



Data transfer within a finite cell remeshing approach applied to large deformation problems

Roman Sartorti¹ · Alexander Düster¹

Received: 30 January 2024 / Accepted: 10 April 2024
© The Author(s) 2024

Abstract

The present work is a comparative study of different data transfer techniques in the context of the finite cell method (FCM) in combination with remeshing for hyperelastic problems undergoing large deformations. The FCM is an immersed-boundary method that uses Cartesian grids for the discretization so as to avoid the generation of boundary conforming meshes. To overcome problems with heavily distorted meshes at large deformation states, we apply a remeshing procedure. During the remeshing, the data containing the deformation history has to be transferred between the meshes. In the present study, different methods are considered and compared: radial basis functions without and with polynomial extension, inverse distance weighting, and L_2 -projection applying the shape functions used in the FCM for the trial and test functions.

Keywords Finite cell method · Data transfer · Radial basis function · Interpolation · Large deformation

This contribution is dedicated to the occasion of Professor Jörg Schröder's 60th birthday. I (Alexander) first met him in 1999 at the Institute of Mechanics at the University of Stuttgart, where he was a Post-Doctoral Researcher in Professor Miehe's group. At the time, I was a Ph.D. student at the Technical University of Munich, and our mutual friends Andreas Koch and Wolfram Volk invited me to Stuttgart to give a presentation. Since then, I have had the pleasure of meeting Jörg Schröder on numerous occasions, including GAMM meetings and jointly organized sections and workshops. I also had the opportunity to participate in the Priority Programme (SPP) 1748: Reliable Simulation Techniques in Solid Mechanics, which Jörg Schröder coordinated. His extensive knowledge and expertise in the broad field of mechanics make him an inspiring colleague. Additionally, his pleasant personality makes spending time with him a true joy. On this special occasion, we both, Alexander and Roman, extend our heartfelt congratulations to Professor Schröder and wish him all the best for the future.

Roman Sartorti and Alexander Düster have contributed equally to this work.

✉ Roman Sartorti
roman.sartorti@tuhh.de

Alexander Düster
alexander.duester@tuhh.de

¹ 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

1 Introduction

Numerical simulations of complex structures are widely applied in the field of product development. In static analysis, often the main interest is to identify stress hotspots relevant for static strength requirements and to check the stiffness/deformation given specific loads. With the development of new manufacturing techniques, the designers have gained more freedom, further supported by structural optimization techniques. This leads to increased complexity of the model and, hence, the effort to create boundary conforming meshes of good quality increases as well.

On the other hand, the simulation of structures obtained from 3D- or CT-scans, such as foams or bones, can be challenging for the same reason. Therefore, fictitious domain or immersed-boundary methods are an attractive alternative to standard finite element discretization methods. Different immersed-boundary methods exist: the finite cell method (FCM) [1, 2], cutFEM [3–5], or cutIGA [6].

The FCM has already been further developed for various fields of application. Thermoelastic [7], geometrically nonlinear [8, 9], and elastoplastic [10–12] analyses have been performed in previous works. Moreover, the FCM has successfully been applied to biomechanics [13–16], vibroacoustics [17], as well as to implicit [18] and explicit structural dynamics [19].

In the present work, we focus on the data transfer that is required during the remeshing procedure presented in [9] in the context of large deformation analysis. The remeshing is applied to overcome problems caused by severely distorted cells, which could otherwise lead to convergence problems in the incremental/iterative solution procedure. Whenever a new mesh is generated, some of the data stored at the integration points has to be transferred to incorporate the deformation history. To this end, radial basis function (RBF) interpolation can be used for the data transfer. In other research fields—like level-set-based topology optimization [20, 21] or the RBF finite difference (RBF-FD) method [22, 23]—it is common practice to use RBFs with polynomial extension. The combination of polyharmonic splines (PHS) with the polynomial extension tends to yield improved accuracy for different numerical examples, especially in RBF-FD.

Nevertheless, the RBF interpolation methods come at the price of having to solve additional linear systems of equations. Therefore, we further test the accuracy of the inverse distance weighting function (IDW) in order to consider a numerically cheaper approach to data transfer.

The IDW as well as the RBF are point-based interpolation methods. Another possibility is to use a mesh-based approximation. Thereby, the data is first projected onto the finite element/cell ansatz space using an L_2 -projection and then evaluated at the required integration points of the new FCM mesh.

The rest of the paper is organized as follows: in Sect. 2, the FCM is introduced for non-linear problems of solid mechanics. Then, the remeshing is explained, followed by a detailed explanation of the different data transfer techniques mentioned before. In Sect. 3, several numerical examples are presented in order to investigate the properties of the different approaches to data transfer. Finally, the conclusions are presented in Sect. 4.

2 Finite cell method for problems with large deformations

2.1 Basic formulation

The basic idea of the FCM is to extend the physical domain Ω such that the resulting domain Ω_e is simply shaped. Then, the extended domain can be discretized using Cartesian grids in 2D or 3D, see also Fig. 1. To distinguish between the physical and the fictitious domain, an indicator function $\alpha(\mathbf{x})$ is introduced

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

Thereby, α_0 is typically a small positive value, e.g., $\alpha_0 = \{10^{-6}, \dots, 10^{-3}\}$, used to stabilize the fictitious domain [2].

To solve problems of solid mechanics, the weak form of equilibrium is defined as

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

where $\mathbf{P}(\mathbf{d})$ is the first Piola–Kirchhoff stress tensor depending on the displacement field \mathbf{d} . The tractions $\hat{\mathbf{t}}$ are applied on the boundary Γ_N , and $\boldsymbol{\eta}$ represents the test function. Please note that, for the sake of clarity, volumetric loads are not considered but could be included without any problems. From Eq. (2), it can be seen that if $\alpha_0 = 0$, the weak form of the extended domain (2) reduces to the weak form of the physical domain [1].

The Newton–Raphson procedure is applied to solve Eq. (2). Thus, the weak form is linearized w.r.t. the displacement increment $\Delta \mathbf{d}$ at deformation state $\bar{\mathbf{d}}$. The directional derivative is given by

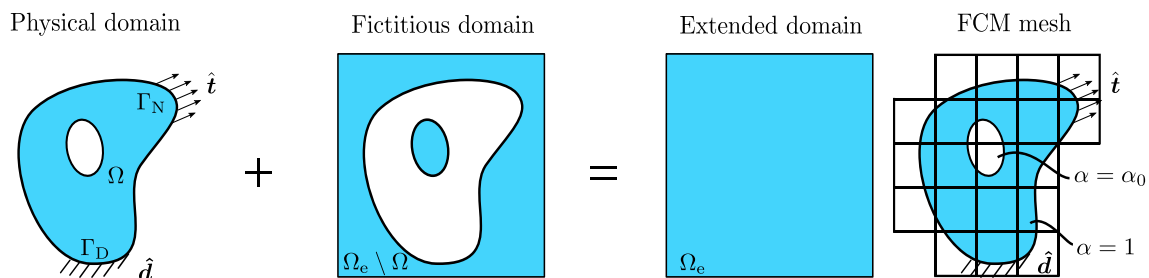


Fig. 1 Visualization of the finite cell method. The physical domain Ω is embedded into an extended domain Ω_e that can easily be discretized. Cells that are completely empty are disregarded. In the fictitious domain, α_0 is a small positive value for stabilization purposes

$$DG_e^\alpha(\bar{d}, \eta) \cdot \Delta d = \int_{\Omega_e} \alpha A(\bar{d}) \text{Grad} \Delta d \cdot \text{Grad} \eta \, d\Omega \quad (3)$$

where

$$A(\bar{d}) = \left. \frac{\partial \mathbf{P}}{\partial \mathbf{F}} \right|_{d=\bar{d}}. \quad (4)$$

This leads to the linearized weak form

$$\begin{aligned} & \int_{\Omega_e} \alpha A(\bar{d}) \text{Grad} \Delta d \cdot \text{Grad} \eta \, d\Omega \\ &= \int_{\Gamma_N} \hat{\mathbf{t}} \cdot \boldsymbol{\eta} \, d\Gamma - \int_{\Omega_e} \alpha \mathbf{P} \cdot \text{Grad} \eta \, d\Omega. \end{aligned} \quad (5)$$

Then, the displacement field and the test functions are approximated by $\bar{d} = \mathbf{N}U$ and $\boldsymbol{\eta} = \mathbf{N}V$, respectively, where \mathbf{N} denotes the shape function matrix. The approach is very similar to the standard non-linear finite element procedure, see for example [24, 25].

To evaluate the integrals, different quadrature schemes are available. Thereby, the immersed interface has to be taken into account. One approach is the spacetree method [1, 26, 27] where cells are subdivided if they are intersected by the boundary. This division is performed until a certain spacetree depth is reached. This method, however, introduces a lot of integration points and can, therefore, increase the overall simulation time drastically. Another approach is the so-called moment-fitting method [28–31], where new integration weights are computed for each broken

cell. Even though this procedure requires the spacetree integration method during the computation of the new weights, it is appealing especially if applied to time-dependent or nonlinear problems. Then, the overhead of the moment fitting method is distributed across the time/load steps, and the reduced number of integration points increases the efficiency significantly. However, this method may introduce negative weights that can be problematic for nonlinear analysis. Therefore, non-negative moment fitting (NNMF), as introduced in [31, 32], ensures that all weights remain positive. In the present investigation, we employ the octree method, which is a three-dimensional version of the spacetree method, and the NNMF method. For further details, we refer to the cited literature.

2.2 Material model

The applied hyperelastic material model in the present study is based on the Neo-Hookean strain energy function [25]

$$\Psi = \frac{\mu}{2} (\text{tr}C - 3) + \frac{\lambda}{4} (J^2 - 1) - \left(\frac{\lambda}{2} + \mu \right) \ln J, \quad (6)$$

from which the first Piola–Kirchhoff stress tensor can be obtained:

$$\mathbf{P} = 2\mathbf{F} \frac{\partial \Psi}{\partial \mathbf{C}} = \frac{\lambda}{2} (J^2 - 1) \mathbf{F}^{-\top} + \mu (\mathbf{F} - \mathbf{F}^{-\top}). \quad (7)$$

The material parameters are set to $\lambda = 28.846$ MPa and $\mu = 19.231$ MPa. Additionally, the derivative of the first

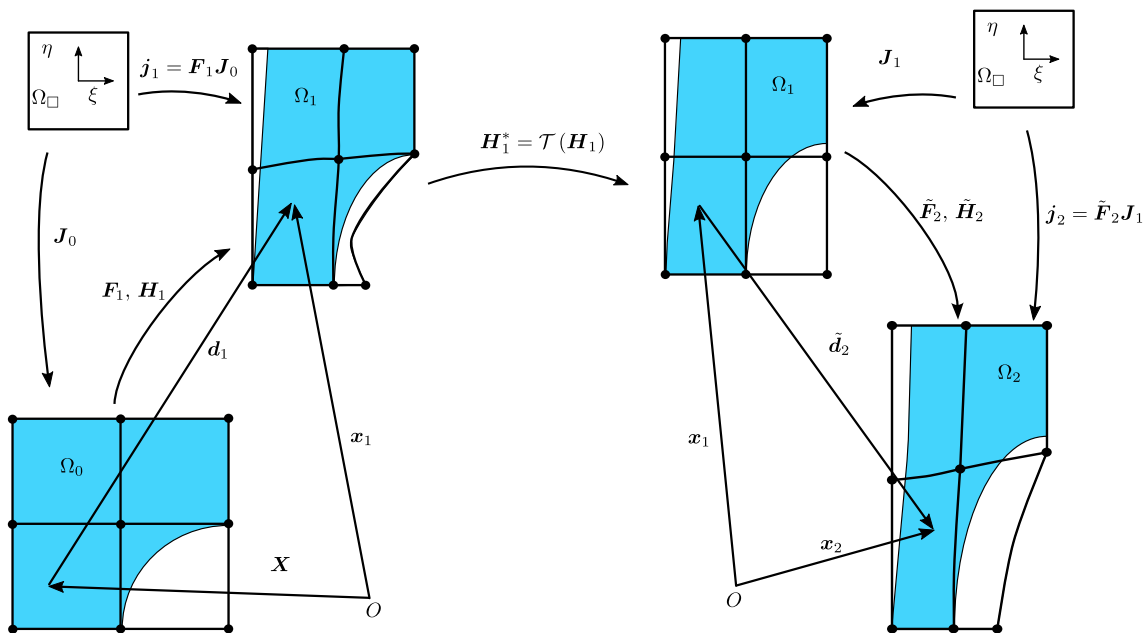


Fig. 2 Visualization of the remeshing procedure and the involved kinematic relations. The structure is deformed until deformation state Ω_n . Then, the displacement gradient is transferred to the new mesh applying the transfer method T

Piola–Kirchhoff stress tensor w.r.t. the deformation gradient is required for the tangent stiffness matrix computation. To this end, the computer algebra software *Mathematica* [33] is used with the *AceGen* extension [34].

2.3 Remeshing

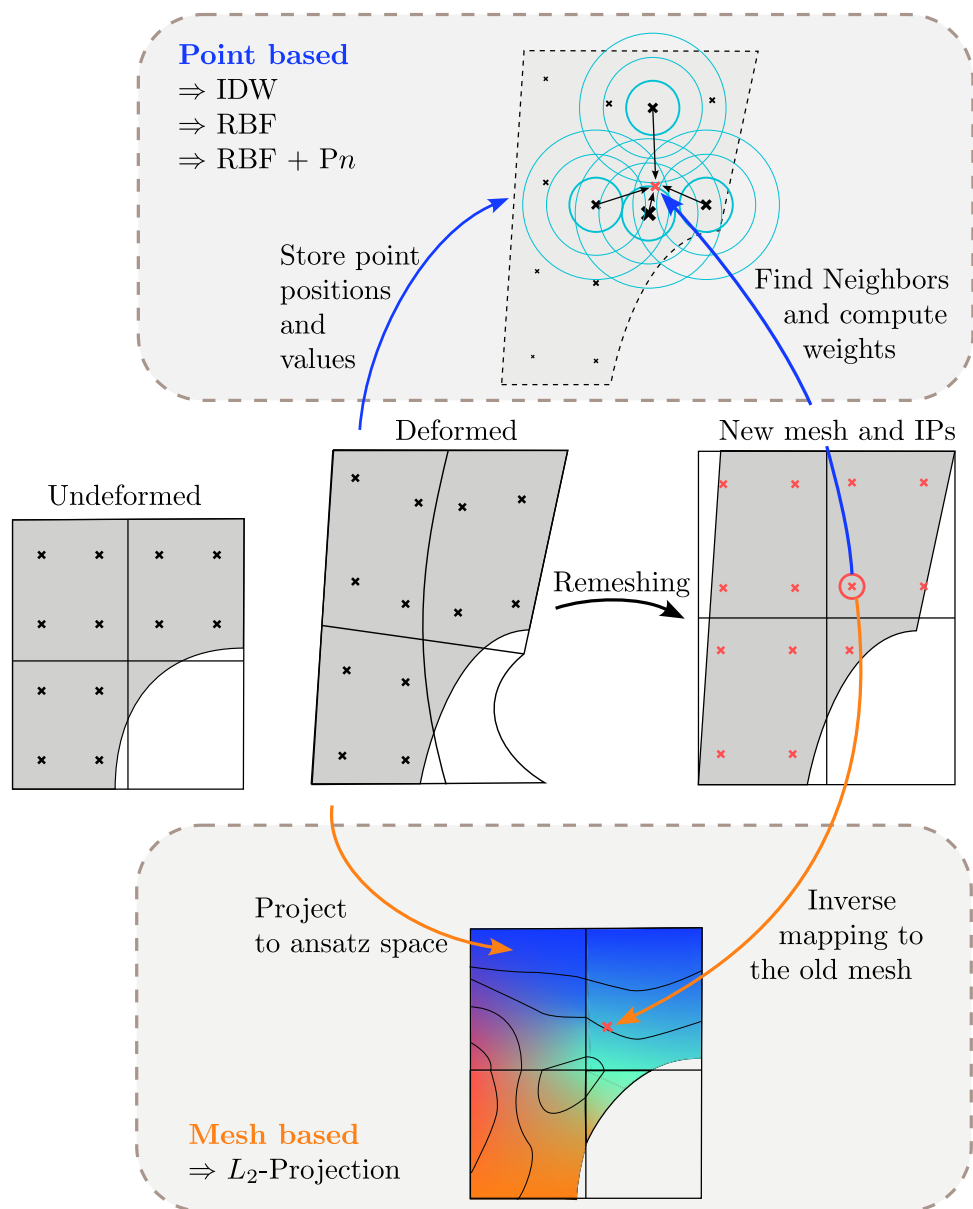
Remeshing has been successfully applied in the context of the finite cell method in [9]. The principle procedure is shown in Fig. 2. The initial configuration is discretized, and the load is applied incrementally until deformation state Ω_n is reached. Then, a new mesh is created based on the deformed structure, and the data is transferred to the new mesh by applying the data transfer \mathcal{T} . In the present study, we transfer the dis-

placement gradient \mathbf{H} from the old to the new mesh. Note that one could also consider the deformation gradient \mathbf{F} instead. However, this increases the error during the transfer, caused by the constant offset that may not be recovered by the data transfer method. A more detailed discussion regarding this aspect is given in Sect. 3.1.

2.4 Data transfer

The data transfer is one key aspect of the present paper. Therefore, the different methods that are applied in the present study are briefly introduced in this section. Additionally, the different methods are visualized in Fig. 3.

Fig. 3 Visualization of the data transfer process for the different methods. The point-based methods rely only on the information stored at the source points. Then, the source values v^s are weighted based on the radial distance r , either by distance weighting (IDW) or by solving a linear system of equations (RBF, RBF + P_n) to compute the target value v^t . In contrast, the mesh-based method first projects the values to the degrees of freedom and, then, an inverse mapping to evaluate the shape functions at the target point location in the old mesh



2.4.1 Radial basis function interpolation with polynomial extension

Radial basis functions are applied for various applications. Surface reconstruction [35], strain determination for additively manufactured parts [36], level-set-based topology optimization [20, 21], and RBF finite difference (RBF-FD) schemes [22, 23] are only a few examples. In the present study, we use the RBF for data interpolation. The computational speed can be increased by following the local RBF approach, where only the N^s nearest neighbors for each target point are chosen.

The standard RBF interpolation approach can be derived as follows: We start by expressing the interpolated value v^t at a specific target point \mathbf{x}^t by the basis functions $\phi(r)$ with radial distance from source point at $r = \|\mathbf{x}^s - \mathbf{x}^t\|$. This can be written as

$$v^t(\mathbf{x}^t) = \sum_{i=1}^{N^s} w_i \phi(\|\mathbf{x}_i^s - \mathbf{x}^t\|) \tag{8}$$

The weights are initially unknown but can be computed by assembling the matrix, specifying the interpolation for each source point value v^s . This can be denoted in matrix–vector notation as

$$\mathbf{Q}\mathbf{w} = \mathbf{b} \tag{9}$$

where

$$Q_{ij} = \phi(\|\mathbf{x}_i^s - \mathbf{x}_j^s\|) \quad i, j = 1, \dots, N^s$$

$$b_j = v_j^s, \quad j = 1, \dots, N^s$$

Next, we want to extend the interpolation by a polynomial term, i.e.

$$\bar{v}^t(\mathbf{x}^t) = \sum_{i=1}^{N^s} w_i \phi(\|\mathbf{x}_i^s - \mathbf{x}^t\|) + \sum_{k=1}^{N^p} \ell_k P_k(\mathbf{x}^t) \tag{10}$$

Following the same steps as before, this would lead to an underdetermined system of equations of size $N^s \times (N^s + N^p)$. By introducing the constraint

$$\sum_{k=1}^{N^p} \ell_k P_k(\mathbf{x}_j^s) = 0 \quad \forall \ell_k, \tag{11}$$

the system can be written as follows

$$\begin{bmatrix} \mathbf{Q} & \mathbf{P} \\ \mathbf{P}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \boldsymbol{\ell} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} \tag{12}$$

where \mathbf{P} contains the polynomial terms.

Table 1 Polynomial extensions that are added for each polynomial degree

degree	1D	2D	3D
0	1	1	1
1	x	x, y	x, y, z
2	x^2	x^2, y^2, xy	x^2, y^2, xy, yz, zx

In a previous study [9], we applied the inverse multi-quadratic (invMQ) RBF

$$\phi(\hat{r}) = \frac{1}{\sqrt{1 + (c\hat{r})^2}}. \tag{13}$$

The normalized radius \hat{r} is thereby defined as

$$\hat{r} = \frac{r}{\bar{r}} \tag{14}$$

with \bar{r} being the mean distance of the N^t nearest neighbors. In the present work, the parameter c is set to one in all examples. In [22], it was shown that applying a polyharmonic spline (PHS) of the form

$$\phi(r) = r^m \tag{15}$$

as RBF with polynomial extension is of advantage for RBF-FD simulations, where m is an odd number, e.g. $m = 3, 5, 7$. One benefit of using PHS with polynomial extension is that this does not require any additional parameters that have to be set to specific values. Therefore, we will investigate the PHS as well as the invMQ RBF in combination with the polynomial extension (Pn) of polynomial extension degree n , see Table 1.

2.4.2 Inverse distance weighting

Another point-based interpolation method is the inverse distance weighting (IDW) method, sometimes also referred to Shepard’s interpolation method [37]. In contrast to the RBF interpolation, no system of equations has to be solved to obtain the weighting of source point values $v^s(\mathbf{x}_i^s)$. For each target point \mathbf{x}^t , the value v^t is computed by

$$v^t(\mathbf{x}^t) = \begin{cases} \frac{\sum_{i=1}^{N^s} w_i(\mathbf{x}^t) v^s(\mathbf{x}_i^s)}{\sum_{i=1}^{N^s} w_i(\mathbf{x}^t)} & , \text{ if } r(\mathbf{x}_i^s, \mathbf{x}^t) \neq 0 \\ v^s(\mathbf{x}_i^s) & , \text{ if } r(\mathbf{x}_i^s, \mathbf{x}^t) = 0 \\ \text{for all } i = 1, \dots, N^s & \text{for any } i = 1, \dots, N^s \end{cases} \tag{16}$$

where the weights w_i are obtained from the inverse distance by

$$w_i(\mathbf{x}^t) = \frac{1}{r(\mathbf{x}_i^s, \mathbf{x}^t)^\beta}. \tag{17}$$

The parameter β can be modified to control the influence of the neighboring points. In the present work, we apply $\beta = 2$. Likewise, again only the N^s nearest source points are considered for the interpolation to increase the efficiency of the method.

2.4.3 L_2 -projection

In contrast to the previous interpolation schemes, the L_2 -projection for the data transfer is an approximation method and works on a global level, meaning that data from the entire model is taken into account. Thereby, the data is projected onto the ansatz space of the FCM and then evaluated at each target point \mathbf{x}^t by an inverse mapping from the new mesh to the old mesh. The L_2 -projection is computed as follows:

$$\int_{\Omega} N^T N \, d\Omega \, \mathbf{w} = \int_{\Omega} N^T v^s(\mathbf{x}) \, d\Omega \tag{18}$$

where again, N is the matrix of the shape functions. The integral on the left-hand side can be identified as the mass matrix of the physical domain. To evaluate the integral on the right-hand side, the source values v^s are required at location \mathbf{x} . However, since the integral is evaluated numerically, those values are known because they are stored and, therefore, available at each integration point. After solving the system of equations, the inverse mapping is carried out for each target point \mathbf{x}^t using a Newton–Raphson scheme. Then, the value at the target point is computed by

$$v^t(\mathbf{x}^t) = N(\mathbf{x}^t)\mathbf{w}. \tag{19}$$

3 Numerical examples

In the subsequent sections, we will present different numerical examples. First, we will investigate the RBF with and without extension in the 1D case with the invMQ and the PHS RBF. Secondly, we will discuss different 3D examples, starting with a simple cube followed by the well-known perforated plate. Lastly, we will address the simulation of a single pore of a foam obtained from a 3D-scan.

3.1 Interpolation of functions

This example investigates the influence of using polynomial extension along with RBF interpolation. We start by defining two functions

$$f_1 = 1 + 0.1x \tag{20}$$

$$f_2 = 0.1x. \tag{21}$$

These two functions are then interpolated by means of RBF and RBF + P_n with a constant polynomial extension denoted by RBF + P0. First of all, the invMQ function is chosen as the RBF in this example, where $N^s = 10$ neighbors are used for interpolation and three neighbors N^t are used to compute the radial distance scaling \bar{r} according to Eq. (14).

The results of the interpolation are shown in Fig. 4. One can clearly see that the error increases by adding the constant term to the function to be interpolated. This also supports the results of previous investigations [9, 38] where it was found that interpolating the displacement gradient

$$\mathbf{H} = \frac{\partial \mathbf{u}}{\partial \mathbf{X}} \tag{22}$$

is preferable compared to the deformation gradient \mathbf{F} defined by

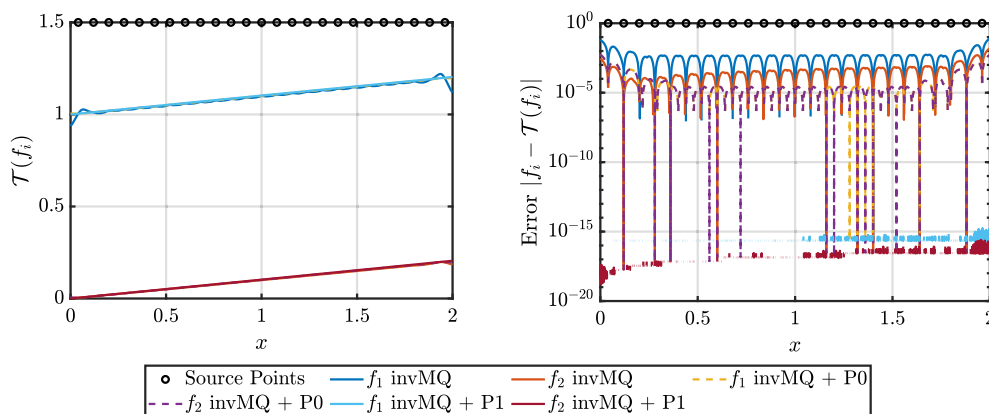
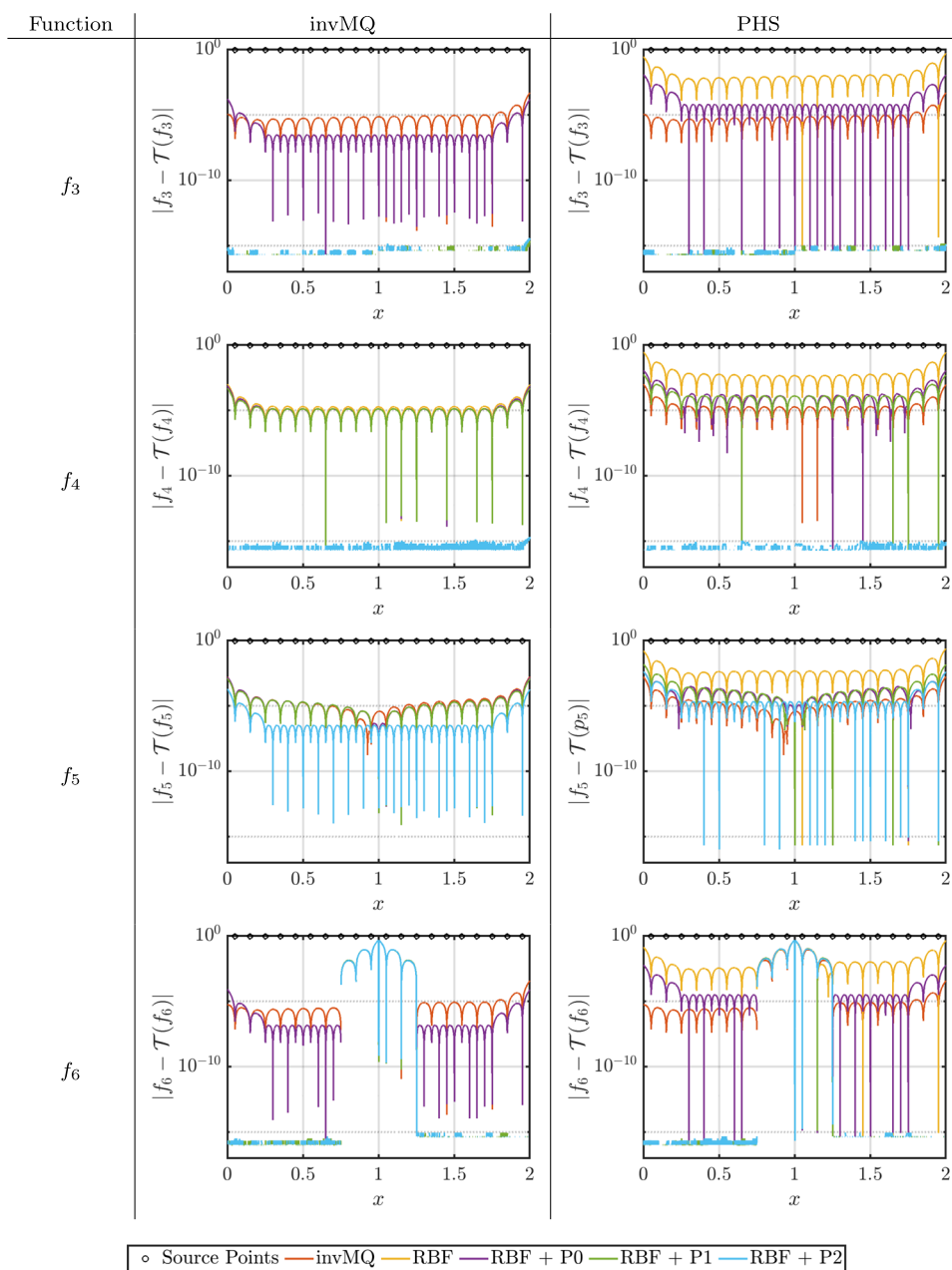


Fig. 4 The plot on the left shows the interpolation of the functions applying the different methods. The other plot shows the error between the interpolation and the exact function. The circles at the top indicate the position of the source points

Table 2 Error for the different functions defined in Eqs. (24)–(27)



$$\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} = \mathbf{I} + \frac{\partial \mathbf{u}}{\partial \mathbf{X}}. \tag{23}$$

From the above equation, it is evident that the only difference is the second-order identity tensor \mathbf{I} , which adds a constant term to the main diagonal. Note that using the L_2 -projection could also recover the constant offset because the linear shape functions can represent the constant properly.

Next, we investigate the PHSs and compare them to the invMQ functions. Thereby, we set $N^s = 6$ and disable the scaling with \bar{r} . Thereby, the following functions are considered for interpolation:

$$f_3 = 1 + x \tag{24}$$

$$f_4 = 1 + (x - 1)^2 \tag{25}$$

$$f_5 = 1 + (x - 1)^3 \tag{26}$$

$$f_6 = 1 + 0.5x + 0.5 \tanh(1000(x - 1)). \tag{27}$$

The RBFs are tested with different polynomial extension degrees P_n .

The absolute error for the different functions is shown in Table 2. For functions f_3 and f_4 , the error is reduced to machine accuracy whenever the polynomial extension is at least the polynomial degree of the function. For the present

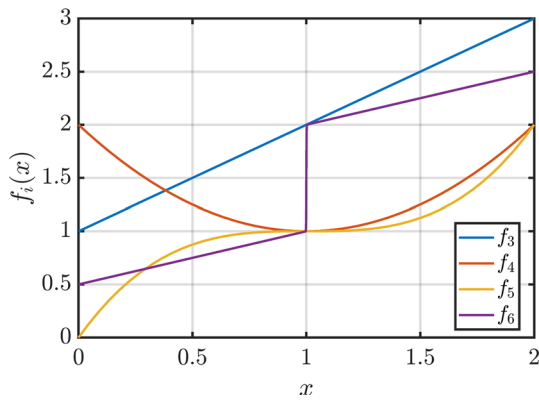


Fig. 5 Functions (24)–(27) used to test the data transfer

examples, the invMQ is always slightly better than PHS. None of the suggested methods can exactly represent f_5 . Even though the extension reduces the error, it cannot reach zero because the polynomial extension has lower degree than the function to be interpolated.

The last function f_6 includes a jump induced by the hyperbolic tangent function. This jump may also occur in the FCM analysis because the displacement gradient is discontinuous at the cell boundary. Again, the extension reduces the error, but the error in the regime close to the jump cannot be reduced further. Nevertheless, the error is again zero if the majority of the nearest neighbors lie in the continuous part of the function, see also Fig. 5. Note that the results are similar if the exponent of the PHS is set to 5 or 7.

3.2 Cube

The first example is a simple cube, as shown in Fig. 6. The displacements are prescribed at all boundary faces. All surfaces are fixed in normal direction, and a displacement boundary condition is applied at the top surface. In doing so, we mimic a 1D compression test that leads to a constant deformation and displacement gradient across the entire structure.

Thereby, the displacement is applied in six equally spaced load increments until $\bar{d}_y = 12$ mm is reached. The physical domain is discretized by 64 finite cells with shape function ansatz order $p = 2$. In this example, we apply the invMQ RBF with and without polynomial extension. The number of

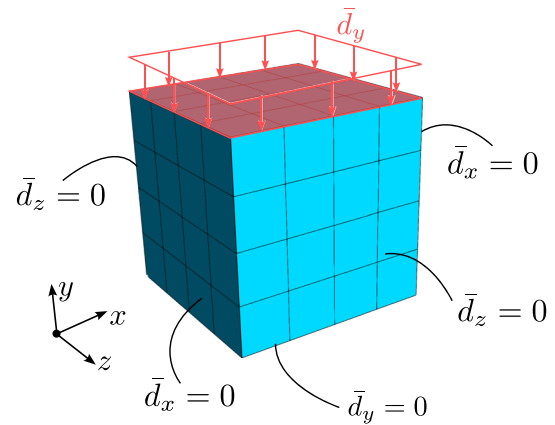


Fig. 6 Cube model discretized by 64 cells. Displacement in normal direction is prescribed at all faces

nearest neighbors is set to $N^s = 40$ for RBF interpolation and to $N^s = 4$ for the IDW interpolation.

The relative error in energy norm is thus computed by

$$\|e\|_{\mathcal{U}} = \sqrt{\frac{|\mathcal{U} - \mathcal{U}^*|}{\mathcal{U}}} \times 100\% \quad (28)$$

where \mathcal{U} and \mathcal{U}^* are the strain energies of the system before and after remeshing, respectively. The error in the reaction force is computed by

$$e_{F_r} = \left| \frac{F_r - F_r^*}{F_r} \right| \times 100\% \quad (29)$$

with F_r and F_r^* being the reaction force before and after remeshing. The results of this study are summarized in Table 3. All methods except the RBF data transfer have zero error in energy norm as well as in the reaction force. Therefore, the methods work as expected, since the polynomial extension should be able to capture the constant distribution. Similarly, the IDW and the L_2 -projection-based data transfer are able to represent a constant distribution, thus showing zero error for the present example.

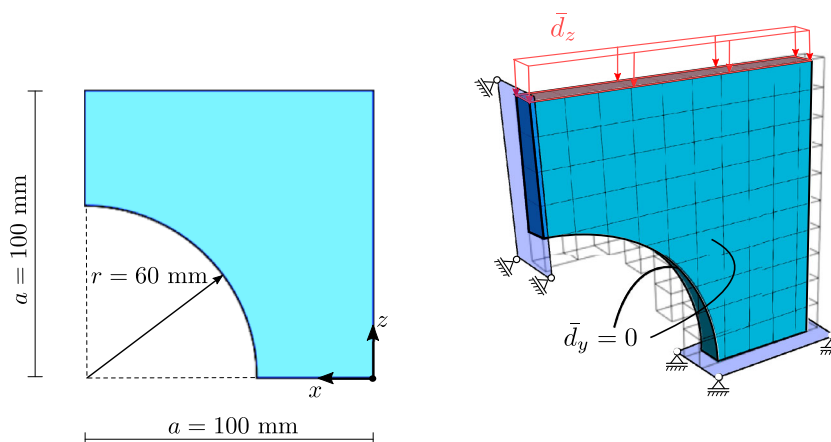
3.3 Plate with a cylindrical hole

The second example is shown in Fig. 7. The domain is discretized using 79 cells, and the zero Dirichlet boundary

Table 3 Summary of the error for the cube example under uni-axial displacement

Data transfer	Error energy norm $\ e\ _{\mathcal{U}}$ [%]	Error reaction force e_{F_r} [%]
IDW	0.0	0.00
RBF	5.3	0.14
RBF-P0	0.0	0.00
RBF-P1	0.0	0.00
L_2 -Projection	0.0	0.00

Fig. 7 Model and boundary conditions of the plate with cylindrical hole



conditions are defined as shown in the Figure. Additionally, the model is considered as plane strain by fixing the degrees of freedom (DOF) in y -direction. The load is applied incrementally by $\Delta d_z = 1.5$ mm until $\bar{d}_z = 15$ mm is reached. After applying remeshing, an equilibrium step is performed following the data transfer to ensure that the structure is in equilibrium. In the equilibrium step, the structure is recomputed without introducing any additional loading. This serves to remove the error in equilibrium due to the data transfer. For the RBF data transfer, the invMQ RBF is applied in all cases.

3.3.1 Preliminary investigations

To find appropriate settings for the data transfer, we first investigate the influence of the tuning parameters for the point-based interpolation methods. First, we investigate the influence of the number of nearest neighbors N^s on the error. Secondly, we fix N^s while varying N^r . In both models, the shape function ansatz order is fixed to $p = 3$. In all cases, we

measure the relative error in energy norm between the results before and after remeshing.

The investigations are performed for both quadrature methods—the octree and the NNMF. This is of interest because the number and positions of the integration points in the broken cells are completely different. To this end, it is not clear how the parameters and accuracy change between the two methods.

To begin with, we fix the number of neighbors N^s used to compute the scaling \bar{r} to $N^r = 3$ and vary the number of nearest neighbors in the range $N^s = \{5, 10, \dots, 50\}$. Thereby, we compute the relative error in energy norm between the results before and after remeshing to capture the error introduced by the remeshing process, see Eq. (28). The results of the study are shown in Fig. 8a. As can be seen, the extension with a constant (RBF + P0) reduces the error for both the octree and the NNMF integration. Additionally, the error remains constant for a wide range of N^s compared to the default RBF without extension. If the octree integration scheme is applied, the first order polynomial extension (RBF + P1) increases the error. This is not observed if the NNMF quadrature is used.

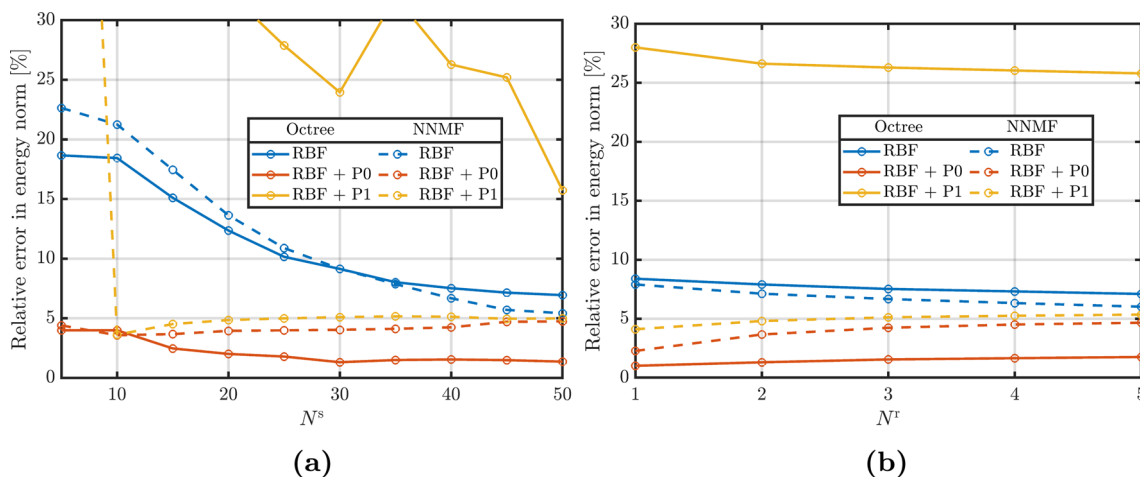
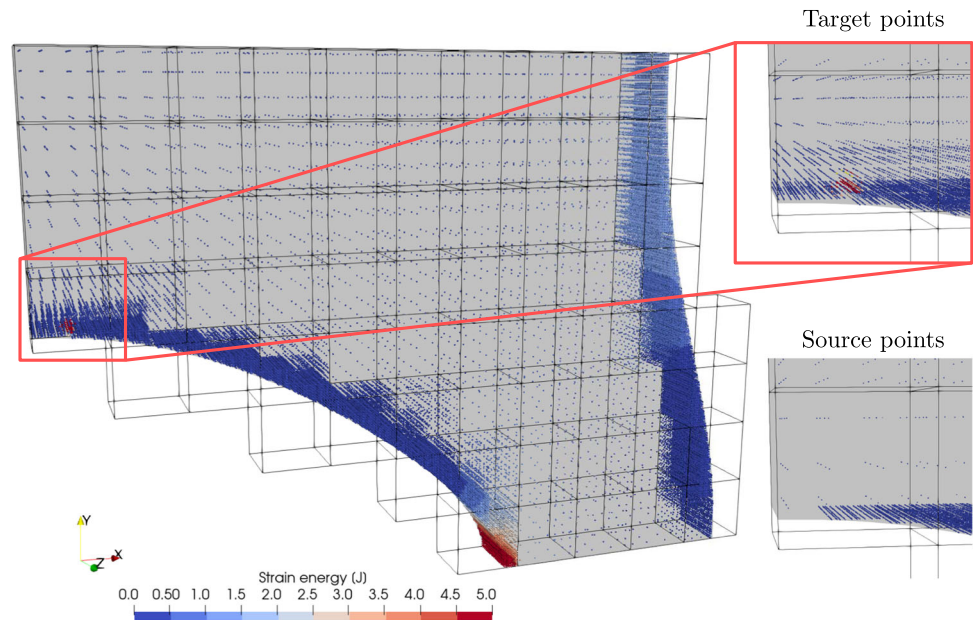


Fig. 8 **a** Error in energy norm after remeshing source points N^s and **b** when varying the number of neighbors for scaling N^r

Fig. 9 Strain energy evaluated at the integration points after remeshing. In the highlighted area, artificial energy was introduced for the data transfer with RBF + P1. The source points for that region can be seen on the lower right. In the critical regime, the source points form a plane where the y -coordinates are almost constant. This leads to an ill-conditioned system of equations, leading to increased error during the interpolation



This indicates that the error may be introduced by the location of the source points. As shown in Fig. 9, the octree quadrature refines towards the edge, leading to a high number of points close to the boundary. The problem is that some target points only have neighbors that are nearly in-plane. Therefore, the linear term in y -direction is almost constant—which leads to ill-conditioning of the system, and, hence, has a negative impact on the accuracy of the interpolation. Next, we investigate the influence of the number of nearest neighbors N^r chosen for scaling in order to compute the scaling radius \bar{r} . This time, N^s is fixed and N^r varies between one and five.

Again, Fig. 8b shows the results for both integration schemes. For both quadrature methods, the error varies only slightly when N^r changes. Adding the constant polynomial extension leads to a reduction in error. For all methods, it can be seen that the simulation is less sensitive to the choice of N^r than to the choice of the number of nearest neighbors N^s .

From this investigation, we see that adding the constant polynomial extension helps to improve the accuracy of the data transfer. Additionally, the constant extension reduces the sensitivity to the number of nearest neighbors N^s . However, adding the linear extension can reduce the accuracy when it is applied in combination with the octree quadrature.

3.3.2 Data transfer comparison

Now that we have a parameter set that gives good results for the point-based data transfer, we will further investigate how well the methods perform when the shape function ansatz

Table 4 Overview of the degrees of freedom and source points for different shape function ansatz orders p

Ansatz order p	DOF	Source points	
		Octree	NNMF
1	600	7,376	998
2	1,968	30,504	3,696
3	3,336	68,832	9,143
4	5,712	130,956	17,562
5	9,096	244,278	29,668

order varies from $p = 1$ to $p = 5$ and, hence, the number of available source points for the interpolation is changed. In order to capture the influence of the integration method, we apply the octree and the NNMF integration method. The number of source points given for each shape function ansatz order and the number of degrees of freedom in the initial mesh are shown in Table 4.

The relative error in energy norm as well as the relative error in the reaction forces are shown in Figs. 10 and 11, respectively. First, we consider the relative error in energy norm shown in Fig. 10. Note that all methods introduce some error in the energy norm. All methods except for the RBF + P1 produce errors around 5%. For both quadrature schemes, the constant extension P0 reduces the error compared to the pure RBF data transfer. The L_2 -projection falls between the two of them, while the IDW sometimes performs better and sometimes worse than the other methods.

Next, Fig. 11 shows the error in the reaction forces induced during the remeshing. The error is significantly reduced when

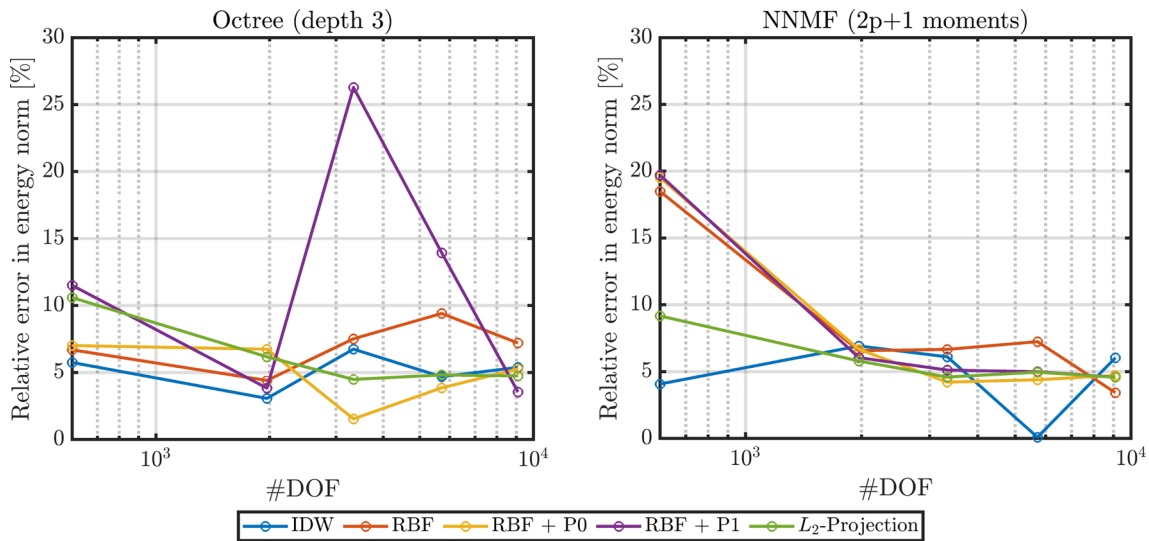


Fig. 10 Relative error in energy norm induced by the remeshing

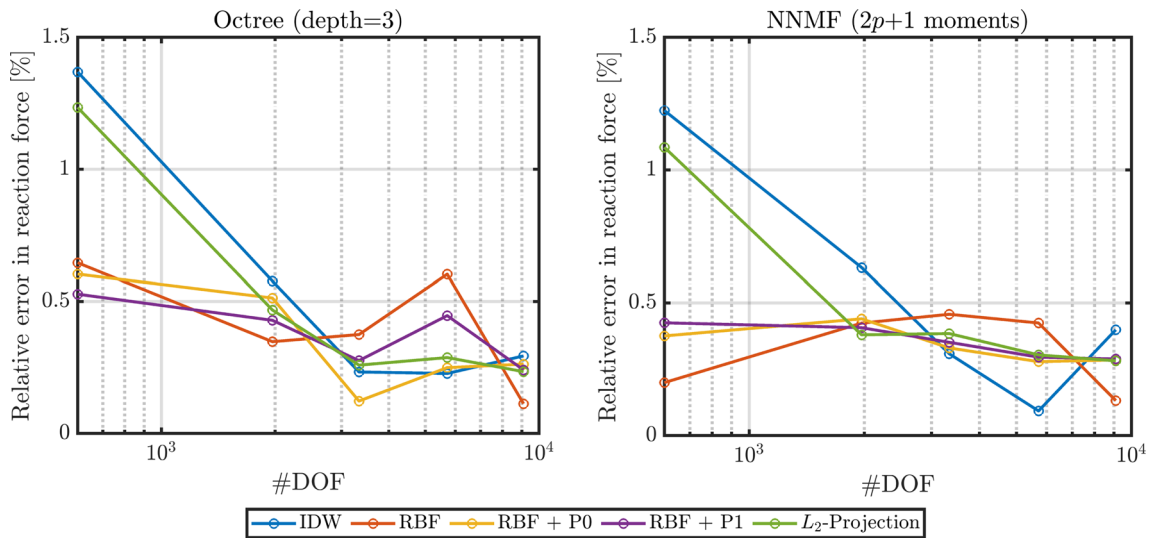


Fig. 11 Relative error in the reaction forces induced by the remeshing

$p > 1$ for all methods, and it stays below 0.5% in most cases. For the NNMF integration, the RBF + P0 and the RBF + P1 data transfer show a slightly lower error than the RBF without extension in most cases.

Lastly, Fig. 12 visualizes the error in the displacement field. The shown displacements are caused by the error introduced during the remeshing step. If the data transfer were exact, the displacement would have to be zero since equilibrium is fulfilled. As this is not the case, some displacements occur to balance out inaccuracies. The RBF and IDW tend to show oscillations in a wide part of the structure. On the other hand, the other methods show a smoother distribution where the hotspot at the circle radius is smoothed out. Interestingly, the displacements field of the L_2 -projection-based data transfer is quite similar to that of the RBF + P0 and the

RBF + P1 data transfers. This may indicate that the ability of representing constants similarly affects the result for both methods.

As explained in the previous parametric study, the error in energy norm is increased if RBF + P1 is applied. From the displacement fields, it can be seen that some displacements occur on the inner circle at the left edge. As already discussed, the reason for this behavior is influenced by the location of the source points.

3.4 Single pore of a foam

Lastly, the single pore of a foam shown in Fig. 13 is analyzed, and, again, the different data transfer methods are compared. The boundary is given as a triangulated surface generated

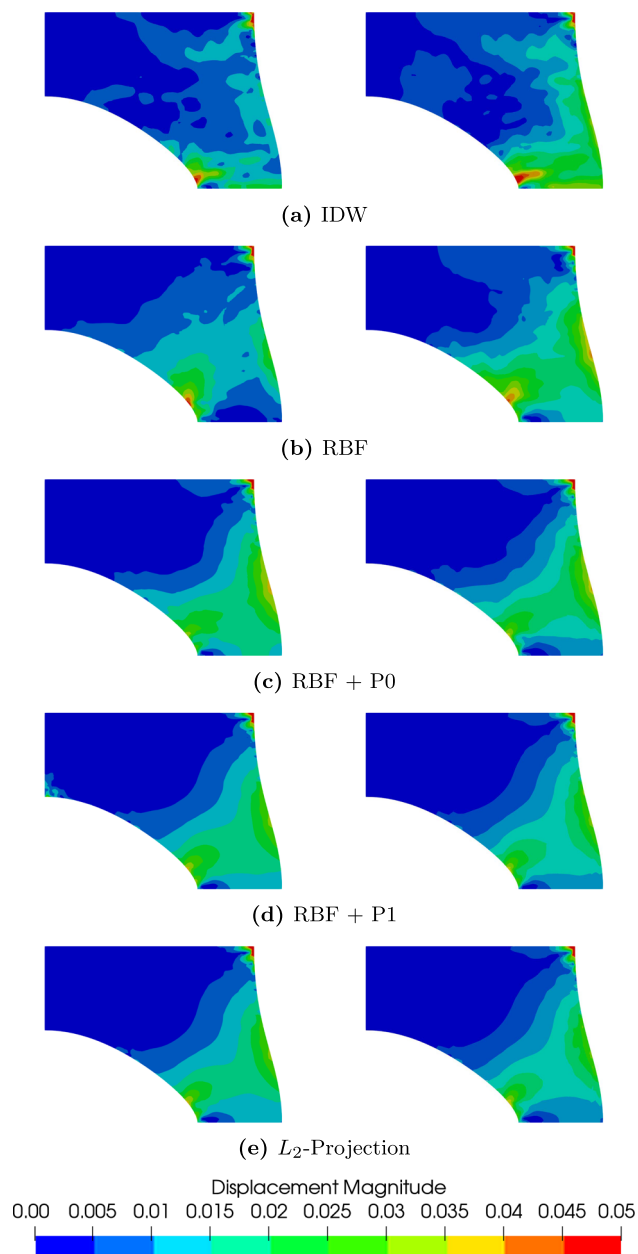


Fig. 12 Resulting residual displacement field after the equilibrium step for $p = 5$. On the left, the octree quadrature is applied for the numerical integration, while the right side shows the outcome of the NNMF quadrature

from a 3D scan. Here, we only apply the NNMF quadrature scheme for spatial integration. The domain is discretized by 438 finite cells, and the shape function ansatz order is varied between values $p = \{1, 2, \dots, 5\}$. The bottom of the foam is clamped, and the top surface is fixed in the in-plane direction. Then, a compressive displacement $\bar{d}_y = 2$ mm is incrementally applied where the displacement increment is set to $\bar{d}_y = 0.02$ mm.

The error in energy norm and the reaction forces are given in Fig. 14. The RBF-based data transfer leads to the smallest error in energy norm, and the IDW and the L_2 -projection-based interpolation show significantly larger errors. Most significantly, the L_2 -projection-based data transfer failed for $p = 2$ during the inverse mapping.

In contrast, the reaction forces are more accurate for the IDW interpolation, while the other methods show slightly larger but similar error. With increasing number of degrees of freedom, the error is reduced to less than 2.5% for all methods.

4 Conclusion

We considered different schemes for data transfer, to be performed during remeshing, to overcome problems with mesh distortion. From the investigations carried out in this contribution, we conclude that in case of pure RBF interpolation the displacement gradient should be considered for interpolation. At this point, however, we can already see the beneficial influence of the polynomial extension. When adding the constant part, the choice of the interpolated data (deformation gradient or displacement gradient) is arbitrary because the offset of 1 can be recovered by the constant part of the polynomial extension. Based on the comparison between the different approaches for data transfer (RBF, RBF + P_n , inverse distance weighting, least squares fit) for the considered problems of hyperelasticity, we recommend to apply the RBF + P_0 , i.e. radial basis functions with a constant polynomial extension. In most of the cases, this approach turned out to be the best strategy to transfer the history data during the remeshing procedure.

Fig. 13 Single foam pore with boundary conditions and the initial mesh

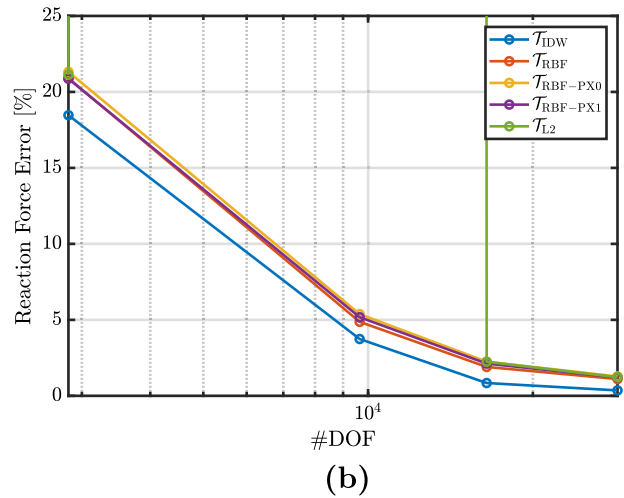
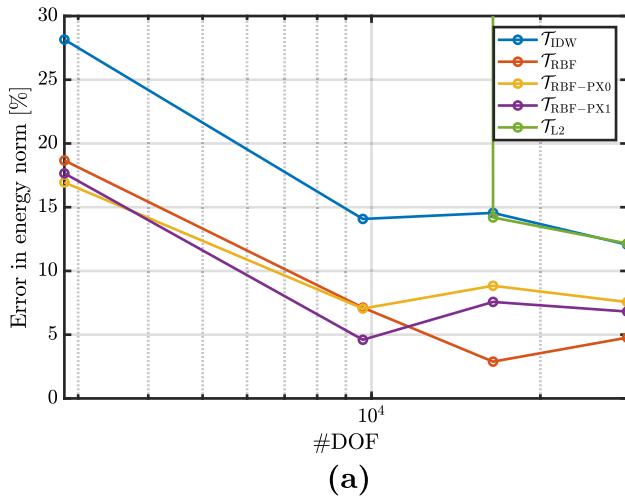
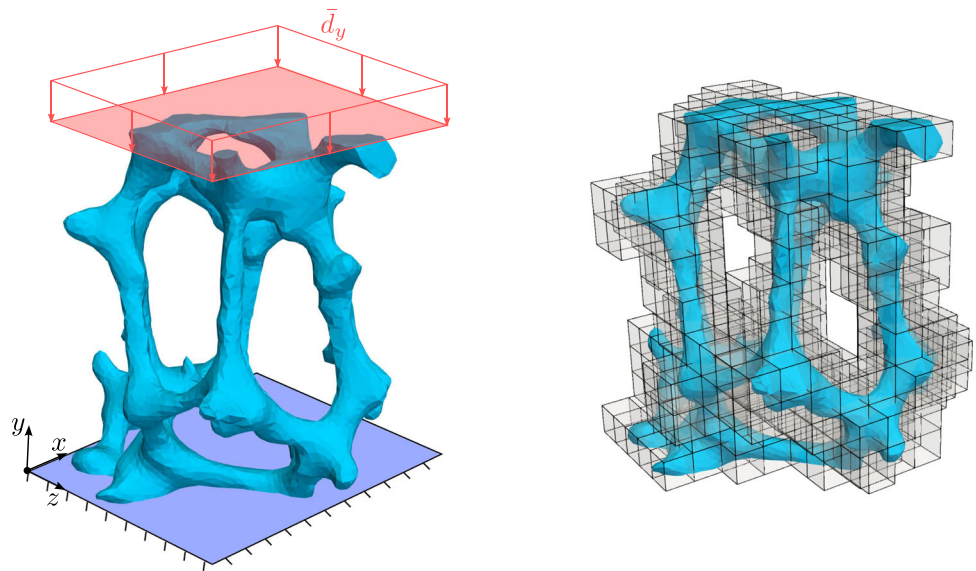


Fig. 14 **a** Error in energy norm and **b** error in the reaction force after remeshing for the single pore of foam example for different shape function ansatz orders

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.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data availability Data generated during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors have no financial or proprietary interests in any material discussed in this article.

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 (2007) Finite cell method - h- and p-extension for embedded domain problems in solid mechanics. *Comput Mech* 41:121–133
2. Düster A, Parvzian J, Yang Z, Rank E (2008) The finite cell method for three-dimensional problems of solid mechanics. *Comput Methods Appl Mech Eng* 197:3768–3782

3. Burman E, Hansbo P (2010) Fictitious domain finite element methods using cut elements: I. A stabilized Lagrange multiplier method. *Comput Methods Appl Mech Eng* 199(41–44):2680–2686
4. Burman E, Hansbo P (2012) Fictitious domain finite element methods using cut elements: II. A stabilized Nitsche method. *Appl Numer Math* 62(4):328–341. <https://doi.org/10.1016/j.apnum.2011.01.008>
5. Burman E, Hansbo P, Larson MG (2015) A stabilized cut finite element method for partial differential equations of surfaces: the Laplace-Beltrami operator. *Comput Methods Appl Mech Eng* 285:188–207
6. Elfverson D, Larson MG, Larsson K (2018) CutIGA with basis function removal. *Adv Model Simul Eng Sci* 5(1):6. <https://doi.org/10.1186/s40323-018-0099-2>
7. Zander N, Kollmannsberger S, Ruess M, Yosibash Z, Rank E (2012) The finite cell method for linear thermoelasticity. *Comput Math Appl* 64(11):3527–3541. <https://doi.org/10.1016/j.camwa.2012.09.002>
8. Schillinger D, Ruess M, Zander N, Bazilevs Y, Düster A, Rank E (2012) Small and large deformation analysis with the p- and B-spline versions of the finite cell method. *Comput Mech* 50:445–478. <https://doi.org/10.1007/s00466-012-0684-z>
9. Garhuom W, Hubrich S, Radtke L, Düster A (2020) A remeshing strategy for large deformations in the finite cell method. *Comput Math Appl* 80:2379–2398. <https://doi.org/10.1016/j.camwa.2020.03.020>
10. Abedian A, Parvizian J, Düster A, Rank E (2013) The finite cell method for the J_2 flow theory of plasticity. *Finite Elem Anal Des* 69:37–47
11. Taghipour A, Parvizian J, Heinze S, Düster A (2018) The finite cell method for nearly incompressible finite strain plasticity problems with complex geometries. *Comput Math Appl* 75:3298–3316. <https://doi.org/10.1016/j.camwa.2018.01.048>
12. Hubrich S, Düster A (2019) Numerical integration for nonlinear problems of the finite cell method using an adaptive scheme based on moment fitting. *Comput Math Appl* 77:1983–1997. <https://doi.org/10.1016/j.camwa.2018.11.030>
13. Ruess M, Tal D, Trabelsi N, Yosibash Z, Rank E (2012) The finite cell method for bone simulations: verification and validation. *Biomech Model Mechanobiol* 11:425–437
14. Yang Z, Kollmannsberger S, Düster A, Ruess M, Garcia E, Burgkart R, Rank E (2012) Non-standard bone simulation: interactive numerical analysis by computational steering. *Comput Vis Sci* 14(5):207–216. <https://doi.org/10.1007/s00791-012-0175-y>
15. Verhoosel CV, Zwieten GJ, Rietbergen B, Borst R (2015) 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
16. Hug L, Dahan G, Kollmannsberger S, Rank E, Yosibash Z (2022) Predicting fracture in the proximal humerus using phase field models. *J Mech Behav Biomed Mater* 134:105415. <https://doi.org/10.1016/j.jmbbm.2022.105415>
17. Radtke L, Marter P, Duvigneau F, Eisenträger S, Juhre D, Düster A (2024) Vibroacoustic simulations of acoustic damping materials using a fictitious domain approach. *J Sound Vibr* 568:118058. <https://doi.org/10.1016/j.jsv.2023.118058>
18. Elhaddad M, Zander N, Kollmannsberger S, Shadavakhsh A, Nübel V, Rank E (2015) Finite cell method: high-order structural dynamics for complex geometries. *Int J Struct Stab Dyn* 15(7):1540018. <https://doi.org/10.1142/S0219455415400180>
19. Duczek S, Joulaian M, Düster A, Gabbert U (2014) Numerical analysis of Lamb waves using the finite and spectral cell method. *Int J Numer Methods Eng* 99:26–53. <https://doi.org/10.1002/nme.4663>
20. Dijk NP, Maute K, Langelaar M, Keulen F (2013) Level-set methods for structural topology optimization: a review. *Struct Multidiscip Optim* 48(3):437–472. <https://doi.org/10.1007/s00158-013-0912-y>
21. Azari Nejat A, Held A, Seifried R (2023) A fully coupled level set-based topology optimization of flexible components in multi-body systems. *Struct Multidiscip Optim*. <https://doi.org/10.1007/s00158-023-03603-y>
22. Fornberg B, Flyer N (2015) *A Primer on Radial Basis Functions with Applications to the Geosciences*. Society for Industrial and Applied Mathematics, SIAM, Philadelphia, PA. <https://doi.org/10.1137/1.9781611974041>
23. Tóth B, Düster A (2022) h -Adaptive radial basis function finite difference method for linear elasticity problems. *Comput Mech*. <https://doi.org/10.1007/s00466-022-02249-9>
24. Bonet J, Wood RD (1997) *Nonlinear continuum mechanics for finite element analysis*. Cambridge University Press, New York
25. Wriggers P (2008) *Nonlinear finite element methods*. Springer, Berlin
26. 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. *Comput Mech* 60:863–881. <https://doi.org/10.1007/s00466-017-1441-0>
27. Petö M, Garhuom W, Duvigneau F, Eisenträger S, Düster A, Juhre D (2022) Octree-based integration scheme with merged sub-cells for the finite cell method: application to non-linear problems in 3D. *Comput Methods Appl Mech Eng* 401:115565. <https://doi.org/10.1016/j.cma.2022.115565>
28. Müller B, Kummer F, Oberlack M (2013) Highly accurate surface and volume integration on implicit domains by means of moment-fitting. *Int J Numer Methods Eng* 96:512–528. <https://doi.org/10.1002/nme.4569>
29. Joulaian M, Hubrich S, Düster A (2016) Numerical integration of discontinuities on arbitrary domains based on moment fitting. *Comput Mech* 57:979–999. <https://doi.org/10.1007/s00466-016-1273-3>
30. Hubrich S, Düster A (2018) Numerical integration for nonlinear problems of the finite cell method using an adaptive scheme based on moment fitting. *Comput Math Appl*. <https://doi.org/10.1016/j.camwa.2018.11.030>
31. Legrain G (2021) Non-negative moment fitting quadrature rules for fictitious domain methods. *Comput Math Appl* 99:270–291. <https://doi.org/10.1016/j.camwa.2021.07.019>
32. Garhuom W, Düster A (2022) Non-negative moment fitting quadrature for cut finite elements and cells undergoing large deformations. *Comput Mech* 70:1059–1081. <https://doi.org/10.1007/s00466-022-02203-9>
33. Wolfram Research, Inc.: *Mathematica*, Version 14.0. Champaign, IL, 2024. <https://www.wolfram.com/mathematica>
34. Korelc J, Wriggers P (2016) *Automation of finite element methods*. Springer, Berlin. <https://doi.org/10.1007/978-3-319-39005-5>
35. Carr JC, Beatson RK, Cherrie JB, Mitchell TJ, Fright WR, McCallum BC, Evans TR (2001) Reconstruction and representation of 3d objects with radial basis functions. In: *Proceedings of the 28th Annual conference on computer graphics and interactive techniques*. SIGGRAPH '01, pp. 67–76. Association for computing machinery, New York, NY, USA. <https://doi.org/10.1145/383259.383266>
36. Hartmann S, Müller-Lohse L, Tröger J-A (2021) Full-field strain determination for additively manufactured parts using radial basis functions. *Appl Sci*. <https://doi.org/10.3390/app112311434>
37. Shephard MS, Dey S, Flaherty JE (1997) A straightforward structure to construct shape functions for variable p-order meshes. *Comput Methods Appl Mech Eng* 147:209–233
38. Garhuom W, Hubrich S, Radtke L, Düster A (2021) A remeshing approach for the finite cell method applied to problems with large deformations. *Proc Appl Math Mech* 21:202100047. <https://doi.org/10.1002/pamm.202100047>

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