# X-ray CT based Multi-layer Unit Cell Modeling of Carbon Fiber-Reinforced Textile Composites: Segmentation, Meshing and Elastic Property Homogenization

**Yuriy Sinchuk [1,*], Oxana Shishkina [2], Mikael Gueguen [3], Loic Signor [3], Carole Nadot [3], Herve Trumel [4], and Wim Van Paepegem [1]**

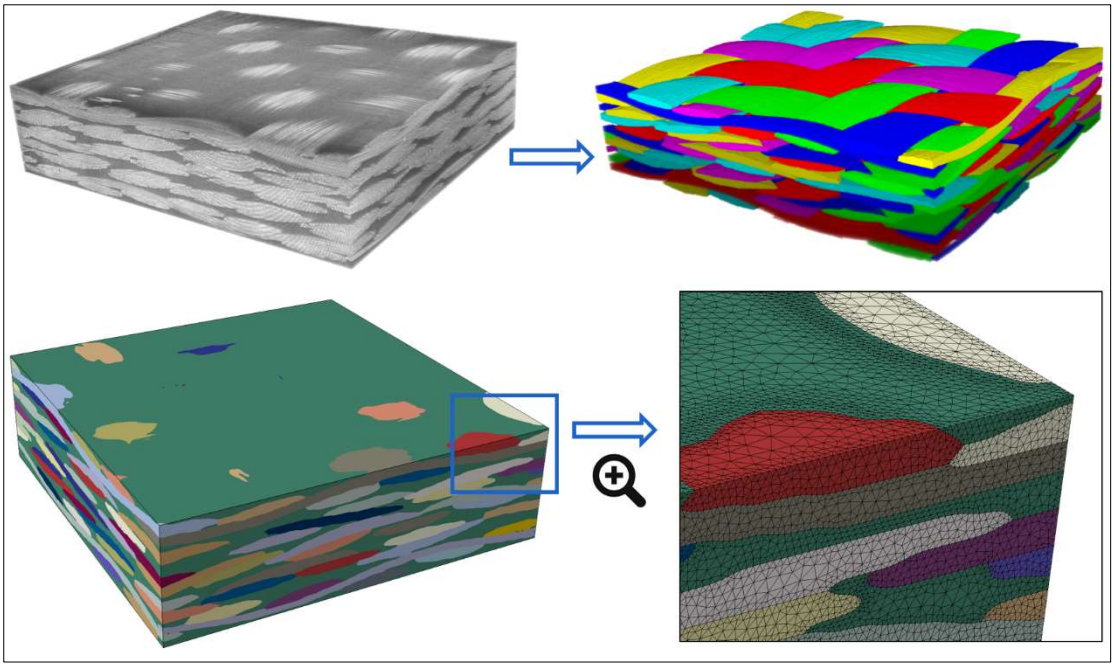
[1] Department of Materials, Textiles and Chemical Engineering, Faculty of Engineering and Architecture, Ghent University, Technologiepark Zwijnaarde 46, 9052 Zwijnaarde, Belgium; E-Mail: wim.vanpaepegem@ugent.be

[2] Siemens Industry Software NV, Interleuvenlaan 68, 3001 Leuven, Belgium; E-Mails: oxana.shishkina@siemens.com

[3] Dpt Physique et Mécaniques des Matériaux, Institut Pprime, Université de Poitiers, ISAE-ENSMA, Téléport 2, 1 avenue Clément Ader – BP40109, 86961 Futuroscope Chasseneuil cedex, France; E-Mails: carole.nadot@lmpm.ensma.fr

[4] CEA, DAM, Le Ripault, 37260 Monts, France; E-Mails: herve.trumel@cea.fr

* Correspondence: yuriy.sinchuk@ugent.be



**Abstract**

Generation of realistic finite element method (FEM) geometry of a textile composite material at tow scale remains a challenging stage of material modeling. In this paper, a FE model generation approach is introduced, based on micro-Computed Tomography (μ-CT) images of a carbon fiber reinforced textile composite with multiple layer orientations. For the specified laminate layup made of plain weave fabric layers, a tow instance segmentation method based on deep learning was proposed. The fiber local orientation field was extracted from the tow centerlines approximation. For the image-based finite element meshing, a material interface reconstruction algorithm was proposed. The algorithm leads to a conformal and smooth finite element (FE) mesh with high-quality elements. FE models with voxel and tetrahedral meshes considering different approximations of the material orientation field were generated and used for numerical homogenization of the composite material's elastic properties.

**Keywords**: *Material modeling, X-ray computed tomography, Textile composites, Finite element analysis (FEA), Deep learning*

## 1. Introduction

Developments in manufacturing industries have led to the need for the growing variety of composite material applications and improvement of the material performance. Carbon fiber-reinforced polymer (CFRP) composites are very strong and light materials that are commonly used where a high strength-to-weight ratio is needed (aerospace engineering, automotive engineering, sports goods manufacture, and many other fields of industries). One of the common groups of CFRP materials are textile composite (woven fabric) laminates. This material often has many application-specific advantages over corresponding unidirectional or short fibers composites (like resistance to fracture due to weaving, better draping capabilities or higher impact strength).

FE modeling of the composites (like numerical homogenization of mechanical properties) plays an important role in the material manufacturing processes optimization and quality prediction. Any composite manufacturing process introduces local changes (variability) in the internal composite structure, e.g., reinforcement misalignment, thickness variations, defects (e.g., voids) [1-3]. In order to achieve a higher accuracy of composite part simulations, more reality (details) should be introduced in the modeled composite geometry [4, 5]. That directly affects predictions of the local material stiffness and, consequently, the mechanical properties of the part itself due to the composite hierarchical nature [3, 6].

The most realistic composite material models can be obtained with the help of X-ray micro-computed tomography (μ-CT) and image processing techniques for data segmentation [2, 4, 6, 7]. However, such a modeling framework includes two challenging sub-problems: image segmentation and finite element (FE) mesh generation, which need to be addressed for the accurate model development.

The gradient-based approaches like structure tensor are often used for the segmentation of μ-CT images of textile composites [6, 8, 9]. The quality of such segmentation is mainly determined by the accuracy of the local fiber gradient, therefore, it is difficult to use this method for images with poor fiber (texture) visibility. Nowadays the deep-learning segmentation becomes a rapidly developing alternative in the X-ray image-based modeling. Many challenging problems in the μ-CT image processing domain were solved using a convolutional neural network (CNN) with the U-net architecture [10-13]. This method is able (potentially) to handle arbitrary bad quality μ-CT images of a textile, where fibers gradient and tow/matrix interface are difficult to recognize because of noise and low phases' contrast [14]. In [15], the μ-CT image segmentation problem was solved using the DeepLab v3+ deep convolutional neural network with ResNet18 as the backbone [16]. Some of the recent studies are concentrated on solving the fiber bundles (instance) segmentation problem by different deep learning approaches [17-19]. For example, R-CNN neural network [20] was used in [18] for tow's cross-section instance segmentation based on synthetic μ-CT images as a training

dataset. In its turn, the synthetic CT is a result of the U-net prediction. The segmentation result was later converted into a FE conformal mesh using a meshing algorithm [21].

The simplest way to convert a segmented image into a FE model is using the voxel mesh method [22]. However, this method often has the following drawbacks: the resulting model is computationally expensive (a big number of mesh nodes), its material interface is stair-wise (not smooth) [23] which results in artificial stress concentrations. As a solution to the latter problem, image-based FE meshing methods for woven geometry reconstruction by triangular smooth surface can be used. These methods are roughly split into two groups: general-purpose methods that can be applied to any kind of 3D image input (multi-label datasets) [24-26] and methods that are dedicated especially to woven geometry input [27-30]. The general purpose remeshing and mesh optimization algorithm [31, 32] can also be used as a critical part of a woven structure meshing workflow [33, 34]. Although many image-based meshing approaches exist, usually each of them has its strong and weak sides, therefore, the problem of FE model generation from the image input continues to be an active research area.

In this study, a workflow for the generation of a realistic FE model of a CFRP laminate with multiple layer orientations starting from a μ-CT image input is introduced. Firstly, the warp/weft/matrix regions are segmented based on the deep learning approach. After that, an algorithm is proposed and applied to the tow's instance segmentation with the reconstruction of the local material orientation (fiber direction) as a result. Then, the segmentation result is used as input data for the image-based FE meshing. The mesh generation step consists of the proposed algorithm for surface meshing (material interface reconstruction) and the volume meshing (tetrahedralization) using open source TetGen software [35]. The surface meshing algorithm is implemented in Python based on a multi-material marching cubes algorithm [26].

The developed segmentation and the mesh generation results were used to create FE models of the composite unit cell (UC) of an automotive composite B-pillar and predict its homogenized stiffness using automated workflows realized with Siemens Simcenter 3D Materials Engineering [5, 36]. The FE simulation based on the tetrahedral mesh was compared against the corresponding voxel-

mesh model result. Additionally, the influence of out-of-plane fiber orientation on the elastic properties was evaluated.

The proposed modeling workflow is summarized in the next sections as follows:

1. μ-CT image (semantic) segmentation using deep learning approach;
2. Tow instance segmentation (tows splitting);
3. Local fiber orientation reconstruction;
4. Image-based meshing;
5. Modeling material elastic properties.

## 2. Materials

The μ-CT image stack was acquired from a composite B-pillar (car's component) [37] which was a sandwich structure made of two CFRP textile composite skins and a machined foam insert separating the inner and outer sides of the B-pillar (**Fout! Verwijzingsbron niet gevonden.**a). Each skin was composed of five plies of the balanced twill weave carbon fiber fabric Chomarat C-WEAVE™ 285T 3K HS impregnated with the epoxy resin Sicomin SR1710 with hardener SD8731 [38] and had a layup of (0, 30, 60, 30, 0). The layer angles correspond to rotation from Y to Z axis (Figure 1b, 1c). The textile pattern of each ply is 2/2 twill weave. The structure consists of tows with six orientations: -60/-30/0/30/60/90 degrees. The scanned specimen was cut from a central part of the B-pillar outer skin (**Fout! Verwijzingsbron niet gevonden.**a, white rectangle) where the shearing of the fabric was minimal, and the layup was close to the designed configuration. The μ-CT images were acquired and reconstructed by TESCAN (Belgium). The volume was scanned using the TESCAN UniTOM XL X-ray CT system with a voxel size of 4.5 μm.

After the preprocessing steps (rotation and cropping) the resulting image (Fig. 1b, 1c) had a size of $S_X \times S_Y \times S_Z = 485 \times 2016 \times 1696$ voxels. This image was used as an input for further processing which is described in the next sections.

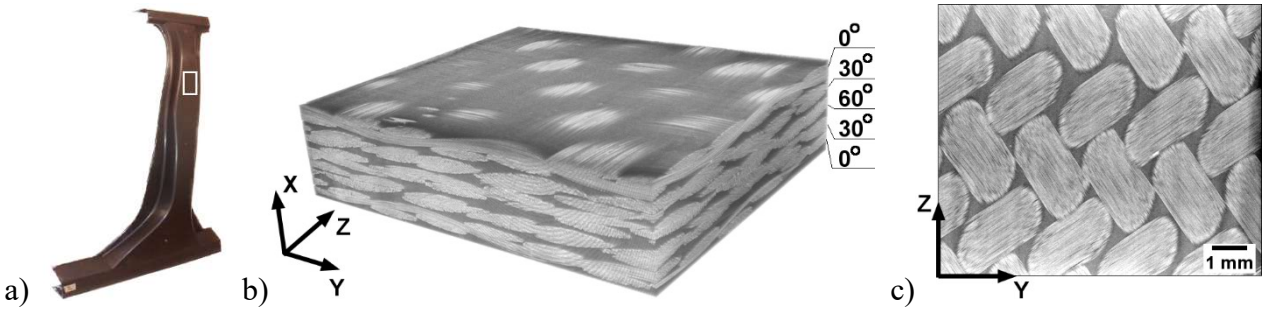



**Fig. 1.** Composite B-pillar (a); the input μ-CT image: 3D view (b), 2D slice of 30° orientation ply (c)

The constituent elastic properties used in the simulations are summarized in Table 1. The studied fabric [39] was composed of untwisted Toray T300B carbon fibers [40] and impregnated with the Sicomin epoxy resin [41]. In literature, the transverse elastic modulus of carbon fibers varies in the range of 5.25-28.7 GPa [42] depending on the measurement technique or back-calculation procedure. In this study, a value of 15 GPa was adopted, which falls in the middle of the aforementioned range and was also reported in another research work [43]. In Table 1, the subscripts 1 and 2, 3 indicate the longitudinal and transverse fiber directions, respectively; shear modulus G and $G_{23}$ were extracted from corresponding values of Young's modulus and Poisson's ratio; the epoxy Poisson's ratio was taken from Siemens Simcenter 3D Materials Library.

**Table 1.** Elastic constants of the material constituents

| | **Carbon Fiber T300** | | **Epoxy Sicomin** | |
|---|---|---|---|---|
| *Mass Density (kg/m$^3$)* | $\rho = 1760$ | [40] | $\rho = 1148.3$ | [41] |
| *Young's Modulus (GPa)* | $E_1 = 230$<br>$E_2 = E_3 = 15$ | [40, 42, 43] | $E = 3.3$ | [41] |
| *Poisson's Ratio* | $\nu_{12} = \nu_{13} = 0.256$<br>$\nu_{23} = 0.3$ | [42] | $\nu = 0.37$ | |
| *Shear Modulus (GPa)* | $G_{12} = G_{13} = 27.3$<br>$G_{23} = 5.796$ | [42] | $G = 1.2$ | |

## 3. CT Image Processing

### *3.1. Semantic Segmentation of Collinear Tows*

Despite the good quality of the input μ-CT image and relatively low level of noise (Fig. 1c), the tow (semantic) segmentation remains a challenging problem for a morphological image processing approach because the local fiber texture is not clear enough to extract the gradient

accurately for all voxels. Therefore, to solve the segmentation problem, the deep learning approach [14] used for the bi-directional textile (tows with 0/90° orientations) was adopted here for the multiple ply orientations case. Similar to the method in [14], the neural network in this study was trained using manually annotated 2D image slices. The slices were oriented in-plane of the sample as depicted in Fig. 1c. For the training dataset, 17 from 485 slices (roughly 3 slices per ply) were selected. All of the slices were annotated manually using ImageJ software [44]. The neural network with the U-net architecture [10] (depth level 5, no batch normalization) was used here. The count of filters in the first convolution layer was 32 and this value increased twice on each next layer, thus, leading to 1024 in the middle of the network (bottlenecks layer).

In order to simplify the training process, the 6 tow orientations were grouped into 2 classes (labels) as demonstrated in Fig. 2. It decreased the count of output classes from 7 (6 tow orientations plus matrix) to 3 and led to more accurate neural network prediction on the test data. Note that the orientations were clustered in a way to have only one boundary value which splits these two clusters (45° and 15° for configurations in Fig. 2a and Fig. 2b respectively).

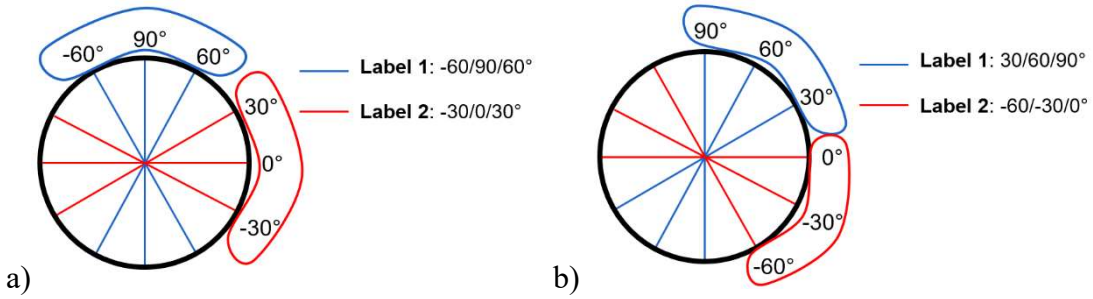



**Fig. 2.** Orientation labels: group A (a), group B (b)

In total, three different ways of the tow orientations labeling were used:

1. Group A (Fig. 2a): -60°, 90°, 60° are label 1, -30°, 0°, 30° are label 2;
2. Group B (Fig. 2b): 30°, 60°, 90° are label 1, -60°, -30°, 0° are label 2;
3. Group C: all tows are label 1 and matrix is label 0.

The neural network was trained for each of the label groups. The training processes had the following characteristics: Adam optimization method with learning rate 0.0003, patch size of 2016x1696 (whole slice), batch size of 1, number epochs of 150 (chosen empirically). The training

on a Nvidia Tesla T4 16Gb GPU took about one hour (for each label group). The example of the raw image slice and the corresponding network predictions are presented in Fig. 3.

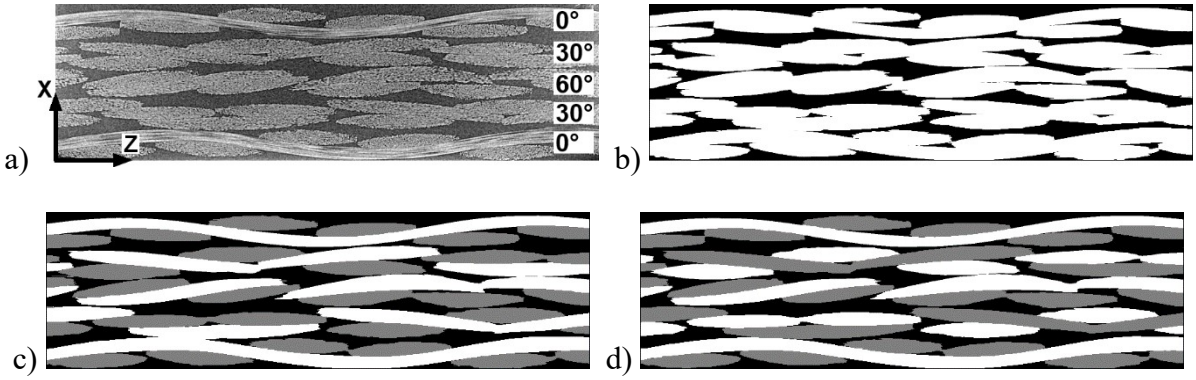



**Fig. 3.** Input image slice (a); neural network predictions: tows/matrix segmentation (labels group C) (b), segmentation for labels group A (c), segmentation for labels group B (d)

The tows/matrix segmentation (labels group C) was optional. Here it was used for the accuracy improvement and the matrix region unification of the segmentations A and B. As for the post-processing steps, the small, isolated regions were removed from the segmentation C; and the remaining matrix of C was assigned as "true matrix" with the consecutive removal of "original matrix" in A and B. After that, a mask for each ply was extracted from the multiplication A*B (Fig. 4a). These ply masks together with the segmentation A provided resulting labels of the collinear tows (Fig. 4b).

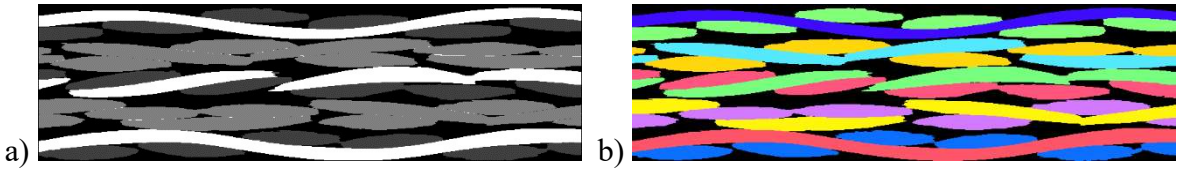



**Fig. 4.** Multiplication of segmentations A and B (a); the resulting collinear tow labels (b)

*3.2. Tow Instance Segmentation*

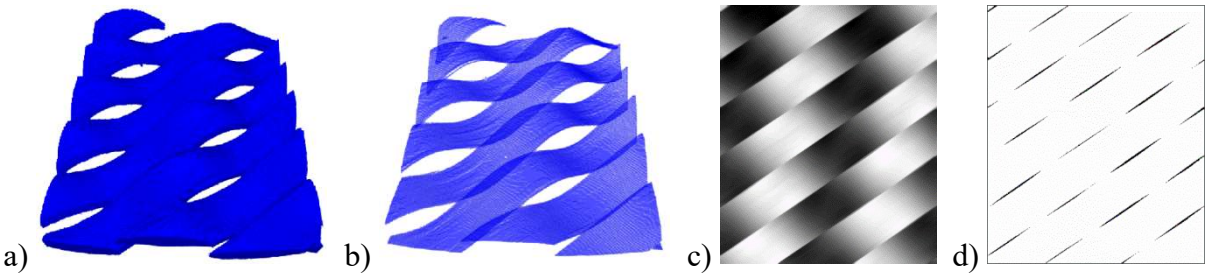
Let us call the collinear tows within a single ply a "tow family". An example of such a tow family is visualized in Fig. 5a. In the total volume, there were 10 "families" (2 for each of 5 plies), which correspond to 10 different label colors in Fig. 4b. For labeling the individual tow instances within a "family", the watershed segmentation over boundary distance transform was used with *tow*

*mean surface* marks. These marks and the corresponding result of the watershed segmentation are presented in Fig. 5g and 5h. The tow mean surfaces were obtained as a segmentation result of the *"family" mean surface* (Fig. 5b).

Let us define all voxel coordinates of the image as grid points $I_{ijk} = [i, j, k]$. For example, index $i$ corresponds to image dimension X and takes values from 0 to $S_X - 1$. The semantic segmentation result can be defined as a function $A_{ijk} = A(I_{ijk})$ that takes values from 0 to 10 and a tow "family" image with label index $f = 1,2, \dots ,10$ can be defined by a binary function $F_{ijk} = F(I_{ijk})$ that takes the value 1 if $A_{ijk} = f$. Then for each $(j, k)$ the tow "family" is approximated by a mean integer value (Fig. 5c)

$$M_{jk} = \left\lfloor \frac{1}{2} + \sum_{i=0}^{S_X-1} I_{ijk} F_{ijk} / \sum_{i=0}^{S_X-1} F_{ijk} \right\rfloor,$$

where $\lfloor \cdot \rfloor$ is the round down function. Finally, the binary function of the tow "family" mean surface (Fig. 5b) can be introduced as $V_{ijk} = \bar{V}_{ijk} F_{ijk}$, where the binary function $\bar{V}_{ijk} = \bar{V}(I_{ijk})$ is non-zero only in voxels $[M_{jk}, j, k]$. The sum $\sum_{i=0}^{S_X-1} V_{ijk}$ gave us a 2D image where non-zero values are white and zero values form black dashes (Fig. 5d). These dash voxels were clustered using the DBSCAN algorithm [45] and connected by lines as demonstrated in Fig. 5e. The lines split the surface projection into tow projection regions (Fig. 5f). Using the heightmap data $M_{jk}$ (Fig. 5c), we returned to three-dimensionality from the projections and obtained the required approximation of tow mean surfaces (Fig. 5g). Then, these tow mean surfaces were used as input for the watershed segmentation of the "families" (Fig. 5h).



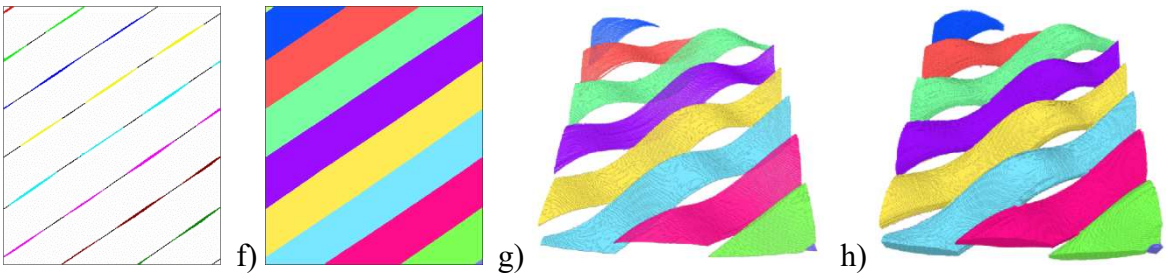



**Fig. 5.** Tows splitting workflow: input tow "family" (a), tow "family" mean surface (b) and its heightmap (c), surface projection (d), gaps clustering (e), mean surface segmentation in 2D (f) and 3D (g), tow "family" segmentation (h)

*3.3. Local Orientation Analysis*

The material properties of a tow are strongly anisotropic and defined by the local fiber orientation field. The following algorithm was implemented to compute tow orientation vectors required for FE modeling:

*For each tow:*

1. *Find centerline projection using "family" orientation angle (Fig. 6a);*
2. *Find the tow mean surface (voxel representation);*
3. *Find tow centerline voxels (based on step 1 and 2 results);*
4. *Get tow centerline curve by a spline fitting to the centerline voxels (Fig. 6b);*
5. *Find fiber orientation for centerline points as tangent vectors;*
6. *Get orientation for each tow voxel as the orientation of the closest centerline point.*

Firstly, the 2D in-plane projection of the tow centerline was found as the straight line through the tow projection mean point with the "family" orientation slope. Fig. 6a shows a tow projection (yellow), its mean point (green) and the resulting centerline projection for "family" slope 34° (red line). This approach provides the same slope for each tow in a "family" including small boundary pieces which can be approximated not accurate by another method. The "family" slopes extracted from the known lay-up (0°, 30°, 60°, 30°, 0°) can be inaccurate, therefore, the new values were computed by averaging tow centerline slopes with length weights. The averaging with the length

weight means that each tow contributes its slope into the average values proportional to its length. For example, the found "family" slopes which correspond to the plies lay-up were [1.01°, 33.84°, 63.77°, 34.06°, 1.05°], which shows more than 4° difference compared to the datasheet value for 4$^{th}$ ply.

In step 3, the voxels of the 2D centerline projections were mapped back to 3D as a corresponding voxel subset of the tow mean surface. The centerline voxels were decimated and fitted by the 2$^{nd}$ order spline curve using SciPy spline fitting functionality [46]. The decimation (decreasing count of points) simplified the spline fitting procedure. The curve tangents (for discrete points set) were computed and then used for the orientation interpolation in tow voxels (steps 5 and 6 of the algorithm). Fig. 6b plots the resulting centerline curve (red), the points used for fitting (green) and the tow boundary profile.

In this way, the orientation for all tow voxels was extracted. The statistical analysis of the obtained orientation distribution showed that the fibers had a small out-of-plane inclination angle (less than 10 degrees for most of the voxels).

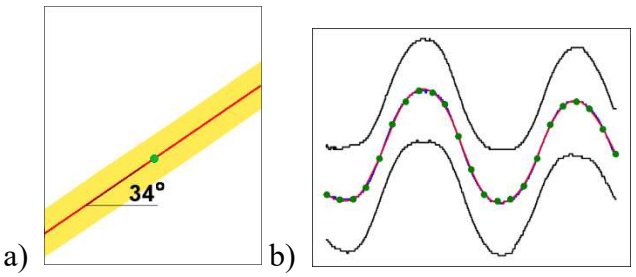



**Fig. 6.** Tow centerline in-plane projection (a) and spline interpolation (b)

## 4. Finite Element Mesh Generation

The proposed tetrahedral mesh generation workflow (for multi-materials labeled image input) includes the following main stages:

1. *image preparation;*
2. *generation of an initial interface mesh (Fig. 7c) based on multi-materials marching cubes algorithm [26];*

3. *initial surface smoothing (Fig. 7d);*
4. *surface remeshing (Fig. 7e – 7g) based on edge split/collapse/flip and tangential smoothing operations;*
5. *volumetric tetrahedral meshing (Fig. 7h) using TetGen open-source software [35].*

The algorithm steps for a test image (10x10x10 voxels, 3 materials) are visualized in Fig. 7 and described in detail in the following subsections. The algorithm was implemented in Python based on Numpy/SciPy library packages [46].

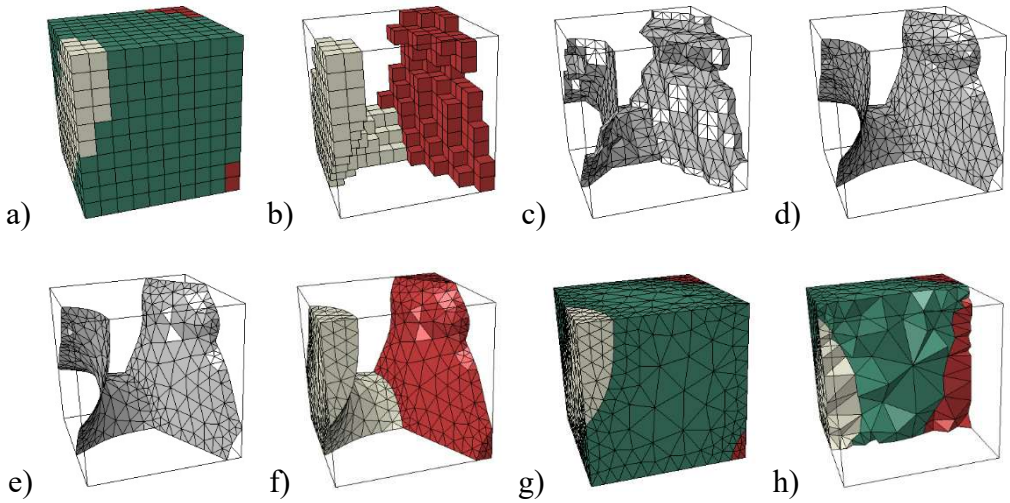



**Fig. 7.** A test input image and its internal view (a, b); initial surface (c); smoothing result (d); remeshed surfaces (e, f, g); volumetric meshing result (h)

*4.1. Image Preparation*

Often the voxel representation of geometry is ambiguous, which can lead to ambiguity or incorrect topology in the following surface reconstruction step. Such ambiguity occurs when voxels with the same label have only one common edge or vertex. For example, Fig. 8a and 8b demonstrate ambiguous configurations: if the green voxels have an identical label and the grey voxels have any label different from the green voxel label.

In this work, the ambiguities were resolved by changing color of one of two critically connected voxels to a neighboring voxel color (Fig. 8c and 8d). The changing of the voxel color can lead to new ambiguities. Therefore, the procedure was performed iteratively until all ambiguities were resolved.

Another part of the image preparation step is image boundary padding by one voxel thick layer. This layer possesses 26 new artificial labels (6 faces, 12 edges, and 8 corners) and serves to fix the image boundary features (edges and corners) during the next surface reconstruction step.

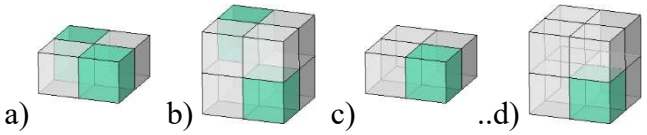



**Fig. 8.** Edge (a) and vertex (b) type of voxel ambiguity; resolving ambiguity (c, d)

*4.2. Initial Surface Reconstruction*

The interface mesh was reconstructed using a multi-material marching cubes algorithm [26]. The original algorithm converts the input voxel grid (labeled image) to a cell grid with nodes in voxel centers. Thus, one cell is a voxel-size cube with the vertexes in the centers of 8 adjacent voxels. Each cell structure stores 8 numbers to identify colors (materials) that correspond to its vertices. This algorithm has the same idea as the bi-material marching cubes algorithm (popular in computer graphics) [47] in which a set of triangles (a pattern) is predefined for all possible cell configurations as a look-up table (LUT). The LUT is used to map the image cell grid directly to the triangles which form the image interface mesh. The high performance of the bi-material marching cube algorithm is achieved based on effective storing of all possible triangle patterns in a small size LUT. In the case of two materials, the count of LUT records is 256 (2 materials with the power of 8 vertices). In the multi-materials case, a cell has 8 with the power of 8 different configurations that results in a cumbersome LUT. In this work, a method for significant reduction of the LUT size was proposed, thus, leading to performance improvement.

The triangle patterns were classified based on the count of the cell faces with a new central node (count of face nodes) [26]. The count of patterns for less than 3 face nodes was reduced from 942,880 to 7,287 unique LUT records. All basis surfaces used for the pattern construction are presented in Appendix (Fig. A).

For the special case of 4 face nodes in the same plane, we introduced cell triangulations without the new node in the cell center (demonstrated in Fig. 9). Such configuration occurs 751,800 times (3,288 unique LUT records).

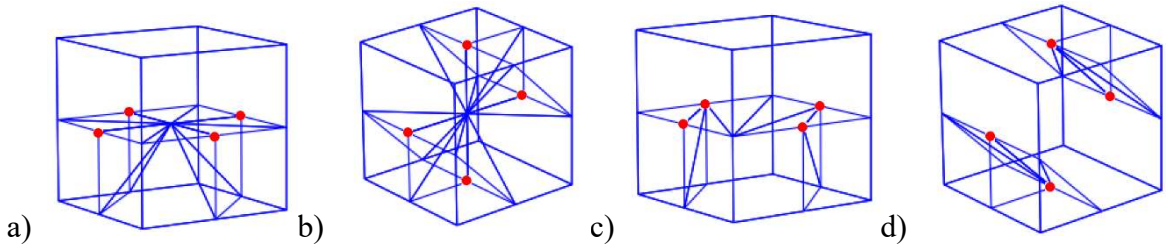



**Fig. 9.** Patterns for cells with 4 face nodes (red dots): original algorithm (a, b) and proposed modification (c, d)

The remaining cell configurations in which the algorithm adds a new node into the cell center [26], were compressed from 15,082,536 to only 256 records. That was achieved by splitting the cell triangulation into 6 parts such that each part belonged to one of the 6 cell faces. Such splitting procedure is demonstrated in Fig. 10. The resulting 256 face patterns (4 materials with the power of 4 vertices) were further reduced to only 60 unique LUT records.

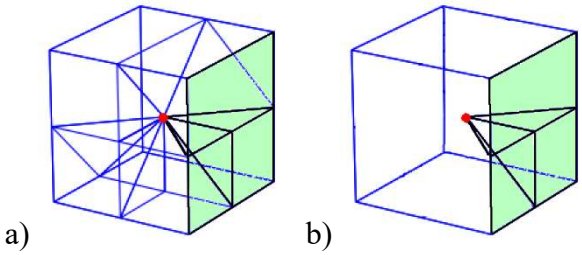



**Fig. 10.** Example of triangulation splitting: cell with central node (a), green face (sub-)pattern (b)

*4.3. Initial Surface Smoothing*

Let us introduce several terms. A manifold interface between only two materials will define a *bi-material surface*. A set of consequently connected edges that are shared between three or four material regions will form a *multi-material segment*. A node that is the endpoint of a multi-material segment will represent a *multi-material corner*. Additionally, a dihedral angle with multi-material

segment edge will be called a *multi-material dihedral angle.* Finally, a dihedral angle with an internal bi-material surface edge will be named a *bi-material dihedral angle*.

The initial surface reconstruction provided a conformal non-manifold interface mesh while keeping the original voxel size resolution. For smoothing this surface, the Laplacian method was employed:

$$X_L = \alpha X + (1-\alpha)\sum X_A /V,$$

where $X$ and $X_L$ are the coordinates of the current and updated nodes respectively; $X_A$ are the coordinates of the node that is adjacent to $X$; $V$ is the count of the adjacent nodes; $\alpha$ is the control parameter of the smoothing speed (value 0.5 was used). The smoothing operation is visualized in Fig. 11a.

During the smoothing process, the multi-material corners were always fixed, multi-material segments were smoothed independently on the bi-material surface nodes, and the bi-material surfaces were smoothed with fixed multi-material segment nodes. The minimal value of the multi-material dihedral angle was limited for preventing surface intersection and avoiding poor quality elements for FE analysis. Therefore, both smoothing operations (for multi-material segments and surfaces) were equipped with an "undo" functionality which controlled the multi-material dihedral angles and set corresponding nodes back to their previous iteration positions if their movement leads to smaller than limited dihedral angles. Additionally, during the smoothing iteration, the following mesh characteristics were limited: the minimal bi-material dihedral angle, the maximal node distance to its initial position (< 0.8 voxels), and the minimal node distance to the neighbor nodes (> 0.3 voxels).

### *4.4. Surface Remeshing*

The remeshing aimed at increasing the quality and decreasing the node count of the smoothed mesh. Here this process was applied only to the bi-material surfaces; the algorithm for optimization of the multi-material segments discretization is not currently implemented. Therefore, only remeshing of a manifold surface with fixed boundary segments (if existing) is considered in this subsection.

However, to the authors' knowledge, there are no theoretical obstacles that hinder the algorithm extension for treatment of the multi-material segments as well.

The surface remeshing algorithm iteratively repeats four local mesh processing operations: *edge split*, *edge collapse*, *edge flip*, and *tangential smoothing* [25, 31]. Basically, the *edge split* adds a new node in the middle of an edge that is longer than the desired limit; *edge collapse* removes the edge which is shorter than the desired limit; *edge flip* changes the edge adjacency if it improves triangles quality (see Fig. 11b). The *tangential smoothing* is the Laplacian smoothing with the following node projection on the surface tangent plane. It is computed according to the following formula:

$$X_T = X_L + N_V[N_V^T \cdot (X - X_L)],$$

where $N_V$ is the surface normal in point $X$ and $X_L$ is the Laplacian smoothing of $X$.

In this work, the edge length limits for each node were computed as the closest distance from this node to other bi-material surface nodes, which were not bigger than a predefined parameter $L_{max}$ and not smaller than the shortest length of the multi-material edge. The efficiency improvement of the edge flip operation was realized based on the shape factor criteria. According to that, an edge was flipped if it led to an increment of the minimal *shape factor* of its adjacent triangles. In turn, the triangle shape factor is a ratio of the triangle area and the area of the equilateral triangle with the same circumradius. Similar to the initial smoothing operation, the edge flip/collapse and the tangential smoothing operation were implemented to produce the surface with a predefined limit on the minimal bi-material dihedral angle.

At the final stage of the FE mesh generation algorithm, the surface remeshing result was used as the input boundary for the open-source software TetGen [35] to get a tetrahedral mesh while preserving the input surface. Therefore, the interface mesh was not modified by TetGen as a fixed boundary. TetGen automatically assigns attributes to tetrahedra in different regions that were used for the material properties assignment.

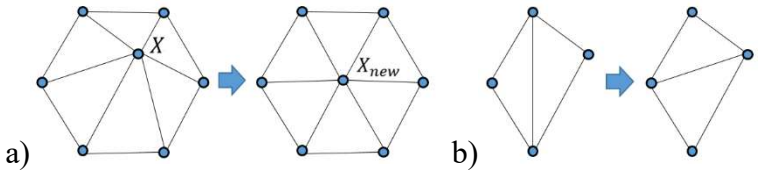



**Fig. 11.** Local mesh operation: smoothing (a), edge flip (b)

*4.5. Meshing Results*

A periodic structure cannot be extracted from the given input image because each ply has its own period. Therefore, we just cut out the tow's instance segmentation (Section 3.2) to have the maximal equal size in-plane (1696 voxels). After that, some of the boundary slices were removed as they usually have less segmentation quality. The new image size was 480x1600x1600 voxels and it was further downsampled with coefficients 0.1 and 0.125 resulting in the volumetric images of 48x160x160 and 60x200x200 voxels respectively. Then, voxel and tetrahedral meshes were generated for both input images. Further in the text, the voxel meshes will be titled vox_x100 and vox_x125 that corresponds to the downsampling coefficients of 0.1 and 0.125. Analogically, the tetrahedral mesh names will be tet_x100 and tet_x125. The choice of the downsampling coefficients is motivated by keeping the balance between the geometry accuracy on one hand and the computational efforts on the other. Only vox_x100 and tet_x100 meshes were used for FE simulation and vox_x125 and tet_x125 meshes were used to demonstrate the tetrahedral meshing results and track the influence of the downsampling coefficients on the tow volume fraction (TVF).

For the tetrahedral mesh generation, the algorithm parameters were set as follows. The count of the initial smoothing iterations was 7; the count of the remeshing iterations was 7; the minimal multi-material and bi-material dihedral angle limits were 20 and 40 degrees respectively (smoothing and remeshing steps had the same values). The maximal edge length limits $L_{max}$ was set to 3 and 4.5 voxels for generation of tet_x100 and tet_x125 meshes respectively.

The voxel and tetrahedral meshes (vox_x125 and tet_x125) are presented in Fig. 12 using Simcenter 3D visualization. The detailed mesh information is collected in Table 2. The tetrahedral

mesh quality was verified using Simcenter 3D functionality. The quality report is presented in Table 3, where the aspect ratio is the ratio between the longest and the shortest edge of an element, and the tetrahedral shape factor is defined analogically to the triangle shape factor (subsection 4.4).

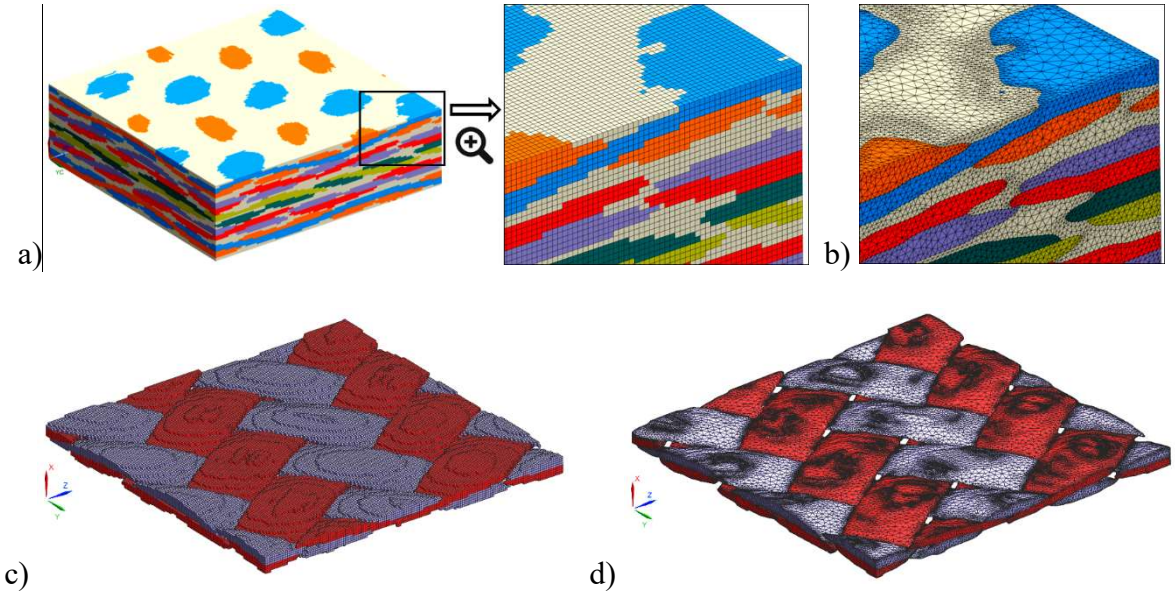



**Fig. 12.** Resulting meshes for voxel (a, c) and tetrahedral (b, d) elements; ply elements selection (c, d)

**Table 2**: Meshes size and tow's volume fraction. The first column (called original) corresponds to the original resolution image with the size of 480x1600x1600 voxels

| | original | vox_x100 | vox_x125 | tet_x100 | tet_x125 |
|---|---|---|---|---|---|
| nodes ($\cdot 10^6$) | 1233 | 1.3 | 2.5 | 0.61 | 0.87 |
| elements ($\cdot 10^6$) | 1229 | 1.2 | 2.4 | 3.48 | 4.97 |
| TVF (in %) | 70.27 | 70.35 | 70.28 | 69.62 | 69.74 |

**Table 3**: Tetrahedral mesh elements quality report

| | min faces angle | | max faces angle | | aspect ratio | | shape factor | |
|---|---|---|---|---|---|---|---|---|
| | average | worst | average | worst | average | worst | average | worst |
| tet_x100 | 36.3 | 7.2 | 90.9 | 163.0 | 1.74 | 8.23 | 0.6078 | 0.0001 |
| tet_x125 | 36.2 | 5.6 | 91.1 | 167.3 | 1.74 | 7.31 | 0.6055 | 0.0001 |

The computational time of tet_x125 mesh generation was about 75 minutes on an Intel Core i7 1.9GHz CPU and 32 Gb RAM. The time was distributed between the main steps of the algorithm as follows: the initial mesh generation (multi-material marching cubes) took 7 seconds, the Laplacian

smoothing step (7 iterations) took ~ 2 minutes, remeshing (7 iterations) took ~ 64 minutes, and, finally, TetGen used 8 minutes for the tetrahedral meshing.

## 5. Numerical Homogenization of Mechanical Properties

### *5.1. Material Properties and Boundary Conditions*

Four simulations were set up on vox_x100 and tet_x100 meshes. For each mesh type, two fields of the tow local orientations were considered. In the first case, all the vectors were located in-plane (YZ plane in Fig. 13a), the out-of-plane component of the orientation vectors was zero (*in-plane model*). In the second case, the orientation was fully 3D and followed the bundles undulation (*out-of-plane model*). The unit cell dimensions were measured from the cropped segmentation and were equal to 2.16×7.2×7.2 $mm^3$. For each model, the fiber volume fraction in the composite was computed based on the sample thickness of 2.16 mm and accounting for five layers of the fabric with the areal density of 285 $g/m^2$ [39], which resulted in a value of 37.48%. Afterwards, the fiber volume fraction inside the tows (TFVF) was calculated as 37.48/69.62≈54% for the tetrahedral mesh and 37.48/70.35≈53% for the voxel mesh. The TFVF for each FE model is reported in Table 4. The tow elastic properties were estimated using Chamis formulae [48] and the constituent properties from Table 1.

To verify the derived TFVF values, it was also evaluated as the ratio of tow cross-section fiber area and the average tow cross-section area. The approximate tow cross-section area (0.204 $mm^2$) was computed based on manual segmentation of 17 cross-sections on the initial μCT image. 3000 fibers in a tow with a diameter of 7 μm gave us 0.115 $mm^2$ of the fiber area in a tow. The resulting ratio of these areas equals ~56% that is close to the values computed based on the areal density.

Siemens Simcenter 3D Materials Engineering allows the user to predict homogenized elastic properties of a periodic UC in an easy and automated manner [5]. The software has a functionality which applies 3D periodic boundary conditions (PBCs) on FE meshes and defines six loading cases: three in tension and three in shear [49], by imposing displacements on three specially created

reference points which provide the necessary degree-of-freedoms and control the UC mechanical response. The process also supports non-identical meshes at opposite boundary faces, which is the case for the tetrahedral mesh. When solutions of the defined six cases are found, the homogenized elastic properties of the modeled composite UC can be calculated. This step is also automatically performed with the dedicated tool of the Simcenter 3D Materials Engineering.

*5.2. Stiffness Homogenization Results*

The μ-CT based models were derived from a real composite material that always has structural deviations. The latter leads to the lack of strict model periodicity making the choice of boundary conditions not obvious [6]. In this work, PBCs were still used, however, bearing in mind that this boundary condition type will result in artificial stress peaks at the unit cell boundary surfaces. In order to estimate the effect of these stress concentrations on the homogenized elastic properties, manual homogenization was performed on a voxel model with the out-of-plane orientation. The out-of-plane orientation was selected because it delivered the highest boundary stress peaks (Fig. 13c). In the manual homogenization, five boundary slices of the FEs were not considered ("cropped") in the volume stress averaging. The difference between homogenized stiffness of the full and "cropped" models was found to be negligible, less than 3.6% for Young's moduli and shear moduli, except for $G_{YZ}$ where the error was about 7%. Thus, further in this work, results obtained on the full models are reported for the sake of convenience.

The results of the FE-based stiffness homogenization on the four studied models are summarized in Table 4, where subscript indexes X, Y, and Z denote the axes of the global coordinate system (Fig. 13a). An example of the resulting stress distribution for the in-plane and out-of-plane models are depicted in Fig. 13b and 13c respectively. They illustrate shear stress (component 31) in the local tow coordinate system (1 represents the fiber direction and 3 denotes the transversal, through-yarn thickness direction). These images correspond to tensile load along the Y axis.

**Table 4: Predicted homogenized properties**

| | in-plane model | | out-of-plane model | |
|---|---|---|---|---|
| | **vox-mesh** | **tet-mesh** | **vox-mesh** | **tet-mesh** |
| *TFVF* | 53.28% | 53.84% | 53.28% | 53.84% |
| $\rho$ *[kg/m$^3$]* | 1377.60 | 1377.57 | 1377.60 | 1377.57 |
| $E_X$ *[GPa]* | 7.093 | 7.062 | 7.076 | 7.053 |
| $E_Y$ *[GPa]* | 27.962 | 28.802 | 29.527 | 31.115 |
| $E_Z$ *[GPa]* | 27.155 | 27.939 | 29.060 | 30.599 |
| $\nu_{XY}$ | 0.081 | 0.078 | 0.081 | 0.077 |
| $\nu_{YZ}$ | 0.287 | 0.290 | 0.293 | 0.297 |
| $\nu_{XZ}$ | 0.084 | 0.082 | 0.084 | 0.080 |
| $G_{XY}$ *[GPa]* | 2.348 | 2.349 | 2.460 | 2.481 |
| $G_{XZ}$ *[GPa]* | 2.346 | 2.347 | 2.469 | 2.493 |
| $G_{YZ}$ *[GPa]* | 9.633 | 9.969 | 10.268 | 10.858 |

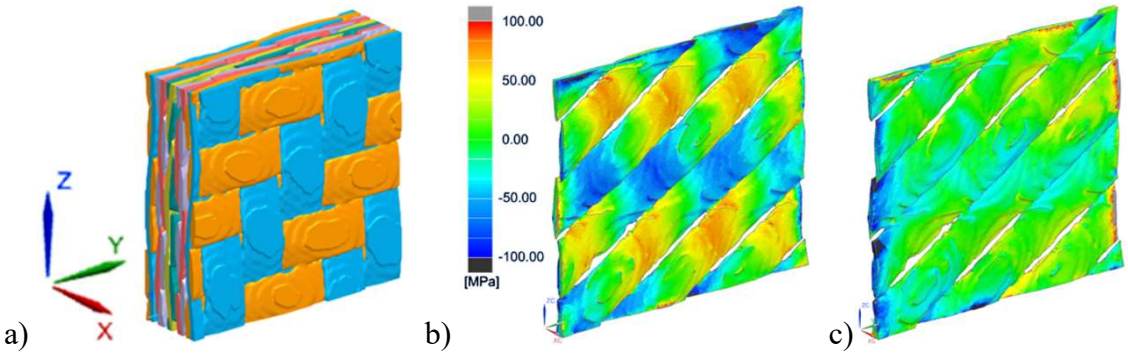



**Fig. 13.** Tet-mesh model: tow layers in the global coordinate system (a); shear stresses in the local coordinate system for in-plane (b) and out-of-plane (c) models

## 6. Discussion

The accuracy of the used deep learning approach for the semantic segmentation step was shown in the work [14]. The proposed idea in the current study to reduce the count of tow labels (clustering orientations) simplifies the neural network training and can also be adopted to a layer stack with other orientations layup. All six tow orientations (labels) were recovered in the post-processing step using three segmentations with individual neural networks training. Training the network more than once is not necessarily an issue for the current approach because the training time can be relatively short (in this research, it took about 1 hour per network).

For the watershed based individual tow segmentation, using morphological erosion appears a simpler alternative to the mean surface marks. However, the erosion-based segmentation can lead to

inaccurate tows splitting and lose small tow pieces at the image corners. On the other hand, it is very rare that the proposed mean surface segmentation in 2D loses a small corner. Thus, this small tow piece can be considered as a single structure together with a larger neighboring tow, or it can be treated in a more accurate way by detecting the local minimum at the corners using information about tow's width and orientation.

Considering the mesh generation step, one could argue that the proposed solution of the voxel ambiguity by modifying the input image is a "rough" method, and a more accurate approach could be resolving ambiguity on the sub-voxel level. However, in practice, this accuracy impact can be insignificantly small. For example, for the tet_x125 image input, only about 300 voxels (0.01%) with critical connectivity were detected which could not change the geometry or material volume fraction significantly even if all the voxels were not modified optimally.

In the resulting tetrahedral meshes, the tow volume fraction was changed very slightly when compared with the original value (see TVF row in Table 2). The difference was less than 0.1% for the voxel meshes and about 2% for the tetrahedral meshes, which was insignificant and could be corrected at the FE modeling stage by adjusting the fiber volume fraction in the tows. In general, preserving the material volume fraction of the input (segmented) image during the interface reconstruction and smoothing/remeshing steps remains a challenging problem. However, the volume fraction accuracy can be improved, for example, by introducing a surface curvature-based sizing field [50] in the remeshing algorithm.

The further development of the remeshing algorithm can include the multi-material segments processing for reducing the node count in the resulting FE mesh. Additionally, the overall mesh generation performance can be improved by implementing it in C++ instead of Python. Currently, most of the computation time was spent on the remeshing step. The edge split/collapse/flip operations are very time-consuming in the Python implementation because of loops over all the mesh edges.

The numerical homogenization results showed that the computed elastic properties were consistent between the considered mesh types. The homogenized elastic moduli were slightly bigger

(less than 5%) for the tetrahedral mesh, which could be attributed to a more accurate composite geometry representation. As a result of a more realistic material orientation, the out-of-plane FE models had a slightly higher stiffness compared to the in-plane models (Table 4). In the latter, dominant deformation in the fiber bundles occurred in shear due to a smaller in-plane shear stiffness of the tows compared to their longitudinal stiffness. The tow undulation included in the out-of-plane models prevented such unrealistic shear deformation (Fig. 13b, 13c).

## 7. Conclusions

A new image segmentation method for processing μ-CT data of 2D textile composites with multiple layer orientations layup was proposed in this study. The method successfully used a deep learning approach for the semantic segmentation of collinear tows. After that, the individual tow's instances were segmented with high accuracy based on the mean surface splitting algorithm. The material's orientation field was computed using tow's centerlines approximation by spline curves.

The segmentation result was used as an input for the FE mesh generation algorithm. The algorithm combined the following main steps: resolving voxel ambiguity, initial mesh reconstruction, smoothing, remeshing and tetrahedral meshing. The multi-material marching cubes algorithm was improved and used here for the initial mesh reconstruction. The classical Laplacian smoothing algorithm was adapted to process the non-manifold reconstruction result avoiding surface self-intersection and sharp dihedral angle issues. Then, the local mesh processing operation was applied for the non-manifold surface remeshing, which significantly decreased the node count of the initial mesh reconstruction result. Finally, the interface surface mesh was used as input for the open-source software TetGen to provide the tetrahedral mesh for FE modeling. This shows the efficiency of the proposed algorithm for the generation of the realistic FE model with high-quality mesh from the accurate image segmentation result.

The segmentation and the mesh generation output were used to generate FE models and to predict homogenized stiffness of the B-pillar composite sample using Siemens Simcenter 3D

Materials Engineering. The FE simulation results for the models based on the voxel and tetrahedral meshes were compared.

**Acknowledgement**

This work was funded by the Research Foundation — Flanders in the Strategic Basic Research Programme (FWO-SBO), file number S003418N.

Siemens Industry Software NV acknowledges SIM (Strategic Initiative Materials in Flanders) and VLAIO (Flemish government agency Flanders Innovation & Entrepreneurship) for their support of the collaborations with: (1) Flanders Make, Sirris Leuven-Gent Composites Application Lab and Com&Sense in creating the B-pillar component in the context of the SBO (n°120024) and ICON (n°150533) projects and (2) TESCAN for acquiring μ-CT images of the B-pillar in the framework of the Innovation Mandate (HBC.2017.0189). **[TOADD** ENSMA Acknowledgement**]**

**Declaration of competing interest**

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

**References**

[1] M. Mehdikhani, N.A. Petrov, I. Straumit, A.R. Melro, S.V. Lomov, L. Gorbatikh, The effect of voids on matrix cracking in composite laminates as revealed by combined computations at the micro-and meso-scales, Composites Part A: Applied Science and Manufacturing 117 (2019) 180-192.
[2] I. Straumit, Prediction of the effective properties of textile composites based on X-ray computed tomography data, KU Leuven, Leuven, Belgium, 2017.
[3] A. Vanaerschot, Multi-scale modelling of spatial variability in textile composites. Uncertainty quantification based on experimental data of internal geometry, KU Leuven, Leuven, Belgium, 2014.
[4] S. Cortelli, VirtualCT: realistic composite material modeling using micro-CT-based voxel approach [online] https://community.plm.automation.siemens.com/t5/Simcenter-Blog/VirtualCT-realistic-composite-material-modeling-using-micro-CT/ba-p/592131, May 09, 2019. (Accessed May 15 2019).
[5] O. Shishkina, A. Matveeva, S. Wiedemann, K. Hoehne, M. Wevers, S. Lomov, L. Farkas, X-Ray Computed Tomograhy-based FE-Homogenization of sheared organo sheets, Proc. of the 18th European Conf. on Composite Materials, Athens, Greece, 2018.

[6] W. Wijaya, P. Kelly, S. Bickerton, A novel methodology to construct periodic multi-layer 2D woven unit cells with random nesting configurations directly from μCT-scans, Composites Science and Technology (2020) 108125.
[7] W. Huang, R. Xu, J. Yang, Q. Huang, H. Hu, Data-driven multiscale simulation of FRP based on material twins, Composite Structures 256 (2021) 113013.
[8] N. Naouar, E. Vidal-Sallé, J. Schneider, E. Maire, P. Boisse, Meso-scale FE analyses of textile composite reinforcement deformation based on X-ray computed tomography, Composite structures 116 (2014) 165-176.
[9] I. Straumit, S.V. Lomov, M. Wevers, Quantification of the internal structure and automatic generation of voxel models of textile composites from X-ray computed tomography data, Composites Part A: Applied Science and Manufacturing 69 (2015) 150-158.
[10] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, International Conference on Medical image computing and computer-assisted intervention, Springer, 2015, pp. 234-241.
[11] P. Galvez-Hernandez, K. Gaska, J. Kratz, Phase Segmentation of Uncured Prepreg X-Ray CT Micrographs, Composites Part A: Applied Science and Manufacturing (2021) 106527.
[12] H. Yang, W.-F. Wang, J.-C. Shang, P.-D. Wang, H. Lei, H.-s. Chen, D.-N. Fang, Segmentation of computed tomography images and high-precision reconstruction of rubber composite seal based on deep learning, Composites Science and Technology (2021) 108875.
[13] Y. Chen, Y. Chen, D. Wang, S. Ai, Generating 3D digital material twins for woven ceramic-matrix composites from μCT images, Journal of the American Ceramic Society n/a(n/a).
[14] Y. Sinchuk, P. Kibleur, J. Aelterman, M.N. Boone, W. Van Paepegem, Variational and Deep Learning Segmentation of Very-Low-Contrast X-ray Computed Tomography Images of Carbon/Epoxy Woven Composites, Materials 13(4) (2020) 936.
[15] M.A. Ali, Q. Guan, R. Umer, W.J. Cantwell, T. Zhang, Deep learning based semantic segmentation of μCT images for creating digital material twins of fibrous reinforcements, Composites Part A: Applied Science and Manufacturing 139 (2020) 106131.
[16] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation, Proceedings of the European conference on computer vision (ECCV), 2018, pp. 801-818.
[17] S. Blusseau, Y. Wielhorski, Z. Haddad, S. Velasco-Forero, Instance segmentation of 3D woven fabric from tomography images by Mathematical Morphology and Deep Learning methods, (2021).
[18] A. Mendoza, R. Trullo, Y. Wielhorski, Descriptive modeling of textiles using FE simulations and deep learning, Composites Science and Technology 213 (2021) 108897.
[19] Y. Sinchuk, P. Kibleur, J. Aelterman, M.N. Boone, W. Van Paepegem, Geometrical and deep learning approaches for instance segmentation of CFRP fiber bundles in textile composites, Composite Structures 277 (2021) 114626.
[20] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, Proceedings of the IEEE international conference on computer vision, 2017, pp. 2961-2969.
[21] A. Rassineux, Robust conformal adaptive meshing of complex textile composites unit cells, Composite Structures (2021) 114740.
[22] Y. Sinchuk, Y. Pannier, R. Antoranz-Gonzalez, M. Gigliotti, Analysis of moisture diffusion induced stress in carbon/epoxy 3D textile composite materials with voids by μ-CT based Finite Element Models, Composite Structures 212 (2019) 561-570.
[23] A. Doitrand, C. Fagiano, F.-X. Irisarri, M. Hirsekorn, Comparison between voxel and consistent meso-scale models of woven composites, Composites Part A: Applied Science and Manufacturing 73 (2015) 143-154.

[24] H.-C. Hege, M. Seebass, D. Stalling, M. Zöckler, A generalized marching cubes algorithm based on non-binary classifications, (1997).
[25] C. Jamin, P. Alliez, M. Yvinec, J.-D. Boissonnat, CGALmesh: a generic framework for delaunay mesh generation, ACM Transactions on Mathematical Software (TOMS) 41(4) (2015) 1-24.
[26] Z. Wu, J.M. Sullivan Jr, Multiple material marching cubes algorithm, International Journal for Numerical Methods in Engineering 58(2) (2003) 189-207.
[27] G. Grail, M. Hirsekorn, A. Wendling, G. Hivet, R. Hambli, Consistent Finite Element mesh generation for meso-scale modeling of textile composites with preformed and compacted reinforcements, Composites Part A: Applied Science and Manufacturing 55 (2013) 143-151.
[28] M.Y. Matveev, L.P. Brown, A.C. Long, Efficient meshing technique for textile composites unit cells of arbitrary complexity, Composite Structures 254 (2020) 112757.
[29] A. Mazumder, Y. Wang, C.-F. Yen, A structured method to generate conformal FE mesh for realistic textile composite micro-geometry, Composite Structures 239 (2020) 112032.
[30] G. Zhou, Q. Sun, D. Li, Z. Meng, Y. Peng, Z. Chen, D. Zeng, X. Su, Meso-scale modeling and damage analysis of carbon/epoxy woven fabric composite under in-plane tension and compression loadings, International journal of mechanical sciences 190 (2021) 105980.
[31] N. Faraj, J.-M. Thiery, T. Boubekeur, Multi-material adaptive volume remesher, Computers & Graphics 58 (2016) 150-160.
[32] P.-O. Persson, G. Strang, A simple mesh generator in MATLAB, SIAM review 46(2) (2004) 329-345.
[33] N. Naouar, E. Vidal-Salle, J. Schneider, E. Maire, P. Boisse, 3D composite reinforcement meso FE analyses based on X-ray computed tomography, Composite Structures 132 (2015) 1094-1104.
[34] B. Wintiba, B. Sonon, K.E.M. Kamel, T.J. Massart, An automated procedure for the generation and conformal discretization of 3D woven composites RVEs, Composite structures 180 (2017) 955-971.
[35] H. Si, TetGen, a Delaunay-based quality tetrahedral mesh generator, ACM Transactions on Mathematical Software (TOMS) 41(2) (2015) 1-36.
[36] S. Donders, L. Farkas, H. Erdelyi, F. Souza, H. Cornwell, Virtual material and product design. Simcenter 3D helps manufacturers comply with regulations and meet customer expectations [Online] https://siemens.highspot.com/items/609a8240659e934c7e2aefba#1, May 11, 2021. (Accessed Nov. 04 2021).
[37] Siemens Industry Software, Achieving lighter vehicles with the use of composites [online] https://blogs.sw.siemens.com/simcenter/Achieving-lighter-vehicles-with-the-use-of-composites/, Aug. 21, 2019.
[38] K. Vanclooster, Infusion simulation of the RTM process with experimental validation by using integrated optical sensors during the manufacturing, EuroMech (Colloquium 602: Composite manufacturing processes. Analyses, modelling and simulations), Lyon, France, Mar. 13, 2019.
[39] Chomarat, C-WEAVETM 285T 3K HS. Data sheet. [online] https://composites.chomarat.com/wp-content/uploads/sites/2/2017/03/CW035-2.pdf. (Accessed Oct. 20 2021).
[40] Toray Industries, TORAYCATM Carbon Fiber. Data Sheet. [online] https://www.cf-composites.toray/resources/data_sheets/pdf/ds_torayca_yarn.pdf, Mar. 08, 2021. (Accessed Oct. 20 2021).
[41] Sicomin, SR 1710 Injection / SD 882x. Structural epoxy system for Resin Transfer Moulding. Data sheet. [online] http://sicomin.com/datasheets/product-pdf93.pdf, Sep. 01, 2019. (Accessed Jun. 14 2021).

[42] H. Miyagawa, C. Sato, T. Mase, E. Drown, L.T. Drzal, K. Ikegami, Transverse elastic modulus of carbon fibers measured by Raman spectroscopy, Materials Science and Engineering: A 412(1-2) (2005) 88-92.
[43] D. Garoz, F. Gilabert, R. Sevenois, S. Spronk, W. Van Paepegem, Material parameter identification of the elementary ply damage mesomodel using virtual micro-mechanical tests of a carbon fiber epoxy system, Composite Structures 181 (2017) 391-404.
[44] C.A. Schneider, W.S. Rasband, K.W. Eliceiri, NIH Image to ImageJ: 25 years of image analysis, Nature methods 9(7) (2012) 671-675.
[45] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, Scikit-learn: Machine learning in Python, the Journal of machine Learning research 12 (2011) 2825-2830.
[46] P. Virtanen, R. Gommers, T.E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, SciPy 1.0: fundamental algorithms for scientific computing in Python, Nature methods 17(3) (2020) 261-272.
[47] W.E. Lorensen, H.E. Cline, Marching cubes: A high resolution 3D surface construction algorithm, ACM siggraph computer graphics 21(4) (1987) 163-169.
[48] C.C. Chamis, Simplified composite micromechanics equations for hygral, thermal and mechanical properties, (1983).
[49] A. Matveeva, D. Garoz, R. Sevenois, M. Zhu, L. Pyl, W. Van Paepegem, L. Farkas, Effect of intra-ply voids on the homogenized behavior of a ply in multidirectional laminates, IOP Conference Series: Materials Science and Engineering, IOP Publishing, 2018, p. 012009.
[50] M. Dunyach, D. Vanderhaeghe, L. Barthe, M. Botsch, Adaptive remeshing for real-time mesh deformation, Eurographics 2013, The Eurographics Association, 2013.

## Appendix

Two groups of basis surfaces for the multi-material marching cubes algorithm are presented in Fig. A, where numbers from 1 to 8 indicate the count of the boundary segments. It includes surfaces of the cell without face nodes and with exactly two face nodes. Any other surface of these groups can be obtained by rotation, symmetrical flipping and combination of the presented surfaces.

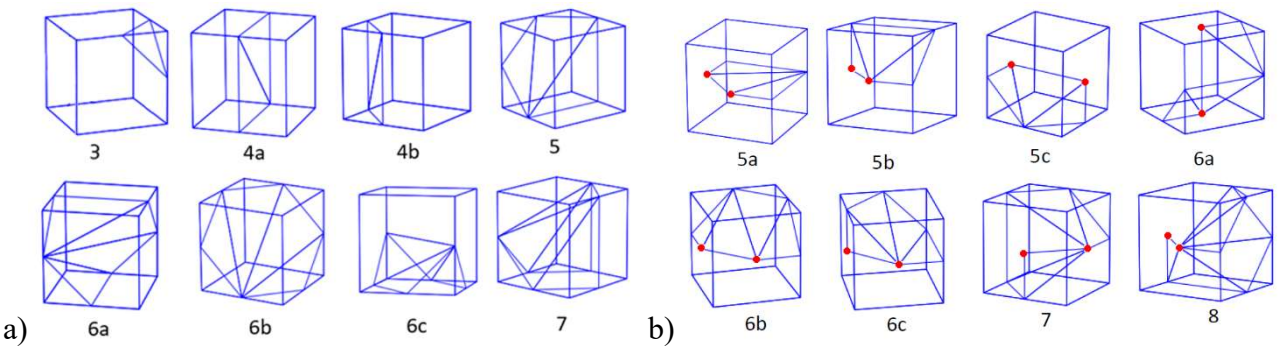



**Fig. A.** Surfaces of the cell without face node (a) and with 2 face nodes (marked by red dots) (b)

## Figure Captions

## Tables