Advances in Engineering Software 99 (2016) 177–188

Contents lists available at ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

Research Paper

Automatic three-dimensional geometry and mesh generation of periodic representative volume elements for matrix-inclusion composites

Konrad Schneider[a,*], Benjamin Klusemann[b,c], Swantje Bargmann[a,c]

[a] Institute of Continuum Mechanics and Material Mechanics, Hamburg University of Technology, Hamburg, Germany
[b] Institute of Product and Process Innovation, Leuphana University of Lüneburg, Lüneburg, Germany
[c] Institute of Materials Research, Helmholtz-Zentrum Geesthacht, Geesthacht, Germany

ARTICLE INFO

Article history:
Received 7 December 2015
Revised 21 April 2016
Accepted 4 June 2016
Available online 14 July 2016

Keywords:
Periodic mesh
Conforming mesh
Random sequential absorption
Representative volume element
Meshing strategy
Homogenization
Matrix-inclusion composite

ABSTRACT

This paper introduces an efficient method to automatically generate and mesh a periodic three-dimensional microstructure for matrix-inclusion composites. Such models are of major importance in the field of computational micromechanics for homogenization purposes utilizing unit cell models. The main focus of this contribution is on the creation of cubic representative volume elements (RVEs) featuring a periodic geometry and a periodic mesh topology suitable for the application of periodic boundary conditions in the framework of finite element simulations. Our method systematically combines various meshing tools in an extremely efficient and robust algorithm. The RVE generation itself follows a straightforward random sequential absorption approach resulting in a randomized periodic microstructure. Special emphasis is placed on the discretization procedure to maintain a high quality mesh with as few elements as possible, thus, manageable for computer simulations applicable to high volume concentrations, high number of inclusions and complex inclusion geometries. Examples elucidate the ability of the proposed approach to efficiently generate large RVEs with a high number of anisotropic inclusions incorporating extreme aspect ratios but still maintaining a high quality mesh and a low number of elements.

1. Introduction

A crucial aspect in the development and optimization of high performance materials is the utilization of heterogeneous materials such as particle or fiber reinforced composites. For simulating and predicting the mechanical deformation behavior most accurately, it is essential to incorporate information of the underlying microstructure. The research field of computational micromechanics deals with this topic and the issues of how this should be conducted. A major interest in this field lies in the prediction of effective material properties or in the deduction of constitutive laws via multiscale methods [14,18,35]. One sophisticated approach is addressed to the investigation of unit cells that act as RVEs[1] of the material of interest [17,18]. The guiding idea is the computation of material responses on sample RVEs by solving a boundary value problem using numerical methods. The numerical method of choice throughout this work is the finite element method representing a state-of-the-art technology in computational engineering. For employing finite element simulations in this context, a discretization of the unit cells is inevitable. This rises the important question on what the RVE should look like and what the requirements of a proper finite element mesh are.

The geometric information of real microstructures gained from experimental observations, such as image reconstruction, are only partially suitable for numerical simulations due to their abundance and complexity [7,13]. For simulation purposes it is often necessary to artificially generate geometrically simpler RVEs which feature relevant properties of the real material. In this regard, the microstructures may be interpreted as the result of a stochastic process [35]. Generating a RVE can therefore be sourced by an artificial stochastic process which in turn leads to artificial microstructures. To verify and to ensure the quality and compliance of the generated geometry a stochastic equivalence between the real and artificial microstructures is desirable [28]. Considering the general

* Corresponding author.
E-mail address: Konrad.Schneider@tuhh.de (K. Schneider).

[1] Following [2] we associate the terminus unit cell with any volume that is capable of forming a periodic microstructure via congruence mappings (translation, mirroring, rotation). As a limiting case a unit cell might be a RVE.

http://dx.doi.org/10.1016/j.advengsoft.2016.06.001

computational feasibility, only a few meaningful geometric parameters and their corresponding distributions might be included in such a process. A common approach to generate randomized artificial RVEs is the utilization of random sequential absorption processes [15,26,33] that successively built up the microstructure.

One major class of microstructured heterogeneous materials are matrix-inclusion composites featuring non-overlapping inclusions to which this paper addresses. Many authors dedicated their work to these materials considering various types of inclusions such as spherical particles [3,4,11,22,23], ellipsoids [3,5,12,24] or cylinders [1,8]. A cumbersome task in the generation process is placing the inclusions randomly while preserving the non-overlapping requirement. For this task intersection tests or distance queries between two inclusions need to be conducted. Especially the generation of unit cells with high volume fractions is difficult to manage [see e.g. 3, 24, 8]. More sophisticated approaches like molecular dynamic methods [12], geometric adaption of the particles [1], dynamic simulation of densification [34] or simulated annealing [26] are necessary to circumvent this problem.

Besides the geometry generation itself the geometric discretization is an important step for subsequent simulations. Especially for finite element simulations this becomes a non-trivial task. In context of the underlying boundary value problem, with respect to homogenization purposes, the application of periodic boundary conditions is favorable [18,21]. A drawback of this method is the inevitable requirement of a periodic mesh topology. Although there exist some software packages that feature a periodic mesh generation, e.g., Netgen [27] or commercial meshing packages which allow mesh copying and constrained meshing, there is no straightforward way of generating such meshes. Problems may arise from the restriction to very simple inclusion geometries, such as spheres. Unstable boolean operations, which are likely to fail in the process of constructing the geometry, might be an inevitable obstacle. On the other hand, the mesh size can easily increase, resulting in finite element models too large for efficient simulations. In [3–5] the authors investigated random RVEs with different types of inclusions using Netgen. It was revealed that for small number of inclusions very large number of elements result from the discretization process (e.g., 15 inclusions yield 100.000 elements). To circumvent this problem, [24] divided the meshing process into several steps. However, they were only able to consider spheroids with aspect ratios smaller than three. The recent work of [8] shows an approach featuring a large number of inclusions with infinite length. A combination of multiple software packages is applied to circumvent the drawback of using boolean operations. However, neither the generated microstructure nor the mesh feature a periodic topology. In this regard, the publication of [32] reveals a promising approach. By successively treating each inclusion individually a periodic mesh is obtained. However, their method requires the utilization of polyhedral finite elements, which requires non-standard software.

Another noticeable class of microstructured heterogeneous materials are polycrystals with a pronounced grain topology. [10,17,25] investigated their generation, discretization and effective mechanical properties by approximating the granular structure via Voronoi-diagrams. Again, only non-standard methods [10] result in a periodic mesh topology suitable for periodic boundary conditions.

These examples highlight the demand for a proper method for generating randomized matrix-inclusion RVEs featuring a periodic mesh topology. The central contribution of this paper is an algorithm that automatically generates a periodic tetrahedralization of cubic matrix-inclusion RVEs for the use in finite element simulations. The outline of this paper is as follows: First, we describe the microstructure geometry generation process. Thereafter, the individual inclusions are incorporated into a constructive solid geometry model, thus, taking care of potential intersections with the unit cell. The difficulties of discretization are solved by meshing the inclusions successively, hence, breaking down the meshing process into smaller subtasks. Therefore, the generated surface meshes of the constructive solid geometry representations of all inclusions are periodically distributed in the RVE, master edges and surfaces are created, resulting in a waterproof surface mesh. Afterwards, a volume mesh of high quality tetrahedrons is generated to obtain a discretization of the whole structure. Finally, we elucidate the quality of the generated mesh, and compare it to meshes from available software packages, e.g., Netgen.

2. Microstructure generation

Exemplarily, the considered microstructures possess ellipsodial inclusions of revolution, namely spheroids. The examined RVEs exhibit a cuboid like shape featuring translational periodicity with these spheroids inside. Before the actual microstructure generation process is explained, we introduce an accurate description of the geometric setting, the significant geometric primitives and their randomized placement.

2.1. Mathematical description of geometric setting

One corner of the cuboid shaped RVE is located at the origin of the global coordinate system with edges aligned parallel to the coordinate axes as shown in Fig. 1. The side lengths of the cuboid are denoted by a_x, a_y and a_z. The faces of the RVE are interpreted as subsets of planes P_i. We describe these planes by the Hessian normal form

$$P_i = \{\boldsymbol{x} \in \mathbb{R}^3 \,|\, \boldsymbol{x}^{\mathrm{T}} \cdot \boldsymbol{n}_p + d = 0\}, \tag{1}$$

with $\boldsymbol{n}_p$ being the outward pointed normal vector and $|d|$ being the distance of the plane from the origin. With this description the six faces of the cuboid are addressed by

$$\begin{aligned}
P_{x_0} &= \{\boldsymbol{n}_{x_0} = [-1, \quad 0, \quad 0]^{\mathrm{T}}; d = 0\},\\
P_{x_1} &= \{\boldsymbol{n}_{x_1} = [\quad 1, \quad 0, \quad 0]^{\mathrm{T}}; d = -a_x\},\\
P_{y_0} &= \{\boldsymbol{n}_{y_0} = [\quad 0, -1, \quad 0]^{\mathrm{T}}; d = 0\},\\
P_{y_1} &= \{\boldsymbol{n}_{y_1} = [\quad 0, \quad 1, \quad 0]^{\mathrm{T}}; d = -a_y\},\\
P_{z_0} &= \{\boldsymbol{n}_{z_0} = [\quad 0, \quad 0, -1]^{\mathrm{T}}; d = 0\},\\
P_{z_1} &= \{\boldsymbol{n}_{z_1} = [\quad 0, \quad 0, \quad 1]^{\mathrm{T}}; d = -a_z\}.
\end{aligned} \tag{2}$$

The edges of the cuboid are defined by subsets of lines L_i

$$L_i = \{\boldsymbol{a} + r\,\boldsymbol{n}_L \,|\, r \in \mathbb{R}\}, \tag{3}$$

where $\boldsymbol{a}$ is a point on the line and $\boldsymbol{n}_L$ the direction vector. Additionally, all 12 edges of the cube are interpreted as subsets of the intersection of two planes and expressed as

$$\begin{aligned}
&\left.\begin{aligned} L_{y_0z_0} &= \{P_{y_0} \wedge P_{z_0}\}, L_{y_0z_1} = \{P_{y_0} \wedge P_{z_1}\},\\ L_{y_1z_1} &= \{P_{y_1} \wedge P_{z_1}\}, L_{y_1z_0} = \{P_{y_1} \wedge P_{z_0}\}\end{aligned}\right\} \text{ with } \boldsymbol{n}_L = \begin{bmatrix}0\\1\\0\end{bmatrix},\\
&\left.\begin{aligned} L_{x_0z_0} &= \{P_{x_0} \wedge P_{z_0}\}, L_{x_0z_1} = \{P_{x_0} \wedge P_{z_1}\},\\ L_{x_1z_1} &= \{P_{x_1} \wedge P_{z_1}\}, L_{x_1z_0} = \{P_{x_1} \wedge P_{z_0}\}\end{aligned}\right\} \text{ with } \boldsymbol{n}_L = \begin{bmatrix}0\\1\\0\end{bmatrix},\\
&\left.\begin{aligned} L_{x_0y_0} &= \{P_{x_0} \wedge P_{y_0}\}, L_{x_0y_1} = \{P_{x_0} \wedge P_{y_1}\},\\ L_{x_1y_1} &= \{P_{x_1} \wedge P_{y_1}\}, L_{x_1y_0} = \{P_{x_1} \wedge P_{y_0}\}\end{aligned}\right\} \text{ with } \boldsymbol{n}_L = \begin{bmatrix}0\\0\\1\end{bmatrix}.
\end{aligned} \tag{4}$$

The eight corners C_i of the cuboid are described as the intersection of three planes by

$$\begin{aligned}
C_{x_0y_0z_0} &= \{P_{x_0} \wedge P_{y_0} \wedge P_{z_0}\}, \quad C_{x_0y_1z_0} = \{P_{x_0} \wedge P_{y_1} \wedge P_{z_0}\},\\
C_{x_1y_1z_0} &= \{P_{x_1} \wedge P_{y_1} \wedge P_{z_0}\}, \quad C_{x_1y_0z_0} = \{P_{x_1} \wedge P_{y_0} \wedge P_{z_0}\},\\
C_{x_0y_0z_1} &= \{P_{x_0} \wedge P_{y_0} \wedge P_{z_1}\}, \quad C_{x_0y_1z_1} = \{P_{x_0} \wedge P_{y_1} \wedge P_{z_1}\},\\
C_{x_1y_1z_1} &= \{P_{x_1} \wedge P_{y_1} \wedge P_{z_1}\}, \quad C_{x_1y_0z_1} = \{P_{x_1} \wedge P_{y_0} \wedge P_{z_1}\}.
\end{aligned} \tag{5}$$

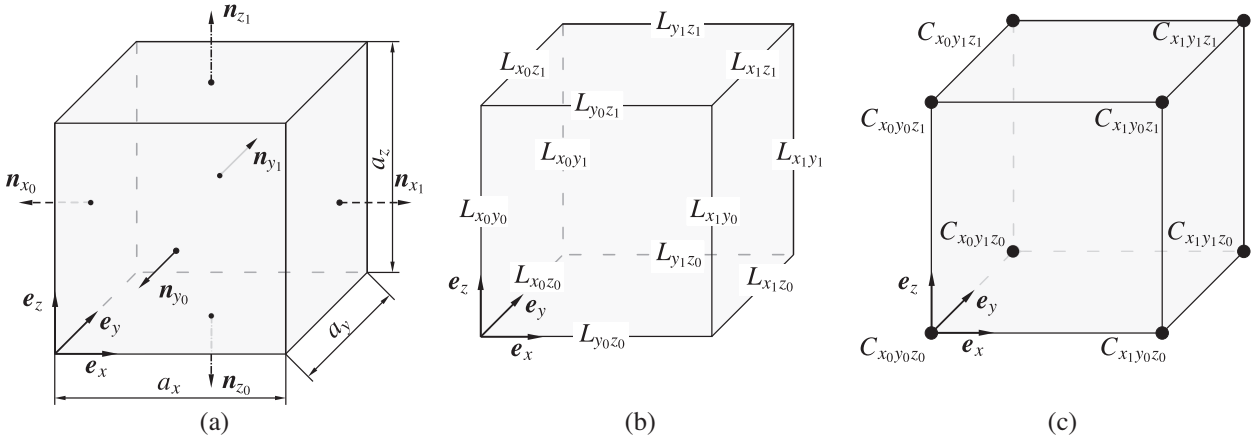


Fig. 1. Description of cuboids: (a) normal vectors $\boldsymbol{n}$ defining faces P (see Eq. (2)), (b) edges L (see Eq. (4)) and (c) corners C (see Eq. (5)).

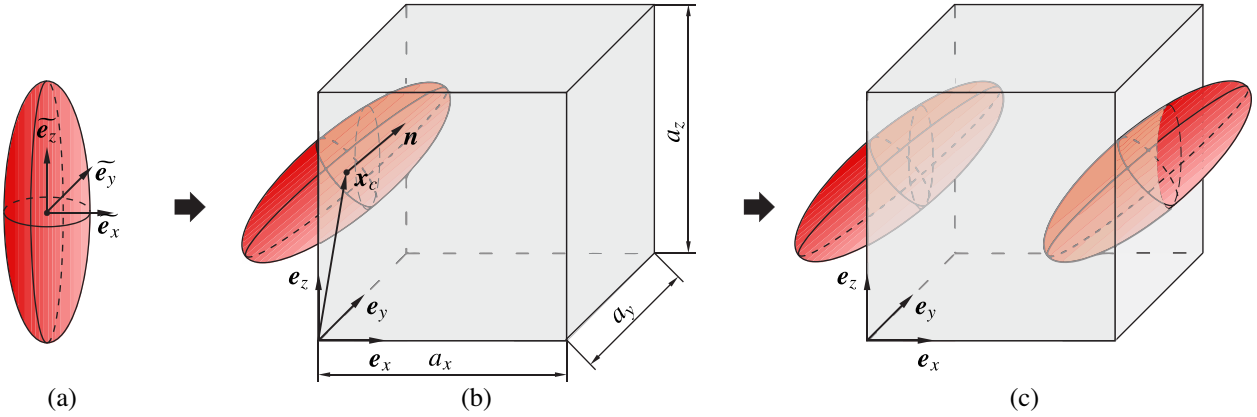


Fig. 2. Placement procedure of a spheroid from the initial state (a) described in local tilde-coordinate system into the unit cell, (b) described in global coordinate system and (c) its periodization.

With these definitions of the geometric primitives of the cube, the inclusions geometric primitive are described in the following. In this work, we focus on ellipsoids of revolution, namely spheroids; an extension to other types of inclusions is, however, straightforward. We represent a spheroid by its half-axis $h_x = h_y$ and h_z defining its shape. The position is specified by the center $\boldsymbol{x}_c$. The orientation vector $\boldsymbol{n}(\varphi, \theta)$ is pointing in direction of the h_z-axis (see Fig. 2(a)). We have

$$h_x, h_z \in \mathbb{R}^+, \boldsymbol{x}_c \in \mathbb{R}^3, \quad \boldsymbol{n} = [\sin\theta\cos\varphi, \sin\theta\sin\varphi, \cos\theta]^{\mathrm{T}} \quad \text{with} \quad \varphi \in [-\pi, \pi], \theta \in [0, \pi]\ . \tag{6}$$

φ and θ are the azimuth and polar angle with respect to the global coordinate system. Due to the transversal isotropic properties of spheroids we describe their orientation by only two parameters. A general spheroid is mathematically defined by the set

$$\begin{aligned} E_i(\boldsymbol{x}_c, h_x, \alpha, \boldsymbol{n}(\varphi,\theta)) = \{&\boldsymbol{x} \in \mathbb{R}^3, h_x \in \mathbb{R}^+, \alpha \in \mathbb{R}^+, \varphi \in [-\pi,\pi], \\ &\theta \in [0,\pi]\,|\,\boldsymbol{x}^{\mathrm{T}}\cdot\boldsymbol{A}\cdot\boldsymbol{x} + \boldsymbol{b}^{\mathrm{T}}\cdot\boldsymbol{x} + c \le 0\} \\ \text{with} \quad \boldsymbol{A} = \boldsymbol{Q}^{\mathrm{T}}\cdot\widetilde{\boldsymbol{A}}\cdot\boldsymbol{Q}, &\boldsymbol{b} = -2\boldsymbol{A}\cdot\boldsymbol{x}_c, c = \boldsymbol{x}_c^{\mathrm{T}}\cdot\boldsymbol{A}\cdot\boldsymbol{x}_c - 1, \\ \widetilde{\boldsymbol{A}} = \mathrm{diag}&\left(\frac{1}{h_x^2}, \frac{1}{h_x^2}, \frac{1}{[\alpha h_x]^2}\right), \\ \boldsymbol{Q} = 2\boldsymbol{v}\cdot\boldsymbol{v}^{\mathrm{T}} - \boldsymbol{1}, \quad &\boldsymbol{v} = \frac{\boldsymbol{n} + \boldsymbol{e}_z}{||\boldsymbol{n} + \boldsymbol{e}_z||_2}. \end{aligned} \tag{7}$$

Here, $\boldsymbol{Q}$ represents a proper orthogonal matrix resulting from Rodrigues' rotation formula with a rotation angle of 180°, $\boldsymbol{v}$ is the normalized axis of rotation, $\boldsymbol{e}_z$ the unit vector in z-direction of the global coordinate system, $||\bullet||_2$ the L^2-norm and diag($\bullet$) the operator that generates a diagonal matrix from a vector.

The RVE generation process is conducted by placing spheroids inside the cuboid until the desired number of inclusions or volume fraction f is obtained. Geometric compatibility requires a periodic RVE, which necessitates that all inclusions intersecting with the boundary of the unit cell must have periodic counter pairs on opposing faces. The randomized placement of inclusions is conducted by interpreting the placement procedure as a stochastical process. Random positions and orientations are generated leading to seven stochastic parameters in the most general case: three coordinates of the inclusion center $\boldsymbol{x}_c$, two angles describing the orientation vector $\boldsymbol{n}(\varphi, \theta)$, one half-axis h_x and the aspect ratio α, relating the half-axes by $\alpha h_x = h_z$ (cf. Eq. (6)). This parameterization allows the generation of arbitrary distributions of randomly orientated spheroids. Finally, four random generator functions $\mathcal{P}(\Omega), \mathcal{R}(\Omega), \mathcal{S}(\Omega), \mathcal{T}(\Omega)$, corresponding to arbitrarily chosen distributions, with $\Omega \in [0, 1]$ being the probability space, are introduced. They are employed to the stochastic parameters via

$$\begin{aligned} &\boldsymbol{x}_c \propto [a_x\,\mathcal{P}(\Omega), a_y\,\mathcal{P}(\Omega), a_z\,\mathcal{P}(\Omega)], \\ &\alpha \propto [\alpha_{max} - \alpha_{min}]\,\mathcal{R}(\Omega) + \alpha_{min}, \\ &\varphi \propto 2\pi\,\mathcal{S}(\Omega) - \pi, \\ &\theta \propto \pi\,\mathcal{T}(\Omega)\ , \end{aligned} \tag{8}$$

with α_{max} and α_{min} being the maximum and minimum aspect ratios of the spheroids.

2.2. Placement procedure

A random sequential absorption method [15,33] is the fundamental process for the placement procedure. It generates hardcore

Algorithm 1: Periodic inclusion positioning inside a RVE via random sequential absorption method.

Input: number of inclusions N, inclusion parameter set $[h_x, \alpha_{max}]$, RVE dimensions $[a_x, a_y, a_z]$
Output: arrangement of inclusions stored in: *Inclusions*

```
1  Inclusions ← initialize
2  while #Inclusions ≠ N do
3      Inclusion ← generate_Inclusion([h_x, α_max], [a_x, a_y, a_z])
4      Periodic_Inclusions ← periodize_Inclusion(Inclusion, [a_x, a_y, a_z])
5      foreach Inclusion_i of Periodic_Inclusions do
6          foreach Inclusion_j of Inclusions do
7              admissibility_flag ← check_admissibility(Inclusion_i,Inclusion_j)
8              if admissibility_flag == FALSE then
9                  reject Inclusion and go to 3
10             end
11         end
12     end
13     if admissibility_flag then
14         Inclusions += Periodic_Inclusions_i
15     end
16 end
```

arrangements (non-overlapping inclusions) of lower to moderate volume fractions[2]. Algorithm 1 depicts a schematic concept of our approach. Its input is the number of target inclusions N, the parameter set characterizing the shapes of the spheroids $[h_x, \alpha_{max}]$ and the dimensions of the RVE $[a_x, a_y, a_z]$. After an inclusion is randomly placed as outlined above, periodization follows. If a generated inclusion intersects the boundary of the RVE, copies of this inclusion are distributed around the RVE in a periodic manner (cf. Fig. 2). Next, the newly created spheroids are tested for admissibility against all already placed inclusions and the RVE boundary.

If all checks reveal positive, the inclusion and its periodic copies are approved and added to the global assembly.

Crucial parts of this algorithm are the `periodize_Inclusion()` and `check_admissibility()` functions. The `periodize_Inclusion()` function checks if the inclusion intersects with the boundary of the RVE by computing the maximal and minimal ellipsoid-plane distances to all six planes of the cuboid. By computing all stationary points and their corresponding objective function values of the constrained optimization problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^3} \ \boldsymbol{x}^{\mathrm{T}}\cdot\boldsymbol{n}_p + d \quad \text{s.t.} \quad \boldsymbol{x}^{\mathrm{T}}\cdot\boldsymbol{A}\cdot\boldsymbol{x} + \boldsymbol{b}^{\mathrm{T}}\cdot\boldsymbol{x} + c = 0 \ , \tag{9}$$

the distances are set. In the optimization problem Eq. (9), the objective function represents the Hessian normal form of the planes P_i: $\{\boldsymbol{x}\in\mathbb{R}^3 | \boldsymbol{x}^{\mathrm{T}}\cdot\boldsymbol{n}_p + d = 0\}$, emanating from the definition of the RVE faces as shown in Fig. 3. The solution of Eq. (9) leads to a quadratic equation with two distinctive realvalued solutions[3]. With these solutions the minimal and maximal signed distances d and d_0 between the plane and the ellipsoid, as shown in Fig. 3(a), are calculated. If the two distances have different signs, plane and inclusion intersect and the spheroid is copied accordingly. If multiple plane intersections are simultaneously present, multiple copies of the inclusions are required to establish periodicity[4].

The `check_admissibility()` routine calculates the distances from the processed inclusion to all already placed inclusions, all RVE face-planes as well as to all RVE corners and compares them with a given threshold distance $d_{min} > 0$. If any distance is smaller than d_{min}, the current inclusion is rejected (e.g., Algorithm 1 line 9). These checks ensure the non-overlapping requirement for inclusions. Furthermore, very small and badly shaped volumes are avoided, if the threshold is chosen large enough. This is extremely beneficial for the subsequent meshing procedure by preventing the amount of poor and degenerated elements and, thus, resulting in more high quality elements. Eventually, the minimum distance between the spheroid and a plane is determined by computing the optimal value of Eq. (9). The distance calculation between two spheroids is more complex [9,19] since the constrained optimization problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^3,\boldsymbol{y}\in\mathbb{R}^3} ||\boldsymbol{x}-\boldsymbol{y}||_2 \quad s.t. \quad \boldsymbol{x}\in E_0 \ , \ \boldsymbol{y}\in E_1 \tag{10}$$

is nonlinear and non-convex. An analytical solution of problem (10) is not possible. [20] suggested an effective iterative algorithm for solving the ellipsoid-ellipsoid distance problem. The fundamental idea of their method is shown in Fig. 3(c) and explained in the following. Balls are inserted into E_0 and E_1 such that they lie completely inside their corresponding spheroid and tangential to the respective boundaries. The iterative process starts by generating the connecting line between both ball centers and calculating its penetration points on the boundaries of E_0 and E_1. Then, the two balls are repositioned such that they touch the penetration points while still being tangential to the spheroid boundaries. This process is repeated until the direction of the connection line coincides with the normal vector of the spheroid boundaries at the penetration points. Accordingly, the resulting connection line represents the shortest connection path between two inclusions with its magnitude being the desired distance. This intuitive method has an excellent convergence behavior and is straightforward to implement. For the spheroid-corner distance calculation we modify the ellipsoid-ellipsoid algorithm as depicted in Fig. 3(a). Instead of changing the position of both balls only one ball evolves inside the ellipsoid. The center of the other ball coincides with the considered corner and remains fixed throughout the iteration process. If the query point is inside the ellipsoid, the advancing ball is positioned outside the ellipsoid[5].

The presented approach for the generation of matrix-inclusion RVEs is implemented into Python. For demonstration purposes we prescribe a fixed aspect ratio $\alpha = \alpha_{max} = \alpha_{min} = 5$ as well as uniform distributions $\mathcal{P}(\Omega) = \mathcal{S}(\Omega) = \mathcal{T}(\Omega) = \mathcal{U}(\Omega)$ for generating the location and orientation of the spheroids[6]. Furthermore, the overall RVE dimensions are exemplarily set to $a_x = a_y = a_z = 1$ resulting in a cubeshaped volume. The target inclusion volume fraction is set to $f = 20\%$. Three microstructures with 10, 50 and 150 inclusions are depicted in Fig. 4.

[2] For a packing of spheres with equal radii, [6] proposed an asymptotic volume fraction of 38% using a random sequential absorption method.

[3] The Lagrangian multipliers of problem (9) lead to the following expression $\lambda_{0,1} = \pm\sqrt{\frac{\boldsymbol{n}^{\mathrm{T}}\boldsymbol{A}\cdot\boldsymbol{n}}{\boldsymbol{b}^{\mathrm{T}}\boldsymbol{A}^{-1}\cdot\boldsymbol{b}-4c}} = \pm\frac{\sqrt{\boldsymbol{n}^{\mathrm{T}}\boldsymbol{A}\cdot\boldsymbol{n}}}{2}$. With $\boldsymbol{A}$ being positive definite and $|\boldsymbol{n}| \neq 0$ by definition, this always leads to two distinct solutions.

[4] If a spheroid intersects with two planes (near edges) 3 copies are needed. On the other hand, if a spheroid is placed in the corner of the RVE it intersects with 3 planes and, hence, 7 copies are required to maintain periodicity.

[5] If the query point lies inside the ellipsoid (cf. Fig. 3(a), right), the evolving ball is located on the ellipsoids outside. The space surrounding the ball is non-convex (complement of three-dimensional space and ellipsoid), and thus convergence problems might occur. By choosing the starting point of the iteration adroitly global convergence is ensured. Therefore, we compute the penetration points of the line defined by the ellipsoid midpoint and the query point and select the closer one, with respect to the query point, as the initial guess.

[6] Particular attention needs to be paid to the generation of the angles to establish a uniform orientation distribution, thus, leading to $\theta = \mathrm{acos}(2\mathcal{U}(\Omega) - 1)$.

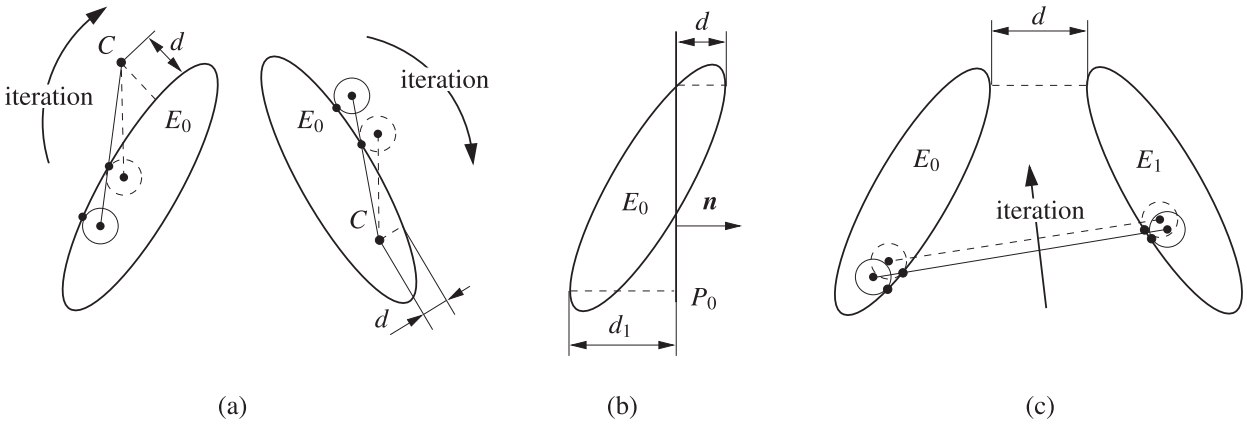


Fig. 3. Two dimensional schematic representation of determining minimum distances d between a spheroid E_0 and: (a) a point (e.g., E_0 and a corner C outside of E_0 or E_0 and a corner C inside of E_0), (b) a plane P_0 and (c) a spheroid E_1.

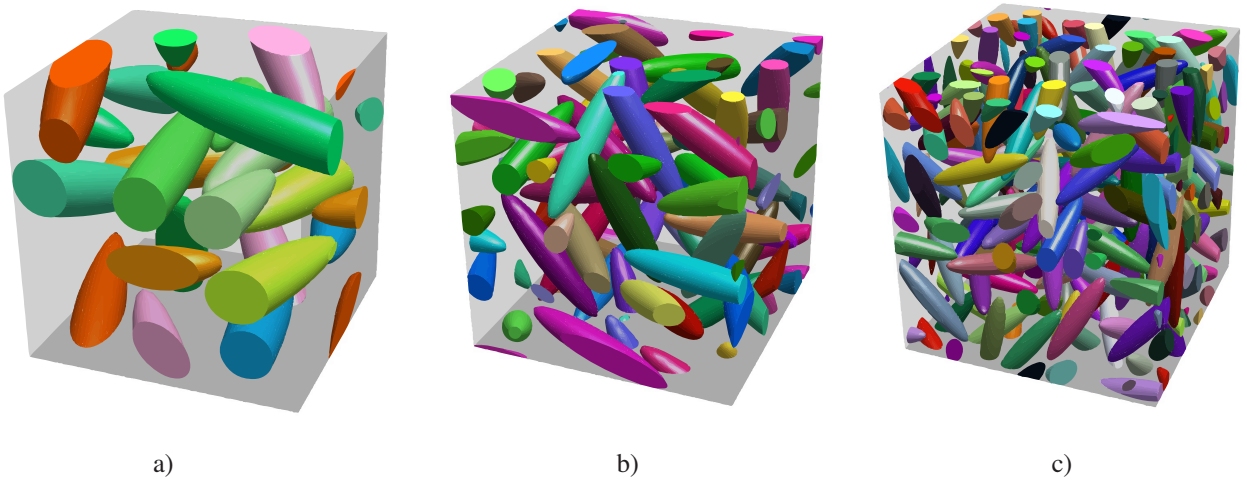


Fig. 4. Periodic microstructures with an inclusion volume fraction of 20% with (a) 10, (b) 50 and (c) 150 spheroidal inclusions. The periodically continued parts of inclusions intersecting the RVE boundary exhibit identical colors.

3. Meshing procedure

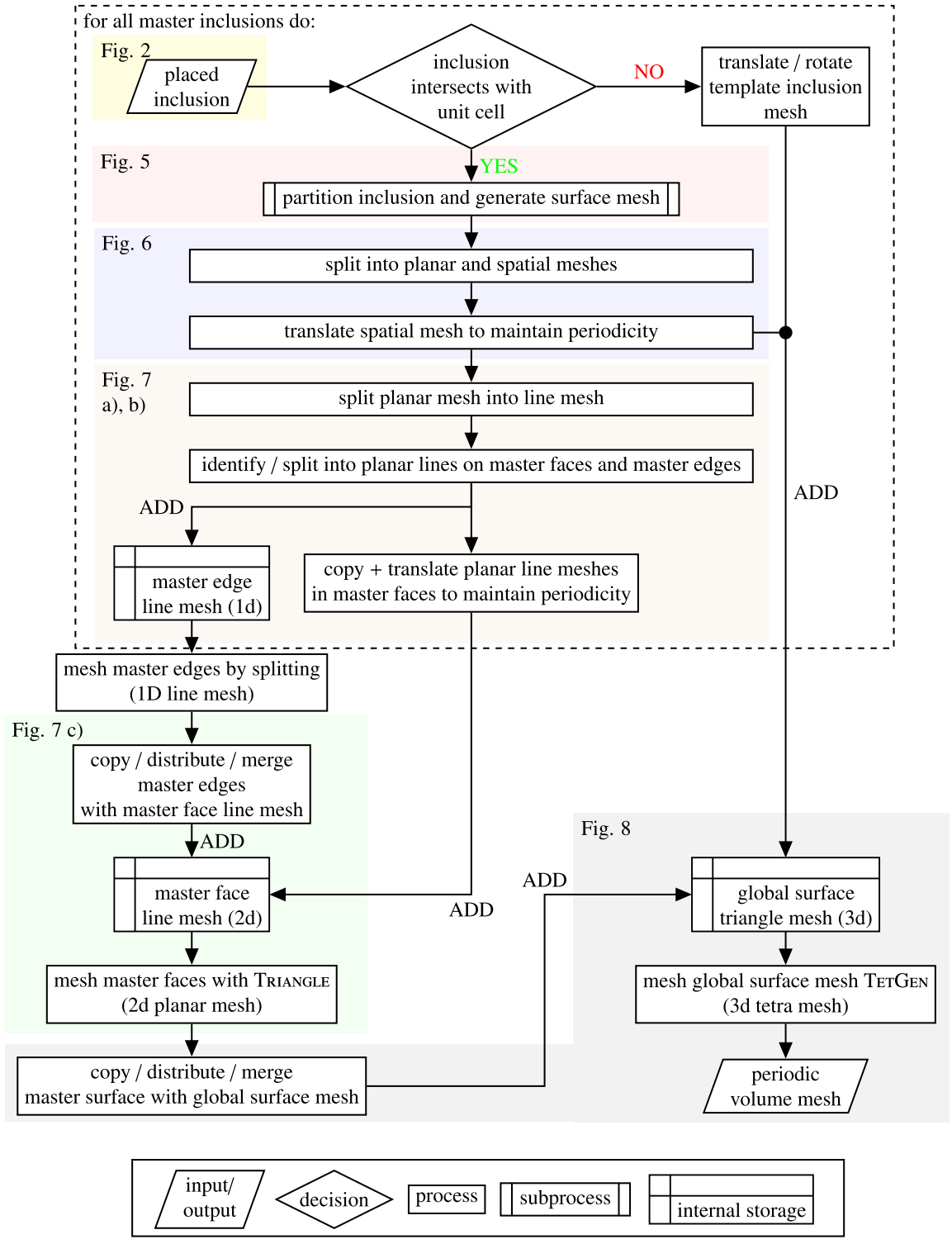
In this section, we present a novel and efficient meshing approach to obtain a high quality periodic discretization of matrix-inclusion microstructures with as few elements as possible. Algorithm 2 depicts the corresponding flow chart. With the findings of Section 2 as an input to the algorithm, the core idea is to process the geometric information successively. One by one, a surface mesh of each individual spheroid is generated. Respecting potential unit cell-inclusion-intersections, the meshes are split and distributed (translated) around the RVE in accordance to periodicity requirements. Following a hierarchical meshing procedure similar to [10], the final discretization is built starting with corners to edges over faces and, finally, ending with the volume mesh. Within the process master edges and faces are generated and copied, hence serving as constrains for the triangulations (master face mesh) and, final tetrahedralization (volume mesh). By utilizing individual source meshes emanating from the individual inclusions, mesh periodicity is automatically obtained. We implement the algorithm in Python allowing a straightforward incorporation of third party libraries via scripts. Throughout the whole algorithm mesh information is split into geometric information of the nodes (positions of the nodes) and topology information in terms of element connectivity tables which significantly increases numerical stability[7].

In the following, we illustrate the developed approach on basis of an example for a detailed description. Fig. 5(a)[8] shows a microstructure which consists of one inclusion having its center at $\boldsymbol{x}_c = [0,\ 0.5,\ 0]^T$ in a cubic unit cell with an edge length of $a_x = a_y = a_z = 1$. Periodicity of the mesh topology is an overall goal, which necessitates the surface triangulations of opposing RVE faces to coincide. We enforce this condition by assigning P_{x_0}, P_{y_0} and P_{z_0} to master faces. The generated meshes on these faces are later copied to their opposing counterparts P_{x_1}, P_{y_1} and P_{z_1}. The creation of master faces requires a waterproof source line mesh, emanating from the boundary of the inclusions and the unit cell edges. At first the processed inclusion is identified as a master or slave inclusion. Master inclusions are completely inside the RVE or intersect solely with one or more of the master faces. They exclusively act as the source for the further meshing process. If a master inclusion is completely inside the RVE, a template surface mesh of the inclusion is translated and rotated according to its properties. In our example, the inclusion intersects with planes P_{x_0} and P_{z_0},

[7] Queries and manipulations on the geometric data (node coordinates) are critical in terms of finite precisions of a computer. By splitting the mesh information those operations are reduced to a minimum.

[8] For demonstration purpose the three copies, demanded by periodicity, are not displayed.

Algorithm 2
Automatic generation of a high quality periodic mesh with a small number of elements.



qualifying it as a master inclusion. Therefore, three copies (slave inclusions) are required to ensure periodicity. Their centers are shifted with respect to the master inclusion center. Three translational shift vectors given by

$$\boldsymbol{v}_1 = [1\ 0\ 0]^{\mathrm{T}}, \quad \boldsymbol{v}_2 = [0\ 0\ 1]^{\mathrm{T}}, \quad \boldsymbol{v}_3 = [1\ 0\ 1]^{\mathrm{T}}, \tag{11}$$

are used to compute slave inclusion centers via $\boldsymbol{x}_{c_i} = \boldsymbol{x}_c + \boldsymbol{v}_i\ , \quad i \in \{1, 2, 3\}$. Next, partitioning of the master inclusion with respect to the intersection planes follows, resulting in four parts, as shown in Fig. 5(b). The emerged pieces relate to the volumes of the combined set of master and slave inclusions being inside the RVE. Thereafter, a surface mesh of linear triangles of the fragmented structure is generated as shown in Fig. 5(c).

Compatibility of the mesh at leading edges is vital for the desired mesh periodicity. Since intersection entities can be expressed

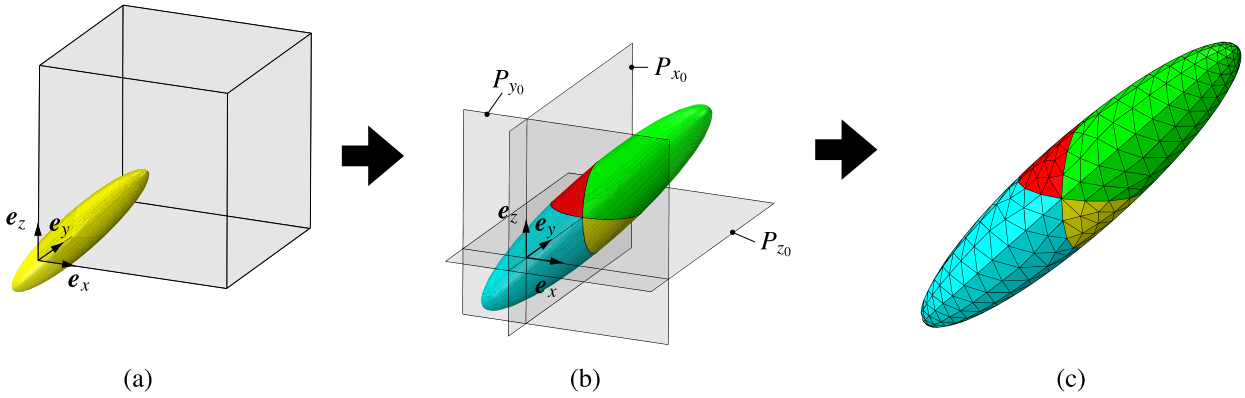
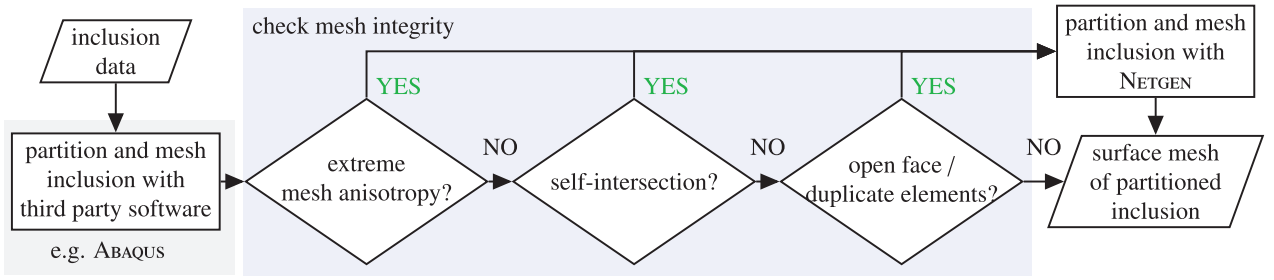


Fig. 5. Procedure of placement and meshing of the master inclusion: (a) random placement of the master inclusion, (b) partitioning by boolean operations and (c) resulting surface mesh.

Algorithm 3
Generation of surface meshes of a partitioned inclusions.



analytically only for simple geometries (e.g., spheres, polyhedra) and meshing of curved surfaces is non-trivial, we employ third party softwares for this task. Modern finite element packages offer the capabilities of modeling and meshing complicated geometries[9]. By means of boolean operations emanating from the field of constructive solid geometry, the dividing process is realized followed by a preferably isotropic surface triangulation. In some cases small volumes or sharp angles are present and lead to degenerated surface meshes that are highly anisotropic, posses self-intersections or are non-waterproof (featuring of open faces). To capture these possible errors, a mesh integrity check follows each surface triangulation as shown in Algorithm 3[10].

It is found that NETGEN [27] provides the most robust mesh generator for our purposes[11]. However, due to the employed advancing front algorithm, the resulting meshes are very dense and therefore, under the viewpoint of obtaining as few elements as possible, NETGEN is not our choice[12]. In the first step a commercial software, e.g. ABAQUS, is used to create a reasonable dense mesh and NETGEN acts as a guarantee in case of mesh degeneration to eventually obtain a linear surface mesh consisting of planar and curved regions, ready for further processing.

In the next step, the surface mesh is split and distributed across the RVE as shown in Fig. 6. The triangulated master inclusion mesh is non-manifold and firstly separated into its components, which are the planar and spatial sections, cf. Fig. 6(a). To distinguish the planar triangle elements on planes from spatial ones we operate on the geometric information of the nodes (***nodes***[...]) by using a numerical tolerance[13] δ via

$$
\begin{aligned}
P_{x_0}P_{y_0}^{+}P_{z_0}^{+}&: \begin{cases} |\boldsymbol{node}[i]\cdot\boldsymbol{n}_{x_0}| \le \delta \wedge \\ \boldsymbol{node}[i]\cdot\boldsymbol{n}_{y_0} \le \delta \wedge \boldsymbol{node}[i]\cdot\boldsymbol{n}_{z_0} \le \delta \end{cases} \\
P_{x_0}P_{y_0}^{-}P_{z_0}^{-}&: \begin{cases} |\boldsymbol{node}[i]\cdot\boldsymbol{n}_{x_0}| \le \delta \wedge \\ \boldsymbol{node}[i]\cdot\boldsymbol{n}_{y_0} \ge \delta \wedge \boldsymbol{node}[i]\cdot\boldsymbol{n}_{z_0} \ge \delta \end{cases} \\
P_{z_0}P_{x_0}^{+}P_{y_0}^{+}&: \begin{cases} |\boldsymbol{node}[i]\cdot\boldsymbol{n}_{y_0}| \le \delta \wedge \\ \boldsymbol{node}[i]\cdot\boldsymbol{n}_{x_0} \le \delta \wedge \boldsymbol{node}[i]\cdot\boldsymbol{n}_{y_0} \le \delta \end{cases} \\
P_{z_0}P_{x_0}^{-}P_{y_0}^{-}&: \begin{cases} |\boldsymbol{node}[i]\cdot\boldsymbol{n}_{z_0}| \le \delta \wedge \\ \boldsymbol{node}[i]\cdot\boldsymbol{n}_{x_0} \ge \delta \wedge \boldsymbol{node}[i]\cdot\boldsymbol{n}_{y_0} \ge \delta \end{cases}
\end{aligned}
\tag{12}
$$

with $i = \{1..n_{nodes}\}$ and n_{nodes} being the total number of nodes. The element sets $P_{x_0}P_{y_0}^{+}P_{z_0}^{+}$, $P_{x_0}P_{y_0}^{-}P_{z_0}^{-}$, $P_{z_0}P_{x_0}^{+}P_{z_0}^{+}$ and $P_{z_0}P_{x_0}^{-}P_{y_0}^{-}$ impose constraints on the later generated master face meshes and are shifted accordingly as shown in Fig. 6(b). Analogously to the slave inclusions and their shift vectors the appropriate regions of the spatial mesh are translated according to Eq. (11) and contribute to the interior of the global surface mesh, see Fig. 6(c).

Before the hierarchical meshing procedure commences, all planar element sets on all master faces are transformed into a line mesh of the exterior edges as shown in Fig. 7(a)[14]. This results

[9] Even for a simple geometry like a spheroid the intersection edges and surface with respect to planes are complex and must be computed numerically [27].

[10] The mesh anisotropy is checked by a heuristic threshold of the ratio of maximum to minimum element area $a_{max}/a_{min} > 250$. The self-intersection test and the open face check is conducted by utilizing the options `-d` and `-V` in TETGEN [30, 31] and verifying its output.

[11] NETGEN uses an implicit function representation of geometric primitives resulting in accurate and robust boolean operations.

[12] Commercial softwares like COMSOL or ABAQUS represent the geometry in a different way to NETGEN, which may result in errors concerning the boolean operations. On the other hand, they employ more sophisticated meshing methodologies which produce more anisotropic meshes leading to fewer elements. However, in some cases the resulting mesh is degenerated and shows a high mesh anisotropy, self-intersections or is open. These faults result from degenerated partitions that produce very small volumes or sharp angles, resulting in nodes treated as coincident due to the finite precision of computers.

[13] Choosing δ correctly is a demanding challenge due to the finite precision in computers. For our purposes, we found $\delta = 10^{-9}$ to be an adequate choice.

[14] This operation is conducted by working on the element connectivity information.

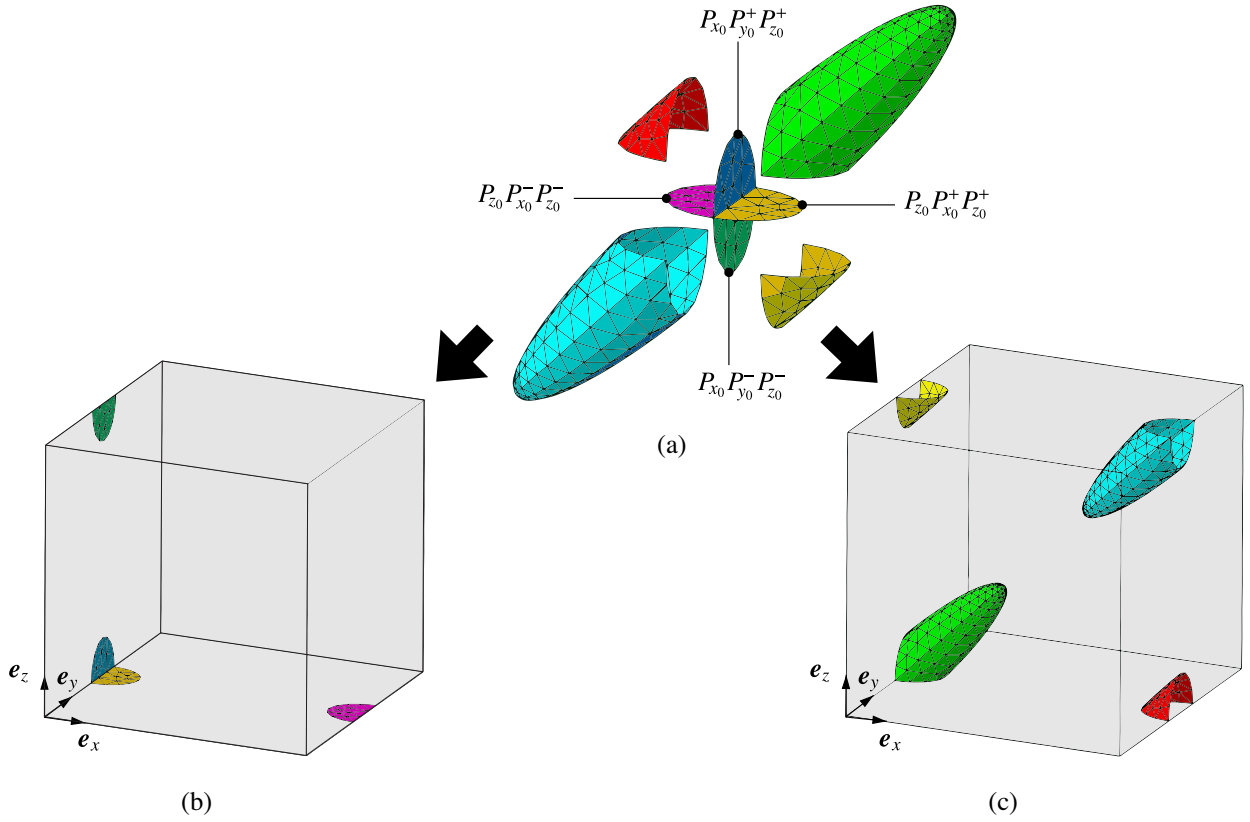


Fig. 6. Splitting and distribution process of partitioned master inclusion surface mesh: (a) split surface mesh, (b) partitioned master face mesh and (c) interior surface mesh.

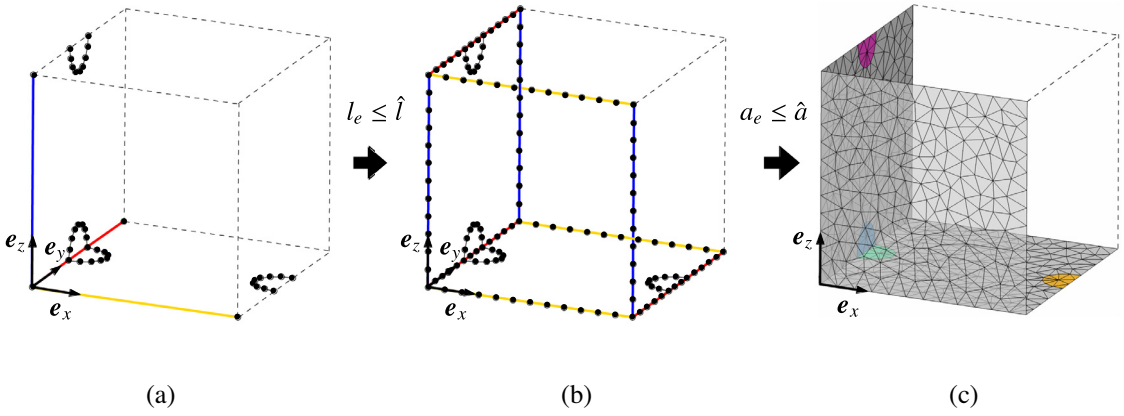


Fig. 7. Meshing of master edges and faces: (a) finding master edge intersection points and generating master face line mesh, (b) mesh master edges and copy them around master faces and (c) mesh master faces. The meshing process is triggered by a maximum line length $\hat{l}$ and a maximum element area $\hat{a}$.

in a two dimensional skeleton line mesh on the individual master faces emanating from the inclusions and, thus, conforming with the spatial triangulation of Fig. 6(c). For obtaining the desired periodic mesh, opposite edges of master faces must have an identical discretization. To this end, we define the master edges $L_{x_0y_0}$ (blue), $L_{y_0z_0}$ (yellow) and $L_{x_0z_0}$ (red) which are copied before the master face meshing is conducted, cf. Fig. 7(a) and (b). Intersection points between the inclusions and the master edges need to be incorporated into the master edge mesh. An identification of these points follows a similar approach to Eq. (12), employing a threshold δ via

$$\begin{aligned} L_{x_0y_0}: &\quad |\boldsymbol{node}[i]\cdot\boldsymbol{n}_{x_0}| \leq \delta \quad \wedge \quad |\boldsymbol{node}[i]\cdot\boldsymbol{n}_{y_0}| \leq \delta, \\ L_{y_0z_0}: &\quad |\boldsymbol{node}[i]\cdot\boldsymbol{n}_{y_0}| \leq \delta \quad \wedge \quad |\boldsymbol{node}[i]\cdot\boldsymbol{n}_{z_0}| \leq \delta, \\ L_{x_0z_0}: &\quad |\boldsymbol{node}[i]\cdot\boldsymbol{n}_{x_0}| \leq \delta \quad \wedge \quad |\boldsymbol{node}[i]\cdot\boldsymbol{n}_{z_0}| \leq \delta. \end{aligned} \tag{13}$$

In our example the master edge $L_{x_0z_0}$ (red) is divided into three partitions by the inclusion. Before the line meshing commences the hitherto described process is repeated for all inclusions.

Next, all master edges are divided by a line splitting algorithm, which divides all lines with a length larger than a predefined maximum length $\hat{l}$. The generated line meshes are copied (two times per edge) to the slave edges, as depicted in Fig. 7(b). For the subsequent face meshing a fully enclosed line mesh is required. Therefore, all duplicate nodes of the entire line mesh (the combination of the skeleton line meshes of the inclusions and the master edge meshes) are merged by joining all node pairs within a distance smaller than a threshold δ. Finding all these pairs is a cumbersome topic and can be treated by nearest-neighbor-lookup algorithms. Our algorithm incorporates a k-d tree by utilizing the `scipy.spatial.cKDTree` class from the `scipy` library [16],

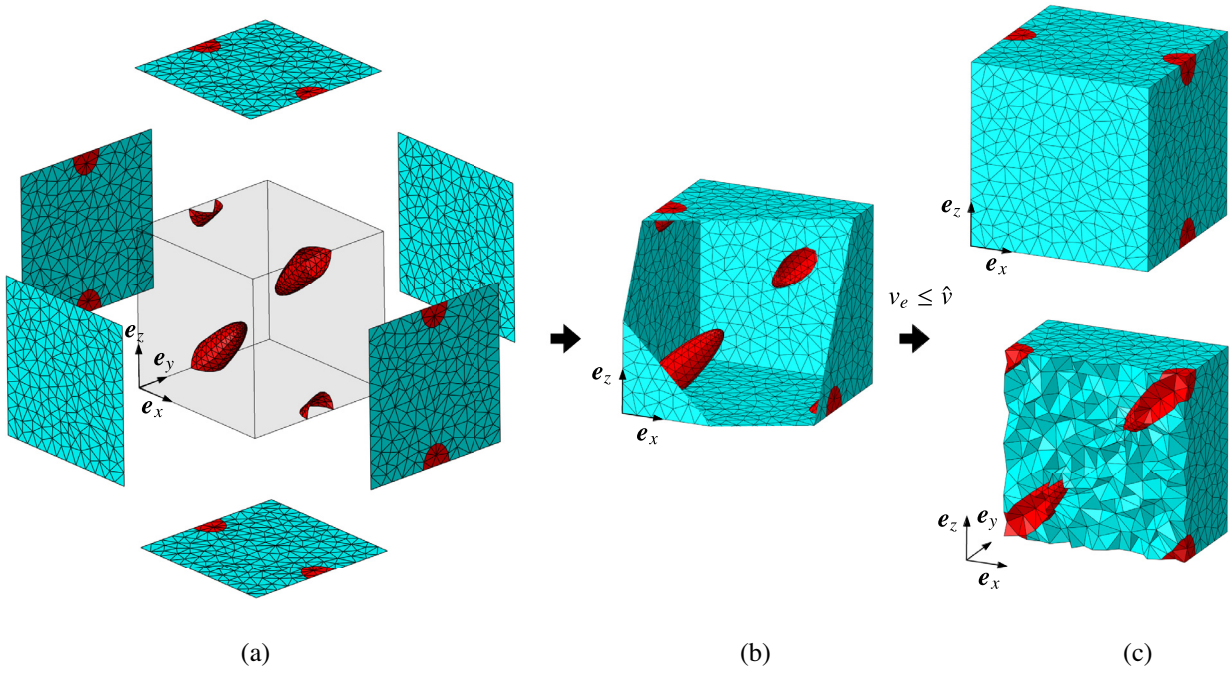


Fig. 8. Merging surface meshes and generating three-dimensional constrained tetrahedralization: (a) assembly and merge of master faces and interior mesh, (b) closed (waterproof) three-dimensional surface mesh and (c) final three-dimensional tetrahedral mesh as well as a cut through the unit cell. Volume meshing is triggered by specifying a maximum element volume $\hat{v}$.

allowing very fast point queries[15]. Identification, removal and reindexing of the nodes and their corresponding element connectivity entries eventually results in the desired enclosing line mesh which serves as input for master face triangulations. Employing the very robust and fast `Triangle` library from [29] all three master faces are meshed via constrained Delaunay triangulations[16]. By defining a maximum triangle area $\hat{a}$ we trigger the mesh size and promote an isotropic mesh. Fig. 7(c) depicts the resulting discretized master faces of the example microstructure.

Finally, the algorithm copies all master face meshes to their opposing sides (P_{x_1}, P_{y_1} and P_{z_1}), cf. Fig. 8(a). Together with the stored interior surface triangles of the inclusions they form the input for the volume meshing. Again, duplicated nodes are merged by the described k-d tree approach, resulting in a waterproof surface mesh, cf. Fig. 8(b). Subsequently, a constrained Delaunay tetrahedralization is conducted using the robust and fast software `TetGen` [30, 31][17]. By constraining bounding triangles, the surface mesh is preserved, resulting in the required mesh periodicity consisting of quadratic tetrahedral elements, cf. Fig. 8(c). By means of specifying a maximal element volume $\hat{v}$ the discretization process is triggered, supporting the formation of an isotropic mesh. Since inclusions form cavities in the input surfaces mesh, `TetGen` automatically assigns element sets, which can be used for material section assignment. For triggering the total number of elements, the length $\hat{l}$, area $\hat{a}$ and volume $\hat{v}$ restrictions in the meshing procedure are related via the edge length of an equilateral tetrahedron and triangle [10] through

$$\hat{v} = \frac{V_{RVE}}{n_{ele}}\,, \quad \hat{l} = \sqrt[3]{\frac{12}{\sqrt{2}}\hat{v}}\,, \quad \hat{a} = \frac{\sqrt{3}}{4}\hat{l}^2\,. \tag{14}$$

to a favored total number of elements n_{ele} and the RVE volume V_{RVE}.

In the presented form the algorithm prodcues isotropic meshes. In some cases, however, it might be of interest to refine the discretization in regions where high gradients would be anticipated, e.g. at material interfaces. By utilizing a finer surface triangulation of the inclusion surfaces this can be achieved. The subsequent constrained Delaunay triangulation via `TetGen` has then to be triggered by a larger intended tetrahedron volume assuring a smooth transition from the fine surface meshes to intended coarser mesh regions of the bulk volume.

4. Examples and comparison

In this section we examine the effectiveness and applicability of the proposed method. Regarding the limited capabilities of computers the total number of elements cannot be arbitrarily large. On the other hand, a fine discretization captures the geometry more precisely and highly distorted elements, which may lead to error-prone numerical simulations, can be circumvented. Therefore, a proper tradeoff between quality demands and computation time is always difficult to find. From that perspective, it is of major importance to quantify the mesh quality in an objective manner. Among many element quality criteria, and the general uncertainty of a most meaningful one, we analyze the shape factor $\eta = v/v_{opt}$, relating the element volume v to the optimal volume v_{opt} of an equilateral tetrahedron with the same circumradius, to evaluate the resulting discretizations.

In this regard, example meshes generated via NETGEN, a software library which is capable of producing periodic meshes automatically and, therefore, considered as state of the art, and the proposed algorithm are analyzed in the following. We create all

[15] The `cKDTree.query_pairs` function is used for the pair search.

[16] The triangulation is separated into two steps. Firstly a coarse constrained triangulation is conducted (`Triangle` option `-pYYVa`). This serves as input for the second meshing procedure resulting in high quality elements (`Triangle` option `-pqrjYYVa`).

[17] Similar to the two-dimensional triangulation the three-dimensional tetrahedralization is separated into two stages. Firstly, a coarse mesh serving as the input of the finer high quality mesh is generated. The corresponding `TetGen` options are `-pYQA` and `-pqYQArao2`.

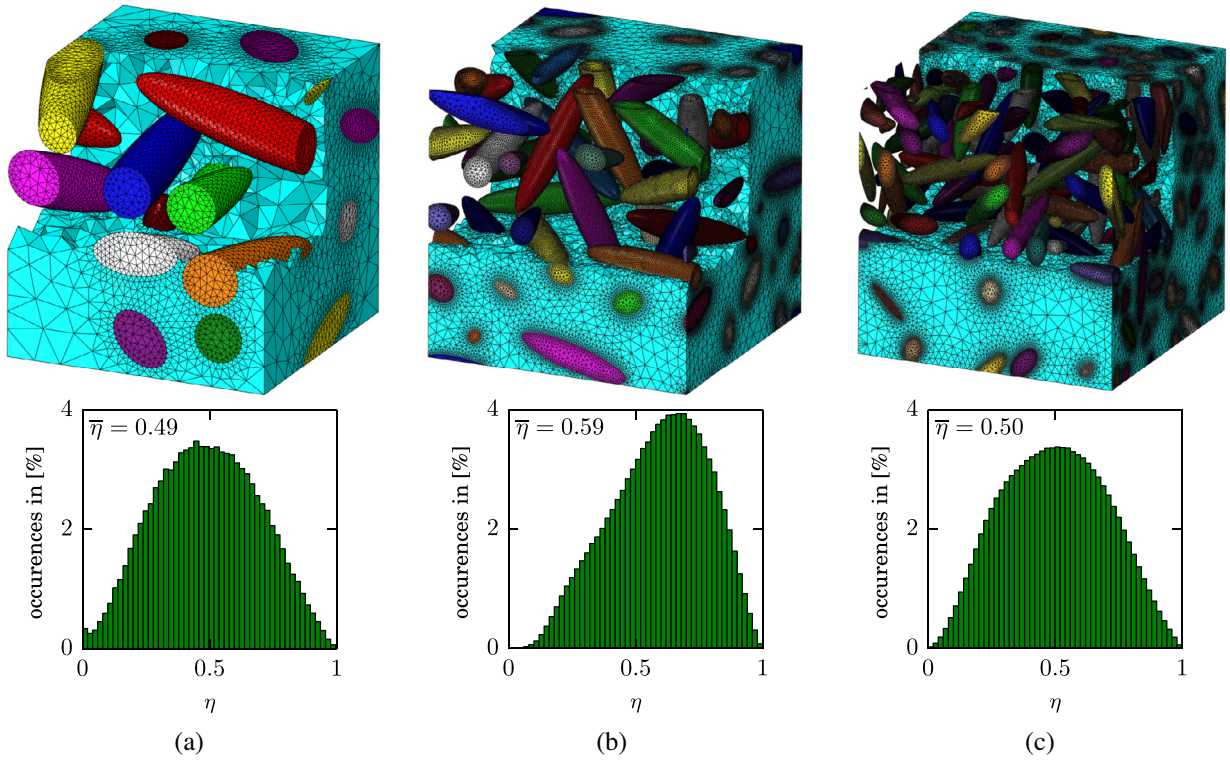


Fig. 9. Three example meshes of the microstructures from Fig. 4 generated with NETGEN, featuring a volume fraction of approximately 20% together with mesh statistics featuring the average element shape factor $\overline{\eta}$: (a) 10 inclusions with 144214 elements (b) 50 inclusions with 1966106 elements and (c) 150 inclusions with 7314188 elements. The periodically continued parts of inclusions intersecting the RVE boundary exhibit identical colors.

meshes under the premises of achieving a small number of elements but maintaining a high overall mesh quality[18]. Taking discretizations emanating from NETGEN as a reference a comparison to our method is conducted.

4.1. Mesh generation with NETGEN software library

Fig. 9 shows discretizations of the three example microstructures from Fig. 4 meshed with NETGEN. To obtain as few elements as possible the mesh granularity option in NETGEN is chosen to *very coarse* or *coarse*, depending which meshing process finished successfully. Additionally mesh statistics, featuring a frequency distribution of the shape factor η as well as the average shape factor $\overline{\eta}$, are presented. All discretizations show an excellent shape factor. However, the average shape factor as a global measure on its own is not sufficient for mesh validation, since already few badly shaped elements may ruin a proper finite element simulation due to numerical problems. Due to inclusion-RVE intersections and their periodic continuations, unwanted volumetric shapes are unavoidable and thus, poorly shaped elements usually occur. From the element shape factor frequency distributions of the different meshes we deduce the trend of finer meshes leading to a decrease of poor elements.

4.2. Mesh generation with our newly developed method

Fig. 10 depicts discretizations and mesh statistics of the three example microstructures (see Fig. 4) generated with our novel method. Again, excellent average shape factors are observed in combination with an increase in high quality elements for finer meshes. Additional distance queries and thresholds between inclusions and the RVE boundary, as described in Section 2, provoke the formation of good-natured volumes leading to a superb mesh quality.

4.3. Comparison

RVEs with NETGEN show mesh refinements at the inclusion interfaces. It is not possible to generate equivalently isotropic meshes with NETGEN, which might be attributed to the advancing front meshing algorithm and its implementation concering the rules of insertion of new triangles on the ellipsoid surfaces, cf. [27]. RVEs generated with our method reveal a more isotropic element size distribution. Analyzing the presented discretizations in terms of mesh statistics we observe no significant differences between both approaches. As depicted in Table 1 average shape factors and relative frequencies of $\eta \leq 10^{-4}$, as a quantitative measure of poor elements, are of similar size. Our method reveals a slight benefit regarding microstructure appoximation for sample (a) and (c) of Fig. 4. The significant improvement of our approach emerges with respect to the total number of elements n_{el}. Already from the visual examination a highly increased amount of elements resulting in a very dense mesh is observable for the NETGEN discretizations. Theses meshes have a limited applicability with respect to finite element simulations in terms of a manageable computation time due to their enormous size. In case that many inclusions are needed for a RVE, it is almost impossible to conduct simulations with a mesh generated by NETGEN, especially if the underlying model captures many physical effects and, therefore, is of a certain complexity. In contrast, our method reduces the total number of elements by a factor up to 10, while still maintaining excellent mesh quality.

Another aspect worth mentioning aims at the approximate geometric representation of the microstructure through the mesh. Due to the curved nature of the inclusions their boundaries cannot be

[18] We consider a mesh to be sufficiently good, if an average element shape factor of $\overline{\eta} \gtrsim 0.5$ is maintained.

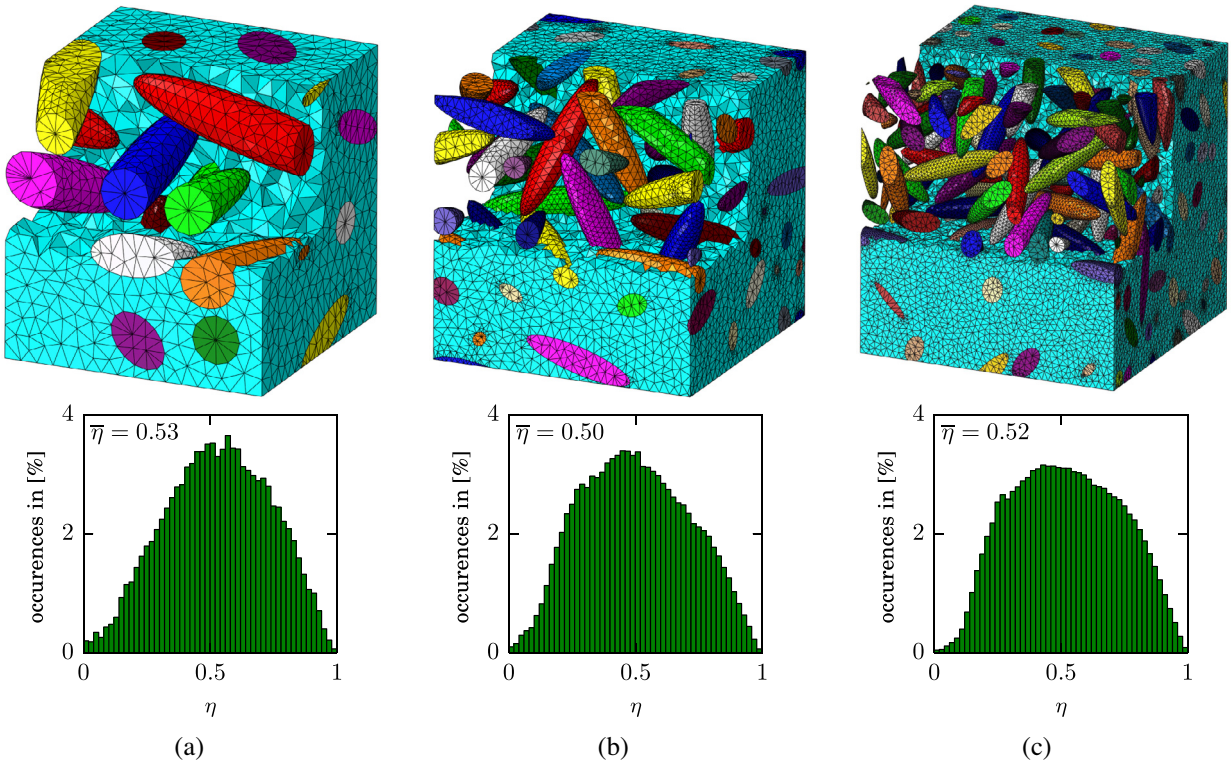


Fig. 10. Three example meshes of the microstructures from Fig. 4 with a volume fraction of approximately 20% together with mesh statistics featuring the average element shape factor $\overline{\eta}$: (a) 10 inclusions with 42211 elements (b) 50 inclusions with 199955 elements and (c) 150 inclusions with 732992 elements. The periodically continued parts of inclusions intersecting the RVE boundary exhibit identical colors.

Table 1
Comparison of our method to NETGEN with respect to the example microstructures with different number of inclusions and an inclusion volume fraction of $f = 0.2$.

	10 inclusions (Fig. 4(a))				50 inclusions (Fig. 4(b))				150 inclusions (Fig. 4(c))			
	$\overline{\eta}$	$\eta \leq 10^{-4}$	n_{ele}	f	$\overline{\eta}$	$\eta \leq 10^{-4}$	n_{ele}	f	$\overline{\eta}$	$\eta \leq 10^{-4}$	n_{ele}	f
Alg. 2	0.53	2×10^{-2} %	42211	0.192	0.50	2×10^{-3} %	199955	0.192	0.52	0 %	732992	0.193
NETGEN	0.49	4×10^{-2} %	144214	0.198	0.59	2×10^{-4} %	1966106	0.199	0.50	4×10^{-4} %	7314188	0.199

described by tetrahedrons exactly. Additionally, the convexity of the considered inclusions always implicates a discretization with a reduced inclusion volume fraction. As shown in Table 1 the volume fraction is slightly smaller than $f = 0.2$ for both approaches. Due to the larger number of elements NETGEN is slightly closer to the exact volume fraction. However, a minor adjustment in the described approach of Section 2, by increasing the volume fraction f in accordance to the assumed lost amount due to meshing, resolves this problem.

5. Summary

In the present work we propose an algorithm capable of automatically generating and meshing three-dimensional, random, non-overlapping matrix-inclusion RVEs to be used in finite element simulations. We maintain a periodic topology of the microstructure as well as the discretization throughout the whole process, allowing a straightforward application of periodic boundary conditions. The randomized, periodic microstructure itself is generated via a random sequential absorption method. Key features of this algorithm are distance queries between the inclusions themselves and between the RVE boundary and inclusions. The meshing process follows a hierarchical discretization approach starting from points, over line meshes and face meshes to the final tetrahedralization. Differentiating between master and slave inclusions in combination with a successive processing of the geometric data, the algorithm proves itself as very effective.

By choosing a very general approach, an adjustment to all types of inclusions without any restrictions is possible. The proposed algorithm is extremely efficient, very robust and automized. Further, it produces high quality meshes while keeping the total number of elements at a manageable minimum under computational viewpoints. Thus, it is applicable to high volume concentrations, a large number of inclusions (incorporating extreme aspect ratios) and complicated inclusion geometries, while still maintaining a high quality mesh.

Acknowledgement

Financial support by the German Research Foundation (DFG) via SFB 986 “M3” (project B6) is gratefully acknowledged.

References

[1] Bailakanavar M, Liu Y, Fish J, Zheng Y. Automated modeling of random inclusion composites. Eng Comput 2014;30(4):609–25.
[2] Böhm H. J. A short introduction to basic aspects of continuum micromechanics 1998.
[3] Böhm HJ, Eckschlager A, Han W. Multi-inclusion unit cell models for metal matrix composites with randomly oriented discontinuous reinforcements. Comput Mater Sci 2002;25(1):42–53.

[4] Böhm HJ, Han W. Comparisons between three-dimensional and two-dimensional multi-particle unit cell models for particle reinforced metal matrix composites. Model Simul Mater Sci Eng 2001;9(2):47.
[5] Brassart L, Doghri I, Delannay L. Homogenization of elasto-plastic composites coupled with a nonlinear finite element analysis of the equivalent inclusion problem. Int J Solids Struct 2010;47(5):716–29.
[6] Cooper DW. Random-sequential-packing simulations in three dimensions for spheres. Phys Rev A 1988;38:522–4.
[7] de Andrade Silva F, Williams JJ, Müller BR, Hentschel MP, Portella PD, Chawla N. Three-dimensional microstructure visualization of porosity and Fe-rich inclusions in SiC particle-reinforced Al alloy matrix composites by X-ray synchrotron tomography. Metall Mater Trans A 2010;41(8):2121–8.
[8] Dirrenberger J, Forest S, Jeulin D. Towards gigantic RVE sizes for 3D stochastic fibrous networks. Int J Solids Struct 2014;51(2):359–76.
[9] Philip S, Eberly DH. Geometric tools for computer graphics. Morgan Kaufmann; 2002.
[10] Fritzen F, Böhlke T. Periodic three-dimensional mesh generation for particle reinforced composites with application to metal matrix composites. Int J Solids Struct 2011;48(5):706–18.
[11] Ghossein E, Lévesque M. A fully automated numerical tool for a comprehensive validation of homogenization models and its application to spherical particles reinforced composites. Int J Solids Struct 2012;49(11):1387–98.
[12] Ghossein E, Lévesque M. Random generation of periodic hard ellipsoids based on molecular dynamics: a computationally-efficient algorithm. J Comput Phys 2013;253(0):471–90.
[13] Groeber M, Haley B, Uchic M, Dimiduk D, Ghosh S. 3D reconstruction and characterization of polycrystalline microstructures using a FIB–SEM system. Mater Charact 2006;57(4):259–73.
[14] Gusev AA. Representative volume element size for elastic composites: a numerical study. J Mech Phys Solids 1997;45(9):1449–59.
[15] Illian J, Penttinen A, Stoyan H, Stoyan D. Statistical analysis and modelling of spatial point patterns, vol. 70. Chichester: John Wiley & Sons; 2008.
[16] Jones E., Oliphant T., Peterson P. SciPy: open source scientific tools for Python. URL http://www.scipy.org 2014.
[17] Kanit T, Forest S, Galliet I, Mounoury V, Jeulin D. Determination of the size of the representative volume element for random composites: statistical and numerical approach. Int J Solids Struct 2003;40(13-14):3647–79.
[18] Kouznetsova V, Brekelmans WAM, Baaijens FPT. An approach to micro-macro modeling of heterogeneous materials. Comput Mech 2001;27(1):37–48.
[19] Lennerz C, Schomer E. Efficient distance computation for quadratic curves and surfaces. In: Geometric modeling and processing, 2002 proceedings; 2002. p. 60–9.
[20] Lin A, Han S-P. On the distance between two ellipsoids. SIAM J Optim 2002;13(1):298–308.
[21] Miehe C, Koch A. Computational micro-to-macro transitions of discretized microstructures undergoing small strains. Arch Appl Mech 2002;72(4-5):300–17.
[22] Nazarenko L, Bargmann S, Stolarski H. Influence of interfaces on effective properties of nanomaterials with stochastically distributed spherical inclusions. Int J Solids Struct 2014;51(5):954–66.
[23] Nazarenko L, Bargmann S, Stolarski H. Energy-equivalent inhomogeneity approach to analysis of effective properties of nanomaterials with stochastic structure. Int J Solids Struct 2015;59:183–97.
[24] Pierard O, Gonzalez C, Segurado J, LLorca J, Doghri I. Micromechanics of elasto-plastic materials reinforced with ellipsoidal inclusions. Int J Solids Struct 2007;44(21):6945–62.
[25] Quey R, Dawson P, Barbe F. Large-scale 3D random polycrystals for the finite element method: Generation, meshing and remeshing. Comput Methods Appl Mech Eng 2011;200(17):1729–45.
[26] Rintoul MD, Torquato S. Reconstruction of the structure of dispersions. J Colloid Interface Sci 1997;186(2):467–76.
[27] Schöberl J. Netgen an advancing front 2D/3D-mesh generator based on abstract rules. Comput Vis Sci 1997;1(1):41–52.
[28] Schröder J, Balzani D, Brands D. Approximation of random microstructures by periodic statistically similar representative volume elements based on lineal–path functions. Arch Appl Mech 2011;81(7):975–97.
[29] Shewchuk JR. Triangle: engineering a 2D quality mesh generator and delaunay triangulator. In: Applied computational geometry towards geometric engineering. Springer; 1996. p. 203–22.
[30] Si H. Tetgen: a quality tetrahedral mesh generator and three-dimensional delaunay triangulator. Technical Report 9, Weierstrass Institute for Applied Analysis and Stochastics 2006:22–3.
[31] Si H. Tetgen, a delaunay-based quality tetrahedral mesh generator. ACM Trans Math Softw (TOMS) 2015;41(2):11.
[32] Sohn D, Park JY, Cho Y-S, Lim JH, Lee H. Periodic mesh generation for composite structures using polyhedral finite elements. J Comput Struct Eng Inst Korea 2014;27(4):239–46.
[33] Stoyan D. Simulation and characterization of random systems of hard particles. Image Anal Stereol 2002;1(21):41–8.
[34] Stroeven P, Stroeven M. Assessment of packing characteristics by computer simulation. Cement Concrete Res 1999;29(8):1201–6.
[35] Torquato S. Random heterogeneous materials: microstructure and macroscopic properties, vol. 16. New York: Springer Science & Business Media; 2002.