in the highlighted area. In contrast, the

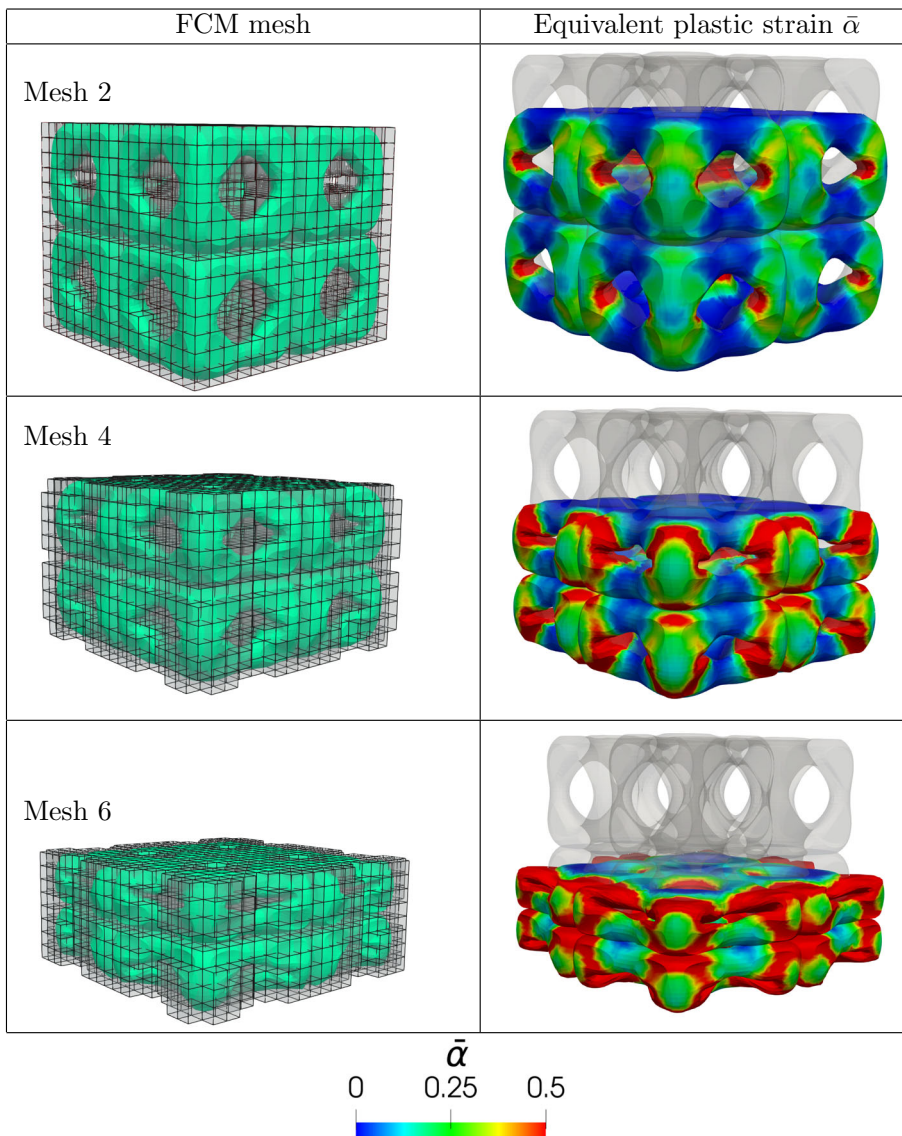


**Fig. 11** **a** Load–displacement curves for the complex cube connector using ansatz order  $p = 2$ . **b** The remeshing criteria at different load steps for the complex cube connector using the IDW interpolation scheme with  $N^s = 20$  and  $\beta = 2$



**Fig. 12** **a** Physical domain and the applied boundary conditions. **b** Initial finite cell mesh applied for the simulation

**Table 5** The equivalent plastic strain  $\bar{\alpha}$  for the complex cube connector at different load steps using the inverse distance weighting interpolation scheme with  $N^s = 20$  and  $\beta = 2$

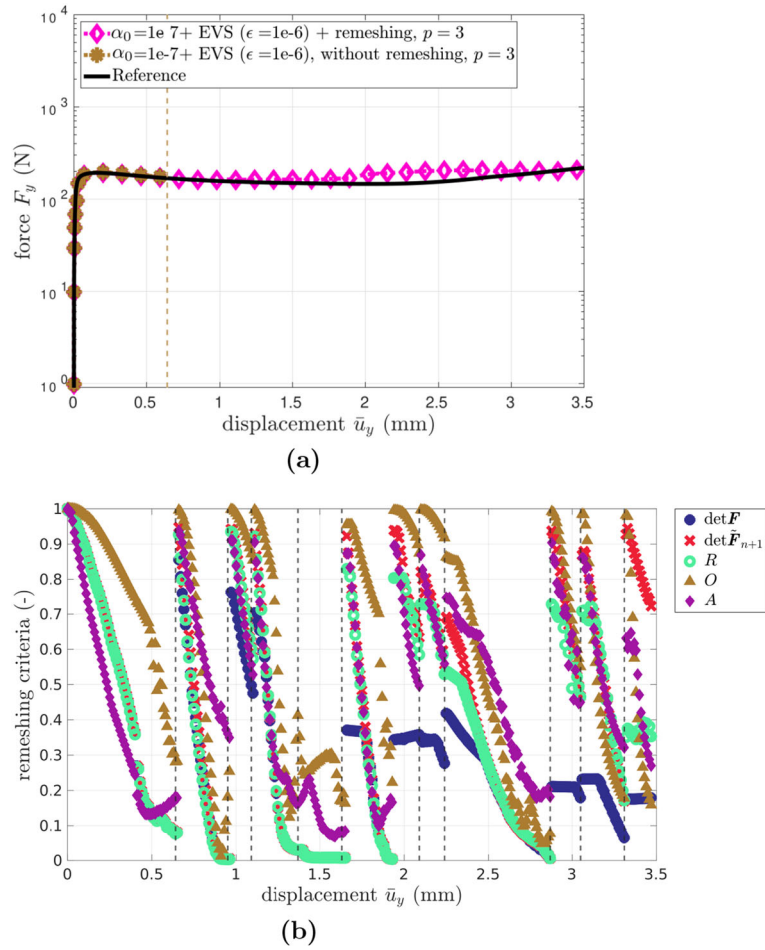


FCM simulation exhibits a type of sticking contact due to remeshing. When a structure penetrates itself during remeshing, the domain boundary is no longer recognized and is treated as a solid region. Consequently, the material remains stuck together.

## 6 Conclusion

Simulating structures using immersed boundary methods such as the FCM poses certain challenges—especially for large deformations. The present work shows that the Eigenvalue stabilization as presented in [20] increases the robustness of the solution procedure. In combination with the remeshing procedure presented in [9], larger deformation states can be reached for hyperelastic materials.

In this work, we presented different examples of increasing complexity, considering a finite strain elasto-plastic material. The combination of  $\alpha$ -stabilization, eigenvalue stabilization and remeshing turned out to be a powerful combination to enhance the stability of the nonlinear simulations. Even large deformation states

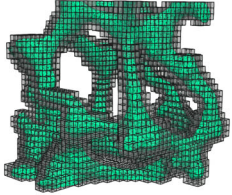
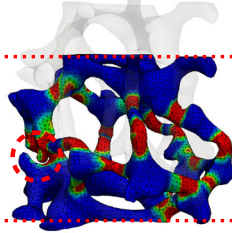
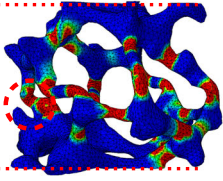
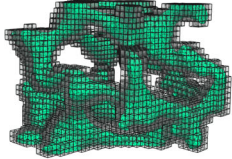
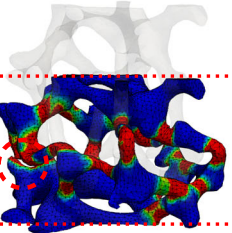
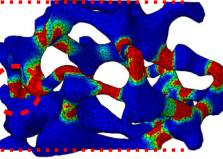
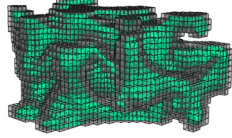
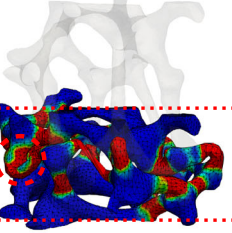
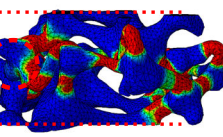



**Fig. 13** **a** Load–displacement curves for the single pore of a foam. **b** Evolution of the remeshing criteria during the simulation

were reached and the obtained solutions were in good agreement to the FEM reference solution computed by using commercial FE-software Abaqus.

The single foam pore motivates the next step that has to be considered: the self-contact. This remains an open challenge because it raises the question of how to identify contact pairs, especially considering that this task needs to be performed for the entire model.

**Table 6** The equivalent plastic strain  $\bar{\alpha}$  for the single pore of a foam at different load steps

FCM mesh	Equivalent plastic strain $\bar{\alpha}$	
	FCM simulation	Abaqus (FEM)
Mesh 4 		
Mesh 7 		
Mesh 11 		

$\bar{\alpha}$   
 0   0.03   0.07   0.1  


The FCM simulation is done using shape function order  $p = 3$  and the FEM simulation is done with second-order tetrahedral elements. One region where self-contact appears is highlighted with the dashed circle

**Acknowledgements** The authors gratefully acknowledge the support provided by the Deutsche Forschungsgemeinschaft (DFG) under the Grant Number DU 405/21-1 and the Project Number 505137962.

**Author contribution** R. Sartorti: Conceptualization, methodology, software, writing—original draft preparation, writing—review and editing. W. Garhuom: Conceptualization, methodology, software, writing—review and editing. A. Düster: Conceptualization, methodology, funding acquisition, resources, supervision, writing—review and editing.

**Funding** Open Access funding enabled and organized by Projekt DEAL.

**Data availability** The data that support the findings of this study are available from the corresponding author upon reasonable request.

**Declarations**

**Conflict of interest** The authors declare no conflict of interest.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Parvzian, J., Düster, A., Rank, E.: Finite cell method—h- and p-extension for embedded domain problems in solid mechanics. *Comput. Mech.* **41**, 121–133 (2007)
2. Düster, A., Parvzian, J., Yang, Z., Rank, E.: The finite cell method for three-dimensional problems of solid mechanics. *Comput. Methods Appl. Mech. Eng.* **197**, 3768–3782 (2008)
3. Burman, E., Hansbo, P.: Fictitious domain finite element methods using cut elements: I. A stabilized Lagrange multiplier method. *Comput. Methods Appl. Mech. Eng.* **199**(41–44), 2680–2686 (2010)
4. Burman, E., Hansbo, P.: Fictitious domain finite element methods using cut elements: II. A stabilized Nitsche method. *Appl. Numer. Math.* **62**(4), 328–341 (2012). <https://doi.org/10.1016/j.apnum.2011.01.008>
5. Burman, E., Hansbo, P., Larson, M.G.: A stabilized cut finite element method for partial differential equations of surfaces: the Laplace–Beltrami operator. *Comput. Methods Appl. Mech. Eng.* **285**, 188–207 (2015)
6. Elfverson, D., Larson, M.G., Larsson, K.: CutIGA with basis function removal. *Adv. Model. Simul. Eng. Sci.* **5**(1), 6 (2018). <https://doi.org/10.1186/s40323-018-0099-2>
7. Zander, N., Kollmannsberger, S., Ruess, M., Yosibash, Z., Rank, E.: The finite cell method for linear thermoelasticity. *Comput. Math. Appl.* **64**(11), 3527–3541 (2012). <https://doi.org/10.1016/j.camwa.2012.09.002>
8. Schillinger, D., Ruess, M., Zander, N., Bazilevs, Y., Düster, A., Rank, E.: Small and large deformation analysis with the p- and B-spline versions of the finite cell method. *Comput. Mech.* **50**, 445–478 (2012). <https://doi.org/10.1007/s00466-012-0684-z>
9. Garhuom, W., Hubrich, S., Radtke, L., Düster, A.: A remeshing strategy for large deformations in the finite cell method. *Comput. Math. Appl.* **80**, 2379–2398 (2020). <https://doi.org/10.1016/j.camwa.2020.03.020>
10. Abedian, A., Parvzian, J., Düster, A., Rank, E.: The finite cell method for the  $J_2$  flow theory of plasticity. *Finite Elem. Anal. Des.* **69**, 37–47 (2013)
11. Taghipour, A., Parvzian, J., Heinze, S., Düster, A.: The finite cell method for nearly incompressible finite strain plasticity problems with complex geometries. *Comput. Math. Appl.* **75**, 3298–3316 (2018). <https://doi.org/10.1016/j.camwa.2018.01.048>
12. Hubrich, S., Düster, A.: Numerical integration for nonlinear problems of the finite cell method using an adaptive scheme based on moment fitting. *Comput. Math. Appl.* **77**, 1983–1997 (2019). <https://doi.org/10.1016/j.camwa.2018.11.030>
13. Ruess, M., Tal, D., Trabelsi, N., Yosibash, Z., Rank, E.: The finite cell method for bone simulations: verification and validation. *Biomech. Model. Mechanobiol.* **11**, 425–437 (2012)
14. Yang, Z., Kollmannsberger, S., Düster, A., Ruess, M., Garcia, E., Burgkart, R., Rank, E.: Non-standard bone simulation: interactive numerical analysis by computational steering. *Comput. Vis. Sci.* **14**(5), 207–216 (2012). <https://doi.org/10.1007/s00791-012-0175-y>
15. Verhoosel, C.V., Zwieten, G.J., Rietbergen, B., Borst, R.: Image-based goal-oriented adaptive isogeometric analysis with application to the micro-mechanical modeling of trabecular bone. *Comput. Methods Appl. Mech. Eng.* **284**, 138–164 (2015)
16. Hug, L., Dahan, G., Kollmannsberger, S., Rank, E., Yosibash, Z.: Predicting fracture in the proximal humerus using phase field models. *J. Mech. Behav. Biomed. Mater.* **134**, 105415 (2022). <https://doi.org/10.1016/j.jmbbm.2022.105415>
17. Radtke, L., Marter, P., Duvigneau, F., Eisenträger, S., Juhre, D., Düster, A.: Vibroacoustic simulations of acoustic damping materials using a fictitious domain approach. *J. Sound Vib.* **568**, 118058 (2024). <https://doi.org/10.1016/j.jsv.2023.118058>
18. Elhaddad, M., Zander, N., Kollmannsberger, S., Shadavakhsh, A., Nübel, V., Rank, E.: Finite cell method: high-order structural dynamics for complex geometries. *Int. J. Struct. Stabil. Dyn.* **15**(7), 1540018 (2015). <https://doi.org/10.1142/S0219455415400180>
19. Duczek, S., Jouliaian, M., Düster, A., Gabbert, U.: Numerical analysis of Lamb waves using the finite and spectral cell method. *Int. J. Numer. Methods Eng.* **99**, 26–53 (2014). <https://doi.org/10.1002/nme.4663>
20. Garhuom, W., Usman, K., Düster, A.: An eigenvalue stabilization technique to increase the robustness of the finite cell method for finite strain problems. *Comput. Mech.* **69**, 1225–1240 (2022). <https://doi.org/10.1007/s00466-022-02140-7>
21. Loehnert, S.: Stabilizing the XFEM for static and dynamic crack simulations. *Proc. Appl. Math. Mech.* **15**, 137–138 (2015). <https://doi.org/10.1002/pamm.201510059>
22. Garhuom, W., Hubrich, S., Radtke, L., Düster, A.: A remeshing approach for the finite cell method applied to problems with large deformations. *Proc. Appl. Math. Mech.* **21**, 202100047 (2021). <https://doi.org/10.1002/pamm.202100047>
23. Sartorti, R., Düster, A.: Remeshing and data transfer in the finite cell method for problems with large deformations. *Proc. Appl. Math. Mech.* **23**, 202300088 (2023). <https://doi.org/10.1002/pamm.202300088>
24. Szabó, B.A.: The use of a priori estimates in engineering computations. *Comput. Methods Appl. Mech. Eng.* **82**, 139–154 (1990)
25. Legrain, G.: Non-negative moment fitting quadrature rules for fictitious domain methods. *Comput. Math. Appl.* **99**, 270–291 (2021). <https://doi.org/10.1016/j.camwa.2021.07.019>
26. Simo, J.C., Armero, F.: Geometrically non-linear enhanced strain mixed methods and the method of incompatible modes. *Int. J. Numer. Methods Eng.* **33**, 1413–1449 (1992)
27. *Inelasticity, Computational: Interdisciplinary Applied Mathematics*, vol. 7. Springer, New York (1998). <https://doi.org/10.1007/b98904>
28. Simo, J.C., Miehe, C.: Associative coupled thermoplasticity at finite strains: formulation, numerical analysis and implementation. *Comput. Methods Appl. Mech. Eng.* **98**, 41–104 (1991)
29. Korelc, J., Stupkiewicz, S.: Closed-form matrix exponential and its application in finite-strain plasticity. *Int. J. Numer. Methods Eng.* **98**(13), 960–987 (2014). <https://doi.org/10.1002/nme.4653>
30. Eterovic, A.L., Bathe, K.-J.: A hyperelastic-based large strain elasto-plastic constitutive formulation with combined isotropic-kinematic hardening using the logarithmic stress and strain measures. *Int. J. Numer. Methods Eng.* **30**(6), 1099–1114 (1990). <https://doi.org/10.1002/nme.1620300602>
31. Weber, G., Anand, L.: Finite deformation constitutive equations and a time integration procedure for isotropic, hyperelastic–viscoplastic solids. *Comput. Methods Appl. Mech. Eng.* **79**(2), 173–202 (1990). [https://doi.org/10.1016/0045-7825\(90\)90131-5](https://doi.org/10.1016/0045-7825(90)90131-5)

32. Boer, A., Zuijlen, A.H., Bijl, H.: Review of coupling methods for non-matching meshes. *Comput. Methods Appl. Mech. Eng.* **196**, 1515–1525 (2007)
33. König, M., Radtke, L., Düster, A.: A flexible C++ framework for the partitioned solution of strongly coupled multifield problems. *Comput. Math. Appl.* **72**, 1764–1789 (2016). <https://doi.org/10.1016/j.camwa.2016.07.031>
34. Bentley, J.L.: Multidimensional binary search trees used for associative searching. *Commun. ACM* **18**(9), 509–517 (1975). <https://doi.org/10.1145/361002.361007>
35. Buhmann, M.D. (ed.): *Radial Basis Functions: Theory and Implementations*. Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, Cambridge (2003). <https://doi.org/10.1017/CBO9780511543241>
36. Shephard, M.S., Dey, S., Flaherty, J.E.: A straightforward structure to construct shape functions for variable p-order meshes. *Comput. Methods Appl. Mech. Eng.* **147**, 209–233 (1997)
37. Abedian, A., Parvizian, J., Düster, A., Rank, E.: Finite cell method compared to  $h$ -version finite element method for elastoplastic problems. *Appl. Math. Mech.* **35**(10), 1239–1248 (2014). <https://doi.org/10.1007/s10483-014-1861-9>
38. Schröder, J., Wick, T., Reese, S., Wriggers, P., Müller, R., Kollmannsberger, S., Kästner, M., Schwarz, A., Igelbüscher, M., Viebahn, N., Bayat, H.R., Wulfinghoff, S., Mang, K., Rank, E., Bog, T., D'Angella, D., Elhaddad, M., Hennig, P., Düster, A., Garhuom, W., Hubrich, S., Walloth, M., Wollner, W., Kuhn, C., Heister, T.: A selection of benchmark problems in solid mechanics and applied mathematics. *Arch. Comput. Methods Eng.* **28**, 1–39 (2020). <https://doi.org/10.1007/s11831-020-09477-3>
39. Dassault Systèmes: *Abaqus 2021 Documentation* (2021). <https://www.3ds.com/products-services/simulia/products/abaqus/>
40. Wriggers, P., Schröder, J., Schwarz, A.: A finite element method for contact using a third medium. *Comput. Mech.* **52**, 837–847 (2013)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.