Composite Structures 254 (2020) 112757

Contents lists available at ScienceDirect

Composite Structures

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

Efficient meshing technique for textile composites unit cells of arbitrary complexity

Mikhail Y. Matveev [*], Louise P. Brown, Andrew C. Long

Composites Research Group, Faculty of Engineering, Advanced Manufacturing Building, Jubilee Campus, The University of Nottingham, Nottingham, UK

ARTICLE INFO

Keywords:
Textile composites
Finite element modelling
Meshing

ABSTRACT

Meso-scale unit cell models are often used to simulate mechanical behaviour of textile composites. Apart from reliable ways to create *meso*-scale geometries, such simulations require reliable meshing algorithms. While the former is made possible via dedicated textile pre-processors or high-fidelity weaving simulations, the meshing remains quite problematic for complex textiles and geometries. Even though, with a lot of user input, it is possible to create very complex meshes using meshing pre-processors, this approach remains infeasible for cases when a large number of models need to be analysed.

This paper presents a meshing approach based on the combination of local octree-refinement with surface smoothing. This allows nearly conformal meshes to be generated for geometries of any complexity which achieve accuracy comparable to that of conformal meshes. A range of unit cells was analysed using the new approach and it was shown that the error in local stresses is within 10% of the reference solution and the average error is below 7%. It was found that the computational cost of the analysis using the new meshing technique is not considerably higher than for an analysis which uses a conventional conformal mesh yet the new approach allows analysis of any geometry.

1. Introduction

Most of the numerical methods for modelling of composites with textile reinforcements rely on *meso*-scale modelling techniques. Therefore, generation of finite element (FE) meshes is an essential step in these procedures. Generic problems, arising during mesh generation, have been described in one of the seminal books on tetrahedral meshing [1] (all-hexahedral meshing still remains a very challenging problem [2]) and can be summarised as the requirement on elements to respect internal and external boundaries, have good shape and not to be too numerous. These requirements on the meshes are closely linked to the initial geometry and its features such as curvature of surfaces, closeness of the internal and external surfaces to each other and the overall size of the geometry. Meso-scale geometries of textile composites contain two of these features: high double curvature of yarn surfaces and yarns being close to or touching each other as shown in Fig. 1. The latter often triggers excessive local mesh refinement and therefore leads to a large number of elements. The aforementioned problems and also this paper focus only on the meshing itself and leave aside problems such as generation of CAD geometry and its accuracy and consistency.

A new technique presented in this paper can create a near-conformal mesh for complex geometries ensuring control over the minimum/maximum size of elements as well as their total number. The algorithm, which employs local octree refinement of voxel meshes as well as mesh smoothing, is built into the TexGen textile pre-processor [4] which makes it immediately available for many types of geometries and performs automatic assignment of local fibre orientation and local fibre volume fraction within the yarns. An example of a mesh generated with this algorithm is shown in Fig. 2.

In principle, a tetrahedral mesh generation algorithm can generate a mesh for almost any geometry. In practice, the meshing algorithms often struggle to mesh matrix domains which occur in modelling of textile composites, in particular narrow regions between two yarns, because of the limitations on the minimum size of element or on the total number of elements (from either memory restrictions or solution time). Setting a mesh tolerance equal to the minimum distance between the yarns does not always solve the problem as it can create volumes of odd shape. Alternatively, manual interventions in the meshing are not desirable as it makes the approach impractical when a large number of models need to be analysed. The meshing of textile composite unit cells has been addressed with various approaches.

[*] Corresponding author.
E-mail address: Mikhail.Matveev@nottingham.ac.uk (M.Y. Matveev).

https://doi.org/10.1016/j.compstruct.2020.112757
Received 7 November 2019; Revised 12 March 2020; Accepted 27 July 2020
Available online 31 July 2020
0263-8223/© 2020 The Authors. Published by Elsevier Ltd.

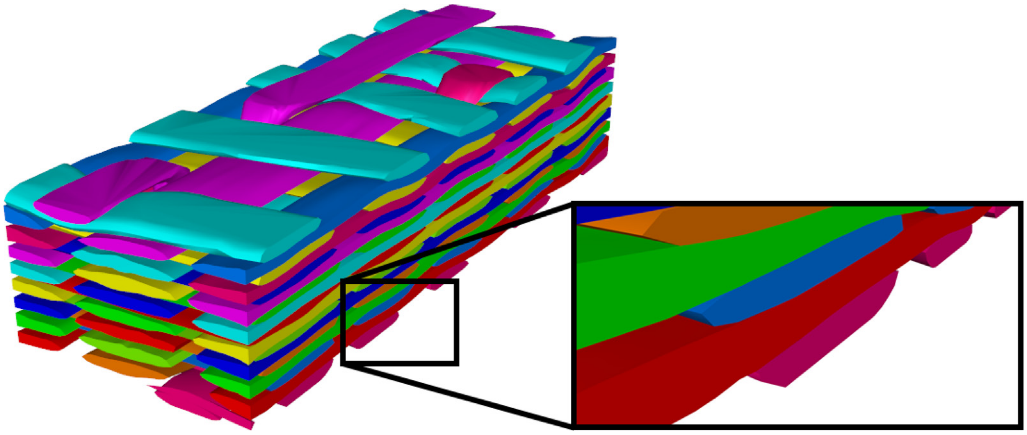


Fig. 1. Realistic unit cell of a 3D woven textile (the model used by Green et al. [3]).

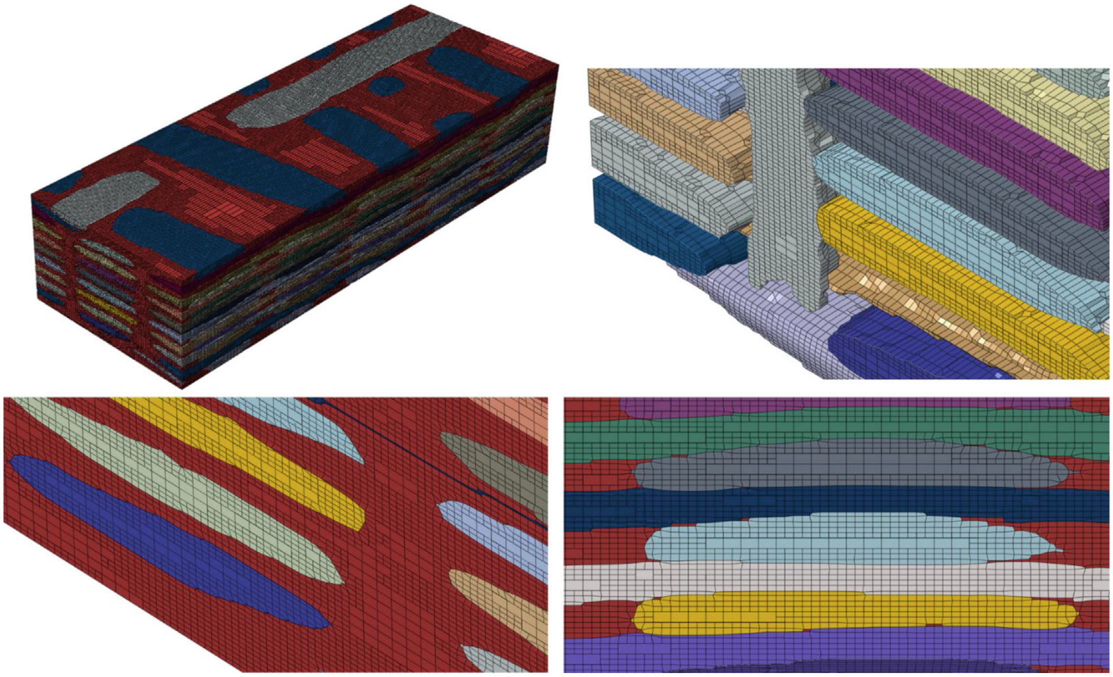


Fig. 2. Mesh of a unit cell of a complex 3D woven composite.

Early research in modelling of textile composites used relatively simple geometry with no matrix pockets which made mesh generation relatively simple [5,6]. The next generation of models [7–9] was based on more realistic models which mainly generated the yarn path using a more complex curve than a trigonometric or piece-wise function. This geometry very often gave complex matrix pockets in between the yarns which were often enlarged to make the meshing easier [8,9]. However, introduction of these pockets into the model makes it less realistic not only because of the changed geometry but also because of the reduction of the yarn volume which needs to be compensated by increase of intra-yarn fibre volume fraction. These difficulties with meshing of complex textile geometries were addressed by several approaches: voxel meshing [10], mesh-free methods [11], embedded meshes (also known as the mesh superposition method) [12] as well as various manipulations with geometry to achieve a high-quality conformal mesh [13,14].

The embedded mesh approach allows the matrix mesh to be as simple as a mesh of a domain with no inclusions. A mesh of yarns is generated using a simple mesh sweeping method and then linked to the domain mesh through a set of linear constraints for the nodes of both meshes. Ease of meshing comes at a cost of stress and strain discontinuity at the boundary of the two meshes and absence of an actual surface between a yarn and matrix.

Generation of high-quality conformal meshes requires the issue with the geometry in the zone where yarns approach each other to be addressed. Grail et al. [13] employed textile compaction simulations to eliminate the gap between yarns and then re-meshed each of the yarns to achieve a conformal mesh. A similar approach was used by Ha et al. [14] who generated the geometric model with allowed intersections which were then eliminated using a special algorithm to create a contact area between yarns. This contact area ensures that the yarn meshes are conformal in this area and that meshes do not contain ill-shaped elements. This geometry preparation procedure applied to a textile with yarns of regular elliptical cross-section resulted in a matrix domain which can be meshed with tetrahedral elements. In summary, both of the approaches rely on a tetrahedral meshing algo-

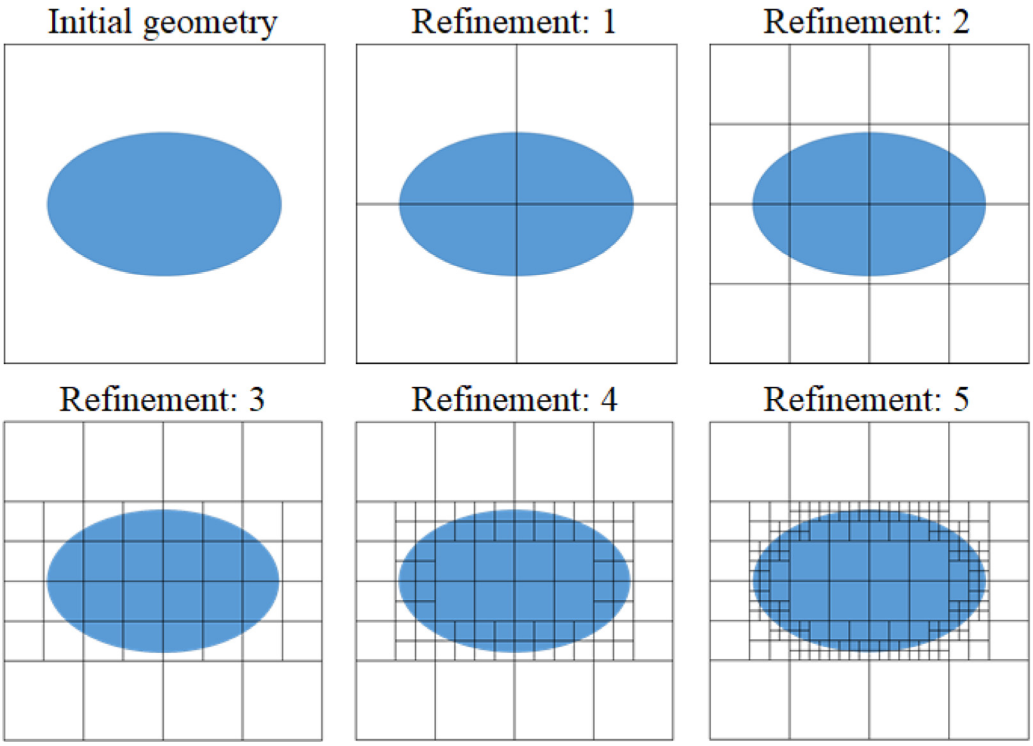


Fig. 3. Octree refinement process.

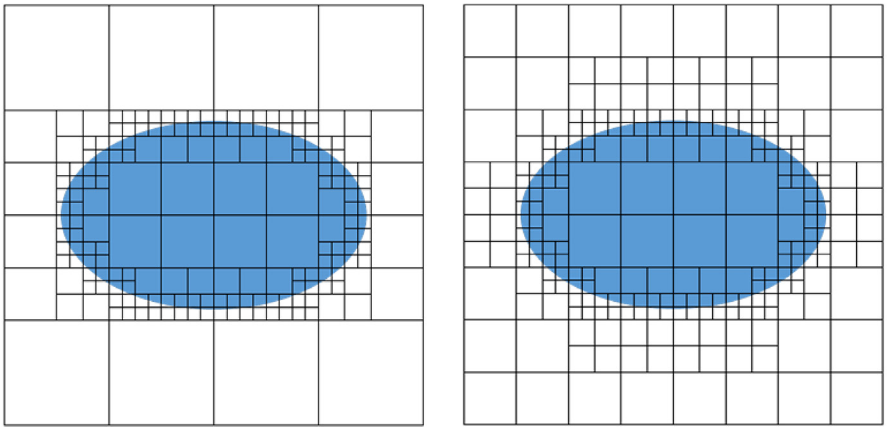


Fig. 4. A non-balanced octree mesh for an elliptic inclusion with a maximum level of refinement of 5 and minimum level of refinement of 2 (left); A balanced octree mesh with 2:1 transition between elements (right).

rithm to generate a tetrahedral mesh using surface meshes of yarns and periodic meshes for domain boundaries as input meshes. While this works in many cases, in the authors' experience, such software as Hypermesh, a commercial software for mesh generation, and TetGen [15], an open-source software, are not always capable of generating a mesh mainly due to the constraints imposed to create periodic boundary meshes. The voxel mesh approach works relatively well for simple textile geometries where voxel elements are mainly aligned with the yarns and load is applied in the yarn direction [16]. The voxel mesh accuracy degrades for geometries not aligned with the voxel mesh therefore more elements or a smoothing technique [17] are required. Increasing the mesh density in a voxel mesh is very inefficient because of the uniform element size. This was addressed by Kim and Swan [10] who implemented local refinement of voxel meshes for textile composites. It was shown that it is possible to reduce the total number of elements without reducing the accuracy of the FE results. However, this refinement still had a jagged surface and local stress concentrations owing to the nature of the voxel mesh. Other studies on voxel meshes (especially those obtained from medical imaging) employed Laplacian smoothing and other algorithms to create surfaces which would be relatively smooth and have a surface close to the original surface.

This paper combines both of these ideas – local octree refinement to approximate the boundaries of yarns as closely as possible, followed by surface smoothing to reduce the effect of stress concentration. The approach is described in detail in Section 2 which is followed by validation of the technique in Section 3. A particular focus is given in Section 4 on comparison of the proposed technique with conventional conformal meshes in terms of accuracy and computational costs. Summary and the plans for future developments of the method are given in Section 5.

2. Octree mesh refinement and mesh smoothing

2.1. Octree refinement of a voxel mesh

Typically mesh refinement is applied to reduce the numerical error introduced by mesh discretisation. Experience shows that surfaces with high curvature need to be discretised with smaller elements otherwise spurious stress concentrations may occur. The refinement employed here aims to represent the geometry as closely as possible but does not aim to have a conformal mesh (where nodes fall exactly on the surfaces of the internal geometry e.g. yarns). The mesh algorithm starts with a simple voxel mesh generated for a unit cell. A refinement algorithm is then applied to each voxel. A voxel is refined by splitting into eight voxels if its nodes belong to different materials. The dimensions of the new voxels are half that of the original voxels as shown in Fig. 3. This process is repeated iteratively until a given level of refinement is achieved.

The mesh obtained using the described refinement procedure can be conveniently stored in an octree data structure, hence its name "octree mesh refinement". An open-source library *p4est* [18] was employed for this purpose. Storing a refined mesh as an octree also makes it simple to balance the mesh i.e. always have 2:1 transition between elements instead of an arbitrary transition as shown in Fig. 4. The mesh refinement implemented in this paper also ensures that the interface between the materials is always represented by elements with the highest level of refinement.

The octree refinement in this paper was implemented such that both the size of elements in the starting voxel mesh and the refined mesh are proportional to the size of the unit cell as $1/2^n$ where n is an integer number between the starting level of refinement and the maximum level of refinement. However, this condition is easy to relax if a geometry is split into arbitrary domains which are then individually meshed using the octree technique. Examples of various octree refinements are shown in Fig. 5. The meshes in this paper will be referred to by their minimum and maximum levels of refinement e.g. suffix '4-8' denotes a mesh with minimum level of refinement of 4 and maximum level of refinement of 8.

The octree mesh refinement allows the mesh to be as close to the real interface as needed without creating additional elements away from an interface. For example, achieving a $1/(2^8)$ precision for the geometry shown in Fig. 5 ('5_8' mesh of elliptic inclusion) requires 675,200 elements with the octree mesh refinement instead of 16,777,216 i.e. $(2^8)^3$ elements for a regular voxel mesh with the same minimum element size.

None of the meshes shown in Fig. 5 are conformal and have multiple hanging nodes, i.e. nodes inside a mesh domain but not connected to all the surrounding elements. Multi-point constraints are used to enable the continuity of the mesh by constraining the hanging nodes to the nodes of larger neighbouring elements. Since the octree is balanced all the transitions between elements are 2:1 and therefore the constraints for hanging nodes which fall on edges of linear hexahedral elements (e.g. C3D8 in Abaqus) can be written as [19]:

$$\boldsymbol{u}_{hanging} - 0.5\left(\boldsymbol{u}^{I}_{master} + \boldsymbol{u}^{II}_{master}\right) = 0 \tag{1}$$

For the nodes which fall on faces the constraints are:

$$\boldsymbol{u}_{hanging} - 0.25\left(\boldsymbol{u}^{I}_{master} + \boldsymbol{u}^{II}_{master} + \boldsymbol{u}^{III}_{master} + \boldsymbol{u}^{IV}_{master}\right) = 0 \tag{2}$$

where $\boldsymbol{u}_{hanging}$ is the vector of displacement of the hanging node and $\boldsymbol{u}^{i}_{master}$ are the vectors of displacement of nodes of a master element.

Constraints similar to (1) and (2) can be written for quadratic elements.

2.2. Mesh smoothing

After octree mesh refinement the refined meshes still exhibit one of the main disadvantages of voxel meshes as they have jagged surfaces which create artificial stress concentrations. The problem can be resolved by applying a smoothing algorithm to the surface nodes. Possible smoothing algorithms include Laplacian smoothing and Laplacian Eigen-decomposition [20]. This work uses Gaussian filtering method for smoothing of meshes as described e.g. by Taubin [21]:

$$p_i = p_i + \lambda L(p_i) \tag{3}$$

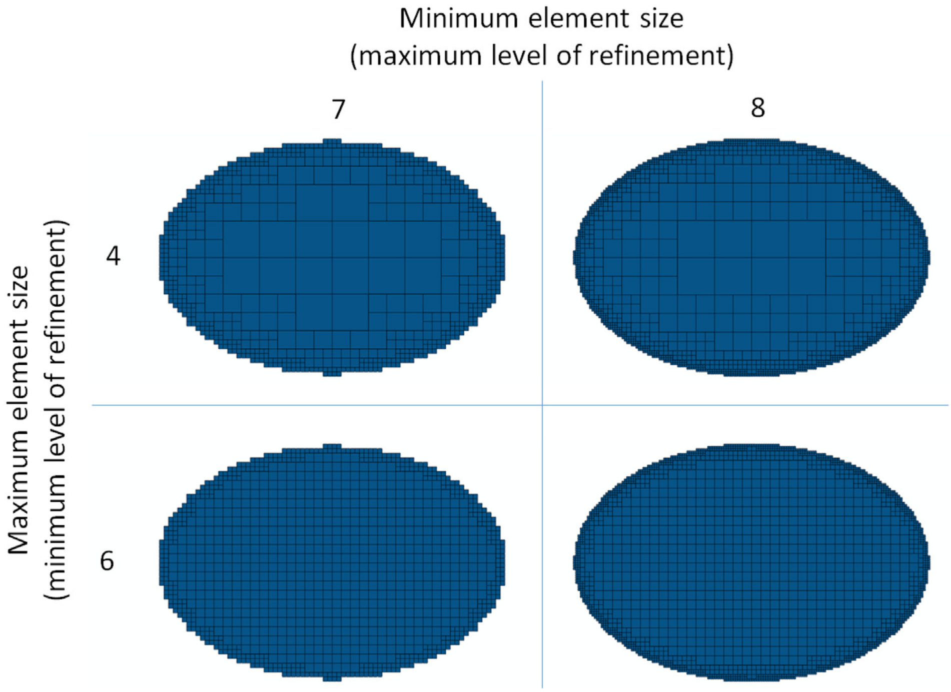


Fig. 5. Examples of octree meshes of an ellipse.

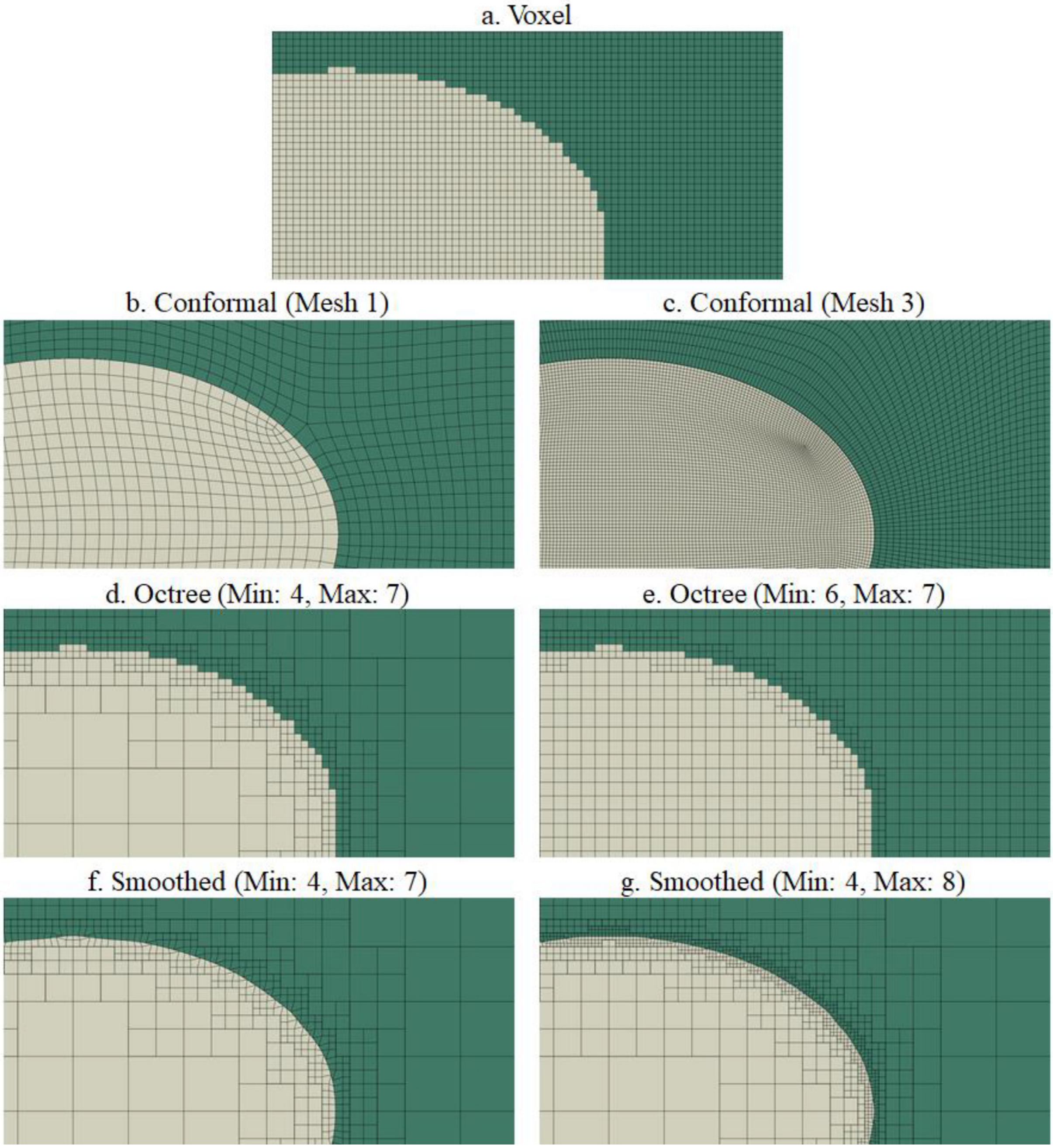

a. Voxel
b. Conformal (Mesh 1)
c. Conformal (Mesh 3)
d. Octree (Min: 4, Max: 7)
e. Octree (Min: 6, Max: 7)
f. Smoothed (Min: 4, Max: 7)
g. Smoothed (Min: 4, Max: 8)

Fig. 6. Example of conformal, voxel, octree-refined and smoothed octree-refined meshes: a. Voxel mesh; b. Conformal mesh (Mesh 1 in Table 1); c. Conformal mesh (Mesh 3 in Table 1); d. Octree-refined mesh (minimum level of refinement 4, maximum – 7); e. Octree-refined mesh (minimum level of refinement 6, maximum – 7); f. Smoothed octree-refined mesh (minimum level of refinement – 4, maximum – 7, smoothing parameter 0.3); g. Smoothed octree-refined mesh (minimum level of refinement – 4, maximum – 8, smoothing parameter 0.3).

Table 1
Convergence of conformal meshes.

Mesh	Number of elements	Max. stress, MPa	$\widehat{\sigma}_{max}$, %	NRMSD 1, % (along major axis)	NRMSD 2, % (outside of inclusion)	NRMSD 3, % (inside inclusion)
1	269,504	33,260	4.43	2.18	0.39	0.068
2	2,183,680	33,930	2.56	0.75	0.32	0.022
3	3,100,188	34,800	–	–	–	–

where $0 < \lambda < 1$ is a weight and $L(p_i)$ is a Laplacian operator whose simplest definition can be written as, essentially, a weighted distance between the node p_i and N neighbouring nodes p_j:

$$L(p_i) = \frac{1}{N}\sum_{j=1}^{N}\left(p_j - p_i\right) \tag{4}$$

Table 2
Convergence of voxel meshes.[a]

Refinement	Number of elements	$\widehat{\sigma}_{max}$, %	NRMSD 1, % (along major axis)	NRMSD 2, % (outside of inclusion)	NRMSD 3, % (inside inclusion)
64 × 64 × 64	262,144	25.03	2.03	7.59	5.89
128 × 128 × 128	2,097,152	31.12	1.63	1.30	1.59
256 × 256 × 256	16,777,126	Not feasible to run			

[a] These meshes are equivalent to octree refinement with minimum and maximum refinement equal to: 6 and 6, 7 and 7, 8 and 8.

Table 3
Convergence of octree and smoothed octree meshes.

Refinement		Number of elements	Max. stress error, %		NRMSD 1, % (along major axis)		NRMSD 2, % (outside of inclusion)		NRMSD 3, % (inside inclusion)	
Max	Min		Octree	Smoothed	Octree	Smoothed	Octree	Smoothed	Octree	Smoothed
7	4	164,032	31.26	12.09	1.66	1.75	1.32	0.54	1.56	0.53
	5	183,296	31.29	12.09	1.64	1.73	1.31	0.52	1.56	0.52
	6	376,832	31.32	12.04	1.63	1.72	1.32	0.53	1.56	0.52
8	4	657,280	34.08	8.73	0.47	0.30	0.79	0.39	0.84	0.27
	5	675,200	33.93	8.76	0.39	0.16	0.78	0.39	0.83	0.25
	6	864,256	33.96	8.76	0.38	0.14	0.78	0.38	0.83	0.25
	7	2,555,904	33.91	8.71	0.42	0.17	0.80	0.40	0.82	0.25

The smoothing operator defined by Eq. (4) is the simplest definition as it assumes equal weights between the nodes. The operator can be applied iteratively with coefficient λ taking different values at each step.

Smoothing algorithms have been applied to smooth voxel meshes before but always starting with a regular voxel mesh which is often far from an accurate representation of the boundaries. This leads either to excessive distortions of elements and geometry or to the significant reduction of the geometry volume. Applying the smoothing algorithm to an octree refined mesh resolves the second problem as the elements on the surface are small and their reduction will not result in a significant change of volume. Excessive distortion of the elements is avoided by imposing a threshold on how much a node can move from its original position.

3. Validation of the new technique

3.1. Model of an elliptic cylinder inclusion

An elliptic cylinder inclusion in a matrix has been selected as the simplest example to demonstrate the validity of the approach. The cross-section of the inclusion was an ellipse with major and minor axes equal to 0.3 mm and 0.2 mm, respectively. The matrix was modelled as a cube with a side equal to 1.0 mm. The materials of both the inclusion and matrix were isotropic with the Young's moduli equal to 200 and 100 GPa, respectively. The Poisson ratio for both materials was 0.3. The inclusion and the matrix are assumed to be perfectly bonded at the interface. The boundary conditions applied to the model were uniform displacements 0.1 mm and −0.1 mm applied at two opposite sides of the cube. Other sides of the cube were free of any loads or boundary conditions.

The model was meshed using conformal (within Abaqus/CAE), voxel, octree and smoothed octree meshes. Illustrations of meshes are given in Fig. 6. Details of these types of meshes are given in Tables 1–3. In all cases, C3D8 elements were used for the analysis. Only one set of smoothing parameters has been used in this paper (λ = 0.3, 50 iterations).

Three conformal meshes were generated to give a reference solution which can be used as a benchmark for solutions obtained with other meshes. Convergence of the meshes was determined using the following criteria – convergence of maximum von Mises stresses and normalised root mean square deviation (NRMSD) of local stresses along three paths shown in Fig. 7. The NRMSD was calculated as the root mean square deviation normalised by the mean value:

$$NRMSD = \frac{\sqrt{E\left((\widehat{y}_i - y_i)^2\right)}}{E(y_i)} \tag{5}$$

where y_i are the values of stresses in the reference model, $\widehat{y}_i$ are the values of stresses in the model of interest and $E(\cdot)$ is mean value of a vector.

The maximum stresses and NRMSD along three selected paths in the model converged fast with the conformal mesh refinement as shown in Table 1. The solution obtained using the most refined conformal mesh consisting of approximately 3.1 × 10^6 elements was selected as a reference solution. The maximum stress error in a model, $\widehat{\sigma}_{max}$, is written as a deviation from the value obtained with the reference mesh, σ_{max}, i.e. $(\widehat{\sigma}_{max} - \sigma_{max})/\sigma_{max}$. The second most refined mesh

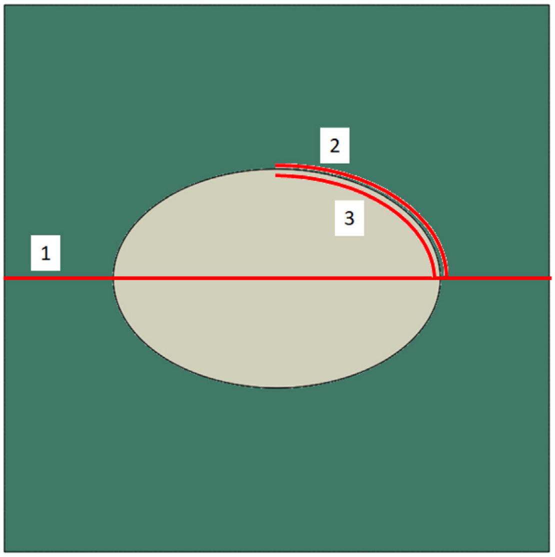
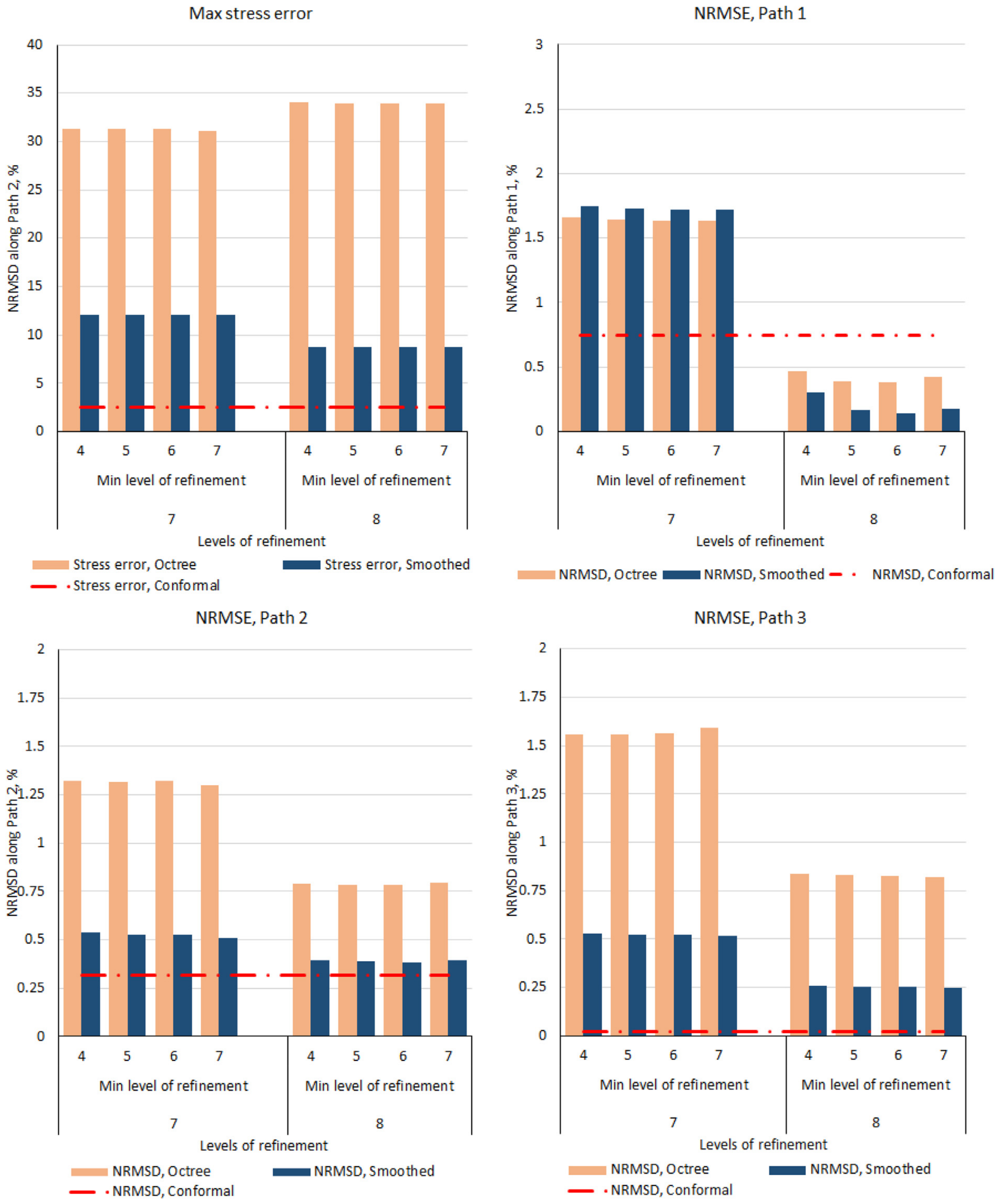


Fig. 7. Paths to query local stresses. Path 1 – along major axis of the inclusion, Path 2 – along an ellipse with sizes 0.025 mm larger than the inclusion, Path 3 – along an ellipse with sizes 0.025 mm smaller than the inclusion. All paths are in the plane which divides the inclusion in two equal halves.



a

Fig. 8. a. Maximum stress error for various meshes; b. Total analysis time (meshing and FE analysis) for various meshes.

has the maximum stress error of 2.56% while other metrics were all below 1%.

Two voxel meshes were compared against the selected reference mesh. The comparison is given in Table 2. The maximum stress error was found to be up to 30% but it should be pointed out that this error is the result of stress concentration at one or several elements. The metrics for average deviations decrease below 2% with increase of voxel mesh density, as expected, but still do not achieve the accuracy of a conformal mesh with a similar number of elements. While it is possible to increase the number of elements further, it becomes more computationally expensive to do so.

The mesh convergence of new meshing approaches was recorded using the same metrics as for conformal and voxel meshes. The octree-based meshes were refined by reducing the maximum element size (increasing the minimum level of refinement from 4 to 7) and reducing the minimum element size (increasing the maximum level

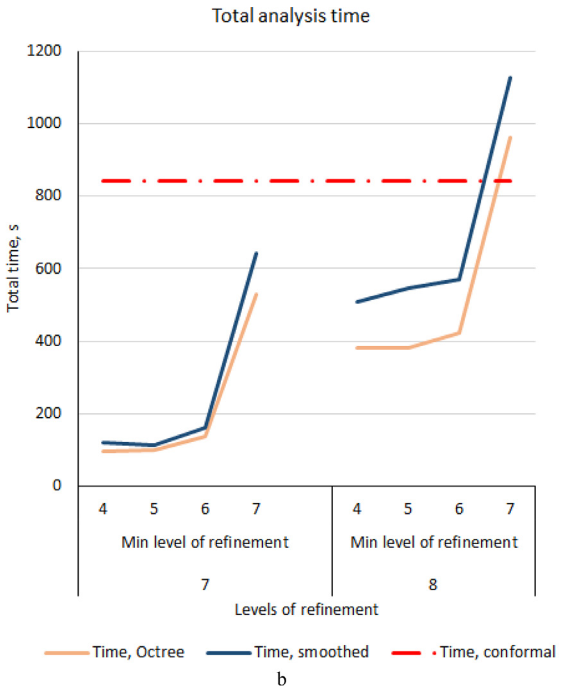


Fig. 8 (*continued*)

of refinement from 7 to 8). The results of the comparison are given in Fig. 8 and Table 3. Reducing the maximum element size reduces the difference between the model and the reference solution but this effect is relatively small. In comparison, both the reduction of the minimum element size and smoothing have a strong effect on the difference between the model and the reference solution. The most refined model (minimum refinement – 7, maximum refinement – 8) has the NRMSD along three selected paths within 0.5% from the reference solution. However, the maximum stress error still remains as high as 8.7% even for the smoothed octree mesh. It is expected that further refinement would reduce the error but would be impractical because of the computational costs. Fig. 8 also shows total computational costs required to generate each of the meshes and obtain a corresponding solution. The costs of obtaining a solution with the reference conformal mesh is also shown in Fig. 8. It is important to note that obtaining solutions with octree or smoothed octree meshes was faster than with the conformal mesh.

Stress fields obtained with the generated meshes are shown in Fig. 9. A magnified view of the same stress fields in some of the meshes is shown in Fig. 10.

It should also be noted that calculation of NRMSD involves several interpolations which can introduce some errors. The first interpolation occurs within Abaqus/CAE when points for comparison are queried at the intersections of the path with the elements. The second interpolation is performed to sample the points from two different meshes to the same grid.

3.2. Textile composites

Section 3.1 showed the applicability of the octree refinement and smoothing techniques for the relatively simple case of an elliptical cylinder inclusion. However, the main motivation of the suggested approach is to apply it to various textile composites. For that purpose, unit cells of a plain weave, a sheared plain weave and an orthogonal 3D woven composite were generated using TexGen. Geometry parameters of the unit cells are given in Tables 4 and 5 and the unit cells are shown in Fig. 11. To allow generation of acceptable conformal meshes all of the unit cells had an artificial gap between yarns.

Similar to the example with an elliptical inclusion, the unit cells of textile composites were meshed with conformal, voxel, octree and smoothed octree meshes. The conformal meshes were generated using TexGen and linear C3D4 elements were used for all of them. C3D8R elements were used for other meshes. Examples of the meshes are shown in Fig. 12. The properties of matrix and yarn materials are given in Table 6.

Convergence studies of conformal meshes were performed for each of the textiles. The convergence was judged using several metrics – stress in a selected cross-section and NRMSD (Eq. (5)) along selected paths. The paths are shown schematically in Fig. 13. As for the example in Section 3.1, the converged solutions obtained with conformal meshes were selected as the reference solutions for each of the unit cells.

The following boundary conditions were applied to all unit cells: uniform displacements of 0.1 mm and -0.1 mm were applied to two opposite sides in the x-direction to simulate an extension in this direction, top surface of the unit cell was constrained in the out-of-plane direction.

Details of convergence studies of the conformal and voxel meshes are given in Appendix A. As in Section 3.1, the maximum stress error converges fast for the conformal meshes but NRMSDs for some paths usually remain above 2%, which can be explained by there being a more complex stress state in the unit cell of a textile composite than in the problem with a single inclusion. The voxel meshes converge more slowly and show up to 11% error in the maximum stress error and NRMSDs. These errors can be reduced by further refining the voxel mesh but it would result in large meshes which are prohibitively expensive to use in a FE analysis.

The solutions obtained with octree and smoothed octree meshes were compared to the reference solutions and detailed results are given in Tables 7–9 for plain weave, sheared plain weave and orthogonal 3D weave unit cells, accordingly. It was found that reducing the maximum element size (increasing the minimum level of refinement) improves accuracy of the solutions more than was observed in Section 3.1. Decreasing the minimum element size (increasing the maximum level of refinement) also has a significant effect on the accuracy. Smoothing also increases the accuracy of the solutions. In summary, deviation less than 7% from the reference solution can be achieved for all these examples. Maximum stress error and NRMSD along path ‘1’ for all unit cells are shown in Fig. 14. The graphs also show the difference between the reference mesh and a conformal mesh with lower refinement. Distributions of von Mises stresses in longitudinal and transverse yarns of the plain weave composite for different levels of refinement are given in Fig. 15.

Total analysis cost for each of the performed simulations is also shown in Fig. 14. It can be seen that reducing the minimum element size, which results in larger meshes, results in growth of the analysis time. The cost of generating an octree mesh and performing analysis using this mesh is always lower than for an equivalent smoothing due to obvious extra operations required for the latter. The actual time to run the simulations in Abaqus were almost identical for both types of meshes as it mainly depends on the number of nodes and constraints rather than on a type of the mesh.

4. Discussion and conclusions

The meshing algorithm presented here can be used for automatic generation of FE meshes of unit cells of woven composites of arbitrary complexity. Combination of an octree voxel mesh and mesh smoothing algorithm helps to create meshes which have accuracy comparable to equivalent conformal meshes. The octree refinement procedure, which is used to create the mesh, approximates the internal geometry of the

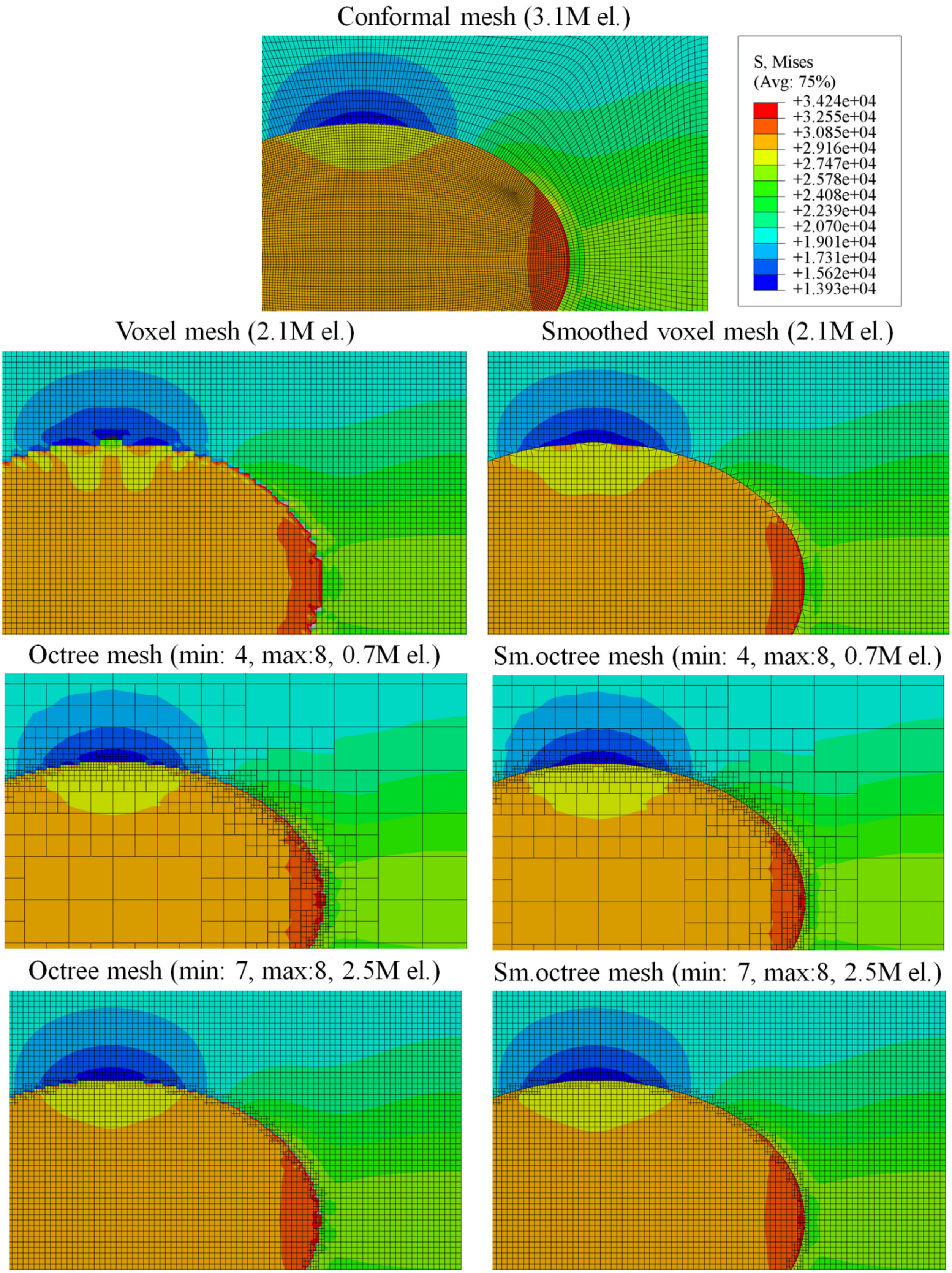


Fig. 9. Distribution of von Mises stresses (in Pa) in models with elliptical inclusion. The colour schemes are scaled to the minimum and maximum values predicted by the conformal mesh.

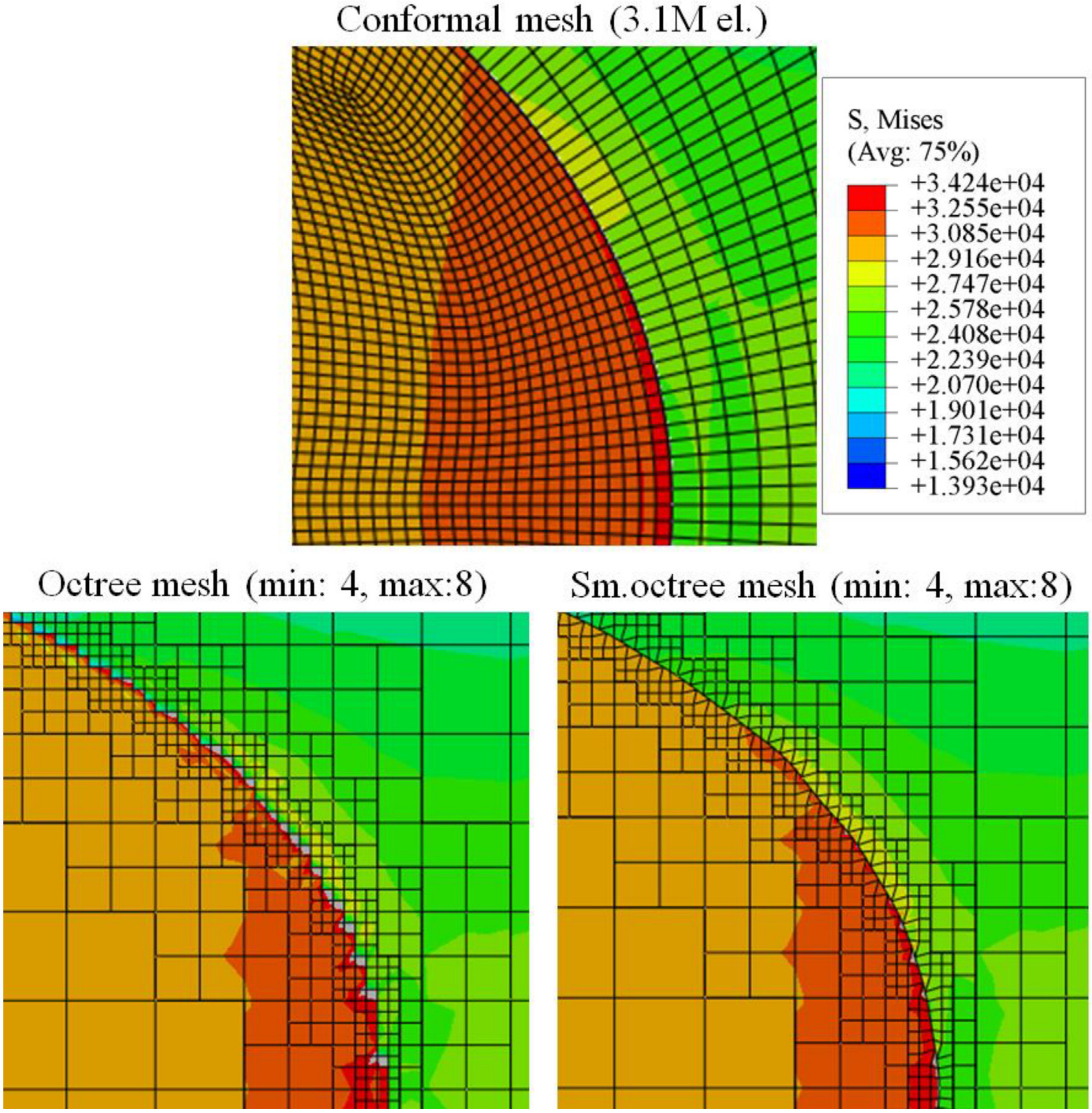


Fig. 10. Magnified view of stress distributions (in Pa) in some of the meshes.

Table 4
Parameters of plain weave reinforcements.

	Yarn spacing, mm	Yarn width, mm	Fabric thickness, mm	Gap size, mm	Shear angle, deg
Non-sheared	1.0	0.8	0.2	0.02	0
Sheared	1.0	0.8	0.2	0.02	20

Table 5
Parameters of 3D woven reinforcement.

	Number of yarns	Number of layers	Yarn spacing, mm	Yarn width, mm	Yarn height, mm	Gap size, mm
Warp	3	2	1.0	0.8	0.1	0.02
Weft	2	3	1.0	0.8	0.1	0.02
Binder	1	–	2.5	0.4	0.05	0.02

unit cell with a given precision without generating an excessive number of elements. The smoothing algorithm reduces the effect of the jagged surface of a voxel mesh and reduces stress concentrations. The main advantage of the technique is its ability to generate a mesh for geometries which are either not possible to mesh using a conformal mesh or where such a mesh would have an excessive number of elements.

The proposed method has been applied to several model problems – a unit cell with an elliptic inclusion and unit cells with plain weave, sheared plain weave and a 3D woven reinforcement. For all the cases

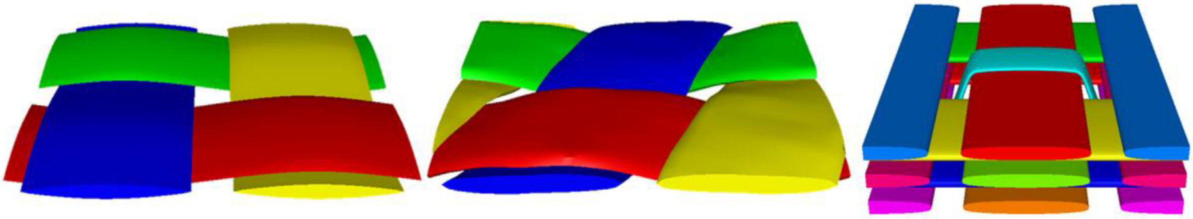


Fig. 11. Unit cells of selected textiles: non-sheared plain weave, sheared plain weave, and orthogonal 3D woven (from left to right).

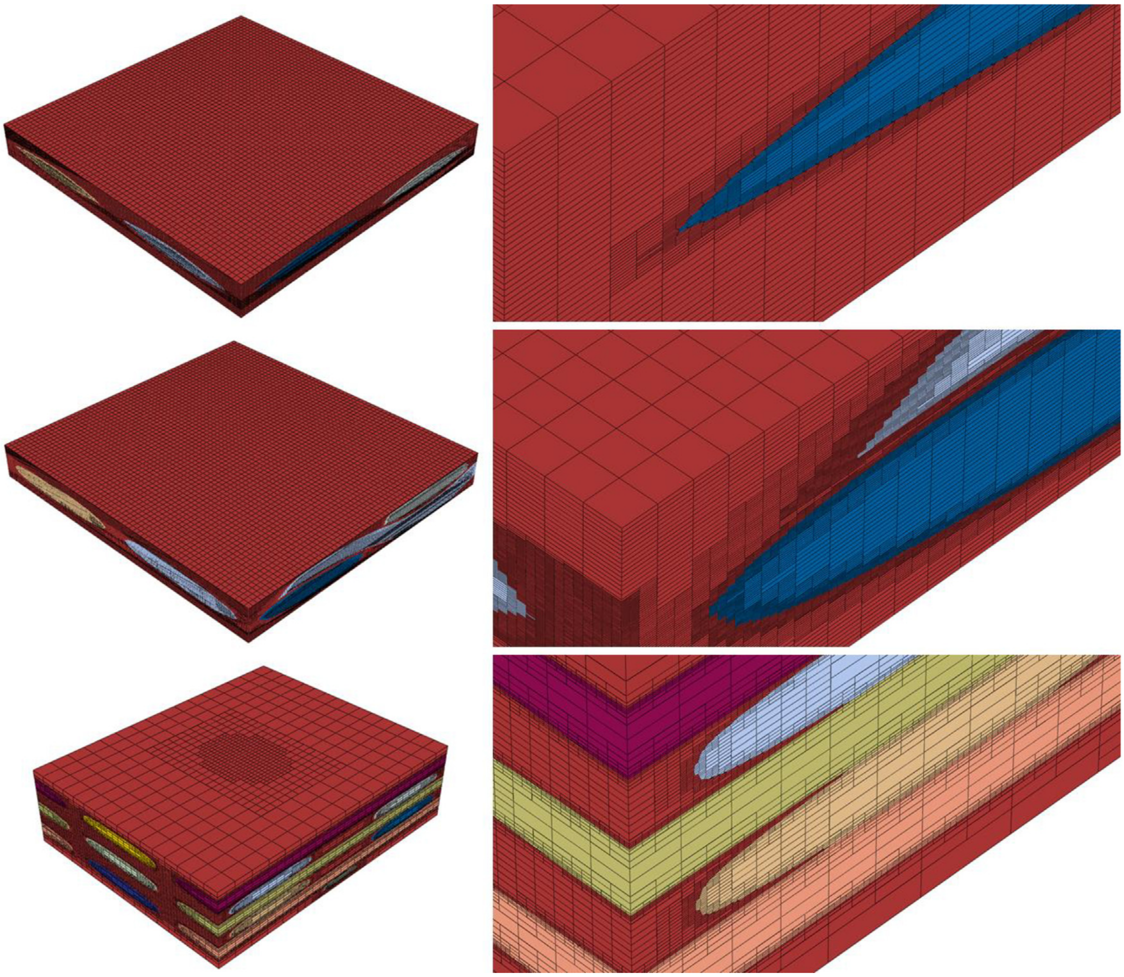


Fig. 12. Examples of meshes.

Table 6
Material properties for the unit cells models.

	E_1, GPa	$E_2 = E_3$, GPa	$\nu_{12} = \nu_{13}$	ν_{23}	$G_{12} = G_{13}$, GPa	G_{23}, GPa
Yarn	200.0	10.0	0.3	0.3	5.0	5.0
Matrix	3.0	3.0	0.2	0.2	1.25	1.25

the proposed method was benchmarked against conventional conformal meshes. It was found that the smoothed octree meshes can estimate local stresses along selected paths with an average deviation of less than 7% when compared to the converged conformal meshes. The error for the maximum stress was estimated to be within 10% for selected cross-sections. Obviously, there might be cases where the presented method does not result in low error e.g. in areas where geometrical features were not meshed with sufficient resolution or the smoothing algorithm did not eliminate stress concentrators. However, it is still expected that average stresses are within an acceptable range and can be achieved within reasonable time cost.

Total analysis time, i.e. the time required to generate a mesh and to perform analysis using these meshes, was also compared. The time to generate octree and smoothed octree meshes increases almost exponentially with reduction of the minimum element size (increasing maximum level of refinement). However, for the meshes with lower

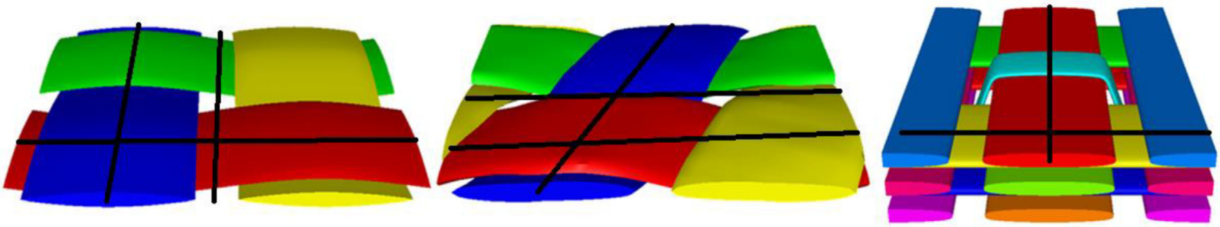


Fig. 13. Position of the cross-section of interest.

Table 7
Comparison of solutions for the composite with plain weave reinforcement.

Refinement		Number of elements	$\widehat{\sigma}_{max}$, %		NRMSD 1, % (along a longitudinal yarn)		NRMSD 2, % (between two longitudinal yarns)		NRMSD 3, % (along a transverse yarn)	
Min	Max		Octree	Smoothed	Octree	Smoothed	Octree	Smoothed	Octree	Smoothed
4	7	976,338	11.72	8.55	13.59	6.03	10.17	8.40	5.22	4.20
5		976,526	11.72	8.55	13.59	6.02	10.17	8.38	5.05	4.01
6		1,033,445	11.72	8.55	9.77	6.02	9.73	8.37	9.91	3.83
7		2,146,689	11.05	5.72	6.18	6.03	9.16	8.30	9.37	3.36
4	8	3,003,344	4.94	3.83	9.12	6.38	4.82	4.82	7.73	6.78
5		4,265,097	4.94	7.38	9.12	6.38	4.82	4.82	7.63	6.68
6		4,314,873	4.94	4.83	7.81	6.38	4.81	4.81	9.43	6.59

Table 8
Comparison of solutions for the composite with sheared plain weave reinforcement.

Refinement		Number of elements	$\widehat{\sigma}_{max}$, %		NRMSD 1, % (along a longitudinal yarn)		NRMSD 2, % (between two longitudinal yarns)		NRMSD 3, % (along a transverse yarn)	
Min	Max		Octree	Smoothed	Octree	Smoothed	Octree	Smoothed	Octree	Smoothed
4	7	1,073,907	3.18	2.34	11.02	5.62	8.00	5.62	8.34	1.56
5		1,073,172	0.68	2.30	9.99	5.55	8.00	5.63	8.35	1.55
6		1,111,441	2.59	2.54	9.99	5.55	7.85	5.39	6.69	1.31
7		2,146,689	7.28	1.42	10.90	5.57	8.70	5.73	8.64	1.38
4	8	4,983,316	1.81	1.42	6.61	4.67	4.88	2.95	4.57	2.77
5		4,982,783	1.76	1.42	6.55	4.60	4.89	2.95	4.57	2.76
6		5,012,507	1.76	1.12	5.42	4.60	4.74	2.64	4.54	2.69

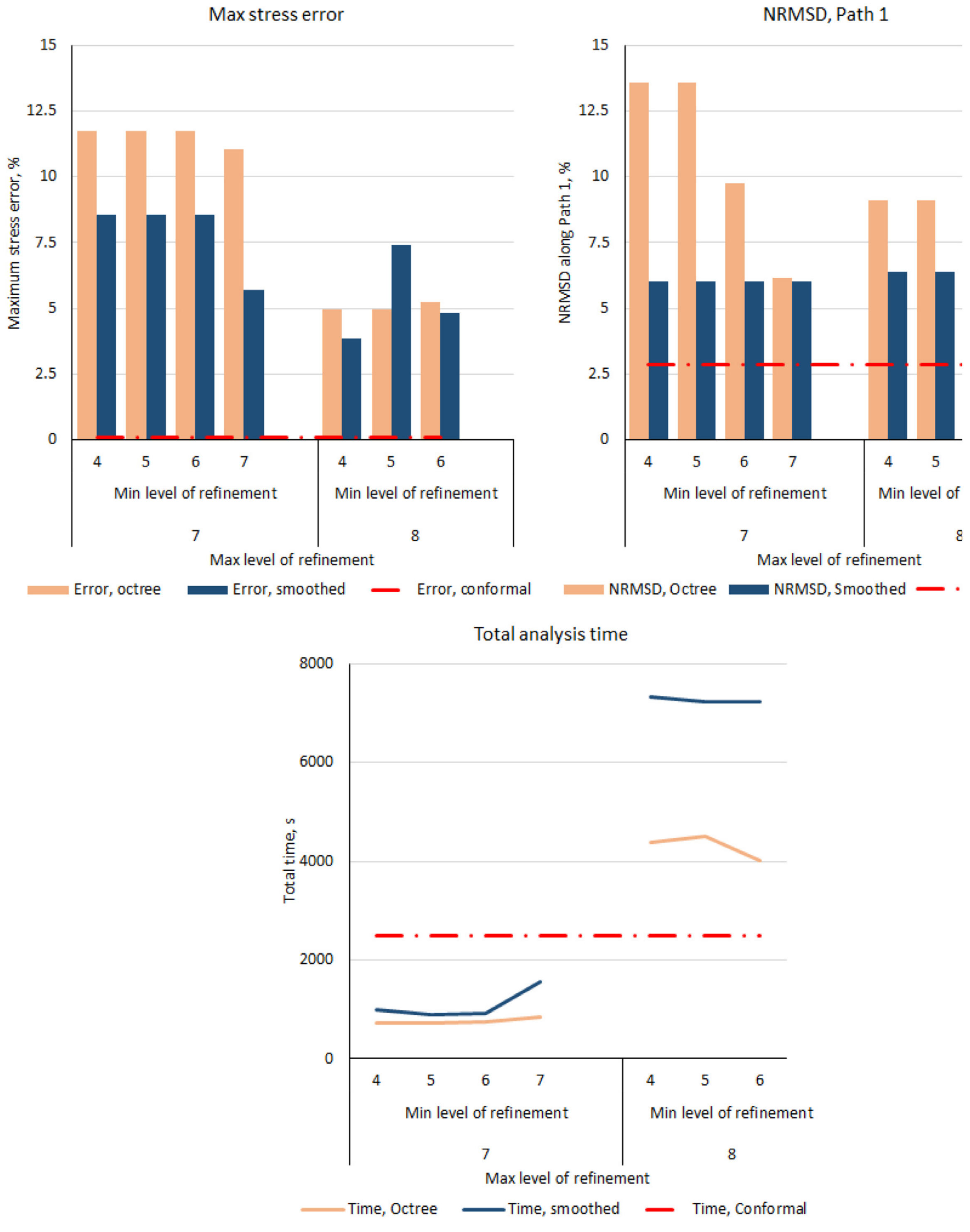
Table 9
Comparison of solutions for the composite with orthogonal 3D weave reinforcement.

Refinement		Number of elements	$\widehat{\sigma}_{max}$, %		NRMSD 1, % (along a longitudinal yarn)		NRMSD 2, % (between two longitudinal yarns)	
Min	Max		Octree	Smoothed	Octree	Smoothed	Octree	Smoothed
4	7	1,024,626	8.27	6.16	0.07	0.63	1.58	2.86
5		1,025,458	8.28	6.16	0.07	0.63	1.51	2.80
6		1,060,026	8.28	6.19	0.05	0.50	1.10	2.81
7		2,146,689	8.36	6.28	0.06	0.47	0.98	2.79
4	8	4,638,876	1.32	1.05	0.07	0.32	1.04	1.27
5		4,639,814	1.32	1.05	0.07	0.32	1.02	1.25
6		4,670,811	1.36	1.07	0.05	0.19	0.43	1.10

level of refinement (as e.g. equal to 7, which was reported here) the time to generate the mesh is often lower than the time required to generate a conformal mesh to achieve similar precision. For the meshes with a high level of refinement, the total analysis time is quite high because of the time required to generate the mesh and to pre-process extra constraints for the hanging nodes. This time will be improved by employing parallel execution of some parts of the code.

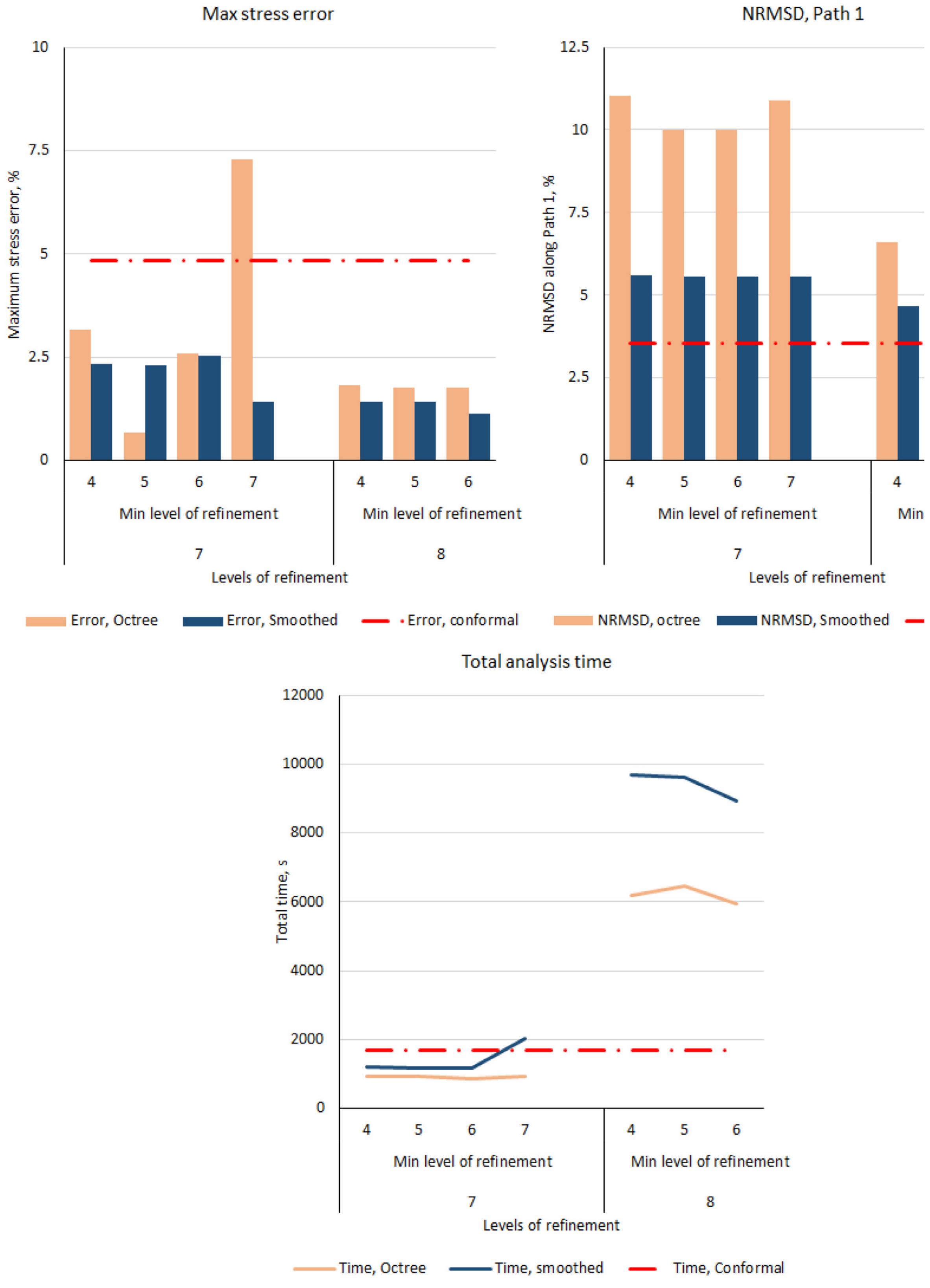
It also should be mentioned that the precision of the solutions obtained with smoothed octree meshes can be reduced because of ill-shaped elements that can occur from the element smoothing at some boundaries. This can be improved by converting the octree mesh into a tetrahedral mesh with no hanging nodes [22,23] as shown in Fig. 16. These tetrahedral meshes can be post-processed after boundary smoothing to reshape the elements by moving internal nodes to improve element quality.

In summary, the presented smoothed octree meshes offer automatic meshing of any complex geometries for which conformal meshes are not always possible. The resulting mesh is not conformal but follows the boundary between materials with a selected precision. The octree refinement of the meshes reduces the computational cost which is advantageous in analysis of large unit cells. It was shown that the smoothed octree meshes can achieve precision comparable to conformal meshes.



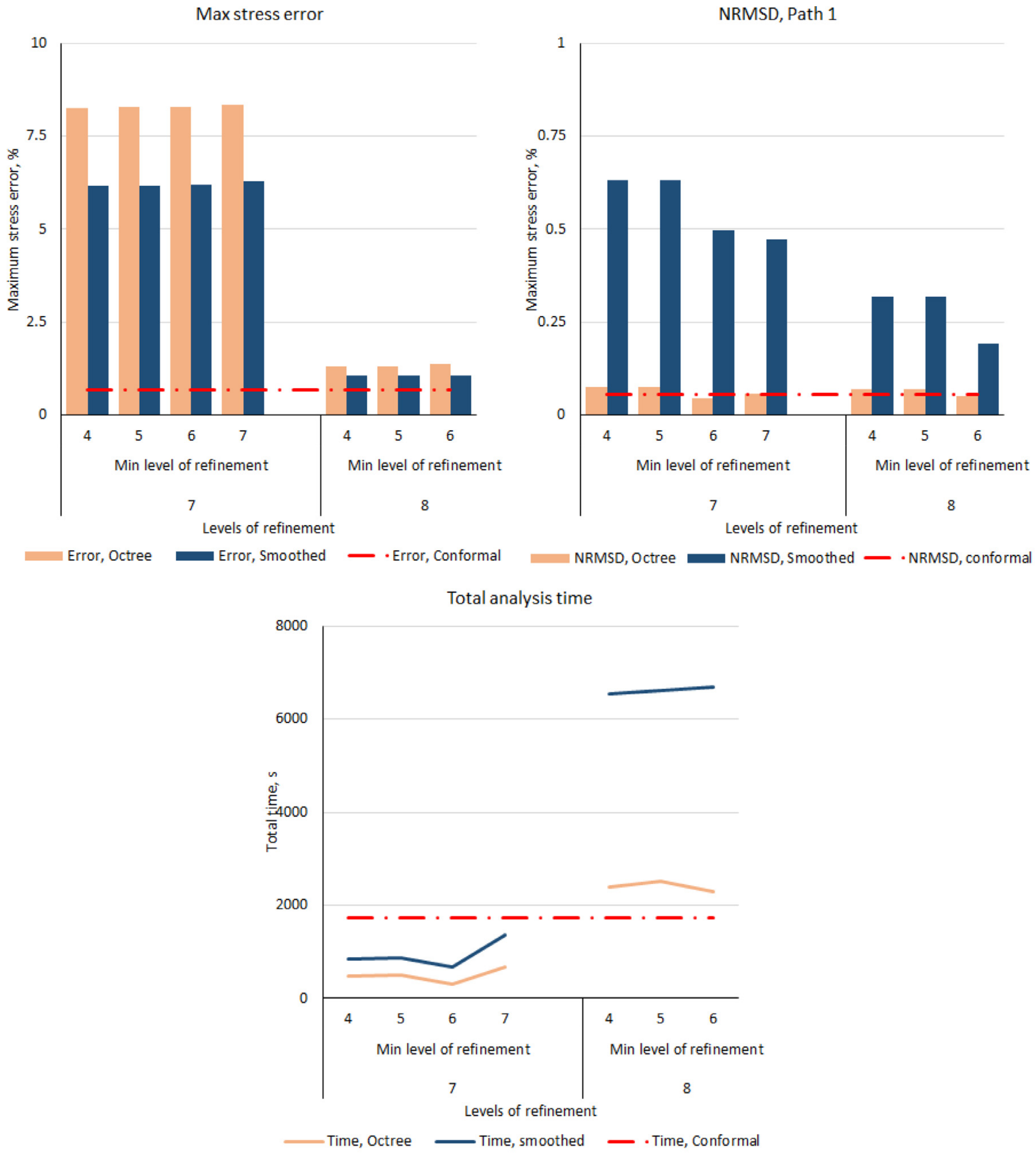
a.

Fig. 14. Comparison of octree and smoothed octree meshes with the reference solution for: a. the plain weave unit cell; b. sheared plain weave; c. 3D weave.



b.

Fig. 14 (*continued*)



c.

Fig. 14 (*continued*)

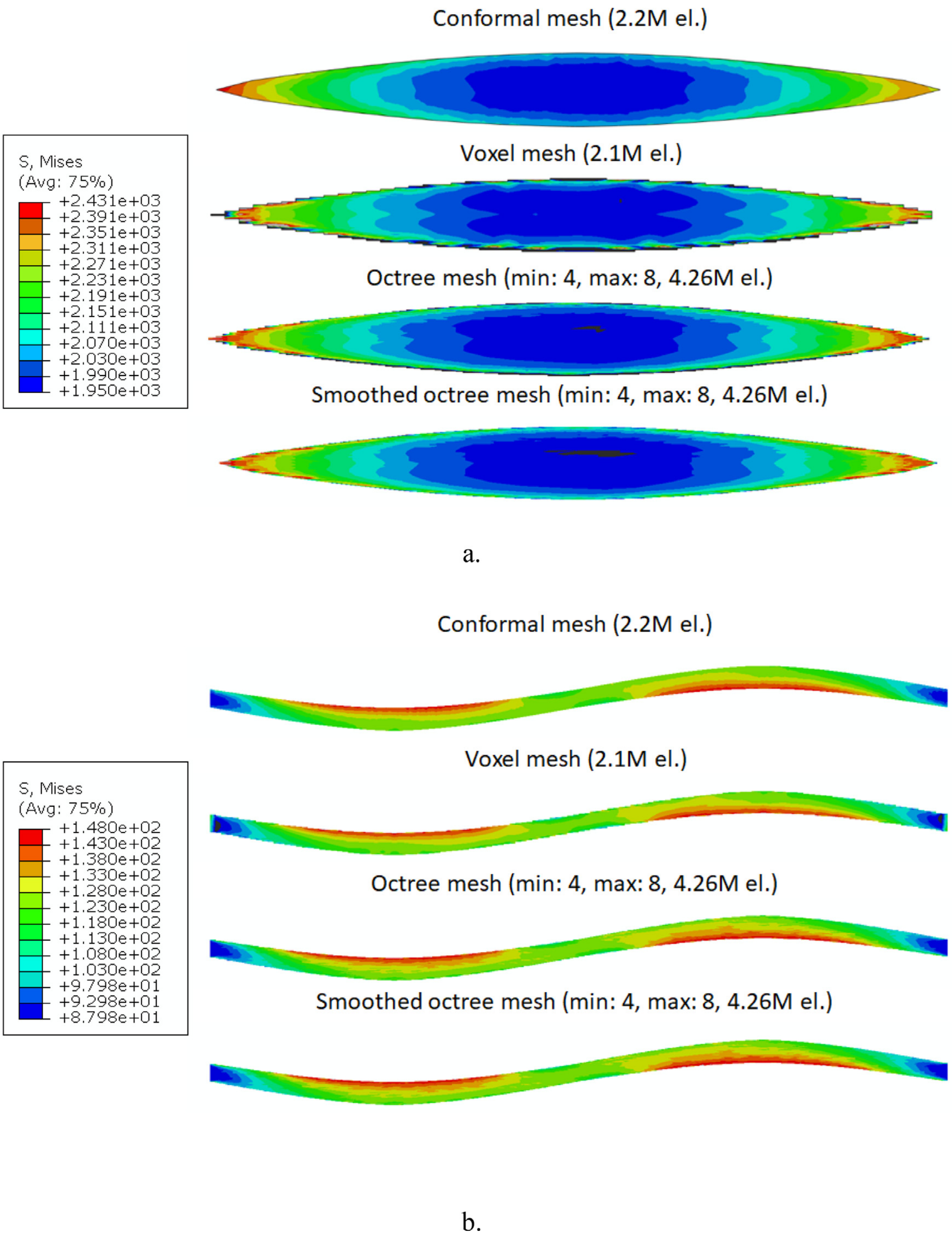


Fig. 15. Distribution of von Mises stresses (in Pa) in a. longitudinal yarns; b. transverse yarns in various meshes.

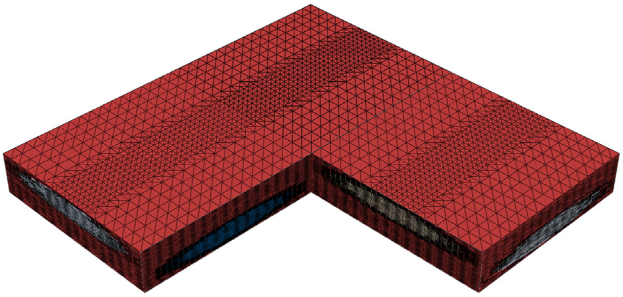


Fig. 16. An example of voxel to tetrahedral conversion using body-centred cubic (BCC) refinement.

CRediT authorship contribution statement

Mikhail Y. Matveev: Conceptualization, Methodology, Investigation, Software, Visualization, Writing - original draft. **Louise P. Brown:** Software, Writing - review & editing. **Andrew C. Long:** Supervision, Writing - review & editing, Funding acquisition.

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.

Acknowledgements

This work was supported by the Engineering and Physical Sciences Research Council [grant number EP/P006701/1]; through the EPSRC Future Composites Manufacturing Research Hub. Software engineering support was funded by the Engineering and Physical Sciences Research Council RSE Fellowship [Grant number: EP/N019040/1].

Appendix A

Tables A1–A6.

Table A1
Convergence of conformal meshes for plain weave textile composite.

	Number of nodes	Maximum stress, MPa	$\widehat{\sigma}_{max}$, %	NRMSD 1, %	NRMSD 2, %	NRMSD 3, %
1	502,519	2431	0.16	6.77	2.03	4.13
2	855,021	2436	0.06	4.01	2.02	2.62
3	1,630,603	2439	<0.01	2.88	3.48	3.33
4	2,240,350	2447	–	–	–	–

Table A2
Convergence of voxel mesh for plain weave textile composite.

Refinement	Number of nodes	$\widehat{\sigma}_{max}$, %	NRMSD 1, %	NRMSD 2, %	NRMSD 3, %
64x64x64	274,625	11.05	18.81	11.52	10.76
128x128x128	2,146,689	11.05	6.18	9.16	9.37
256x256x256	16,974,593	Not feasible to run			

Table A3
Convergence of conformal mesh for sheared plain weave composites.

	Number of nodes	$\widehat{\sigma}_{max}$, %	NRMSD 1, %	NRMSD 2, %	NRMSD 3, %
1	470,293	4.84	2.38	1.94	1.02
2	803,320	4.84	3.55	1.33	0.52
3	1,539,649	–	–	–	–

Table A4
Convergence of voxel mesh for sheared plain weave textile composite.

Refinement	Number of nodes	$\widehat{\sigma}_{max}$, %	NRMSD 1, %	NRMSD 2, %	NRMSD 3, %
64x64x64	274,625	16.90	9.68	8.14	16.34
128x128x128	2,146,689	7.28	10.90	8.70	8.64
256x256x256	16,974,593	Not feasible to run			

Table A5
Convergence of conformal meshes for orthogonal 3D woven textile composite.

	Number of nodes	$\widehat{\sigma}_{max}$, %	NRMSD 1, %	NRMSD 2, %
1	404,855	4.33	0.14	0.70
2	1,799,105	0.67	0.06	0.33
3	3,103,719	–	–	–

Table A6
Convergence of voxel mesh for orthogonal 3D woven textile composite.

Refinement	Number of nodes	$\hat{\sigma}_{max}$, %	NRMSD 1, %	NRMSD 2, %
64x64x64	274,625	17.11	0.08	1.91
128x128x128	2,146,689	8.36	0.06	0.98
256x256x256	16,974,593	Not feasible to run		

Appendix B. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.compstruct.2020.112757.

References

[1] Cheng SW, Dey TK, Shewchuk J. Delaunay mesh generation; 2016. CRC Press.
[2] Livesu M, Vining N, Sheffer A, Gregson J, Scateni R. PolyCut: monotone graph-cuts for PolyCube base-complex construction. ACM Trans Graph 2013;32(6):1–12. https://doi.org/10.1145/2508363.2508388.
[3] Green SD et al. Mechanical modelling of 3D woven composites considering realistic unit cell geometry. Compos Struct 2014;118:284–93.
[4] Brown LP, Matveev MY, Spackman G. louisepb/TexGen: TexGen v3.12.0 (Version v3.12.0). 2020 2020, March 11; Available from: http://doi.org/10.5281/zenodo.3706478
[5] Blackketter DM, Walrath DE, Hansen AC. Modeling damage in a plain weave fabric-reinforced composite-material. J Compos Tech Res 1993;15(2):136–42.
[6] Tang XD, Whitcomb JD. Progressive failure behaviors of 2D woven composites. J Compos Mater 2003;37(14):1239–59.
[7] Ivanov DS et al. Failure analysis of triaxial braided composite. Compos Sci Technol 2009;69(9):1372–80.
[8] Melro AR et al. Numerical simulation of the non-linear deformation of 5-harness satin weaves. Comput Mater Sci 2012;61:116–26.
[9] De Carvalho NV, Pinho ST, Robinson P. Numerical modelling of woven composites: biaxial loading. Compos Part A Appl Sci Manuf 2012;43(8):1326–37.
[10] Kim HJ, Swan CC. Voxel-based meshing and unit-cell analysis of textile composites. Int J Numer Meth Eng 2003;56(7):977–1006.
[11] Li LY, Wen PH, Aliabadi MH. Meshfree modeling and homogenization of 3D orthogonal woven composites. Compos Sci Technol 2011;71(15):1777–88.
[12] Chowdhury NT, Joosten MW, Pearce GMK. An embedded meshing technique (SET) for analysing local strain distributions in textile composites. Compos Struct 2019;210:294–309.
[13] Grail G et al. Consistent Finite Element mesh generation for meso-scale modeling of textile composites with preformed and compacted reinforcements. Compos Part A Appl Sci Manuf 2013;55:143–51.
[14] Ha MH, Cauvin L, Rassineux A. A methodology to mesh mesoscopic representative volume element of 3D interlock woven composites impregnated with resin. CR Mec 2016;344(4–5):267–83.
[15] Si H, Gartner K. Meshing piecewise linear complexes by constrained Delaunay tetrahedralizations. Proceedings of the 14th international meshing roundtable; 2005. p. 147–163
[16] Matveev MY et al. Effects of layer shift and yarn path variability on mechanical properties of a twill weave composite. J Compos Mater 2017;51(7):913–25.
[17] Fang GD et al. Smoothing artificial stress concentrations in voxel-based models of textile composites. Compos Part A Appl Sci Manuf 2016;80:270–84.
[18] Burstedde C, Wilcox LC, Ghattas O. p4est: Scalable algorithms for parallel adaptive mesh refinement on forests of octrees. Siam J Sci Comput 2011;33(3):1103–33.
[19] Panesar A et al. Hierarchical remeshing strategies with mesh mapping for topology optimisation. Int J Numer Methods Eng 2017;111(7):676–700.
[20] Vallet B, Levy B. Spectral geometry processing with manifold harmonics. Comput Graph Forum 2008;27(2):251–60.
[21] Taubin G. A signal processing approach to fair surface design. In: Proceedings of the 22nd annual conference on computer graphics and interactive techniques; 1995, ACM. p. 351–358.
[22] Friess H et al. Tetrahedral mesh generation based on space indicator functions. Int J Numer Methods Eng 2013;93(10):1040–56.
[23] Labelle F, Shewchuk JR. Isosurface stuffing: fast tetrahedral meshes with good dihedral angles. ACM Trans Graphics 2007;26(3).