New algorithms for virtual reconstruction of heterogeneous microstructures

Ming Yang\textsuperscript{a}, Anand Nagarajan\textsuperscript{a}, Bowen Liang\textsuperscript{a}, Soheil Soghrati\textsuperscript{a,b,*}

\textsuperscript{a}Department of Mechanical and Aerospace Engineering, The Ohio State University, USA
\textsuperscript{b}Department of Materials Science and Engineering, The Ohio State University, USA

Abstract

A new set of algorithms is introduced for the virtual reconstruction of heterogeneous material microstructures, in which morphologies of embedded particles/fibers are explicitly represented. As a pre-processing phase, a shape library containing morphologies of heterogeneities, parameterized in terms of Non-Uniform Rational B-Splines (NURBS), is extracted from digital data such as micro-computed tomography images. Two packing algorithms are then introduced to reconstruct an initial (raw) periodic microstructure: In the first approach, a set of hierarchical bounding boxes approximating particle shapes are employed to check for overlap. The second approach, which is specialized for fibrous microstructures, uses the NURBS representation of fiber centerlines during the packing process. An optimization phase is applied to build the final microstructural model, relying either on the Genetic Algorithm to selectively eliminate some of the inclusions or their sequential relocation within the raw microstructure. The objective functions of this optimization phase are designed to replicate the target statistical microstructural descriptors such as the volume fraction, size distribution, and spatial arrangement of inclusions. Several example problems are presented to show the application of these algorithms for synthesizing various heterogeneous microstructures, as well as their finite element modeling and simulation.

Keywords: Microstructure reconstruction, NURBS, Heterogeneous material, Packing algorithm, Mesh generation, CISAMR

*Corresponding author: Assistant professor of Mechanical and Aerospace Engineering & Materials Science and Engineering, The Ohio State University, 201 W. 19th Avenue, Columbus, OH 43210, USA.
Email address: soghrati.1@osu.edu (Soheil Soghrati)
1. Introduction

The computational design of composite materials requires predicting the micromechanical behavior of multiple repeating unit cells (RUCs) of the material with various microstructural features. One of the main challenges involved in this process is the creation of realistic geometrical models of the microstructure and more importantly virtually changing its characteristics such as the volume fraction and spatial arrangement of embedded particles/fibers throughout the design process. To accomplish this, one could convert digital data such as Scanning Electron Microscopy (SEM) or micro-computed tomography (micro-CT) images into geometrical models [1, 2]. For example, the characterization and direct reconstruction of 3D microstructures of a fiber-reinforced polymer from micro-CT data are presented in [3]. However, in addition to the fact that the preparation of imaging data (specially in 3D) is time-consuming and expensive, several challenges are often encountered during the image processing phase due to difficulty in identifying material interfaces, which emanates from low contrast between different phases or their close proximity [4]. More importantly, direct microstructure reconstruction from imaging data can only provide a single microstructural model of an existing material, which would not be useful for the computational design of a new materials system.

Alternatively, one can implement a reconstruction algorithm [5] to virtually create realistic microstructures of composite materials. Significant research has been conducted to address the need for synthesizing material microstructures based on geometrical and statistical data extracted from 2D/3D imaging data. Such methods can be categorized into two main groups: descriptor-based methods [6, 5, 7] and correlation function-based techniques [8, 9, 10, 11]. In the former approach, the goal is to replicate descriptor functions such as those describing the spatial arrangement of heterogeneities (e.g., nearest neighbor distance function), their morphologies, and the volume fraction. Optimization algorithms such as the Genetic Algorithm (GA) [12, 13] and Simulated Annealing (SA) [14, 15, 16] are often utilized during the dispersion reconstruction phase to replicate the desired spatial distribution of inclusions. However, using such iterative algorithms often lead to a computationally demanding reconstruction process for synthesizing complex 3D microstructures.
Correlation function-based approaches such as N-point statistics [9] have also been implemented to reconstruct virtual microstructures for a wide array of heterogeneous materials [14, 17, 18]. Various techniques have been adopted for creating the virtual microstructure in these methods, among which we can mention the random sequential adsorption (RSA) [19, 20, 21], nearest neighbor algorithm (NNA) [22, 23], Voronoi tessellations (VT) [24, 25, 26, 27], and random field-based methods [28, 29, 11, 30]. Unlike NNA and VT, RSA cannot reconstruct microstructures with high volume fractions of embedded inclusions due to the jamming limit [31]. However, all three algorithms are not capable of handling arbitrary-shaped heterogeneities. Further, none of these methods alone can replicate all desired correlation functions, in particular when the spatial arrangement of embedded heterogeneities in the domain is nonuniform [6]. This can be compensated by implementing stochastic optimization techniques such as the Monte-Carlo methods [32, 33, 34], pixel switching [35, 36], and the mass-spring mutation operator [8, 37]. Note that some of these techniques overlook the periodicity of the resulting microstructure, which is of significant importance for imposing appropriate boundary conditions in numerical homogenization and multiscale simulations. Further, the elimination of overlapping inclusions during the optimization phase could lead to a high computational cost and a low convergence rate.

In this article, a set of integrated reconstruction algorithms are presented that relies on Non-Uniform Rational B-Splines (NURBS) for the explicit representation of morphologies of arbitrary-shaped heterogeneities. As the first step, particles/fibers are virtually packed inside the domain to build an initial (raw) periodic model of the microstructure. Two algorithms are introduced to perform this packing process: In the first approach, a set of hierarchical bounding boxes (BBoxes) are defined to approximate the morphology and determine overlaps between new and existing inclusions. The second approach, which is specialized for reconstructing fibrous microstructures, NURBS curves representing fiber centerlines are employed to identify/eliminate intersections between fibers while generating the raw microstructure. A computationally inexpensive optimization phase relying on GA and/or sequential relocation of inclusions is then utilized to transform a virtually packed raw microstructure into the final microstructural model that satisfies all target statistical descriptors.

It must be noted that the explicit description of material interfaces using NURBS in periodic microstructures reconstructed using the proposed algorithms highly facilitates the subsequent creation of conforming finite element
(FE) meshes. Thereby, these algorithms are in particular attractive for the

treatment micromechanical problems involving FE analyses. In order to fur-
ther highlight this advantage, in some of the example problems presented in
this work, a non-iterative mesh generation algorithm [38, 39] is employed to
automatically create the FE model and then simulate the micromechanical
behavior of synthesized microstructures.

The remainder of this manuscript is structured as follows: In Section
2, we describe the image processing phase to extract the shape library and
statistical descriptors needed for the virtual microstructures reconstruction.
Two new packing algorithms relying on BBox and NURBS representation
of embedded inclusions are introduced in Section 3. Two optimization al-
gorithms are then presented in Section 4, which are employed to create the
final virtual microstructure by simulating target statistical microstructural
descriptors. Several numerical examples are provided in Section 5 to demon-
strate the application of these packing and optimization algorithms for the
reconstruction of various heterogeneous microstructures. Final concluding
remarks are summarized in Section 6.

2. Image processing and statistical descriptors

2.1. Building the NURBS shape library

Before synthesizing the virtual microstructure of a given materials sys-
tem, the associated digital data (e.g., SEM or micro-CT images) must be
processed to extract the required statistical and morphological features. The
image processing phase involves several steps including the noise filtration,
smoothing, and segmentation to identify inclusion-matrix interfaces. Note
that the close proximity of inclusions and/or their small characteristic length
scales relative to the voxel size could prohibit identifying all material inter-
faces and thereby the actual morphology of several inclusions during the im-
age processing phase. This is one of the main challenges that could prohibit
the direct transformation of imaging data into appropriate microstructural
models, which is especially pronounced in 3D (e.g., when dealing with micro-
CT images). Note that a clear representation of morphologies of inclusions
in the microstructural model is highly important when the end goal is to
simulate micromechanical behaviors such as plasticity and damage, which
are often initiated along material interfaces. This is because the accurate
prediction of the initiation of such phenomena is directly affected by recov-
ered stress concentrations along material interfaces, which in turn depends
on the shape of each inclusion. The reconstruction algorithm introduced in this manuscript address this issue by virtually packing and then optimizing locations of a set of inclusions with well-defined geometries to synthesize a realistic microstructure. Thus, as a pre-processing phase, we create a shape library containing a representative subset of heterogeneities observed in the imaging data, for which interfaces can be clearly distinguished after image processing.

The morphology of each inclusion in the shape library is characterized using a Non-Uniform Rational B-Spline (NURBS) function. A NURBS curve $\mathbf{C}(u)$ is composed of $n$ B-splines of order $p$, which are functions of a parametric coordinate $u$ [40]. A set of breakpoints $U = \{u_1, u_2, u_3, ..., u_{n+p+1}\}$ is also employed to distinguish initial and end points of $\mathbf{C}(u)$, as well as its sharp corners. Note that closed NURBS curves (surfaces) with identical initial and end breakpoints can properly characterize 2D (3D) morphologies of embedded particles in a heterogeneous microstructure. In addition to breakpoints, a set of control points with coordinates $\mathbf{x}_i$ and weights $w_i$ determine the shape of $\mathbf{C}(u)$ as

$$\mathbf{C}(u) = \sum_{i=1}^{n} \frac{\mathbf{x}_i w_i M_i^p(u)}{\sum_{j=1}^{n} w_j M_j^p(u)},$$

where $M_i^p(u)$ is the $i$th B-spline curve of order $p$. Figure 1 schematically shows a closed NURBS curve, extracted from the SEM image of a particulate composite, which represents the shape of one of the silica particles embedded in its epoxy matrix.

There are several advantages associated with parameterizing particle morphologies in terms of NURBS functions in the shape library used in the proposed reconstruction algorithms. The first benefit of implementing NURBS, which can be observed when dealing with large 3D imaging data (e.g., nano-CT or FIB-SEM) that may require hundreds of gigabytes for storage, is similar to that used for several decades in the computer-aided design (CAD) community. In such cases, a small text file (few megabytes) holding coordinates of control points and the knot vector of NURBS curves, as well as their order, fully quantifies morphologies of all particles. As will be discussed later in this section, this NURBS shape library, together with a set of statistical microstructural descriptors (e.g., volume fraction and size/spatial distribution) is sufficient to fully characterize the material microstructure. The second advantage of NURBS is realized during the virtual packing of
Figure 1: SEM image of a particulate composite with an epoxy matrix and embedded silica particles, parameterized in terms of NURBS functions.

inclusions (Section 3), where they can easily be added to specific locations in the microstructure after applying an orthogonal transformation (translation and rotation) to their control points and knot vectors. Further, for problems such as Integrated Computational Materials Engineering (ICME), one can virtually alter morphologies of inclusions by applying a non-orthogonal transformation (e.g., scaling and elongation). As noted previously, the third major benefit is the explicit (versus pixelated) representation of material interfaces, which highly facilitate the construction of conforming meshes for subsequent FE analyses.

2.2. Statistical microstructural descriptors

In addition to shapes of particles, the virtual reconstruction of a realistic microstructure requires taking into account other statistical microstructural descriptors such as the volume fraction, size distribution, spatial orientation, and spatial arrangement of particles. An appropriate probability distribution function (PDF) is often employed to statistically quantify such microstructural features (e.g., a two-point correlation function for the spatial arrangement). Target statistical descriptors can either be extracted from the imaging data to reconstruct statistically equivalent microstructures or virtually designed/modified to build new microstructural models for applications such as ICME and Uncertainty Quantification (UQ). The appropriate number and types of PDFs for a given microstructure depend on the morphology and
arrangement of inclusions, e.g., spherical-shaped versus arbitrary-shaped or particulate versus fibrous composite. Note that the optimization phase of the proposed reconstruction technique, which will be described in Section 4, does not impose any restriction on the number and types of statistical microstructural descriptors. In the examples provided in this manuscript, in addition to the volume fraction, we have considered several PDFs including the size distribution, spatial orientation distribution, and two-point spatial correlation functions.

A log-normal distribution function is employed to characterize the distribution of size (area/volume of particles in 2D/3D or diameter of fibers) and aspect ratios of inclusions in a heterogeneous microstructure, which can be expressed as

\[
f(\alpha) = \frac{1}{\sqrt{2\pi} S} \exp \left[ -\frac{(\ln \alpha - N)^2}{2S^2} \right], \tag{2}
\]

where \(N\) is the mean value and \(S\) is the standard deviation of variable \(\alpha\) (size or aspect ratio). For fibrous microstructures, a Weibull distribution function \[41\] is utilized to quantify the distribution of lengths of embedded fibers as

\[
f(x|\lambda, k) = \frac{k}{\lambda} \left( \frac{x}{\lambda} \right)^{k-1} \exp \left[ -\left( \frac{x}{\lambda} \right)^k \right], \tag{3}
\]

where scalars \(k > 0\) and \(\lambda > 0\) are referred to as shape and scale parameters, respectively. Note that although realistic morphologies of heterogeneities are already stored in the NURBS library, it would still be necessary to take into account these PDFs during both the packing and optimization phases of the proposed reconstruction algorithm. In the former, it is important to add particles to the virtual microstructure according to the PDF associated with their size, length, and aspect ratio in the target microstructure. Also, since one of the optimization phases presented in the current manuscript involves the selective elimination of virtually packed inclusions, PDFs associated with remaining particles must satisfy the target statistical descriptors.

The spatial arrangement of particles in a given microstructure can be characterized using different PDFs, the most common one being the two-point correlation function \[42, 43\]. This function yields the probability that two randomly selected points in a microstructure belong to the same material phase. In order to determine this function for a two-phase microstructure, an indicator function \(I(x)\) is first employed to distinguish the particle phase
Ω_p at position x form the surrounding matrix as

\[
I(x) = \begin{cases} 
1 & x \subset \Omega_p \\
0 & \text{otherwise,}
\end{cases}
\] (4)

The two-point correlation function \( S_2(r) \) can then be defined as

\[
S_2(x_1, x_2) = \langle I(x_1)I(x_2) \rangle
\] (5)

where \( x_1 \) and \( x_2 \) are two arbitrary points in the microstructure, and \( \langle \cdot \rangle \) is the linear expectation operator. For a material microstructure that satisfies the ergodicity, statistical homogeneity, and isotropy conditions, the two-point probability function can be simplified as

\[
S_2(x_1, x_2) = S_2(|x_1 - x_2|) = S_2(r).
\] (6)

In order to evaluate the reference two-point correlation based on the imaging data, we implement the sampling template algorithm presented in [15]. In this approach, a spherical template with \( 10^3 \) radial points and 20 circumferential points is randomly thrown thousands of times into the medium and corresponding successful hits are counted. The number of sampling points depends on the type of the microstructure (e.g., shapes and size distribution of particles) and must be determined based on a spatial convergence study for \( S_2(x_1, x_2) \). As noted previously, other PDFs such as the lineal-path, Ripley’s K, chord length density, and two-point cluster functions can also be used to characterize the spatial arrangement of inclusions in the current microstructure reconstruction algorithm.

For heterogeneous fibrous materials such as chopped fiber composites, it is also important to take into account spatial orientations of embedded fibers during the reconstruction process. To synthesize such microstructures, we employ a truncated exponential distribution function [44] to statistically quantify orientations of fibers. Assigning an in-plane angle \( \phi \in [0, 2\pi] \) and an out-of-plane angle \( \theta \in [0, \pi] \) to each fiber, this function can characterize the statistical distribution of orientation angles as

\[
f(\beta|\lambda, b) = \frac{\lambda^{-1} \exp(-\beta\lambda^{-1})}{1 - \exp(-b\lambda^{-1})},
\] (7)

where \( 0 < \beta < b \) denotes either \( \phi \) or \( \theta \) and \( \lambda > 0 \) is a scalar variable. Different values of \( \lambda \) in this function correspond to distinct distributions of
orientation angles in the synthesized microstructure. For example, $\lambda \to \infty$ represents a random in-plane distribution when $\beta = \phi$ and $b = 2\pi$, while a more biased distribution toward the out-of-plane direction ($\beta = \theta$, $b = \pi$) can be simulated using a small value for this parameter (e.g., $\lambda = 0.1$). Next, we describe how the NURBS shape library, together with target statistical descriptors extracted from the imaging data, are used in the proposed reconstruction algorithm to synthesize realistic microstructural models.

3. Packing arbitrary-shaped inclusions

3.1. General BBox-based packing algorithm

In this section, we introduce a BBox-based packing algorithm to virtually build heterogeneous microstructures with arbitrary-shaped particles. As noted previously, a raw microstructure synthesized using this algorithm must undergo an optimization phase to create the final microstructural model that replicates all target statistical descriptors. Figure 2a illustrates a sample periodic raw microstructure with a volume fraction of $V_f = 61\%$ generated using the proposed BBox-based algorithm. The NURBS shape library used for the reconstruction of this microstructure is extracted from multiple SEM images of a heterogeneous adhesive (silica particles and epoxy matrix), one of which is shown in Figure 1. Note that complex morphologies of particles in this materials system, which include concave and elongated inclusions, prohibit the application of conventional packing algorithms such as RSA and VT for its reconstruction.

The reconstruction of a raw microstructure requires the efficient detection of overlaps between new and existing arbitrary-shaped particles during the packing process. However, determining whether NURBS functions characterizing morphologies of two particles intersect with one another would be a computationally demanding task (specially in 3D), which requires solving a nonlinear system of equations. Note that for particles with complex geometries, the surface morphology is often represented using a set of NURBS patches rather than a single closed NURBS function, which increases this computational cost even further. To address this issue, we implement a three-level BBox representation of each particle to provide a hierarchical approximation of its morphology. As shown in Figure 2b, in addition to the actual BBox of each particle (primary BBox), this hierarchical representation consists of an enlarged BBox obtained by scaling up the primary BBox, together with a set of smaller bounding boxes (secondary BBoxes) along the
Figure 2: (a) Periodic raw microstructure reconstructed using the BBox-based packing algorithm based on the shape library extracted from a set of SEM images, including that shown in Figure 1; (b) hierarchical BBox representation of a particle.

particle surface. Assuming that $b_x$ and $b_y$ are dimensions of the primary BBox, the enlarged BBox is created with dimensions $b_x + a$ and $b_y + a$, where $a = c(b_x + b_y)$ and $0 < c < 1$. Using the NURBS function representing each particle morphology, a set of points are then created on its surface and used to build secondary BBoxes, as depicted in Figure 2b. The number of points used in creating secondary BBoxes depends on the particle’s shape, i.e., its size and curvature. Note that using an exceedingly large number of secondary BBoxes does not have a notable impact on increasing the chance of packing more particles in the domain and only leads to a higher computational cost.

Before starting the packing process, all the particles stored in the shape library are sorted based on their size (volume, area, or diameter) in a descending order. Using the PDF $f(\alpha)$ associated with the size distribution, particles are then selected one-by-one from the shape library and added to the raw microstructure at a random location until reaching the desired volume fraction. Before trying to add a new particle to the microstructure, it is rotated at an arbitrary angle, followed by the construction of its enlarged, primary, and secondary BBoxes. Assuming that $n - 1$ particles are already packed, the process of adding the $n$th particle begins with checking intersec-
tions between its enlarged BBox and primary BBoxes of existing particles. As shown in Figure 3a, this computationally inexpensive heuristic check immediately identifies particles that might overlap with this new particle, but its main function is to eliminate those that certainly do not overlap with that (particles 1 and 5). Clearly, if no intersection between enlarged and primary BBoxes is detected, the new particle can be added to the microstructure. Otherwise, the next step is to determine whether the primary BBox of a new particle intersects with those of existing particles already overlapping with its enlarged BBox, i.e., particles 2, 3, 4, and 6 in Figure 3a. Again, if no overlap is detected, the new particle is embedded in the microstructure at that location.

![Figure 3](image-url)

Figure 3: (a) Checking intersections between the enlarged BBox of a new particle and primary BBoxes of existing particles during the packing process; (b) moving the new particle by distance \(d_x\) to the left to avoid intersection between its primary BBox and primary BBoxes of existing particles.

If the primary BBox of a new particle intersects with the primary BBox of only one of the existing particles (e.g., particle 4), we first evaluate dimensions \(d_x\) and \(d_y\) of their overlapping rectangular region, as shown in Figure 3a. If either \(d_x < a\) or \(d_y < a\), the new particle can be relocated within its enlarged BBox in the corresponding direction to obviate the intersections between the primary BBoxes. As shown in Figure 3b, if after relocating the new particle, its primary BBox does not intersect with those of existing particles that overlap with its enlarged BBox (i.e., particles 1, 3 or 4), the new particle can
be embedded in the microstructure. A similar approach can be used when the primary BBox of a new particle intersects with primary BBoxes of two or more existing particles, although the probability of it overlap with existing particles in its surrounding after the relocation would be much higher. Thus, in the current algorithm, we limit the relocation phase to case scenarios that the primary BBox of a new particle overlaps with that of only one of the existing particles.

If the primary BBox of a new particle still intersects with primary BBoxes of some of the existing particles after applying the relocation scheme described above, their secondary BBoxes can be utilized to more accurately determine whether the particles actually overlap. Before this task, another computationally inexpensive heuristic check is carried out to assess the probability that a new particle overlaps with one of the existing particles. In this approach, we evaluate the area of the overlapping region of primary BBoxes of new and existing particles (e.g., the rectangle with dimensions $d_x$ and $d_y$ in Figure 3a). If the ratio of $d_x d_y$ to the area of primary BBox of a new particle is large (e.g., $> 20\%$), the likelihood that it overlaps with the corresponding existing particle would be relatively high. Therefore, at initial stages of the packing process ($V_f < 30\%$), checking the intersections of secondary BBoxes is often more computationally demanding than restarting the process by picking a new random location for the particle. Thus, this initial screening phase is employed to determine the chance of success for inserting a new particle before checking intersections between secondary BBoxes. As the volume fraction of particles in the raw microstructure increases during the packing process, we skip this phase, as the probability that primary BBoxes of new and existing particles overlap with one another would be high at other locations as well.

In order to check the intersection between secondary BBoxes, one must first identify whether secondary BBoxes of existing particle(s), for which primary BBox(es) overlap with those of the new particle, intersect the overlapping box(es) or not. For example, in Figure 3a, this requires checking intersections between secondary BBoxes of particle 4 and the highlighted overlapping box with dimensions $d_x$ and $d_y$. As shown in Figure 4a, if no overlap is detected, the new particle can be added to the microstructure. Otherwise, the intersections between secondary BBoxes of the new particle and the overlapping region must be checked. The particle can be added to the raw microstructure if no intersection occurs at this stage. If secondary BBoxes of both new and existing particles intersect with this overlapping
region, it is still possible that the new particle does not actually overlap with the existing particle(s), as shown in Figure 4b. In this case, one must check the intersection between each secondary BBox of a new particle that already intersects with the overlapping region, as well as secondary BBoxes of the existing particle(s) that also intersect with this region. The new particle is virtually embedded in the microstructure if no intersection is detected between any of these BBoxes and otherwise a new location must be randomly selected to restart this process. Note that this phase of the packing process provides the ability to efficiently pack concave-shaped particles in the domain (cf., Figures 4a and 4b).

![Overlapping region of primary BBoxes](image1)
![Secondary BBoxes intersecting with the overlapping region](image2)
![Arbitrary points on surface of the new particle](image3)

Figure 4: (a,b) Checking intersections between secondary BBoxes of new and existing particles with the overlapping region of their primary BBoxes and with one another. (c) Special case: checking whether an arbitrary point on the surface of a new particle is inside an existing particle before checking intersections between their secondary BBoxes.

The algorithm described above for identifying potential overlaps between new and existing particles by checking intersections between their hierarchical BBoxes may not properly function for case scenarios similar to that depicted in Figure 4c. In such special cases, the primary BBox of a new, small particle is fully confined within the primary BBox of an existing, larger particle. Therefore, although no intersection is detected between secondary BBoxes of these particles, the new particle might still overlap with the existing particle. To handle such cases, before checking intersections between secondary BBoxes, a random point is selected on the surface of the new particle to determine whether it is located inside the NURBS curve representing the existing particle morphology or not. Intersections between secondary BBoxes are to be checked only if this point is located outside the existing particle. Otherwise, the two particles are overlapping with one another (Figure 4c) and a different location must be assigned to the new particle.
The hierarchical approach used in the proposed BBox-based packing algorithm yields a high efficiency by implementing multiple initial screenings before advancing to more computationally demanding phases. However, the computational efficacy of this method can be further improved by using a quadtree (octree) search algorithm to expedite adding new particle to a 2D (3D) domain during the packing process. As schematically shown in Figure 5, in this approach, embedded particles in the raw microstructure are stored in a quadtree data structure. To add a new particle, we first recursively check intersections between its enlarged BBox and each quadrant of the tree to identify the leaf (leaves) that overlap with it. Determining whether the new particle can be embedded at the designated location then only requires considering existing particles belonging the leaf (leaves) identified during this process. Note that the quadtree search algorithm operates at $O(\log n)$, which can considerably reduce the total number of calculations by quickly eliminating existing particles that are not in the vicinity of a new particle.

![Figure 5: Using the quadtree search algorithm for the efficient insertion of a new particle in the BBox-based packing algorithm.](image)

This BBox-based packing algorithm easily allows reconstructing heterogeneous microstructures with periodic features. This enables assigning a periodic boundary condition (PBC) in the corresponding FE model, which is one of the best choices for preventing unrealistic stress concentrations and
thereby premature damage localization near boundaries in micromechanical or multiscale simulations [45]. In order to reconstruct a periodic microstructure, if a new particle intersects with one of the domain boundaries, an identical copy of that is created at distance $L$ (length of the domain) on the parallel face. The feasibility of adding both new particles to the microstructure are then simultaneously checked (i.e., none of them are allowed to overlap with existing particles). Note that if a new particle intersects with edges (corners) of a 3D domain, 4 (8) copies of that must be created on all edges (corners) of the domain to maintain the periodicity.

It is worth mentioning that the use of hierarchical BBoxes in the proposed packing algorithm is partially inspired by some of the real-time collision detection algorithms used in computer animation [46]. While such algorithms are designed to fulfill a different objective (collision detection), they can efficiently identify close-to-contact (and not actual contact) moments between arbitrary-shaped objects. Similarly, the virtual packing of particles does not require detecting contact or intersection points between them, and instead must only ensure that they do not overlap with one another. The pseudocode in Algorithm 1 summarizes different steps of this packing algorithm.

3.2. NURBS-based packing algorithm: fibrous microstructures

While the BBox-based packing algorithm can be utilized to reconstruct a variety heterogeneous microstructure, its efficiency could decrease for creating fibrous microstructures such as nanofiber reinforced polymers. In such cases, initial screening phases such as checking intersections between enlarged and primary BBoxes of fibers can still improve the efficiency by identifying existing fibers that are in close proximity of a new fiber. However, due to the high aspect ratio of fibers, using their secondary BBoxes to more accurately determine overlaps between new and existing fibers would no longer be effective. To address this issue, we introduce a NURBS-based reconstruction algorithm, which is specifically designed to synthesize fibrous microstructures. In this approach, rather than using NURBS to represent the surface morphology of fibers in the shape library, a set of 3D NURBS curves is employed to parameterize their centerline geometries. As shown in Figure 6a, a combination of this centerline NURBS curve, together with a given diameter $D$, fully characterizes the morphology of a fiber.

Assuming that $n-1$ fibers are already embedded in the virtual microstructure, the process of adding the $n$th fiber begins by selecting a NURBS curve representing its centerline from the shape library. The Weibull distribution
Algorithm 1 (BBox-based packing of arbitrary-shaped particles)

1: function BBoxpacking($S_p, V_f^{\text{raw}}$, $f(\alpha)$, $\Omega$)
2: $S_p' \leftarrow \text{sort\_particles}(S_p)$ \quad \triangleright sort particles in shape library $S_p$ from largest to smallest size
3: qtree $\leftarrow \text{build\_quadtree}(\Omega)$ \quad \triangleright build a quadtree data structure to represent the domain $\Omega$
4: $V_f \leftarrow 0$ \quad \triangleright set initial volume fraction of raw microstructure as zero
5: while $(V_f < V_f^{\text{raw}})$ do \quad \triangleright while $V_f$ is less than target volume fraction for raw microstructure
6: $P_i \leftarrow \text{particle\_selector}(S_p, f(\alpha))$ \quad \triangleright pick a particle from $S_p$ based on size distribution function
7: $P_i' \leftarrow \text{random\_rotator}(P_i)$ \quad \triangleright rotate the particle at a random angle
8: EBox, PBox, SBox $\leftarrow \text{bbox\_builder}(P_i')$ \quad \triangleright create enlarged, primary, and secondary BBoxes
9: $x_i \leftarrow \text{random\_positionor}(P_i')$ \quad \triangleright assign a random location $x_i$ inside $\Omega$ to particle
10: $S_{\text{leaf}} \leftarrow \text{leaf\_locator}(\text{EBox}, \text{qtree})$ \quad \triangleright find leaves of the quadtree that intersect EBox
11: $S_{\text{EBox}} \leftarrow \text{check\_ebox}(\text{EBox}, S_{\text{leaf}})$ \quad \triangleright find particles in $S_{\text{leaf}}$ whose PBoxes intersect EBox
12: if ($S_{\text{EBox}} = \emptyset$) then
13: break for loop
14: else
15: $S_{\text{PBox}} \leftarrow \text{check\_pbox}(\text{PBox}, S_{\text{EBox}})$ \quad \triangleright find subset of $S_{\text{EBox}}$ whose PBoxes intersect $P_i'$ PBox
16: if ($S_{\text{PBox}} = \emptyset$) then
17: break for loop
18: else if (size($S_{\text{PBox}}$) = 1 and ($V_f > 40\%$)) then
19: $d_x, d_y \leftarrow \text{overlap\_evaluator}(\text{PBox}, S_{\text{PBox}})$ \quad \triangleright find dimensions of overlapping PBoxes
20: $S_{\text{PBox}}' \leftarrow \text{move\_particle}(P_i', S_{\text{PBox}}, d_x, d_y)$ \quad \triangleright move particle inside EBox and update $S_{\text{PBox}}$
21: if ($S_{\text{PBox}} = \emptyset$) then
22: break for loop
23: end if
24: else
25: $S_{\text{SBox}}' \leftarrow \text{check\_sbox}(P_i', S_{\text{PBox}})$ \quad \triangleright find intersecting SBoxes of new/existing particles
26: if ($S_{\text{SBox}} = \emptyset$) then
27: break for loop
28: end if
29: Goto line 9 \quad \triangleright restart process by assigning a new random location to particle
30: end if
31: end if
32: if ($P_i'$ does not intersect with domain boundaries) then
33: add\_particle(qtree, $P_i'$) \quad \triangleright add $P_i'$ to corresponding leaves of quadtree data structure
34: $V_f \leftarrow \text{volume\_updater}(V_f, P_i')$ \quad \triangleright update $V_f$ based on volume of $P_i'$
35: else
36: $P_i''' \leftarrow \text{periodic\_creator}(P_i', \Omega)$ \quad \triangleright create mirror image of particle on opposite boundary
37: if ($P_i'''$ does not overlap existing particles) then
38: add\_particle(qtree, $P_i'''$) \quad \triangleright add both $P_i'$ and $P_i'''$ to leaves of quadtree
39: add\_particle(qtree, $P_i''''$) \quad \triangleright add both $P_i'$ and $P_i''''$ to leaves of quadtree
40: $V_f \leftarrow \text{volume\_updater}(V_f, P_i')$ \quad \triangleright update $V_f$ based on volumes of $P_i'$ and $P_i''''$
41: end if
42: Goto line 9 \quad \triangleright restart process by assigning a new random location to particle
43: end if
44: end if
45: end while
46: end function

function $f(L)$ associated the length distribution of fibers and the log-normal PDF determining the distribution of fiber diameters are then utilized to determine the length and diameter of the fiber, respectively. Also, its in-plane
and out-of-plane orientation angles are determined based on the truncated exponential distribution function \( f(\beta|\lambda, b) \) given in (7), according to which an orthogonal transformation is applied to control points of the NURBS curve. In order to add the new fiber to the microstructure, we first check intersections between its enlarged and primary BBoxes and those of existing fibers (using an octree search algorithm) to quickly identify existing fibers that are in its close proximity. Assuming that the diameter of the new fiber is \( D_j \), determining whether it intersects with a nearby existing fiber with diameter \( D_i \) requires identifying the minimum distance \( d_{\text{min}} \) between the NURBS curves representing their centerlines (Figure 6b). If \( d_{\text{min}} > 0.5(D_i + D_j) \), the fibers are not overlapping and thus the new fiber can be embedded in the microstructure.

Using the control polygons approach presented in [47], the minimum distance \( d_{\text{min}} \) between two NURBS curves can be evaluated by decomposing each curve into piecewise Bézier sub-curves. However, calculating the exact value of \( d_{\text{min}} \) between two Bézier curves requires the use of an iterative Newton-Raphson solver, which could be computationally expensive. Instead, replacing each Bézier sub-curve with a line segment allows the efficient approximation of \( d^l_{\text{min}} \) by evaluating the distance between two line segments. In this approach, \( d^l_{\text{min}} > 0.5\gamma(D_i + D_j) \) indicates no intersection between fibers, where \( \gamma > 1 \) is a scalar parameter (safety factor) that depends on the number of Bézier sub-curves representing the fiber centerline. For an appropriate discretization of the centerline curve, which depends on its curvature (i.e., shorter line segments in regions with higher curvatures), even a small value for \( \gamma \) (in this work, 1.5) ensures that the probability of overlap between

Figure 6: (a) NURBS parameterization of the centerline of a fiber with diameter \( D_i \); (b) evaluating the minimum distance between centerlines of new and existing fibers to determine whether they intersect with one another.
new and existing fibers is negligible. This NURBS-based packing algorithm is summarized in the pseudocode given in Algorithm 2.

**Algorithm 2** (NURBS-based packing of arbitrary-shaped fibers)

1: function NURBS_Packing($S_f$, $V_f^{raw}$, $f(D)$, $f(L)$, $f(\phi|\lambda, 2\pi)$, $f(\theta|\lambda, \pi)$, $\Omega$)
2: while ($V_f < V_f^{raw}$) do ▷ while $V_f$ is less than target volume fraction of raw microstructure
3:     $C_i$ ← centerline_selector($S_f$) ▷ select a NURBS parameterization of centerline from $S_f$
4:     $C_i' ←$ centerline_transfer($C_i$, $f(L)$, $f(\phi|\lambda, 2\pi)$) ▷ assign $L_i$, $\phi$, $\theta$ based on dist. funcs.
5:     $D_i ←$ diameter_selector($f(D)$) ▷ pick $D_i$ based on corresponding prob. dist. func.
6:     $S_i^{seg} ←$ line_segmentor($C_i'$) ▷ subdivide $C_i'$ into a set of line segments
7:     $x_i ←$ random_positionor($C_i'$) ▷ assign a random location $x_i$ inside $\Omega$ to fiber
8:     for ($j = 1; j \neq N_{ex}; x + j$) do ▷ loop over $N_{ex}$ existing fibers in $\Omega$
9:         for ($L_i = S_i^{seg}$(begin); $L_i \neq S_i^{seg}$(end); $x + L$) do ▷ loop over $S_i^{seg}$ of new fiber
10:            for ($L_j = S_j^{seg}$(begin); $L_j \neq S_j^{seg}$(end); $x + L_j$) do ▷ loop over $S_j^{seg}$ of existing fibers
11:                $d_{min} ←$ min_distance($L_i, L_j$) ▷ evaluate min distance between $L_i$ and $L_j$
12:                if ($d_{min} < D_i + D_j$) then ▷ intersection between fibers is highly probable
13:                    Goto line 7 ▷ restart process by assigning a new location to $C_i'$
14:         end if
15:     end for
16:     end for
17:     add_fiber($D_i, C_i'$) ▷ add new fiber to the raw microstructure
18:     $V_f ←$ volume_updater($V_f, D_i, C_i'$) ▷ update $V_f$ based on volume of new fiber
19: end while
20: end function

It is worth mentioning that the proposed NURBS-based packing algorithm can be further generalized to accommodate fibers with ellipse-shaped cross sections. In order to synthesize such microstructures, in addition to finding minimum distances between line segments discretizing fiber centerlines, one must evaluate the direction $v_{min}$ of the line connecting the points corresponding to $d_{min}$ on each sub-segment. Note that $v_{min}$ can be calculated analytically without the need to implement an iterative solver. The criterion used for checking the potential overlap between fibers is then modified to $d_{min} > 0.5\gamma(D_i^{v_{min}} + D_j^{v_{min}})$, where $D_i^{v_{min}}$ and $D_j^{v_{min}}$ are diameters of new and existing fibers in direction $v_{min}$, respectively.

4. Optimizing phase: replicating statistical descriptors

Both the BBox-based and NURBS-based packing algorithms introduced in Section 3 can reconstruct virtual microstructures with realistic particle/fiber shapes, size distribution, spatial orientation, and volume fraction. However, given the random locations assigned to each inclusion during the packing process, these algorithms have no control on the spatial arrangement
of embedded heterogeneities in the resulting virtual microstructure. In other words, neither the BBox-based nor the NURBS-based algorithm is capable of replicating PDFs such as the two-point correlation function associated with the spatial arrangement of heterogeneities. This is the reason we refer to such virtually packed microstructure as raw microstructures, meaning further processing would be necessary to replicate such statistical descriptors. In this section, we introduce two optimization-based algorithms that can be utilized to evolve a raw microstructure into the final microstructural model.

4.1. GA-based optimization algorithm

In the first approach, a multi-objective Genetic Algorithm (GA) is employed to directly synthesize microstructures with low volume fractions (< 35%). This requires the construction of a raw microstructure with a higher volume fraction than the target value. The optimization process aims to eliminate a subset of virtually packed inclusions from the raw microstructure such that the target volume fraction, size distribution, spatial orientation, and spatial arrangement descriptors are satisfied for remaining heterogeneities. The objective functions of this optimization problem are defined to minimize the $L_2$-norm of the error associated with each PDF (e.g., the two-point correlation function and log-normal size distribution), i.e.,

$$E(\alpha) = \|F_i(\alpha) - \tilde{F}_i(\alpha)\|.$$  

where $F_i(\alpha)$ and $\tilde{F}_i(\alpha)$ are the target and simulated PDFs, respectively.

To carry out the GA optimization, we first create an initial population $P_0$ of $N_P$ binary chromosomes $C$, each of which encodes locations of particles/fibers in the raw microstructure. As shown in Figure 7a, in a chromosome represented by 01011101001100..., 1’s and 0’s indicate, respectively, the presence and absence of their corresponding particles in the final microstructure. In order to enhance the GA convergence, the probability of occurrence of 1’s in each chromosome is determined such that the corresponding volume fraction of the decoded microstructure is close to the target value. In this approach, rather than creating a completely random initial population, the probability of occurrence of 1’s in a chromosome is given by $V_f^{\text{raw}}$, where $V_f$ and $V_f^{\text{raw}}$ are volume fractions of target and raw microstructures, respectively. Also, in order to maintain the periodicity, only one of the particles belonging to each pair of periodic particles intersecting with domain boundaries is encoded in $C$ and the second particle simply follows its status (presence or absence) in the evolved microstructure.
In this work, the non-dominated sorting genetic algorithm II (NSGA-II) [48] is employed to evolve the initial population \( P_0 \) into an optimized population. NSGA-II seeks a set of non-dominated optimized individuals in a Pareto optimal front, in which each individual is better than others in at least one objective function but could be worse in others. In each generation \( g \) during the GA evolution, the tournament selection, together with the concept of Pareto-optimality, is employed to identify a set of fittest individuals. The uniform crossover is then applied to each selected individual, followed by the mutation of alleles with a probability of \( p_m = l_C^{-1} \) (where \( l_C \) is the length of \( C \)) to build a new population \( P'_g \). The offspring population \( P_{g+1} \) is formed by first creating \( P_g \cup P'_g \) and then re-applying the tournament selection to pick \( N_p \) fittest individuals. After performing this GA optimization, the final periodic microstructure is reconstructed by decoding one of the optimized chromosomes on the resulting Pareto-optimal front.

4.2. Relocation-based optimization algorithm

The GA-based optimization algorithm presented in Section 4.1 can efficiently and accurately synthesize microstructures with low volume fractions after a few generations. This is due to the fact that at each generation, this

![Figure 7: (a) Binary encoding of the raw microstructure to represent a new virtual microstructure by assigning 1’s (presence) and 0’s (absence) to embedded particles; (b) BBox-based sequential relocation of particles in the raw microstructure to replicate the PDF associated with their target spatial arrangement.](image)
algorithm only requires re-calculting the objective functions and not re-checking overlaps between the embedded inclusions. However, this approach is not suitable for replicating PDFs representing the spatial arrangement of inclusions in microstructures with high volume fractions (e.g., $V_f = 50\%$). In such cases, it would not be feasible to initially reconstruct a raw microstructure with a considerably higher volume fraction (e.g., $V_{f_{raw}} = 80\%$) than the target value using either of the packing algorithms described in Section 3. Hence, a small difference between volume fractions of raw and target microstructures only allows for eliminating a limited number of inclusions during the optimization process, which could easily prohibit the GA convergence or lead to genetic drift [49].

To address this limitation of the GA-based optimization phase, we introduce a sequential optimization scheme that re-utilizes the hierarchical BBox architecture used in the packing algorithm to replicate the target spatial arrangement of heterogeneities in microstructures with high volume fractions. Unlike the GA-based optimization, in this approach, the raw microstructure can be reconstructed with the actual target volume fraction. Given the fact that the size, aspect ratio, and spatial orientation distributions of inclusions can be replicated in the raw microstructure during the packing process, the only unmatched statistical descriptors will be those associated with the spatial arrangement of particles. This relocation-based optimization algorithm can also be applied to an evolved microstructure obtained from the GA optimization phase, in which all PDFs except for those associated with the spatial arrangement of heterogeneities are properly matched. Therefore, in either of these cases, the only statistical descriptors that must still be replicated by relocating embedded inclusions are those associated with their spatial arrangement (e.g., two-point correlation functions).

In order to simulate the spatial arrangement PDFs, the embedded particles in the raw/evolved microstructure are sequentially relocated to optimize their locations such that the $L_2$-norm of the error between target and simulated descriptor functions are minimized. Note that after the construction of the raw microstructure using either of the proposed packing algorithms, it is already known that the enlarged BBox of each inclusion intersects with primary BBoxes of which neighboring particles. For example, as depicted in Figure 7b, the enlarged BBox of particle 1 only intersects with primary BBoxes of particles 2, 3, and 4. Hence, if the relocation of particle 1 is restricted to the area within its enlarged BBox, one only needs to ensure that it does not overlap with particles 2, 3, and 4 after relocation. In the current
algorithm, four possible directions are considered for relocating a particle by a distance of $a = c(b_x + b_y)$ in 2D, as indicated by arrows in Figure 7b. A similar approach can be used in 3D, with the difference that six directions are available for moving a particle parallel to faces of its enlarged BBox. For each vertical or horizontal direction, the feasibility of moving a particle can easily be checked using the same hierarchical approach used during the packing process, i.e., starting by checking intersections between primary BBoxes and then using secondary BBoxes (or centerline NURBS) to determine whether relocated and existing particles overlap with one another or not.

In the proposed relocation-based optimization approach, all particles are visited in a random order to determine the feasibility of relocating them and its impact on the evolving PDF. For a periodic microstructure, the particles cut by domain boundaries are excluded from the relocation process to maintain the periodicity and facilitate the implementation of this algorithm. After identifying an allowable movement direction for a particle (i.e., no overlap with neighboring inclusions), the objective function associated with the spatial arrangement PDF (Eqn. 8) is recalculated based on its new location. If the updated objective function $E(\alpha)$ is reduced, the particle is relocated and otherwise it maintains its current location. Note that in this sequential optimization algorithm, which is in fact an extension of the BBox-based packing process, other statistical descriptors initially replicated in the raw microstructure (e.g., volume fraction and size distribution) remain intact. The pseudocode for the proposed GA-based and relocation-based optimization algorithms are presented in Algorithm 3.

5. Numerical Examples

In this section, we implement the algorithms described in sections 3 and 4 to reconstruct various heterogeneous microstructures, including 2D and 3D particulate composites, a ceramic fiber reinforced aluminum, and a nano-enhanced polymer with embedded carbon nanofibers.

5.1. 2D particulate composite

As the first example, the proposed GA-based optimization algorithm is employed to transform the virtually packed raw microstructure of the 2D particulate composite shown in Figure 2a into a periodic microstructure with a volume fraction of $V_f = 35\%$. The $300 \mu m \times 300 \mu m$ raw microstructure is composed of 1022 embedded silica particles ($V_f^{raw} = 61\%$) in an epoxy
matrix. The target PDFs used for the reconstruction of final RUC, i.e., the size distribution $f(\alpha)$ and the two-point correlation function $S_2(x_1, x_2)$ of particles, are illustrated in Figure 8. This figure also shows the statistical microstructural descriptors associated with the raw microstructure, which are considerably different than target PDFs. In order to synthesize the final microstructure, the GA optimization is carried out with an initial population of 200 individuals, which are evolved for only 20 generations. Note that the initial population is constructed such that each chromosome represents an RUC with a similar volume fraction to the target value. The periodic microstructure associated with one of the individuals in the resulting Pareto-optimal front is depicted in Figure 9a, which is formed after the selective elimination of 499 particles from the raw microstructure. The size distribution and two-point spatial correlation functions corresponding to this RUC are illustrated.
Figure 8: First example problem: initial, target, and optimized probability distribution functions: (a) particles size distribution; (b) two-point spatial correlation function.

Figure 9: First example problem: (a) final periodic RUC with $V_f = 35\%$ after applying the GA optimization phase to the raw microstructure depicted in Figure 2a; (b) small portion of the conforming mesh generated using CISAMR, corresponding to the inbox shown in figure a.

in Figure 8, which have an excellent agreement with target PDFs.

As noted previously, two of the main advantages of the proposed BBox-based packing algorithm, which are maintained after applying the optimization phase, are the explicit representation of the morphology of material inter-
faces and the ability to synthesize periodic microstructures. These features, which are preserved after the completion of the optimization phase, highly facilitate the construction of FE models with PBC for micromechanical analyses. In order to shed more light on these advantages, a new non-iterative mesh generation algorithm, named Conforming to Interface Structured Adaptive Mesh Refinement (CISAMR) [38, 39, 50], is employed to create the FE model of RUC shown in Figure 9a and simulate its micromechanical behavior.

CISAMR automatically detects intersection points of material interfaces with a structured background mesh and transforms that into a high-quality, hybrid, conforming mesh composed of quadrilateral and triangular elements. For 2D problems, this non-iterative transformation is carried out in three steps [38]: (i) \( h \)-adaptive refinement of background elements in the vicinity of material interfaces; (ii) \( r \)-adaptivity of nodes of elements intersecting material interfaces by relocating some of these nodes to the interface; (iii) sub-triangulating the remaining nonconforming elements. Although a detailed description of the 2D CISAMR algorithm is presented in [38], it is worth mentioning that one of the unique advantages of this mesh generation algorithm is the ability to non-iteratively handle material interfaces that are in close proximity or intersecting with one another. This is either carried out by applying additional levels of \( h \)-adaptive refinement to embedded particles that are in close proximity or using a hierarchical \( r \)-adaptivity algorithm to create conforming elements in such regions that material interfaces are in contact or intersecting one another. Note that in cases that the distance between two particles is extremely small, in order to avoid several additional levels of \( h \)-adaptivity, one can assume that corresponding particles are in contact and implement the hierarchical \( r \)-adaptivity scheme to properly transform the background mesh.

A small portion of the conforming mesh generated using CISAMR corresponding to the inbox of the synthesized periodic RUC in Figure 9a is depicted in Figure 9b. A 300×300 structured mesh with two levels of \( h \)-adaptive refinement is utilized to generate this mesh (≈0.64 million elements). The FE approximation of shear stress \( \sigma_{12} \) and normal stress \( \sigma_{22} \) fields in this RUC subject to a macroscopic shear strain of \( \varepsilon_{12}^M = 1 \) is depicted in Figure 10. In these simulations, the modulus of elasticity and Poisson’s ratio of silica particles are assumed to be \( E_p = 524 \) GPa and \( \nu_p = 0.15 \), respectively, while those of the epoxy matrix are considered as \( E_m = 415 \) GPa and \( \nu_m = 0.16 \). PBC is imposed by assigning identical equation numbers to all periodic pairs of nodes located on mutually parallel edges of the domain during the assem-
Figure 10: First example problem: simulated (a) shear stress and (b) normal stress in the FE model of the synthesized periodic RUC generated using CISAMR subject to a macroscopic shear strain of \( \varepsilon_{12}^M = 1 \) and PBC.

bly of stiffness matrix. Based on the asymptotic expansion method, \( \varepsilon_{12}^M \) is then applied to all elements discretizing RUC. As shown in Figure 10, the high-quality, adaptively-refined, periodic mesh generated using CISAMR, together with the use of PBC, yields an appropriate approximation of sites of stress concentration in this synthesized RUC.

5.2. 3D particulate composite

In this example, the BBox-based packing and GA-optimization algorithms are employed to reconstruct a periodic RUC with \( V_f = 25\% \) based on the micro-CT data of a particulate composite, as shown in Figure 11a. After processing the imaging data, the NURBS shape library is built by extracting morphologies of a representative set of particles, some of which are depicted in Figure 11b. The BBox-based packing algorithm is then implemented to build the raw microstructure with a volume fraction of \( V_f^{\text{raw}} = 37\% \) (940 particles), as illustrated in Figure 12a. Next, the GA-optimization phase is utilized to achieve the target volume fraction (\( V_f = 25\% \)), as well as the two-point correlation and size distribution functions depicted in Figure 13. This optimization is carried out with an initial population consisting of 200
individuals, which is evolved for only 10 generations. A periodic RUC corresponding to one of the individuals in the resulting Pareto-optimal front, which contains 532 particles, is depicted in Figure 12b. According to Figure 13, the difference between the statistical descriptors associated with this RUC and target PDFs are negligible.

Figure 11: Second example problem: (a) micro-CT image of the heterogeneous adhesive microstructure; (b) morphologies of a subset of particles stored in the shape library.

Figure 12: Second example problem: (a) raw microstructure of a particulate composite reconstructed using the BBox-based packing algorithm; (b) final periodic RUC obtained after the completion of the GA optimization phase.
Figure 13: Second example problem: initial, target, and optimized probability distribution functions: (a) particles size distribution; (b) two-point spatial correlation function.

5.3. Ceramic fiber reinforced aluminum

This example demonstrates the application of a combination of BBox-based packing, GA optimization, and relocation-based optimization algorithms for synthesizing the periodic RUC of a metal matrix composite (MMC). One of the 2D slices of micro-CT images prepared for this MMC is illustrated in Figure 14a, which in addition to embedded dogbone-shaped ceramic fibers with a volume fraction of $V_f = 50\%$, contains multiple pre-existing voids in its aluminum matrix. The complexity of this microstructure and in particular the presence of voids along fiber-matrix interfaces leads to severe challenges during its virtual reconstruction and subsequently the creation of an appropriate conforming FE model for this problem. In order to overcome these challenges, the BBox-based packing algorithm is first implemented to synthesize the raw microstructure, while neglecting the presence of voids. The resulting periodic RUC with a fiber volume fraction of $V_f^{\text{raw}} = 62\%$ (557 fibers) is depicted in Figure 14b. The shape and size distribution of fiber cross-sections in this initial RUC are extracted from the micro-CT image shown in Figure 14a.

The raw microstructure must then be transformed into the final RUC by replicating the target statistical microstructural descriptors, i.e, the volume fraction and two-point spatial correlation function of fibers. Due to the high value of the targeted volume fraction in this case ($V_f = 50\%$), only a small number of embedded fibers in the raw microstructure can be eliminated dur-
Figure 14: Third example problem: (a) 2D slice of the micro-CT image of the ceramic fiber reinforced aluminum composite; (b) raw microstructure with $V_{\text{raw}} = 62\%$ reconstructed using the BBox-based packing algorithm.

Using the GA-based optimization. Thus, using this algorithm alone would not yield an RUC that can accurately replicate the desired two-point correlation function due to limitation on the number of fibers that can be selectively eliminated. To address this issue, the GA-based optimization algorithm is first employed to form an intermediate stage (Pareto-optimized front) by evolving an initial population of 200 individuals encoded based on the raw microstructure for 15 generations. The resulting encoded microstructures at this stage have a volume fraction of $V_f = 50\%$, can perfectly replicate the target size distribution of fibers, and the error between their two-point correlation functions and the target function is minimized (but not vanished). Figure 15b shows this difference for one of the optimized microstructures obtained from the GA optimization, which contains 462 fibers. The relocation-based optimization phase described in Section 4.2 is then utilized to precisely replicate the target two-point spatial correlation function by moving the fibers within this intermediate microstructure. The final periodic RUC and the corresponding two-point correlation function are depicted in Figure 15, where the latter perfectly matches target PDF.

Figure 15a shows a set of closed NURBS curves representing the pre-existing voids in the MMC microstructure, which are virtually superposed.
on the synthesized RUC at random locations. Similar to the voids observed in the micro-CT image of this materials system (Figure 14a), these NURBS curves are intersecting with those corresponding to silica particles. The integration of the current reconstruction algorithms and CISAMR highly facilitates the creation of an appropriate FE model for this complex microstructure. In this case, CISAMR enables the direct use (i.e., without trimming) of overlapping NURBS representing void and particle phases for the construction of a high-quality conforming FE mesh. As discussed in [38], CISAMR can easily handle intersecting material interfaces via a non-iterative, hierarchical $r$-adaptivity scheme.

The FE approximation of the normal stress field in the $x$-direction in this RUC subject to a macroscopic normal strain of $\varepsilon_{11}^M = 1$ and PBC (plane strain) is depicted in Figure 16a. Material properties used in this model are $E_f = 300$ GPa, $\nu_f = 0.24$ for ceramic fibers and $E_m = 69$ GPa, $\nu_m = 0.3$ for the aluminum matrix. Figure 16b illustrates a small portion of the conforming mesh generated using CISAMR on an initial $320 \times 320$ structured mesh, where 3 levels of $h$-adaptive refinement are applied along material interfaces. This high-quality mesh enables the accurate approximation of stress concentrations in the aluminum matrix, which are in particular magnified in the vicinity of pre-existing voids (cf., Figure 16a).
5.4. Carbon nanofiber reinforced polymer

In this final example problem, the NURBS-based packing algorithm is employed to synthesize a nano-enhanced epoxy RUC with embedded carbon nanofibers (CNFs). In such nano-enhanced composites, a low volume fraction of CNFs (often < 1%) is introduced to the polymer matrix to improve its stiffness, strength, and abrasion resistance [51, 52]. Figure 17 illustrates two virtual nanostructures \( L = 300 \) nm with similar volume fractions of 0.62%, which are synthesized based on four arbitrary-chosen morphologies for CNF centerlines in the shape library. During the packing process, the diameter of each nanofiber is selected within the range of 0.5 nm to 2.0 nm according to a log-normal distribution function \( (N = 0.0285 \text{ and } S = 0.336) \). Also, CNF lengths are determined based on a Weibull distribution function with \( \lambda = 161.7, k = 2.4 \), and within the range of 20 nm to 500 nm [41]. Note that the 2,775 embedded CNFs in the nanostructure depicted in Figures 17a and 17b have random spatial orientations. However, the 2,803 nanofibers virtually packed in RUC shown in Figures 17c and 17d have a biased orientation.
Figure 17: Fourth example problem: synthesized RUCs with (a,b) random CNFs' spatial orientation; (c,d) biased spatial orientation of CNFs toward the $xy$ plane. This is achieved by choosing $\lambda = -0.2$ and $b = \frac{\pi}{2}$ in the truncated exponential distribution function characterizing the spatial orientation of CNFs, as given in (7). It must be noted that in the absence of 3D imaging data to determine PDFs associated with the spatial distribution of fibers (e.g., two-point correlation function), the optimization phase is not applied to the virtually packed nanostructures shown in Figure 17.
6. Conclusion

A set of new algorithms was introduced for the virtual reconstruction of heterogeneous microstructures with arbitrary-shaped embedded inclusions. As a pre-processing phase, a library of heterogeneities' shapes, characterized in terms of NURBS, together with their statistical descriptors (e.g., size and spatial distribution) were extracted from the imaging data. An efficient packing algorithm relying on checking intersections between hierarchical bounding boxes of particles was then employed to synthesize an initial (raw) microstructural model. For fibrous materials, a special version of this algorithm was presented that directly checks intersections between NURBS curves representing fiber centerlines. In order to simulate target statistical microstructural descriptors and in particular those associated with the spatial arrangement of inclusions, two distinct optimization-based algorithms were introduced: In the first approach, a multi-objective GA was utilized to selectively eliminate some of the inclusions, while in the second technique packed inclusions were sequentially relocated within the raw microstructure. Several example problems were presented to demonstrate the application of proposed algorithms for reconstructing realistic particulate and fibrous composite microstructures. We also employed a non-iterative mesh generation algorithm (CISAMR) to create FE models and simulate the micromechanical behavior of resulting microstructures to show three main advantages of these reconstruction algorithms: (i) explicit representation of arbitrary-shaped particle morphologies; (ii) creating periodic microstructures; (iii) ability to handle intersecting interfaces that, for example, could represent a void formed adjacent to an inclusion.

Acknowledgement

This work has been supported by the Air Force Office of Scientific Research (AFOSR) under grant number FA9550-17-1-0350. The authors also acknowledge partial support from the Ohio State University Simulation Innovation and Modeling Center (SIMCenter) and the allocation of computing time from the Ohio Supercomputer Center (OSC).

References


