Received: 29 June 2023 | Accepted: 22 August 2023

DOI: 10.1002/pamm.202300088

RESEARCH ARTICLE

Remeshing and data transfer in the finite cell method for problems with large deformations

Roman Sartorti | **Alexander Düster**

Hamburg University of Technology, Numerical Structural Analysis with Application in Ship Technology (M-10), Hamburg, Germany

Correspondence
Roman Sartorti, Hamburg University of Technology, Numerical Structural Analysis with Application in Ship Technology (M-10), Am Schwarzenberg-Campus 4 (C), 21073 Hamburg, Germany.
Email: roman.sartorti@tuhh.de

Funding information
Deutsche Forschungsgemeinschaft, Grant/Award Numbers: DU 405/21-1, 505137962

Abstract

The simulation of complex structures using standard finite element discretization techniques can be challenging because the creation of the boundary conforming meshes for such structures can be time-consuming. Therefore, fictitious domain methods are attractive alternatives because the underlying mesh does not have to conform to the boundary. One fictitious domain method is the finite cell method where a structured non-geometry conforming Cartesian grid is created and the geometry is then described by a simple indicator function. When solving non-linear problems, for example, if large deformations are taken into account, broken cells are typically heavily distorted and may lead to failure of the overall solution procedure. To tackle this issue remeshing is applied to proceed with the simulation. In previous publications, radial basis functions (RBFs) were applied to map the deformation gradient between the different meshes. However, it is not clear whether this is the best way to transfer data during the remeshing procedure. Therefore, we investigate whether there are alternative methods that can be used instead. This is then compared to the RBF interpolation and applied to different numerical examples with hyperelastic materials.

1 | INTRODUCTION

Numerical simulation of complex structures can be a challenging task, for example, for foam-like structures. Especially, the meshing procedure itself can be time-consuming if a boundary conforming mesh is required, where the finite elements (FEs) have to fulfil certain quality requirements. Therefore, immersed methods, such as the finite cell method (FCM) [1], CutFEM [3, 4] and CutIGA [5], are appealing because they do not require boundary conforming meshes.

However, in large deformation states, these methods may also suffer from highly distorted broken cells. This is similar to standard finite element method (FEM) when it comes to a large deformation state—elements are highly distorted and may break the solution process. To overcome this, one possibility is to stabilise the iterative solver, see, for example, Garhuom et al. [6]. Nevertheless, we may end up at a certain state where we cannot continue the simulation. This is the point where remeshing comes into play.

Remeshing can improve the mesh quality during the simulation. Typically, this is done after a certain state of possibly large deformation, where some cells distort heavily. In order to keep track of the deformation history, the predeformation is stored and transferred to the new mesh. Typically, this predeformation history is stored at the integration points level

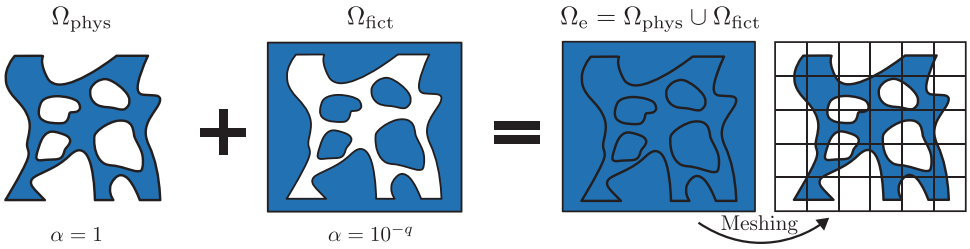

FIGURE 1 Basic idea of the finite cell method. After extending the physical domain by a fictitious domain, mesh creation is straightforward.

and is not available at an arbitrary point in the structure. However, when creating a new mesh, there are new integration points that typically do not match the position of old integration points. Thus, the values at new integration points need to be interpolated from the old values. To this end, radial basis functions (RBFs) are used for the interpolation and have successfully been applied in [2, 7].

As an alternative approach in the present work, we apply an L^2-projection, to project the deformation gradient into the FE ansatz space. By doing so, we can evaluate the deformation gradient at any point in the domain by using the shape functions. One may argue that this is also possible without the projection. However, since we rely on the deformation gradient w.r.t. the initial configuration, this is only possible for the initial mesh. Moreover, the deformation gradient is not continuous at element boundaries and, especially for multiple remeshings, all the previous meshes must be kept to do the inverse mapping through multiple meshes. By projection to the FE-space, we elevate the continuity of the deformation gradient to be at least C^0-continuous which smoothens possible discontinuities at the element boundaries.

2 | THEORY

2.1 | FCM

The principle idea of the FCM is to embed the physical domain Ω_{phys} into a simpler shaped domain $\Omega_e = \Omega_{\text{phys}} \cup \Omega_{\text{fict}}$ that can be easily meshed using quadrilateral or hexahedral cells, see Figure 1.

To distinguish between the physical and fictitious domain, an indicator function is introduced:

$$\alpha(\boldsymbol{x}) = \begin{cases} 1, \text{ for } \boldsymbol{x} \in \Omega_{\text{phys}} \\ \alpha_0 = 10^{-q}, \text{ else,} \end{cases} \tag{1}$$

where α_0 is a small positive value with $q = 6, 7, 8, \ldots$. This indicator function is then used to trim the integral to the physical domain boundary, see also [8]. Choosing $\alpha_0 = 0$ leads to the solution of the original problem. However, for stability reasons, we choose $q = 5$.

Due to the immersed interface inside the cells, a discontinuity is introduced. This needs to be taken into account during integration of system matrices. Different approaches exist, and the most common approaches are the standard Oc-Tree or Quad-Tree [1, 8] and its derivates, moment fitting [9–11] and non-negative moment fitting [12]. In the present work, we apply the non-negative moment-fitting scheme for the numerical integration. For further details on this method, the reader is referred to [12, 13].

2.2 | Remeshing

In large deformation scenarios, the solution may not converge after a certain deformation state is reached. To continue the simulation, we apply remeshing on the last converged solution, similar to reference [7]. The kinematic relations are shown in Figure 2. For simplicity, it is only shown for three deformation states that are involved in a single remeshing procedure. However, the same holds true for multiple remeshing steps. Starting from an initial configuration Ω_0, the structure is deformed until at deformation state Ω_n, where either the solution fails or a remeshing criterion is fulfilled. Then, a new

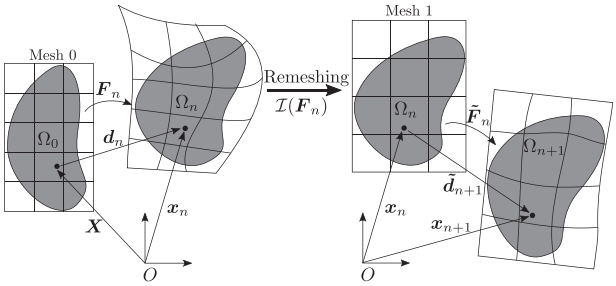

FIGURE 2 Kinematic relations and important quantities in different configurations and meshes that are involved in one remeshing step.

mesh is created based on configuration Ω_n and the simulation is continued to deformation state Ω_{n+1}. The simulation stops either when the final state is reached or another remeshing is initiated.

During this procedure, the total displacement field $\boldsymbol{d}$ is split into the partial displacements $\tilde{\boldsymbol{d}}_i$ arising from the different meshes. Thus, the total displacement at any point of the structure is the sum of the partial displacements, that is,

$$\boldsymbol{d} = \tilde{\boldsymbol{d}}_1 + \tilde{\boldsymbol{d}}_2 + \cdots + \tilde{\boldsymbol{d}}_{n+1} = \boldsymbol{d}_n + \tilde{\boldsymbol{d}}_{n+1} \tag{2}$$

holds. However, every time a new mesh is created, the previous displacement $\boldsymbol{d}_n$ is not known to the new mesh. To account for the previous deformation history, the total deformation gradient $\boldsymbol{F}_n$ of the last state is stored as predeformation variable at the integration points.

To compute the total deformation gradient w.r.t. to the initial configuration, we exploit the chain rule incorporating all the intermediate configurations. This leads to

$$\boldsymbol{F}_{n+1} = \underbrace{\frac{\partial \boldsymbol{x}_{n+1}}{\partial \boldsymbol{x}_n}}_{\tilde{\boldsymbol{F}}_{n+1}} \underbrace{\frac{\partial \boldsymbol{x}_n}{\partial \boldsymbol{x}_{n-1}} \cdots \frac{\partial \boldsymbol{x}_1}{\partial \boldsymbol{X}}}_{\boldsymbol{F}_n}. \tag{3}$$

The current deformation gradient w.r.t. the intermediate configuration Ω_n is computed by

$$\tilde{\boldsymbol{F}}_{n+1} = \boldsymbol{I} + \frac{\partial \tilde{\boldsymbol{d}}_{n+1}}{\partial \boldsymbol{x}_n}. \tag{4}$$

To compute the total deformation gradient $\boldsymbol{F}_{n+1}$, the partial deformation gradients from previous meshes are required. As already mentioned, this necessitates efficient data handling and storage of old discretizations and their corresponding solution fields. Another approach is to interpolate the last converged total deformation gradient from the old integration points to the new integration points of the new mesh. By doing so, old data can be deleted, once the data transfer is completed. Thus, the total deformation gradient is approximated by

$$\boldsymbol{F}_{n+1} \approx \tilde{\boldsymbol{F}}_{n+1} \mathcal{I}(\boldsymbol{F}_n), \tag{5}$$

where $\mathcal{I}$ is representing the interpolation scheme.

2.3 | Interpolation methods

2.3.1 | Radial basis function interpolation

One possibility to perform the data interpolation is the usage of RBFs, similar to references [2, 7]. There exist different RBFs, see, for example, [14]. We use the inverse multiquadric RBF defined by

$$\Phi(r) = \frac{1}{\sqrt{1 + r^2}}. \tag{6}$$

To compute the values of a given quantity f at a target point $\boldsymbol{x}^t$, we first solve a system of equations for N_s source points. In the present work, we use the $N_s = 50$ nearest neighbours for this purpose.

$$\begin{bmatrix} \Phi(r_{11}) & \cdots & \Phi(r_{1N_s}) \\ \vdots & \ddots & \vdots \\ \Phi(r_{11}) & \cdots & \Phi(r_{N_sN_s}) \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_{N_s} \end{bmatrix} = \begin{bmatrix} f(\boldsymbol{x}_1^s) \\ \vdots \\ f(\boldsymbol{x}_{N_s}^s) \end{bmatrix}, \; r_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|. \tag{7}$$

Once the weights w_i are compute, the target value f at target point $\boldsymbol{x}^t$ can be computed by evaluating

$$f(\boldsymbol{x}^t) = \sum_{i=1}^{N_s} \Phi(\|\boldsymbol{x}^t - \boldsymbol{x}_i^s\|) w_i. \tag{8}$$

The benefit of this approach is the straightforward implementation as an add-on solution to an existing FE-code. The only required interface is the access to integration points. Additionally, the process can be parallelised to increase the performance.

2.3.2 | L^2-projection

As an alternative approach, we first map the values from the integration points into the FE-ansatz space of the old mesh by minimising the error in the L^2 norm. To do so, we solve the following system:

$$\underbrace{\int_{\Omega_e} \boldsymbol{N}^\top \boldsymbol{N} \, \mathrm{d}\Omega}_{\boldsymbol{M}} \boldsymbol{\mu} = \int_{\Omega_e} \boldsymbol{N}^\top f \, \mathrm{d}\Omega, \tag{9}$$

where $\boldsymbol{N}$ contains all the ansatz functions. The left-hand side is the standard mass matrix and is computed by numerical integration. The computation of the right-hand side is also straightforward, since the predeformation data are already present at the integration points. Once the projection is computed, the predeformation data on the new integration points are updated by applying inverse mapping of the new integration points into the old mesh. To avoid instabilities during inverse mapping, we set $\det(\tilde{\boldsymbol{F}}_{n+1}) <= 0$ as a remeshing criterion.

In contrast to the RBF interpolation, this approach requires more implementation effort. However, we can directly make use of the ansatz space to compute values at arbitrary points. The main computational effort is the solution of Equation (9) and the inverse mapping between the meshes.

3 | NUMERICAL EXAMPLES

In the following, we compare the presented interpolation methods on several numerical examples. All models assume plane strain conditions. The material is of Neo-Hookean type with the following strain energy density function:

$$\Phi(\boldsymbol{F}) = \frac{\mu}{2}(\mathrm{tr}(\boldsymbol{F}^\top \boldsymbol{F}) - 3) + \frac{\lambda}{2}(J^2 - 1) - \left(\frac{\lambda}{2} - \mu\right) \ln(J), \tag{10}$$

where $J = \det \boldsymbol{F}$. The first Lamé parameter is set to $\mu = 19.231\,\mathrm{N/mm}^2$ and the shear modulus is set to $\lambda = 28.846\,\mathrm{N/mm}^2$.

Evaluation:
To compare the RBF and the L^2-projection approach, the error analysis is carried out as follows:

1. Deform the structure until state $\bar{\boldsymbol{d}}$.
2. Create new mesh.
3. Compute error by integrating over the physical domain in the new mesh.

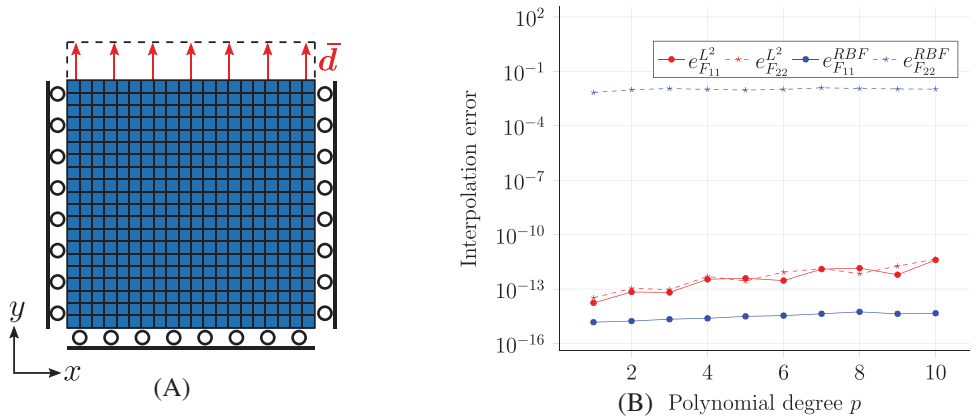

FIGURE 3 (A) Geometry, discretization mesh and boundary conditions. (B) Error plot for different polynomial degrees.

(a) Evaluate interpolated deformation gradient $\mathcal{I}(F_{ij})$.
(b) Compute the exact value F_{ij} in the old mesh via inverse mapping.
(c) Integrate error over physical domain Ω_{phys}

$$e_{F_{ij}} = \sqrt{\int_{\Omega_{\text{phys}}} \left(\mathcal{I}(F_{ij}) - F_{ij}\right)^2 \Omega}. \tag{11}$$

4. Repeat the same test for meshes of different ansatz orders p.

3.1 | Square plate

In this example, we consider a square plate for which we can create a boundary conforming mesh. Therefore, we can exclude any error coming from the immersed interface inside broken cells. The geometry as well as the initial mesh is shown in Figure 3A. Due to the boundary conditions, the deformation state is one dimensional leading to changes only in 22-direction of the deformation gradient.

As can be seen from the Figure 3b, the error remains constant for F_{22} in the RBF approach, whereas the L^2 approach leads to almost numerically zero error. This can be explained from the fact that the RBF cannot represent a constant function, and hence, cannot interpolate the given deformation gradient field exactly. On the other hand, the L^2 projection can represent a constant field. The remaining error is more related to rounding errors during projection and inverse mapping.

3.2 | Plate with hole

Now, we consider a model with an immersed interface. This is the well-known perforated plate example under symmetry boundary conditions. The model and the initial mesh are shown in Figure 4A and the maximum displacement is set to $\bar{\boldsymbol{d}} = 10\,\text{mm}$.

The results are shown in Figure 4B. It is clearly visible that the data transfer does not work properly for low ansatz orders. However, for sufficiently high ansatz order, we reach a state of constant error level. We assume that the remaining error comes from the change of the ansatz space and not the interpolation itself. It should be noted that still the L^2-projection approach leads to smaller interpolation error in total.

3.3 | Foam pore

Single remeshing

As in the previous example, we trigger the remeshing after $\bar{\boldsymbol{d}} = 5\,\text{mm}$. The initial mesh and the error for different polynomial degrees are shown in Figure 5A. As shown, the error decreases for higher polynomial degrees. In case of RBF,

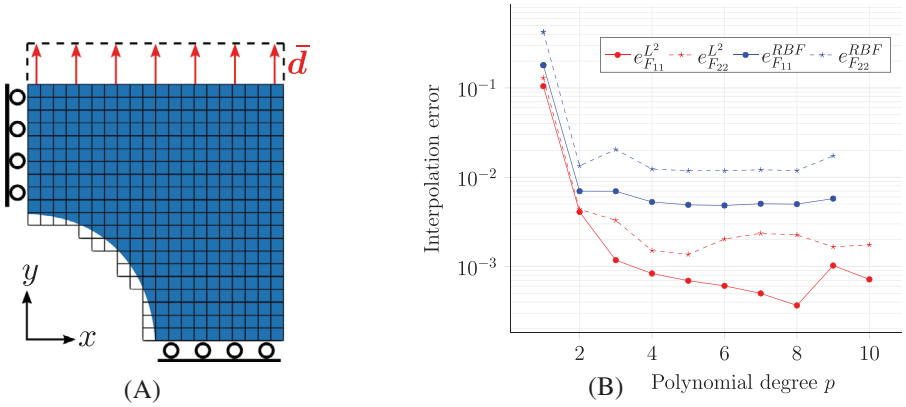

FIGURE 4 (A) Geometry, discretization mesh and boundary conditions. (B) Error plot for different polynomial degrees.

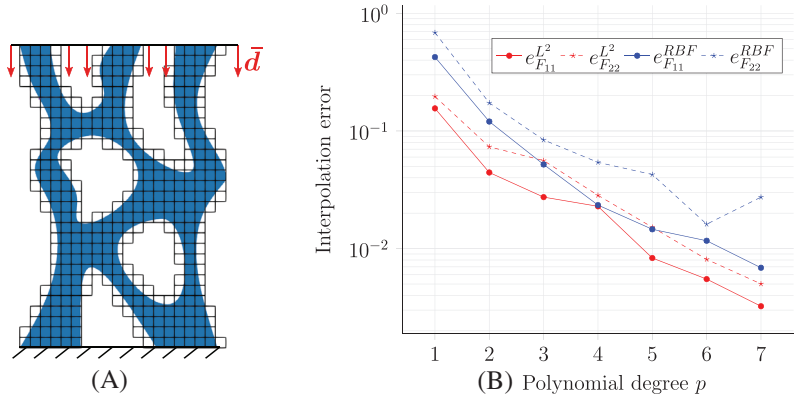

FIGURE 5 (A) Geometry, discretization mesh and boundary conditions. (B) Error plot for different polynomial degrees.

this is caused by increasing number of integration points that are available as source points for the interpolation. In the L^2-projection case, we increase the accuracy by higher order shape functions that are capable to capture the distribution with increasing accuracy. Even though the differences in the error between the two methods are smaller as compared to the previous examples, the L^2-approach is always at least as accurate as the RBF interpolation.

Multiple remeshings and displacement field projection

Finally, we show the projection method for multiple remeshings triggered whenever $\det(\tilde{\boldsymbol{F}}) <= 0$ is reached. The polynomial degree is fixed to $p = 5$ and the load step size is set to $\Delta\bar{\boldsymbol{d}} = 0.2$ mm. Since we can no longer compute the exact value of the total deformation gradient to compare it with the exact value, we measure the error in energy norm by evaluating

$$e_r = \sqrt{\left|\frac{\mathcal{U}_{n+1} - \mathcal{U}_n}{\mathcal{U}_n}\right|}, \tag{12}$$

where $\mathcal{U}_n$ and $\mathcal{U}_{n+1}$ are the strain energy in the mesh n and $n+1$, respectively. To investigate the change in the error during the equilibrium step, the strain energy is computed twice: once directly after the interpolation procedure and once after the equilibrium step.

Additionally, we use a second L^2-projection to transfer the displacement field to the ansatz space of the new mesh. The global error of data transfer is then computed by

$$e_{\boldsymbol{d}_n} = \sqrt{\int_{\Omega_{\text{phys}}} (\mathcal{I}(\boldsymbol{d}_n) - \boldsymbol{d}_n)^{\top} (\mathcal{I}(\boldsymbol{d}_n) - \boldsymbol{d}_n) \Omega}. \tag{13}$$

TABLE 1 Error in displacement and energy norm for multiple remeshing steps triggered at deformation $\boldsymbol{d}_n$. The L^2-projection approach is applied for the data transfer.

Mesh n	d_n	e_{d_n} [mm]	Error in energy norm e_r^{pre}	e_r^{post}
0	6.50 mm	0.0239	1.8%	2.1%
1	10.50 mm	0.0685	0.9%	0.9%
2	13.25 mm	0.1845	2.9%	3.5%
3	16.00 mm	0.1898	4.3%	3.2%
4	16.50 mm	0.0539	2.3%	–

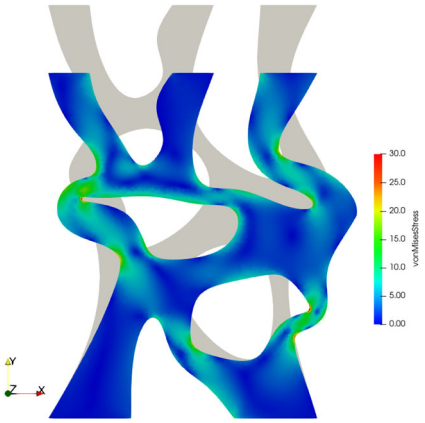

FIGURE 6 von Mises stress in the last deformation state.

The results are summarised in Table 1. As can be seen, the remeshing introduces an error in the strain energy, that is, the method is not energy-conserving. The main driver for the error is the interpolation procedure. The error due to discretization changes is balanced out in the equilibrium step which is done on the new mesh at same load level as before the remeshing. Additionally, it should be noted that the number of load increments between remeshings decreases. We suspect that this is caused by the broken cells, which are prone to self-contact. In addition, the sharp kinks that are present at the buckled structures show high-stress peaks, compare Figure 6. This indicates a large change in the displacement gradients within one cell having a negative impact on the stability of the solution.

4 | CONCLUSION

The present work shows that the L^2-projection is a reasonable alternative to RBF interpolation during the data transfer in the remeshing procedure. This is shown for simple as well as for complex examples in 2D and it can be seen that the proposed method leads to a reduced global error in the deformation gradient interpolation. Moreover, we mapped successfully the deformation field into the new ansatz space.

The main restriction arises from the inverse mapping where we require that the determinant of the deformation gradient remains positive to ensure convergence. Typically, this happens in the fictitious domain. We tackle this by applying remeshing whenever $\det \boldsymbol{F} <= 0$, but this may also be circumvented by selecting a better initial guess for the inverse mapping.

In contrast, the RBF approach works more stable, because the mapping is performed between point sets, that is, the deformation of the mesh has no influence on the interpolation. In future, the accuracy may be improved by polynomial extension of the RBF, for example, done in [15].

As a further natural extension, we aim to include self-contact in the problem formulation. Especially, in foam-like structures, pores will collapse at some point leading to self-contact. This will be investigated in future work.

ACKNOWLEDGMENTS

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.

Open access funding enabled and organized by Projekt DEAL.

ORCID

Roman Sartorti https://orcid.org/0000-0002-3410-8584
Alexander Düster https://orcid.org/0000-0002-2162-3675

REFERENCES

1. Parvizian, J., Düster, A., & Rank, E. (2007). Finite cell method: h- and p-extension for embedded domain problems in solid mechanics. *Computational Mechanics*, *41*(1), 121–133.
2. Garhuom, W., Hubrich, S., Radtke, L., & Düster, A. (2021). A remeshing approach for the finite cell method applied to problems with large deformations. *Proceedings in Applied Mathematics and Mechanics*, *21*(1), e202100047.
3. Burman, E., & Hansbo, P. (2010). Fictitious domain finite element methods using cut elements: I. A stabilized Lagrange multiplier method. *Computer Methods in Applied Mechanics and Engineering*, *199*(41), 2680–2686.
4. Burman, E., Claus, S., Hansbo, P., Larson, M. G., & Massing, A. (2015). CutFEM: Discretizing geometry and partial differential equations. *International Journal for Numerical Methods in Engineering*, *104*(7), 472–501.
5. Elfverson, D., Larson, M. G., & Larsson, K. (2018). CutIGA with basis function removal. *Advanced Modeling and Simulation in Engineering Sciences*, *5*(1), 6.
6. Garhuom, W., Usman, K., & Düster, A. (2022). An eigenvalue stabilization technique to increase the robustness of the finite cell method for finite strain problems. *Computational Mechanics*, *69*(5), 1225–1240.
7. Garhuom, W., Hubrich, S., Radtke, L., & Düster, A. (2020). A remeshing strategy for large deformations in the finite cell method. *Computers & Mathematics with Applications*, *80*(11), 2379–2398.
8. Düster, A., Parvizian, J., Yang, Z., & Rank, E. (2008). The finite cell method for three-dimensional problems of solid mechanics. *Computer Methods in Applied Mechanics and Engineering*, *197*(45), 3768–3782.
9. Joulaian, M., Hubrich, S., & Düster, A. (2016). Numerical integration of discontinuities on arbitrary domains based on moment fitting. *Computational Mechanics*, *57*(6), 979–999.
10. Hubrich, S., Di Stolfo, P., Kudela, L., Kollmannsberger, S., Rank, E., Schröder, A., & Düster, A. (2017). Numerical integration of discontinuous functions: Moment fitting and smart octree. *Computational Mechanics*, *60*(5), 863–881.
11. Hubrich, S., & Düster, A. (2019). Numerical integration for nonlinear problems of the finite cell method using an adaptive scheme based on moment fitting. *Computers & Mathematics with Applications*, *77*(7), 1983–1997.
12. Garhuom, W., & Düster, A. (2022). Non-negative moment fitting quadrature for cut finite elements and cells undergoing large deformations. *Computational Mechanics*, *70*(5), 1059–1081.
13. Legrain, G. (2021). Non-negative moment fitting quadrature rules for fictitious domain methods. *Computers & Mathematics with Applications*, *99*, 270–291.
14. Buhmann, M. D. (2003). *Radial basis functions: Theory and implementations* (1st ed.). Cambridge University Press.
15. Tóth, B., & Düster, A. (2023). H-Adaptive radial basis function finite difference method for linear elasticity problems. *Computational Mechanics*, *71*(3), 433–452.

How to cite this article: Sartorti, R., & Düster, A. (2023). Remeshing and data transfer in the finite cell method for problems with large deformations. *Proceedings in Applied Mathematics and Mechanics*, *23,* e202300088. https://doi.org/10.1002/pamm.202300088