Yannick F. Fangye · Niklas Miska · Daniel Balzani

Automated simulation of voxel-based microstructures based on enhanced finite cell approach

Received: 2 January 2020 / Accepted: 7 June 2020 / Published online: 1 July 2020
© The Author(s) 2020

Abstract A new and efficient method is proposed for the decomposition of finite elements into finite subcells, which are used to obtain an integration scheme allowing to analyse complex microstructure morphologies in regular finite element discretizations. Since the geometry data of reconstructed microstructures are often given as voxel data, it is reasonable to exploit the special properties of the given data when constructing the subcells, i.e. the perpendicularly cornered shape of the constituent interfaces at the microscale. Thus, in order to obtain a more efficient integration scheme, the proposed method aims to construct a significantly reduced number of subcells by aggregating as much voxels as possible to larger cuboids. The resulting methods are analysed and compared with the conventional Octree algorithm. Eventually, the proposed optimal decomposition method is used for a virtual tension test on a reconstructed three-dimensional microstructure of a dual-phase steel, which is afterwards compared to real experimental data.

Keywords Dual-phase steel · Voxel-based microstructures · Finite cells · Multiscale simulation

1 Introduction

Dual-phase steels (DP steels) belong to the important class of advanced high-strength steels (AHSS), which combine attractive properties such as high tensile strength and formability. DP steels are therefore suitable for applications in many engineering fields, e.g. in the automotive industry for an improvement in the crash safety and stability whilst the overall weight is reduced. The advantageous material properties are directly dependent on the interactions of the individual microscopic constituents, which arise from a specific production process, see, e.g. Al-Abbasi and Nemes [1], Pierman et al. [25], Kim and Lee [17] and Davies [6]. The microstructure of DP steel consists of hard martensitic inclusions embedded in a ductile ferritic matrix. A crucial step in the production of DP steels is the rapid cooling of hot-rolled steel that induces the transformation from austenite to martensite. These transformation is responsible for graded material properties in the ferrite matrix as found in Brands et al. [4].

Due to the complexity of their microstructure, the modelling and simulation of structures made of DP steels is not straightforward. Since a phenomenological model to appropriately describe the effective material response is difficult to design, homogenization approaches taking the morphology of the microstructure into account are more suitable, see, e.g. Golanski et al. [14], Moulinec and Suquet [23] or Miehe et al. [21]. A direct two-scale homogenization scheme is the FE^2 method, where a microscopic boundary value problem is solved at each integration point of the macroscopic finite elements, cf., e.g. Smit et al. [33], Feyel [10, 11], Feyel and Chaboche [12] and Schröder [30]. An important aspect of such multiscale approaches is the choice of a suitable representative volume element (RVE) which characterizes the microstructural composition of

---

Y. F. Fangye · N. Miska · D. Balzani (✉)
Chair of Continuum Mechanics, Ruhr-Universität Bochum, Universitätsstraße 150, 44801 Bochum, Germany
E-mail: daniel.balzani@ruhr-uni-bochum.de

the material. One possibility is the direct three-dimensional reconstruction of the microstructure morphology by means of a combination of the electron backscatter diffraction (EBSD) and the focused ion beam (FIB), c.f. Zaefferer et al. [35], Brands et al. [4] or Calcagnotto et al. [5]. However, the complexity and high level of detail of such RVEs require rather small finite elements and thus a large number of degrees of freedom. The arising computational effort can be reduced by using more efficient computational methods such as the fast Fourier transform or the spectral element method, applied, respectively, in Lee et al. [20] and Eisenlohr et al. [9], to predict the macroscopic material response. These methods may, however, be restricted (i) with view to the microstructure discretization and (ii) the material models applied at the microscale. Other approaches focus on the construction of geometrical models with reduced complexity, whilst the overall material response remains similar to the one of the real microstructure. One of these approaches is the construction of a three-dimensional statistically similar RVE (SSRVE) as presented in Balzani et al. [2] and Scheunemann et al. [27]. No matter which computational approach is used at the microscale, the discretization of the microstructure remains one of the major challenges in finite element analysis. In particular for highly complex geometries such as DP steel microstructures, the generation of a suitable finite element mesh may be difficult, time-consuming and not available to automation.

A possible way out is the finite cell method (FCM), which was introduced by Parvizia et al. [24] and Düster et al. [8] for two- and three-dimensional problems in solid mechanics, respectively. The FCM combines the fictitious domain approach with higher-order finite elements. Details concerning the fictitious domain approach can be found in Bishop [3], Ramière et al. [26], Del pino and Pironneau [7] or Glowinski and Kuznetsov [13]. Similar to methods like the extended finite element method (XFEM) or the generalized finite element method (GFEM), the approximation of the primary variables is separated from the approximation of the geometry. This results in a significant simplification of the meshing process, since a Cartesian grid can be used as mesh and the complexity of the geometry is only captured within the integration of the stiffness matrices of the elements. Therefore, the expensive generation of an interface-conforming mesh is not necessary. On the other hand, discontinuities occur within elements which are intersected by the internal constituent interfaces. Since the Gaussian quadrature, which is usually used in the finite element method, does not perform well for non-smooth functions, an adapted integration scheme is required. For this purpose, the intersected elements are decomposed into subcells for integration purposes only. In Düster et al. [8] and Schillinger and Ruess [28], an Octree algorithm is used to construct the subcells, at which the element of interest is recursively subdivided into eight equal-sized regions. Specifically for linear problems and voxel-based data, Yang et al. [34] propose a more efficient approach by making use of preintegration. However, octrees require a comparatively large number of subcells for the sake of accuracy, which implies a high number of integration points and therefore considerable computational effort for the integration. To overcome this problem, a smart Octree method is introduced in Kudela et al. [19] for smooth structures. In Joulaian et al. [16], the moment fitting approach is used, where an individual quadrature rule is used for every broken element. This method is extended in Hubrich et al. [15] by an optimization strategy for the improvement in the quadrature rule.

In this paper, we propose new methods for the subcell decomposition of finite elements to circumvent the mentioned issues of the classical Octree algorithm. We focus on microstructure data, which is given as voxel data, and propose to construct subcells which exploit the nature of such data. The material boundaries are thereby exactly captured with a lower number of subcells, thereby enabling reduced computing times and less memory requirements. The proposed method will be compared with the classical Octree method in terms of efficiency and accuracy in finite element simulations. Furthermore, it will be applied to the simulation of DP steel microstructures. Since our algorithms mainly aim to reduce the number of subcells required for integration, significant savings in computational costs can only be expected whenever the integration domains can suitably be decomposed in the sense of finite cells. This is particularly the case for microstructures with distinct individual phases and/or cavities or pores. If a material's microstructure has graded properties, as, for instance, biological tissues, which may be suitably captured in terms of different properties for each single voxel, merging of finite cells with same properties will hardly be possible. Then, our algorithms will not directly enable a significant reduction in computational effort. However, using a representation where each voxel is integrated by a finite subcell corresponds to the assumption of constant properties within each voxel. This may be considered fine as no additional data in terms of measurements is usually available anyways. But stepwise constant properties may not be considered realistic in microstructures with graded properties, especially in biological tissues. There, it may be more realistic to assume some sort of smoothed interpolation of properties and in fact, a representation using finite cells may not be favoured. More advantageous could be to simply assign different properties to each Gauss point (which is often referred to as Gauss-point method). Thereby, the fields of properties are intrinsically assumed to be continuous and more or less automatically

smoothed by the shape functions. This is actually the approach we follow in our analysis of the real DP steel microstructures, where the graded properties of at least the matrix phase are included this way. Note that specifically for linear mechanical problems, the computational effort can furthermore be significantly reduced by means of precomputing of the cell matrices and vectors, see Yang et al. [34].

The paper is organized as follows: for consistency, the FCM is briefly recapitulated in Sect. 2 and the proposed approach for a more efficient subcell decomposition and several variations therefrom are presented in Sect. 3. Afterwards, the performance of the proposed approach is analysed by means of numerical examples in Sect. 4, and a conclusion including an outlook on future work is given in Sect. 5.

2 The finite cell method

This section briefly recapitulates the basics of the finite cell method (FCM) and the classical subcell decomposition method. This is important as the major novelty presented in this paper is an enhanced subcell decomposition to enable more efficient FC analysis. The FCM is an extension on top of the finite element method (FEM) and was introduced to facilitate the expensive construction of geometry-conforming meshes for the FEM, c.f. Parvizian et al. [24], Düster et al. [8]. Although sophisticated algorithms for the automated mesh generation are available, c.f., e.g. Schneider et al. [29], they may fail for arbitrary complex geometries and hence are not sufficient in the general case. Thus, a core idea of the FCM is to separate the approximation of the primary variables, i.e. the nodal displacements, from the geometry approximation in the sense of integration. For this purpose, the domain of interest is embedded in a larger fictitious domain of simpler shape, e.g. in a cuboid, which allows the use of structured meshes of cuboid-shaped hexahedral finite elements. Since the focus of this work is on the computational homogenization of representative volume elements (RVEs), which are typically of a cuboidal shape already, the construction of a fictitious domain can be omitted and the chosen RVE is directly discretized by a structured grid. However, both approaches yield finite elements, which cut either the border of the original domain in the case of the fictitious domain approach or the interfaces of different microscopic constituents within the RVE. For the integration required for computing the element stiffness matrix $\mathbf{k}^e$, this implies that the integrand becomes discontinuous over the domain of the element due to the variation in the material's properties. In the case of classical finite elements, the stiffness matrix is computed as $\mathbf{k}^e = \int_{\Omega_e} \mathbf{B}^T \mathbb{C} \mathbf{B} \, dV$, in which $\mathbf{B}$ denotes the standard B-matrix, $\mathbb{C}$ the material tangent (derivative of stress tensor with respect to strain tensor) in Voigt matrix notation; Ω_e is the domain of the element. In case of discontinuities in the integrand, the classical Gaussian integration schemes cannot directly be used for the numerical evaluations of these integrals. The fundamental idea of the finite cell method is thus to decompose the element domain into subdomains in such a way that no material discontinuities appear in the subdomains. This is achieved by splitting the element integrals into multiple integrals over the subdomains, wherein the function to be integrated is continuous. As an example, consider an element with two different materials, in which material 1 occupies the subdomain Ω_e^1 and a second material the subdomain Ω_e^2 . If $\Omega_e^1 \cup \Omega_e^2 = \Omega_e$ and $\Omega_e^1 \cap \Omega_e^2 = \emptyset$, the computation of the element stiffness matrix reads

$$\mathbf{k}^e = \int_{\Omega_e^1} \mathbf{B}^T \mathbb{C}_1 \mathbf{B} \, dV_1 + \int_{\Omega_e^2} \mathbf{B}^T \mathbb{C}_2 \mathbf{B} \, dV_2, \quad (1)$$

wherein $\mathbb{C}_1$ and $\mathbb{C}_2$ denote the material tangent of material 1 and material 2, respectively. If the subdomains are in turn chosen to be composed of multiple subcells which have a simple and regular, e.g. hexahedral, shape, existing Gaussian integration schemes can be used to evaluate the individual integrals over the subcells and thus the integrals over the subdomains in Eq. (1). Certainly, the decomposition of the element's domain is not restricted to only two subdomains. In principle, there can be as many subdomains as needed to ensure continuity in the individual integrands. To distinguish finite elements, whose domain is decomposed in subdomains, from classical finite elements, the authors of the original proposal coined the term “finite cell” for this new type of elements. If Eq. (1) is expanded for an arbitrary number of subcells n_{sc} and with the use of the Gaussian integration scheme, the computation of the stiffness matrix for the finite cell fc is written as

$$\mathbf{k}^{fc} = \sum_{sc=1}^{n_{sc}} \sum_{l=1}^{l_{int}} (\mathbf{B}^T \mathbb{C}_{sc} \mathbf{B}) |_{\xi_l} \det \mathbf{J}_{sc} w_{sc,l} \det \mathbf{J}_{fc}, \quad (2)$$

wherein l_{int} denotes the number of Gauss points and w_l denotes the respective weight of integration point l at location ξ_l in the isoparametric space. $\mathbf{J}_{fc}$ represents the Jacobian matrix to map the actual finite cell fc to

Fig. 1 Illustration of **a** the considered geometry of a circular inclusion in a matrix, **b** example of a structured grid used as finite element mesh for this domain and **c** decomposition of the top left finite cell from subfigure **b** based on the application of the quadtree algorithm with a recursion-depth of 4

the isoparametric space and the additional Jacobian matrix J_{sc} which maps the domain of subcell sc within the isoparametric space of fc to its own isoparametric space to utilize the usual integration scheme.

Once the adapted integration scheme using subcells is available, the key question is how to decompose an arbitrary finite cell into hexahedral subcells, especially if the geometry exhibits curved boundaries. For three-dimensional problems, a popular method is the Octree algorithm, which recursively divides a finite cell into eight equal-sized subcells. The generated subcells are checked for geometry discontinuities, such that non-uniform cells can be split again, until a stopping criterion is reached. Such criteria can be, for example, the maximum recursion depth, in the following referred to as level of the Octree, or a threshold on the volume fraction of a second material within a cell, which avoids the decomposition for very small portions of a second material. For two-dimensional problems, the analogous recursion strategy is referred to as quadtree algorithm. A simple example for the principle procedure is shown in Fig. 1. In Fig. 1a, a quadrilateral domain with a circular inclusion is depicted, clearly it is difficult to decompose this domain in rectangular subdomains. Figure 1b represents the regular grid used as FE mesh for the primary variables. Its top-left finite cell is shown in detail in Fig. 1c. There it can be seen, how this particular cell is divided four times by the quadtree algorithm to approximate the curved geometry by quadrilateral subcells. Apparently, the geometry approximation will never be perfect in this case, but with a suitably chosen recursion depth an adequate replication of the geometry can be reached. However, since the primary variable is solved only for the nodes of the regular grid, potential jumps in the displacement field and derived quantities at material boundaries are not resolved as it would be the case for conforming meshes. Thus, usually a straightforward solution is to employ higher-order shape functions in a finite cell to increase the interpolation accuracy of the quantities of interest. In this work, only Lagrangian polynomials are used for this purpose, but other kinds of shape functions known from classical finite elements are as well possible in principle. Still, the major advantage of the finite cell method, the regularity of the FE mesh, allows for efficient computations as less degrees of freedom can be used in total, which will be shown later. Further advantage of the regularity is that automated calculations can be easily performed for sets of RVEs with different microstructure morphologies. This may be important for the quantification of uncertainties which are related to microstructure variation, see [22]. Especially, in the context of microstructure simulations as part of computational homogenization, the regular grids allow for a direct application of periodic boundary conditions. In case of complex domains embedded in a fictitious domain, the boundary conditions can, however, only be applied to the fictitious domain and not to the domain of interest, which requires additional processing of these boundary conditions. Since such applications are out of the scope in the context of this paper, the interested reader is referred to more extensive explanations of the finite cell method, which can be found in, e.g. Schillinger and Ruess [28] or Düster et al. [8].

3 Enhanced subcell decomposition approaches

In the context of the FCM, the decomposition into subcells is mostly based on the Octree algorithm. And certainly, given an arbitrary domain, e.g. parameterized by free-form surfaces, the Octree algorithm is an excellent method to subdivide the given space into subcells as it recursively exploits the domain and refines at locations of geometry discontinuity. However, if the geometry is known to be defined in terms of perpendicular planes, i.e. in terms of voxels or pixels, it may be beneficial to directly aggregate certain voxels resp. pixels

Fig. 2 Subcell decomposition of a finite cell, **a** base representation of the geometry with 5 by 5 pixels, **b** subcell decomposition by means of quadtree until level 4 with intersected subcells and **c** optimal subcell decomposition

to larger cuboids, so that less subcells compared to the application of the Octree are produced. A smaller number of subcells is in principle favourable, as less iterations, and hence, integration point evaluations are required to assemble the element matrices, and thus, the computational effort required to solve the boundary value problem of interest is reduced. To illustrate the aggregation of voxels/pixels, a simple two-dimensional domain of five by five pixels and two different materials as depicted in Fig. 2a is supposed to be decomposed in subcells. In Fig. 2b, the geometry is decomposed by means of a quadtree of level four. Besides many small subcells, it is noticeable that the fourth level is not sufficient to capture the material bounds exactly as many subcells with in principle two materials are visible. As this is not admissible in context of the FCM theory, the corresponding subcells are either assigned with the material having the larger volume fraction within the cell or split by a further Octree iteration, which leads to even smaller and more subcells. Whilst the first option effectively shifts the material bounds and thus modifies the boundary value problem of interest, the latter option leads to undesired high computational demands. These issues could partially be dealt with by the application of a strategy similar to the smart octrees (or analogously smart quadtrees in 2D) introduced in Hubrich et al. [15] or Kudela et al. [19] for smooth structures. Motivated by these approaches, the nodes of the classical octree-based subcells may be relocated in order to directly capture the intersection topology.

Thus, we propose an optimal decomposition of the domain of finite cells as presented in Fig. 2c. Therein, the pixels of the same materials are clustered in such a way that the least number of subcells is created and the material boundaries are captured exactly.

3.1 Optimal Decomposition of finite cells

Although the introductory example was specifically posed such that the Octree algorithm appears unfavourable, it should have become clear that an optimal decomposition can be the more reasonable choice for voxel-based data. Whilst the simple example shown in Fig. 2 could be solved visually, an algorithmic treatment for arbitrary voxelized domains is necessary. The developed algorithm for the Optimal Decomposition (OD) is described in the following. Although the algorithm works for both two- and three-dimensional problems, the explanation will refer mainly to the two-dimensional case for simplicity. If the three-dimensional workflow requires a modified procedure at certain points, this will be explicitly annotated.

As the first step, the pixel located at the lower corner of the finite cell is identified, as depicted in Fig. 3a, and considered as prototype subcell with the dimension of one pixel. Starting therefrom, the neighbouring pixel in the first dimension is searched and checked for compatibility with the prototype subcell. In this context, compatibility implies firstly that both the prototype subcell and the pixel are supposed to exhibit the same material, otherwise an aggregation of both is undoubtedly not reasonable. Secondly, the size of the prototype subcell and the pixel in the remaining, i.e. not in the search-direction, dimensions has to be equal to ensure that the result of an union will be a cuboid and not a more complex shape. If these criteria are met, the prototype subcell is extended by the pixel as displayed in Fig. 3b, otherwise a new prototype subcell is created. This scheme is repeated along the first direction until the boundary of the finite cell is reached; in Fig. 3c the entire lowest row could be joined to one prototype subcell. Subsequently, the algorithm shifts by one pixel in the second dimension and starts the same procedure as before, c.f. Fig. 3d, which results again in an entire joint row, c.f. Fig. 3e in this example. The aggregation of pixels along the first dimension into prototypal subcells

Fig. 3 Principle algorithm of the Optimal Decomposition (OD) for one single finite cell containing 5 by 5 pixels; **a** a prototype subcell, **b** extension in the first search direction, **c** an entire joined row, **d** and **e** iterations on the next row, **f** the result of the extension in the first direction, **g** merge operation in the second direction and **h** final decomposition result. In **i** and **j**, the decomposition result for a permutation of the search direction is shown

is repeated row by row until the boundary of the finite cell is reached in the second dimension as well. As it is visible in the result of the example, see Fig. 3f, a material discontinuity was encountered in the middle row. Therefore, the associated subcell was terminated and a new prototype subcell was started at the material interface. Since a three-dimensional problem can be considered as a stack of two-dimensional layers, the described procedure can easily be repeated for each of the layers, until the boundary of the finite cell is reached in the third dimension as well. This scheme corresponds to the pseudo-code given in lines 3–12 in Algorithm 1.

Regardless of the dimension, once the iteration is accomplished, we end up with a set of subcells, which are tube-shaped to a certain extent since the pixels are merged to subcells only in the first direction. This changes in the next step of the algorithm, in which we iterate over the second direction to merge the subcells whenever possible. Analogous to the first iteration, the individual subcells are checked for compatibility, i.e. the material is checked as well as the dimension of the subcells in non-search directions. The first step of this iteration performed on the example is shown in Fig. 3g, where the two prototype subcells at the bottom are merged to a larger cuboid. The middle row cannot be merged, neither to the larger subcell at the bottom nor to the subcell above, since either the material or the size is incompatible. Thus, the algorithm jumps to the second-last subcell from the top and merges it with the top subcell as depicted in Fig. 3h. At this point, this is the end of the algorithm for the two-dimensional case and the supposed optimal decomposition is found. These steps are reflected in pseudo-code in lines 14–20 in Algorithm 1. For a three-dimensional problem, however, the latter steps are again repeated for each layer in the third direction, so that a set of subcells, which are mostly plate-shaped, originates. Additionally, an iteration over the third direction has to be performed for these three-dimensional problems to merge the plate-shaped subcells to larger cuboids. Following the same pattern, again the material as well as the size in the first and second direction are checked for compatibility, so that box-shaped subcells can be created. After this final iteration, the three-dimensional algorithm has reached its end and an optimal three-dimensional decomposition is found. This last iteration is represented by the pseudo-code in lines 22–28 in Algorithm 1.

So far, this algorithm iterates over the directions in ascending order, and indeed, this choice may not be optimal. Thus, the algorithm is repeated for all possible permutations of directions, resulting in two combinations in the two-dimensional and six possible combinations in the three-dimensional case. These permutations are not explicitly listed in Algorithm 1 in order to keep the listed algorithm simple. Exemplarily, the results of firstly iterating over the second dimension are depicted in Fig. 3i and the results of the subsequent subcell-merge step in the first dimension are shown in Fig. 3j. Evidently, different subcells, but the same number of subcells are created, such that the decomposed integration of the element matrices will yield the same level of efficiency. Nevertheless, for the general case we could not prove a certain order of directions to be favourable over others. Thus, the algorithm analyses each possible permutation of directions to obtain the least number of subcells which in our case is considered optimal. Note that the bounds of the finite cell do not need to match

Algorithm 1: Optimal Decomposition (OD)

```

1 def OptimalDecomposition (finite_cells, voxel_data) :
2   for fc in finite_cells:                                     # loop over all finite cells
3     subcells = []
4     for vz in z-direction:                                   # iteration to aggregate
5       for vy in y-direction:                                 # voxels in first direction
6         for vx in x-direction:
7           vox = voxel_data[vx,vy,vz]                         # get actual voxel
8           last_sc = subcells.Last()                           # get current subcell
9           if CheckCompatibility (vox, last_sc, 1) :
10            last_sc.ExtendBy(vox)                             # extend subcell by voxel
11           else:
12            subcells.Append(vox)                              # start new subcell with voxel
13
14   subcells.Sort()                                           # sort subcells for merge in 2nd direction
15   for i in subcells.length:                                  # loop over all prototypal subcells
16     sc1 = subcells[i]                                       # extract neighbouring subcells
17     sc2 = subcells[i-1]
18     if CheckCompatibility (sc1, sc2, 2) :
19       sc2.ExtendBy(sc1)                                     # merge neighbouring subcells if possible
20       sc1.Remove()
21
22   subcells.Sort()                                           # sort subcells for merge in 3rd direction
23   for i in subcells.length:                                  # loop over all prototypal subcells
24     sc1 = subcells[i]                                       # extract neighbouring subcells
25     sc2 = subcells[i-1]
26     if CheckCompatibility (sc1, sc2, 3) :
27       sc2.ExtendBy(sc1)                                     # merge neighbouring subcells if possible
28       sc1.Remove()
29   return subcells

```

---

the bounds of the voxels, i.e. pixels, as it was the case in the example. Actually, in these cases only the portion of the pixel/voxel located inside the finite cell is incorporated and used as subcell. This is the fundamental reason why the dimension of the corresponding subcells is precisely checked before a merging operation to ensure that only cuboidal-shaped subcells are obtained.

The major drawback of the proposed algorithm may be the potential generation of quite narrow subcells. These subcells are generated when voxels can be merged along one or two directions only, but not along the remaining ones. To investigate the impact of these narrow subcells on the results, a measure of irregularity of the subcells is defined as the ratio of the longest and shortest edge of a subcell resp. as

$$\text{irr}[sc] = \frac{\max\{l^{sc}, w^{sc}, h^{sc}\}}{\min\{l^{sc}, w^{sc}, h^{sc}\}}, \quad (3)$$

wherein l^{sc} , w^{sc} and h^{sc} denote the length, width and height of the subcell sc , respectively.

3.2 Modifications of the Octree algorithm

Given the Optimal Decomposition (OD) algorithm from the previous subsection, it may be meaningful to investigate the combination of the Octree algorithm with the newly proposed OD. Therefore, three different combinations of both algorithms are proposed, which are shortly described in the following.

3.2.1 Octree with subsequent global subcell merge

At first, the Octree algorithm is combined with the subcell-merge step proposed in the Optimal Decomposition, which leads to the OcTree-Merging algorithm (T-M). For this purpose, the Octree is first executed until the desired thresholds, e.g. until level three as depicted in Fig. 4b. Subsequently, the subcell-merge step is performed to reduce the total number of subcells, for which again all possible permutations of directions are tested. An exemplary result is shown in Fig. 4c, in which clearly less subcells than for the pure Octree (Fig. 4b) are observable. As this modified algorithm does not change the Octree algorithm in its core, it is still possible

Fig. 4 Combinations of the Octree with the Optimal Decomposition; **a** base geometry, **b** Octree level 3, **c** the T-M, **d** the T-OD and **e** the T-OD-M algorithms

that the material bounds are not precisely captured, as the bounds of the subcells from the Octree algorithm do not necessarily match the material bounds. In the current example, the original volume fraction of the green inclusion is $v_{\text{inc}} = 13/25 = 52\%$, c.f. Fig. 4a, whereas the Octree algorithm until level three yields $v_{\text{inc}} = 34/64 = 53.125\%$. The volume fraction has been changed by the Octree, and thus, the result of the integration may not be accurate.

3.2.2 Octree with subsequent local Optimal Decomposition in finest level

The procedure at the highest Octree level is altered in a different combination with the OD, and thus, it is named Octree-Optimal-Decomposition (T-OD). Instead of a rather naive assignment of the material with the highest volume fraction within the respective subcell, an Optimal Decomposition is performed within each subcell on the highest Octree level. The fundamental algorithm of the OD is not changed, and only the bounds of the corresponding subcell resulting from the Octree are used instead of the bounds of the finite cell. With this at hand, overly narrow subcells are avoided due to the regular split of the domain as the result of the Octree, combined with an exact approximation of the material bounds by cause of the application of the OD on the finest level. If this modified algorithm is applied to the present example, a subcell decomposition as depicted in Fig. 4d is the result. Compared to applying the original Octree in Fig. 4b, the material bounds are exactly captured at the expense of a higher number of subcells. Additionally, a few narrow subcells are obtained at the material interfaces. Thus, to choose the finest level for the application of the T-OD requires a trade-off between the desired number of subcells and the irregularity of the resulting subcells.

3.2.3 Octree with subsequent local Optimal Decomposition and global subcell merge

The latter two independent modifications of the Octree algorithm are now combined in a third modification, the Octree-Optimal-Decomposition-Merging (T-OD-M). Herein, first the Octree algorithm is executed until the desired recursion depth is reached, and then, the Optimal Decomposition is performed locally in those subcells on the finest level, which still contain two materials, before all subcells will be globally merged if they are compatible. If the T-OD-M is executed on the example, a result as shown in Fig. 4e appears. As can be observed, the number of subcells is indeed reduced, but narrow subcells are created as well. In this case, the bounds of the subcells created by the Octree algorithm and the bounds of the materials are matching, and thus, narrow subcells can hardly be avoided close to the interfaces. The impact of this observed behaviour is investigated by means of numerical examples in Sect. 4.

Remark Note that for voxel-based geometries considered here, the same subcell arrangement as resulting from the Optimal Decomposition algorithm could be obtained by an alternative procedure. This procedure combines two steps: first, movement of an octree decomposition (as one part of the smart octrees introduced in Kudela et al. [19]) and second, merging of subcells.

3.3 Optimized clustering

Since the proposed modifications of the Octree algorithm lead to larger numbers of subcells, they are not directly favourable over the Optimal Decomposition algorithm if narrow subcells are acceptable. However, it is not clear to what extent the OD algorithm indeed leads to the minimal number of subcells. To investigate

Fig. 5 Illustration of the Optimized Clustering algorithm (OC), **a** randomly chosen start pixel, **b** and **c** two possible restricted growth modes, **d** fully grown subcell and **e** resulting subcell decomposition

this issue, an algorithm is introduced, which constructs the subcells by solving an optimization problem. To this end, an arbitrary voxel within a finite cell is picked as starting point, c.f. the two-dimensional, exemplary problem shown in Fig. 5a. From thereon, the algorithm tries to cluster the surrounding voxels to grow the starting voxel into a larger cuboid. First, it tries to grow volumetrically in all directions, resulting in a larger cube. If this is not possible anymore, e.g. due to a voxel of a different material which would be included or due to the bounds of the finite cell, the growth mode is changed. In these cases, the algorithm restricts the expansion. At first, one particular direction is set to grow only in either positive or negative direction, whilst the other two directions are free. This case is depicted in Fig. 5b, c, where two possibilities are shown for the chosen start pixel. There, a volumetric growth is impossible due to the inclusion at the lower left corner. Whenever a growth mode is not successful anymore, the algorithm restricts an additional degree of freedom until no more degrees of freedom are left. Then, the largest subcell for the particular start voxel is found. Subsequently, the algorithm randomly picks another voxel, which is not yet clustered in a subcell, to start the cluster algorithm again, c.f. Fig. 5d, in which the largest possible subcell was found and a new starting voxel at the lower right corner was picked. These steps are repeated, until no more free voxels within a finite cell are left, c.f. Fig. 5e.

One may notice that certain restrictions in growth directions offer multiple possibilities for the subcell to grow. For example, as depicted in Fig. 5b, c, the subcell could either grow to the right or to the top of the finite cell. As it is hard to judge beforehand, which way is the best, the algorithm employs some sort of heuristic and decides randomly, which particular direction of growth is chosen. This heuristic and the randomness in choosing the starting voxels result in the awareness that a particular subcell decomposition may not be optimal for the global subcell decomposition. Thus, a Monte Carlo optimization is applied and the described algorithm is performed a certain number of times for each finite cell to find the certain subcell decomposition with the least number of subcells. Again, the algorithm of the OC is shortly summarized as pseudo-code in Algorithm 2.

Note that Algorithm 2 is mainly introduced here in order to have an estimator for the optimal setting, indicating that Algorithm 1 leads to a finite cell decomposition being close to optimal. It is obvious that the complexity of Algorithm 2 is significantly higher than the one of Algorithm 1 whose complexity is in the order of v^3 (v being a characteristic number of voxels per space direction). Anyhow, the computing time for Algorithm 1 being only a preprocessing problem is orders of magnitude smaller than the computing time required for the solution of the nonlinear mechanical problem, and thus, the efficiency of the preprocessing algorithms themselves does not matter too much. However, significant savings regarding the computing time of the mechanical problem can be reached.

4 Numerical examples

In this section, the accuracy and efficiency of the proposed approaches from Sect. 3 are analysed with regard to their applicability for computational homogenization. To this end, the homogenized mechanical response of different microstructures is computed for macroscopic uniaxial tension conditions using the FCM. The first two examples are devoted to the comparison of the presented approaches with the Octree method. Then, the introduced OD approach is used to simulate a uniaxial tension test of a real microstructure of a DP steel. The results are compared with FE computations and experimental data from Brands et al. [4].

To determine the overall mechanical response of the microstructure, the concept of FE^2 -method is utilized, see, e.g. [10–12, 14, 21, 23, 30, 33]. Therein, a microscopic boundary value problem is solved at each integration point of a macroscopic finite element discretization of a structural problem. The microscopic boundary value problem in turn consists of a finite element discretization of a representative volume element (RVE) with its

Algorithm 2: Optimized Clustering (OC)

```

1 def OptimizedClustering (finite_cells, voxel_data):
2   for fc in finite_cells:                                     # loop over all finite cells
3     subcells = []
4     for idx in monte_carlo_runs:                             # Monte Carlo optimization
5       subcells_tmp = []
6       while unassigned_voxels in fc:                         # loop free voxels
7         start_vox = PickRandomlyFrom(unassigned_voxels)
8         subcells_tmp += GrowSubcells (start_vox, subcells_tmp, voxel_data, 1)
9         if subcells_tmp.size < subcells.size:                 # update list if new minimum found
10          subcells = subcells_tmp
11   return subcells
12
13 def GrowSubcells (startcell, subcell_list, voxel_data, growth_mode):
14   switch growth_mode do                                     # recursive call decides growth mode
15     case 1:
16       trial_area = grow_cube(startcell)
17     case 2:
18       trial_area = grow_cuboidal(startcell)
19     case x:                                                  # last growth mode is simply the startvoxel
20       trial_area = startcell
21
22   if !CheckElementBounds():                                # next mode if element bounds are intersected
23     return GrowSubcells (startcell, subcell_list, voxel_data, growth_mode+1)
24   if !CheckOverlap():                                       # next mode if overlap with existing subcell
25     return GrowSubcells (startcell, subcell_list, voxel_data, growth_mode+1)
26   if !CheckMaterialInterface():                             # next mode if material interface is cut
27     return GrowSubcells (startcell, subcell_list, voxel_data, growth_mode+1)
28
29   unassigned_voxel.RemoveVoxelInside(trial_area)
30   return trial_area

```

Table 1 Material parameters used for the ferritic and martensitic phases

	E (MPa)	ν (—)	y_0 (MPa)	y_∞ (MPa)	α (—)	h (—)
Ferrite	206,000.0	0.3	260.0	580.0	9.0	70.0
Martensite	206,000.0	0.3	1000.0	2750.0	35.0	10.0

domain $\mathcal{B}_0$ and volume V . Here, a homogeneous problem, i.e. uniaxial tension, is considered at the macroscale, and thus, the macroscopic finite element problem just serves to iteratively account for the conditions of vanishing stresses in transverse directions. Therefore, the consideration of only one macroscopic element with just one integration point is sufficient. At the microscale, displacement boundary conditions are defined in terms of the macroscopic deformation gradient $\bar{\mathbf{F}}$. After solving the microscopic boundary value problem, the macroscopic stresses in form of the first Piola–Kirchhoff stress tensor $\bar{\mathbf{P}}$ are computed as volume average of the microscopic counterparts as $\bar{\mathbf{P}} = \frac{1}{V} \int_{\mathcal{B}_0} \mathbf{P} dV$. As microscopic problems, we consider RVEs which are associated with DP steels, and thus, they consist of martensitic inclusions embedded in a ferritic matrix. For these two phases, the finite J_2 elasto-plasticity formulation with isotropic hardening following Klinkel [18] and Simo [32] is used. The material parameters for both phases are taken from Brands et al. [4] and are listed in Table 1.

4.1 One element test

This first example is a rather academic example to demonstrate the accuracy and the efficiency of the proposed OD approach in comparison with the classical Octree algorithm. For this purpose, two simple artificial voxel-based microstructures are constructed. Both microstructures are discretized with only one 27-noded hexahedral element and consist of a martensitic as well as a ferritic phase.

The first microstructure is depicted in Fig. 6a and is described by $16 \times 16 \times 16$ voxels. The martensitic phase is located at a corner of the microstructure and defined by seven voxels in each spatial direction. In Fig. 6b, c, the subcells obtained for the OD and the Octree approach of level four are shown, respectively. The voxel

Fig. 6 **a** A 16^3 voxels model with a ferritic (grey color) and a martensitic (green color) phase, **b** subcells for the Octree of level 4 (323 subcells), **c** subcells for the OD approach (4 subcells), **d** macroscopic stress vs. relative elongation, **e** difference of maximal macroscopic stresses vs. level of Octree

model has been constructed such that the material boundaries coincide exactly with the subcells after four subdivision levels of the Octree method. Further subdivisions do not change the subcell decomposition. Whilst this approach requires a total number of 323 subcells, the OD method only needs four subcells to exactly describe the microstructure. The overall mechanical response of the individual subcell decompositions is plotted in Fig. 6d. As expected, the different subcell decompositions based on different Octree levels provide diverse results. Due to the variation in the volume fraction of the different phases depending on the recursion depth, the material response is stiffer for an increased martensitic fraction. For instance, the integration mesh of level one for the Octree approach with eight equally sized subcells exhibits the stiffest material response, since it has the largest phase fraction of martensite. It can be observed that the finest Octree solution corresponds exactly to the solution resulting from the optimal decomposition. The difference between the maximal stress, i.e. the stress value for an elongation of five percent, of the Octree solution $\bar{P}_{x,max}^T$ and the one of the Optimal Decomposition $\bar{P}_{x,max}^{OD}$ is used to compare both methods in Fig. 6e. Clearly visible is the convergence of the Octree solutions to the solution obtained from the Optimal Decomposition. From level four on, the results of both methods are the same, which is due to the special construction of the microstructure allowing the Octree of level four to exactly reflect the microstructure morphology.

Concerning the Octree decomposition, it may happen that the material boundaries do not fit the subcell borders after a certain number of subdivisions. This is, for instance, the case for the $24 \times 24 \times 24$ voxels model depicted in Fig. 7a. Here, the martensitic phase is represented by a $11 \times 11 \times 11$ voxel-sized subcube. The corresponding subcells for both the OD and Octree approach with six subdivisions are shown in Fig. 7b and in Fig. 7c, respectively. Similar to the previous example, the OD approach requires only four subcells to describe the microstructure exactly. For the Octree split, however, up to six subdivisions with a total of 5923 subcells are not sufficient to exactly capture the material boundaries. In Fig. 7d, the stress–elongation curve is only depicted for simulations with different subcell decompositions, since the pairs of level one and two, four and five, six and seven as well as eight and nine yield the same subcell decomposition of the microstructure, respectively. The resulting behaviour is shown in Fig. 7e, in which the mentioned levels are connected by a horizontal line. Similar to the previous model, the level of Octree subdivision influences the results of the

Fig. 7 **a** A 24^3 voxels model, **b** subcells for the Octree of level 6 (5923), **c** subcells for the OD approach (4), **d** stress vs. relative elongation, **e** difference of maximal stresses vs. level of Octree

simulation. It can be seen that the results of the Octree converge to the one of the OD approach with increasing number of subcells, but even with a recursion depth of nine, the microstructure is not exactly captured.

Discussion: Both examples reveal that the choice of the appropriate level of the Octree is not straightforward, since the convergence to the exact solution is not necessarily monotonic. Additionally, an overkill solution by choosing a high level for the Octree method is not advisable, since Octree level eight and nine yield already 96,069 subcells, rendering this problem very inefficient. These two examples consisting of simple microstructures and a single finite cell demonstrate that the proposed Optimal Decomposition approach can accurately and efficiently capture the geometry with considerably less subcells than the classical Octree decomposition. Note that a *smart* Octree would have led to a much reduced number of subcells than the standard Octree for this simple example. But this is only due to the special nature of the problem consisting of a perfectly cuboidal inclusion in a cube. It is emphasized that the discretization used in the presented examples is not able to accurately solve the boundary value problem. However, for the analysis focusing here on differences of the individual approaches for integration in terms of finite cells, the highest accuracy of solutions to the boundary value problem given a fixated discretization is a finite cell decomposition matching perfectly with the material interface. Such a decomposition is always obtained using the Optimal Decomposition approach.

4.2 Comparison of different approaches for the subcells generation

To confirm the findings from the previous example for more complex microstructures, an SSRVE in the sense of Balzani et al. [2] is used here for further investigations. Different combinations of the Optimal Decomposition and the Octree algorithm as proposed in Sect. 3 are analysed. Commonly, SSRVEs have a lower complexity than the real microstructure morphology whilst still exhibiting a similar mechanical behaviour. They require therefore less computational effort than classical RVEs. However, compared to the simple RVEs in the previous examples, the microstructure of an SSRVE is still much more complex as it is constructed to describe a real material. Here, the SSRVE for DP steel from Brands et al. [4] is considered. Based on a geometry representation with a resolution of $60 \times 60 \times 60$ voxels, a mesh of $12 \times 12 \times 12$ hexahedral finite cells with cubic shape functions is considered. In Fig. 8, the integration mesh of the SSRVE for the OD approach is illustrated exemplarily.

Fig. 8 Subcells for the Optimal Decomposition approach for a SSRVE with a martensitic phase (green colour) and a ferritic phase (grey colour)

The overall macroscopic material response in terms of the maximal stress is plotted against the different level of the Octree algorithm for the presented approaches in Fig. 9a. Again, the abbreviation T is used for Octree in the graphs, as well as T-M for the approach from Sect. 3.2.1, T-OD for the approach from Sect. 3.2.2 and T-OD-M for the approach from Sect. 3.2.3.

Discussion: As expected, approaches T and T-M exhibit the same results, since they exhibit the same material boundaries. Whilst the simulation for T with the level four requires high computational effort due to the large number of subcells, the demand is considerably lower for the T-M approach as shown in Fig. 9c. As a consequence of the high memory demand for the pure Octree approach, we continue the investigations of higher Octree level only for approaches incorporating the “merging”-step “M”, i.e. T-M and T-OD-M. The OD, T-OD and T-OD-M approaches provide the same material response, which is expected since all three approaches reflect the material boundaries exactly. Between T-OD and T-OD-M, a similar behaviour like the one between T and T-M can be observed, namely the reduction in the number of subcells. As in the previous example, it can be observed that the results of the T and T-M approaches tend to the one of the OD, T-OD and T-OD-M with the increasing number of subcells.

The irregularity of the subcells introduced in Sect. 3 is depicted in Fig. 9b. The OD, T and T-OD approaches exhibit a constant and low value of irregularity, whilst the M methods induce a growing irregularity value for increasing Octree levels. Nevertheless, the irregularity seems to have no effect on the accuracy of the integration in this example, since the decisive factor is the conservation of exact phase fraction of the different microscopic material constituents. On the other hand, the computing time presented in Fig. 9d is influenced by the irregularity of the subcells. Although the number of subcells for Octree level one and two is low for the “M”-approaches, their computing time is higher than the one of the corresponding T and T-OD approaches due to a worse convergence behaviour in the Newton iterations. In the present case, the OD method provides the best compromise in terms of accuracy, computing time and irregularity.

4.3 Optimized clustering

The optimized clustering presented in Sect. 3.3 is applied to the SSRVE with the same element discretization as in the previous example. Since one finding of the previous examples is that material-bound-preserving approaches yield the correct result, the optimal clustering is examined in view of the number of created subcells and thus the efficiency of the method. Thus, we restrict ourselves to the preprocessing, i.e. the creation of subcell decompositions and do not perform a complete simulation. For the current microstructure, the optimal decomposition approach generates 6287 subcells and a value of irregularity of 5. The key data of the subcell decompositions resulting from the Optimized Clustering for different numbers of Monte Carlo runs are listed in Table 2.

Concerning the irregularity of the subcells, both methods are equivalent even for different numbers of Monte Carlo repetitions in the optimal clustering. After a closer look to the number of elements and voxels along one edge, the resulting value of irregularity is no surprise, since the value of 5 is the quotient of the

Fig. 9 Comparison of different approaches **a** maximal stresses, **b** irregularity, **c** number of subcells and **d** computing time for different levels of Octree

Table 2 Optimized Clustering for $12 \times 12 \times 12$ Elements

Repetitions	500	1000	5000	10,000	50,000	100,000	500,000	1,000,000
Time in s	66	118	543	1057	4967	9422	45,222	87,101
Subcells	7853	6839	6573	6490	6385	6355	6296	6285
Irregularity	5	5	5	5	5	5	5	5

number of voxels and the number of elements. Thus, both methods created at least one subcell, which spans over one voxel in two directions, whilst the third direction is as long as one element. The total number of subcells resulting from the optimal clustering changes with an increasing number of Monte Carlo repetitions. Here, the least number of 6285 subcells is obtained after 1,000,000 repetitions, which correspond to approximately 24 hours of runtime for the optimal clustering algorithm. Thus, although the optimal clustering algorithm found two subcells less than the optimal decomposition, the OD is assumed to be more efficient than the optimal clustering, since the subcell decomposition was constructed in approximately one second (vs. 24 h of the OC approach).

4.4 Simulation of a real microstructure of a DP Steel

Based on the findings of the previous examples, the OD approach is now further analysed using the voxel data of the real DP steel microstructure from Brands et al. [4]. The considered portion of the microstructure has the dimensions $15.9\mu\text{m} \times 16.45\mu\text{m} \times 5\mu\text{m}$ and is captured by $731 \times 709 \times 50$ voxels.

Fig. 10 **a** Finite element discretization and the corresponding integration subcell mesh as amplified illustration, **b** results of the convergence study

Fig. 11 Number of subcells for **a** 10,240 elements and **b** 19,200 elements

4.4.1 Convergence study

In order to find a sufficiently accurate finite cell resolution for the simulation of this real microstructure, we perform numerical tests with different finite cell discretizations. In Fig. 10a an example of a finite element discretization of the microstructure by 27-noded hexahedral elements and the underlying subcells is depicted. We apply linear displacement boundary conditions at the microscale, because the morphology of the microstructure itself is not periodic, and thus, periodic boundary conditions are difficult.

The resulting stress–elongation curves of the simulations are depicted in Fig. 10b. There, additionally the results from the experiment and from a finite element computation using a discretization with quadratic tetrahedrals which conform with the material interfaces (both from Brands et al. [4]) are plotted. The macroscopic response of all FCM models is stiffer than the included solution of the FEM computation. Nevertheless, the response of the FCM models converges against a certain state with increasing fineness of the finite cell mesh. However, a gap between the FEM and finest FCM solution persists. The reason is most likely associated with the difference in the microstructure which was considered for the simulation. For the conforming FEM calculation, a smoothed morphology had to be constructed from the voxel data first. Thereby, already a difference in martensitic phase fraction of several volume percent was obtained. For the FCM calculation, directly the measured voxel data could be used, and thus, the originally measured phase fraction was included. Therefore, it is disputable which simulation should be considered as more “correct” and the experiment is taken as reference. However, neither the converged FCM nor the FEM simulation match well the experimental stress–strain response, cf. Fig. 10b, which has already been observed in Brands et al. [4]. The reason is that some important physics at the microscale are not captured if locally constant material properties are considered within the ferrite phase. Nevertheless, the converged FC mesh from this example is considered as appropriate discretization for further analysis in the next section. This mesh consists of $40 \times 40 \times 12$ equally spaced finite

cells corresponding to 19,200 hexahedral elements with a total of 430,629 degrees of freedom. For comparison, the FE mesh considered in Brands et al. [4] as appropriate discretization required 612,323 quadratic tetrahedral elements and 2,390,340 degrees of freedom and thus a significantly larger number of unknowns. Note, however, that the approximation accuracy of the solution of a boundary value problem with non-smooth material boundaries cannot be very high when using a non-conforming finite element mesh.

In Fig. 11, the methods presented in Sect. 3 are applied to the real microstructure. We focus on the number of subcells since these values correlate with the computing time. To this end, we consider the two finest mesh discretizations with, respectively, 10,240 and 19,200 finite elements. We can observe that, similar to the example in Sect. 4.2, the proposed approaches offer an advantageous decomposition with less integration effort than the classical Octree. Due to the high number of voxels, the computational effort and the memory requirement for the construction of the subcells are especially high for the octree-based approaches. This is the reason why some mesh generations in the present example abort for the sixth-level Octree. Note that all processes are for reason of fairness performed on the same computing machine.

4.4.2 Enhanced microscopic model based on initial volumetric strain (IVS) Approach

According to Brands et al. [4], the martensitic phase fraction in the measured basis voxel data is too small ($\approx 9.9\%$) in comparison with reality ($\approx 14.4\%$), i.e. the three-dimensionally measured microstructure was found not to be completely representative. An associated correction was included in [4] only in terms of constructing an appropriate SSRVE, where such statistical constraints can easily be incorporated. For the smoothed morphology of the real microstructure, this is not as straightforward, and thus, the model based on the real microstructure was not analysed in [4] regarding a comparison with experimental data.

Since here voxel data is directly used, the adjustment of the volume fraction can be easily achieved by applying a dilation procedure of one voxel layer around the inclusions. This step increases the volume fraction of the martensitic phase to approximately 13.0%. A further application of the dilation procedure provides a volume fraction of 16.1%. Since this is obviously too high, we use the modified microstructure with the martensitic phase fraction of 13.0% which closely matches the measured value of 14.4%. However, just considering this increased volume fraction of the martensitic phase is not sufficient to close the gap between experiment and simulation. Additionally, graded material properties arising from the production process of DP steel need to be taken into consideration in the ferrite phase.

The initial volumetric strain (IVS) approach originally proposed in Schröder et al. [31] and extended in [4] is applied here to capture a physical effect arising from the production process, where the originally austenitic inclusion phase transforms to martensite whilst experiencing a volume jump. At the same time, chemical diffusion processes appear, which mainly modify the carbon content in the ferrite resulting in locally distributed yield properties in the ferrite. To incorporate these modified material properties, the numerical simulation is performed in two steps. In the first step, a realistic isotropic expansion in the martensitic phase is applied to reproduce the volume expansion and to obtain the resulting, locally distributed equivalent plastic strains $\alpha_{IVS}(\mathbf{X})$ at each position $\mathbf{X}$ in the microstructure. This variable is used to compute a modification factor

$$\gamma(\mathbf{X}) := 1 + \bar{\gamma} \frac{\tilde{\alpha}_{IVS}}{\langle \alpha_{IVS}(\mathbf{X}) \rangle} \quad \text{with} \quad \tilde{\alpha}_{IVS} = \begin{cases} \langle \alpha_{IVS} \rangle, & \text{if } \alpha_{IVS}(\mathbf{X}) > \langle \alpha_{IVS}(\mathbf{X}) \rangle \\ \alpha_{IVS}(\mathbf{X}) & \text{else} \end{cases} \quad (4)$$

to shift the yield curve of the ferritic phase to an extent according to the level of experienced plastic strains due to the volume expansion of the martensitic phase. In Eq. (4), $\langle \alpha_{IVS}(\mathbf{X}) \rangle$ denotes the volume average of $\alpha_{IVS}(\mathbf{X})$ over the RVE and $\bar{\gamma}$ is a parameter obtained from experiments. The formulation $\tilde{\alpha}_{IVS}$ is introduced in order to avoid strong influences from numerical singularities onto the modification factor. Here, we use $\bar{\gamma} = 0.8$ which leads to a maximal value of $\gamma = 1.8$ for the modification factor, which corresponds to the increased hardness values maximally measured in [4]. Numerically, the graded properties in the ferrite phase are included by assigning the modified properties to the associated Gauss points. Thereby, the property fields are intrinsically assumed to be continuous and automatically smoothed by the shape functions.

With these modifications and the finite cell discretization from the previous section at hand, the uniaxial tension test is computed again. The resulting distribution of the modification factor over the RVE is depicted in Fig. 12a. In comparison with the phase distribution as depicted in Fig. 10a, the hardened zones in yellow around the martensitic phase inclusions can easily be recognized. The homogenized, macroscopic mechanical response of the microstructure using the FCM is plotted in Fig. 12b. As can be seen, the FCM simulation shows a good agreement with the experimental macroscopic stress–elongation curve. Compared to the FEM

Fig. 12 Application of the IVS approach to the real microstructure: **a** distribution of the modification factor γ , **b** comparison of experimental data with the simulations. Note that for the FCM simulation the IVS approach was applied, whereas the curve of the FEM simulation is taken from [4] where no IVS approach was considered

simulation, however, a significant reduction in the equivalent computing time of several orders of magnitude could be realized. This reduction resulted from the significant savings in degrees of freedom and integration effort. It shows that the FCM method using the OD approach does not only enable an automated, regular and simple discretization, it also allows for much more efficient simulations whilst still ensuring a sufficient accuracy.

5 Conclusion

In this contribution, new and efficient approaches for the subcell decomposition of finite elements in the context of the finite cell method were proposed and analysed. Therein, particular focus was on methods for geometries which are given as voxel/pixel data as they are usually the basis of microstructure simulations. The proposed Optimal Decomposition algorithm exploits the perpendicularity of the material interfaces to construct an optimal number of subcells for an individual element by aggregating voxels to larger cuboids independent from constraints used for the Octree strategy. In that respect, the material bounds exactly capture the ones of the voxel/pixel data and the phase fractions of different constituents are preserved. Numerical examples have illustrated that the OD algorithm performs well for numerical homogenization problems and furthermore, the solutions of the Octree decompositions converge to the solutions of the OD with increasing Octree recursion depth. Based on the Optimal Decomposition and the Octree algorithm, different combinations of both methods were proposed, which partially perform better than the Octree algorithm, but the OD turned out to be the best. As further verification, a subcell decomposition based on a Monte Carlo optimization was proposed, which is principally based on greedy subcell growth from random starting points. It could be shown that the resulting number of subcells converges to the number of subcells resulting from the Optimal Decomposition approach. Eventually, the developed Optimal Decomposition was used in combination with the initial volume strain approach (IVS) from [4] to simulate a tension test on a real microstructure. The results of the simulation were found to be in good agreement with experimental data. However, a significant reduction in computing time of several orders of magnitude could be achieved compared to the FE computation in [4], where a conforming mesh was used. Summarizing, it was shown that the proposed Optimal Decomposition approach is an efficient and accurate subcell decomposition approach for voxel-based geometries, which has the following major benefits:

- A regular discretization can be used, thanks to the finite cell method, which allows for a simple automated mesh construction.
- A significant reduction in subcells and thus assembling effort compared to classical finite cell approaches can be achieved by using the proposed Optimal Decomposition approach.
- The efficiency of microstructure simulations can be improved by several orders of magnitude due to less degrees of freedom compared to conforming finite element meshes and due to the reduced integration effort.

Note that our approach enables a more efficient integration. The potentially steep gradients in the solution fields induced by material interfaces inside the elements still pose challenges regarding the discretization.

Especially, for low-order polynomials, a large number of elements is usually required which does reduce the effectiveness of the finite cell method and thus the potential savings of computing time enabled through the proposed approaches. That is why higher-order shape functions are usually favoured.

Acknowledgements Open Access funding provided by Projekt DEAL. Financial support by the German Research Foundation (DFG) as part of the Priority Program 1648, Project BA2823/11-2 is gratefully acknowledged by the authors.

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/>.

Compliance with ethical standards

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Al-Abbasi, F., Nemes, J.: Micromechanical modeling of dual phase steels. *Int. J. Mech. Sci.* **45**(9), 1449–1465 (2003). <https://doi.org/10.1016/j.ijmecsci.2003.10.007>
2. Balzani, D., Scheunemann, L., Brands, D.: Construction of two- and three-dimensional statistically similar RVEs for coupled micro-macro simulations. *Comput. Mech.* **54**, 1269–1284 (2014). <https://doi.org/10.1007/s00466-014-1057-6>
3. Bishop, J.: Rapid stress analysis of geometrically complex domains using implicit meshing. *Comput. Mech.* **30**, 460–478 (2003). <https://doi.org/10.1007/s00466-003-0424-5>
4. Brands, D., Balzani, D., Scheunemann, L., Schröder, J., Richter, H., Raabe, D.: Computational modeling of dual-phase steels based on representative three-dimensional microstructures obtained from EBSD data. *Arch. Appl. Mech.* **86**, 575–598 (2016). <https://doi.org/10.1007/s00419-015-1044-1>
5. Calcagnotto, M., Ponge, D., Demir, E., Raabe, D.: Orientation gradients and geometrically necessary dislocations in ultrafine grained dual-phase steels studied by 2D and 3D EBSD. *Mater. Sci. Eng., A* **527**(10), 2738–2746 (2010). <https://doi.org/10.1016/j.msea.2010.01.004>
6. Davies, R.G.: Influence of martensite composition and content on the properties of dual phase steels. *Metall. Trans. A* **9**(5), 671–679 (1978). <https://doi.org/10.1007/BF02659924>
7. Del Pino, S., Pironneau, O.: A fictitious domain based general PDE solver. In: Kuznetsov, Y., Neittanmaki, P., Pironneau, O. (eds), *Numerical methods for scientific computing variational problems and applications*. CIMNE, Barcelona (2003)
8. Düster, A., Parvizi, J., Yang, Z., Rank, E.: The finite cell method for three-dimensional problems of solid mechanics. *Comput. Methods Appl. Mech. Eng.* **197**(197), 3768–3782 (2008). <https://doi.org/10.1016/j.cma.2008.02.036>
9. Eisenlohr, P., Diehl, M., Lebensohn, R., Roters, F.: A spectral method solution to crystal elasto-viscoplasticity at finite strains. *Int. J. Plast.* **46**, 37–53 (2013). <https://doi.org/10.1016/j.iplas.2012.09.012>
10. Feyel, F.: Multiscale FE^2 elastoviscoplastic analysis of composite structures. *Comput. Mater. Sci.* **16**(1), 344–354 (1999). [https://doi.org/10.1016/S0927-0256\(99\)00077-4](https://doi.org/10.1016/S0927-0256(99)00077-4)
11. Feyel, F.: A multilevel finite element method (FE^2) to describe the response of highly non-linear structures using generalized continua. *Comput. Methods Appl. Mech. Eng.* **192**(28), 3233–3244 (2003). [https://doi.org/10.1016/S0045-7825\(03\)00348-7](https://doi.org/10.1016/S0045-7825(03)00348-7)
12. Feyel, F., Chaboche, J.-L.: FE^2 multiscale approach for modelling the elastoviscoplastic behaviour of long fibre SiC/Ti composite materials. *Comput. Methods Appl. Mech. Eng.* **183**(3), 309–330 (2000). [https://doi.org/10.1016/S0045-7825\(99\)00224-8](https://doi.org/10.1016/S0045-7825(99)00224-8)
13. Glowinski, R., Kuznetsov, Y.: Distributed lagrange multipliers based on fictitious domain method for second order elliptic problems. *Comput. Methods Appl. Mech. Eng.* **196**, 1498–1506 (2007). <https://doi.org/10.1016/j.cma.2006.05.013>
14. Golanski, D., Terada, K., Kikuchi, N.: Macro and micro scale modeling of thermal residual stresses in metal matrix composite surface layers by the homogenization method. *Comput. Mech.* **19**(3), 188–202 (1997). <https://doi.org/10.1007/s004660050168>
15. Hubrich, S., Di Stolfo, P., Kudela, L., Kollmannsberger, S., Rank, E., Schröder, A., Düster, A.: Numerical integration of discontinuous functions: moment fitting and smart octree. *Comput. Mech.* **60**, 863–881 (2017). <https://doi.org/10.1007/s00466-017-1441-0>
16. Joulaian, M., Hubrich, S., Düster, A.: Numerical integration of discontinuities on arbitrary domains based on moment fitting. *Comput. Mech.* **57**, 979–999 (2016). <https://doi.org/10.1007/s00466-016-1273-3>
17. Kim, S., Lee, S.: Effects of martensite morphology and volume fraction on quasi-static and dynamic deformation behavior of dual-phase steels. *Metallur. Mater. Trans. A* **31**(7), 1753–1760 (2000). <https://doi.org/10.1007/s11661-998-0328-2>
18. Klinkel, S.: *Theorie und Numerik eines Volumen-Schalen-Elementes bei finiten elastischen und plastischen Verzerrungen*. PhD thesis, Universität Karlsruhe, Karlsruhe (2000)
19. Kudela, L., Zander, N., Kollmannsberger, S., Rank, E.: Smart octrees: Accurately integrating discontinuous functions in 3D. *Comput. Methods Appl. Mech. Eng.* **306**, 406–426 (2016). <https://doi.org/10.1016/j.cma.2016.04.006>

20. Lee, S.-B., Lebensohn, R., Rollett, A.: Modeling the viscoplastic micromechanical response of two-phase materials using Fast Fourier Transforms. *Int. J. Plast.* **27**(5), 707–727 (2011). <https://doi.org/10.1016/j.ijplas.2010.09.002>
21. Miehe, C., Schröder, J., Schotte, J.: Computational homogenization analysis in finite plasticity. Simulation of texture development in polycrystalline materials. *Comput. Methods Appl. Mech. Eng.* **171**(3), 387–418 (1999). [https://doi.org/10.1016/S0045-7825\(98\)00218-7](https://doi.org/10.1016/S0045-7825(98)00218-7)
22. Miska, N., Balzani, D.: Quantification of uncertain macroscopic material properties resulting from variations of microstructure morphology based on statistically similar volume elements: application to dual-phase steel microstructures. *Comput. Mech.* **64**(6), 1621–1637 (2019). <https://doi.org/10.1007/s00466-019-01738-8>
23. Moulinec, H., Suquet, P.: A numerical method for computing the overall response of nonlinear composites with complex microstructure. *Comput. Methods Appl. Mech. Eng.* **157**(1), 69–94 (1998). [https://doi.org/10.1016/S0045-7825\(97\)00218-1](https://doi.org/10.1016/S0045-7825(97)00218-1)
24. Parvizian, 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). <https://doi.org/10.1007/s00466-007-0173-y>
25. Pierman, A.-P., Bouaziz, O., Pardoen, T., Jacques, P., Brassart, L.: The influence of microstructure and composition on the plastic behaviour of dual-phase steels. *Acta Mater.* **73**, 298–311 (2014). <https://doi.org/10.1016/j.actamat.2014.04.015>
26. Ramière, R., Angot, P., Belliard, M.: A fictitious domain approach with spread interface for elliptic problems with general boundary conditions. *Comput. Methods Appl. Mech. Eng.* **196**(4), 766–781 (2007). <https://doi.org/10.1016/j.cma.2006.05.012>
27. Scheunemann, L., Balzani, D., Brands, D., Schröder, J.: Construction of statistically similar RVEs. In: Conti, S., Hackl, K. (eds), *Analysis and computation of microstructure in finite plasticity* (Lecture notes in applied and computational mechanics 78), Springer, (2015). doi: <https://doi.org/10.1007/978-3-319-18242-1>
28. Schillinger, D., Ruess, M.: The finite cell method: a review in the context of higher-order structural analysis of CAD and image-based geometric models. *Arch. Comput. Methods Eng.* **22**(3), 391–455 (2015). <https://doi.org/10.1007/s11831-014-9115-y>
29. Schneider, K., Klusemann, B., Bargmann, S.: Automatic three-dimensional geometry and mesh generation of periodic representative volume elements for matrix-inclusion composites. *Adv. Eng. Softw.* **99**, 177–188 (2016). <https://doi.org/10.1016/j.advengsoft.2016.06.001>
30. Schröder, J.: A numerical two-scale homogenization scheme: the FE^2 -method. In: *Plasticity and beyond—microstructures, crystal-plasticity and phase transitions*. CISM lecture notes, volume 550. Springer Berlin, (2014). doi: https://doi.org/10.1007/978-3-7091-1625-8_1
31. Schröder, J., Balzani, D., Richter, H., Schmitz, H., Kessler, L.: Simulation of microheterogeneous steels based on a discrete multiscale approach. In: *Proceedings of the 7th International Conference and Workshop on Numerical Simulation of 3D Sheet Metal Forming Processes*, pp. 379–383 (2008)
32. Simo, J.: Algorithms for static and dynamic multiplicative plasticity that preserve the classical return mapping schemes of the infinitesimal theory. *Comput. Methods Appl. Mech. Eng.* **99**, 61–112 (1992). [https://doi.org/10.1016/0045-7825\(92\)90123-2](https://doi.org/10.1016/0045-7825(92)90123-2)
33. Smit, R., Brekelmans, W., Meijer, H.: Prediction of the mechanical behavior of nonlinear heterogeneous systems by multi-level finite element modeling. *Comput. Methods Appl. Mech. Eng.* **155**(1), 181–192 (1998). [https://doi.org/10.1016/S0045-7825\(97\)00139-4](https://doi.org/10.1016/S0045-7825(97)00139-4)
34. Yang, Z., Ruess, M., Kollmannsberger, S., Düster, A., Rank, E.: An efficient integration technique for the voxel-based finite cell method. *Int. J. Numer. Meth. Eng.* **91**, 457–471 (2012). <https://doi.org/10.1002/nme.4269>
35. Zaefferer, S., Wright, S., Raabe, D.: Three-dimensional orientation microscopy in a focused ion beam-scanning electron microscope: a new dimension of microstructure characterization. *Metallur. Mater. Trans. A* **39**(2), 374–389 (2008). <https://doi.org/10.1007/s11661-007-9418-9>