Received: 20 June 2023 | Revised: 7 July 2023 | Accepted: 22 August 2023

DOI: 10.1002/pamm.202300023

RESEARCH ARTICLE

Block-structured mesh generation from implicit geometries for cardiovascular applications

Domagoj Bošnjak | **Thomas-Peter Fries**

Institute of Structural Analysis, Graz University of Technology, Graz, Austria

Correspondence
Domagoj Bošnjak, Institute of Structural Analysis, Graz University of Technology, Lessingstrasse 25, Graz, Austria.
Email: bosnjak@tugraz.at

Funding information
LEAD-project: Mechanics, Modelling and Simulation of Aortic Dissection

Abstract

Generation of suitable meshes for patient-specific vascular domains remains a major obstacle in the context of numerical simulations relying on the finite element method, as well as many similar methods. Herein, we discuss a skeleton-based structured meshing approach, revolving around *block-structures*. We rely on an implicitly represented surface, either via a signed distance function, or a convolution surface. The mesh generation using the aforementioned concepts is performed through the proper utilisation of transfinite maps. Application-wise, we consider healthy blood vessels, and extensions required to tackle aortas afflicted by *aortic dissection*. The high quality of the resulting meshes is affirmed through common mesh quality metrics. Our approach features large flexibility in terms of spatial resolution, as well as mesh order.

1 | INTRODUCTION

Structured hexahedral mesh generation offers a multitude of advantages compared to the more commonly utilised unstructured meshing approaches. Examples include straightforward control over the number of nodes and elements in the mesh, as well as more nuanced, computational fluid dynamics themes, such as flow-oriented elements and high-quality treatment of boundary layers [1]. However, generating structured meshes for arbitrary domains often remains only a semi-automatic, or rather a non-viable task. Tubular structures, such as patient-specific blood vessels, represent a class of domains where structured mesh generation has shown to be both useful, as well as achievable [2–4]. Furthermore, significant effort is required to tackle aortas plagued by *aortic dissection* [5], a serious cardiovascular condition. The established approaches to meshing similar domains primarily focus on generating linear meshes, whereas we pay special attention to higher-order meshes as well. Additionally, we deem the ability to generate coarse meshes important, as they have proven to be useful for various purposes, for example, problems where a huge number of simulations is required, or *the geometric multigrid* method for solving linear systems, where they are in fact a requirement.

The mesh generation consists of four main steps: skeleton extraction, implicit surface representation, block-structure generation and the final volumetric mesh generation step. This essentially serves as an outline of this work, supplemented by examples.

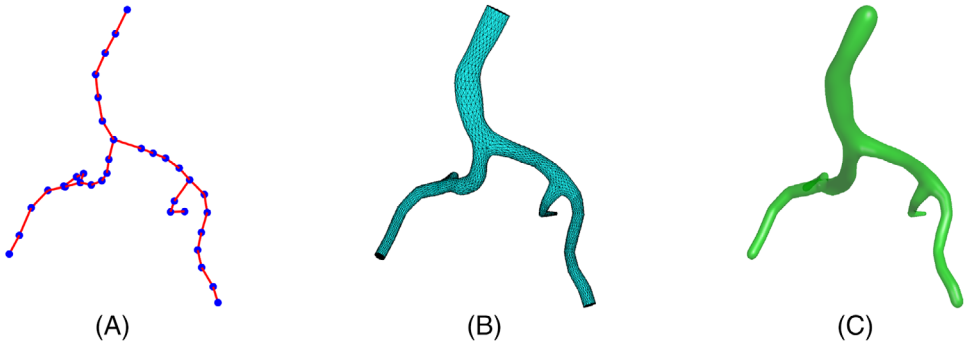

FIGURE 1 The skeleton and the surface of a part of a patient-specific aorta: (A) the skeleton, (B) the triangulated surface for the SDF and (C) the convolution surface.

2 | METHODS

We refer to the *skeleton* as a set of segments representing the centerline of the domain. It is extracted based on the mesh contraction approach from Au et al. [6], with additional details in Bošnjak et al. [7]. The surface description is implicit, that is, we find a function $f : \mathbb{R}^3 \to \mathbb{R}$, such that its zero level-set $\{\boldsymbol{x} \in \mathbb{R}^3 \ : \ f(\boldsymbol{x}) = 0\}$ represents the surface. A viable and well-known approach is the *signed distance function* (SDF), defined as the orthogonal distance of a point to the surface, which is additionally negative if the point is inside the region bounded by the surface, and positive if it is outside. Thus, given a triangulated surface of the domain, one may directly generate the SDF. Alternatively, for healthy blood vessels, we employ *convolution surfaces* [8]. Namely, given a skeleton S consisting of segments S_i, denoted by $S = \bigcup_{i=1}^{n} S_i$, and the corresponding indicator function $\mathbb{1}_S(\boldsymbol{x}) = \begin{cases} 1, \boldsymbol{x} \in S \\ 0, \boldsymbol{x} \notin S \end{cases}$, the convolution surface is defined as the convolution of the indicator function with a filter function h, originally proposed to be $h(x) = e^{\frac{-\|x\|^2}{2}}$:

$$C_S(\boldsymbol{x}) = \int_{\mathbb{R}^3} \mathbb{1}_S(\boldsymbol{y}) h(\boldsymbol{x} - \boldsymbol{y}) d\boldsymbol{y}.$$

Making use of the facts that the skeleton is consisted of segments, which share only individual points, and the definition of the indicator function, the evaluation of the convolution surface boils down to

$$C_S(\boldsymbol{x}) = \sum_{i=1}^{n} \int_0^1 K(\|\boldsymbol{x} - \Gamma_i(s)\|) ds,$$

where $K \in C^1(\mathbb{R}^+)$ is called a *kernel* function, and $\Gamma_i : [0,1] \to \mathbb{R}^3$ is the parametrization of the segment $S_i = [a_i, b_i]$, defined by $\Gamma_i(s) = \boldsymbol{a}_i + s(\boldsymbol{b}_i - \boldsymbol{a}_i)$. The kernel function may be freely chosen, resulting in different properties. We make use of the framework based on a compactly supported kernel function $K(x) = \frac{35}{16}(1 - x^2)^3 \mathbb{1}_{[0,1]}(x)$, presented by Fuentes Suaréz [9], offering provenly good radius control of the resulting surface. In regards to dissected aortas, at this point in time, the SDF is the only viable option of the two. Thus, we utilise convolution surfaces exclusively in the case of healthy blood vessels. Figure 1 shows an example of a skeleton, triangulated surface from which one can generate a SDF, and a convolution surface.

Following the skeleton extraction and the surface representation, we turn to the generation of a *block-structure* [10], that is, a subdivision of the domain into coarse blocks, aimed at capturing the topology of the domain, without necessarily considering all of the geometrical information (e.g. curvature). In our work, blocks are hexahedra, and even valid linear elements themselves, although generally, they need not possess either of the two properties. The block-structure is generated based on the skeleton, via prototypes of specific skeleton configurations. Since the skeleton is a centerline that abstracts the topology of the domain, it fits well with the notion of a block-structure.

One of the biggest hurdles that occurs when modelling vascular networks are bifurcations, that is, skeleton points connected to three other points via segments. Herein, we describe a way to model bifurcations, whose final configuration

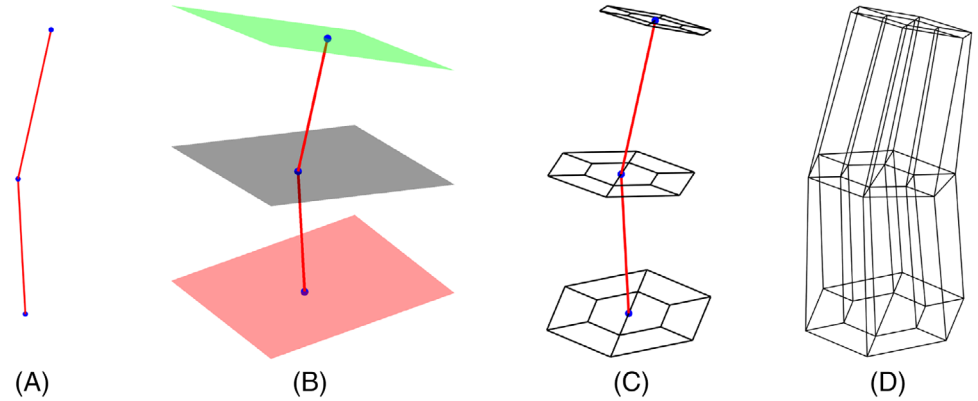

FIGURE 2 Generation of the blocks for non-junction points: (A) a skeleton consisting of two segments, (B) the cutting planes determined for the placement of the cross-sections, (C) the cross-sections and (D) the block-structure.

is similar to the bifurcation approach in Ghaffari et a. [2]. Given a point with two neighbours, which is, therefore, not in a bifurcation, a cross-section is placed around it exactly in the plane between the two segments, as in Bošnjak et al. [7]. Formally, given a skeleton point X with neighbours Y_1 and Y_2, we define $u_i = Y_i - X$, $i = 1, 2$ from which the normal of the aforementioned plane is computed as $n = \frac{u_1 - u_2}{\|u_1 - u_2\|}$. Thereafter, computing $d = -\langle n, X \rangle$ yields the plane which may then be represented as $n_1 x + n_2 y + n_3 z + d = 0$. In case of a terminal point, meaning it has only one neighbour, the given segment between X and Y_1 immediately serves as the normal of the plane for the cross-section. This is illustrated in Figure 2.

Given a bifurcation point X, three neighbouring points are associated with it, denoted by Y_1, Y_2 and Y_3. Looking at the neighbours in pairs, three distinct combinations arise. This yields three normal vectors when applying the aforementioned plane computation for each pair, denoted by n_{12}, n_{23} and n_{31}. The subscripts of the normals denote the corresponding pair of points Y_i. Thereafter, we compute cross-products of the normals, to find out whether the planes in question perhaps intersect on a line. We consider the basic properties of the cross-product, namely anticommutativity, compatibility with scalar multiplication and distributivity over addition:

$$\begin{aligned} n_{12} \times n_{23} &= (u_1 - u_2) \times (u_2 - u_3) = u_1 \times u_2 - u_1 \times u_3 - u_2 \times u_2 + u_2 \times u_3 \\ &= u_1 \times u_2 + u_2 \times u_3 + u_3 \times u_1 \end{aligned}$$

$$\begin{aligned} n_{23} \times n_{31} &= (u_2 - u_3) \times (u_3 - u_1) = u_2 \times u_3 - u_2 \times u_1 - u_3 \times u_3 + u_3 \times u_1 \\ &= u_1 \times u_2 + u_2 \times u_3 + u_3 \times u_1 \end{aligned}$$

$$\begin{aligned} n_{31} \times n_{12} &= (u_3 - u_1) \times (u_1 - u_2) = u_3 \times u_1 - u_3 \times u_2 - u_1 \times u_1 + u_1 \times u_2 \\ &= u_1 \times u_2 + u_2 \times u_3 + u_3 \times u_1, \end{aligned}$$

implying that

$$n_{12} \times n_{23} = n_{23} \times n_{31} = n_{31} \times n_{12} = u_1 \times u_2 + u_2 \times u_3 + u_3 \times u_1.$$

Hence, the three planes in question indeed all intersect at the same line parametrized by the point X and the direction vector $u_1 \times u_2 + u_2 \times u_3 + u_3 \times u_1$. Then, this line is used as a reference towards building a junction configuration, enabling a connection to the cross-sections around Y_i, as shown in Figure 3. The reference lines for non-junction points are then computed by interpolation. In future work, we wish to further develop this notion to accommodate for the case of two or more bifurcations extremely close to one another. This is often encountered in dissected and occasionally healthy aortas.

The bifurcation scheme presented here offers a step towards extending [7] to aortic dissection. To this end, the next task is the generation of a block-structure for the parts of the blood vessel which do not have a circular (nor near-circular) cross-section. Figure 4 shows a dissected aorta, with the false lumen in red, and the true lumen in green. Unlike the cross sections found in a healthy aorta, the cross-sections of the false lumen, and its neighbouring true lumen, are far from

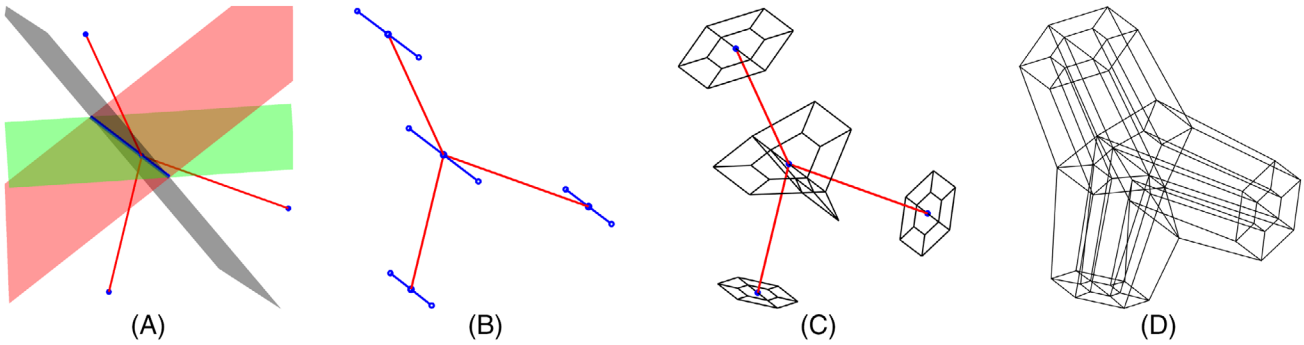

FIGURE 3 Generation of a bifurcation block-structure: (A) intersection (the blue line) of planes (in grey, red and green) determined by taking all 2-combinations of the branches of a bifurcation, (B) reference lines around each point in a bifurcation, (C) cross-sections around the involved points, and (D) the bifurcation block-structure.

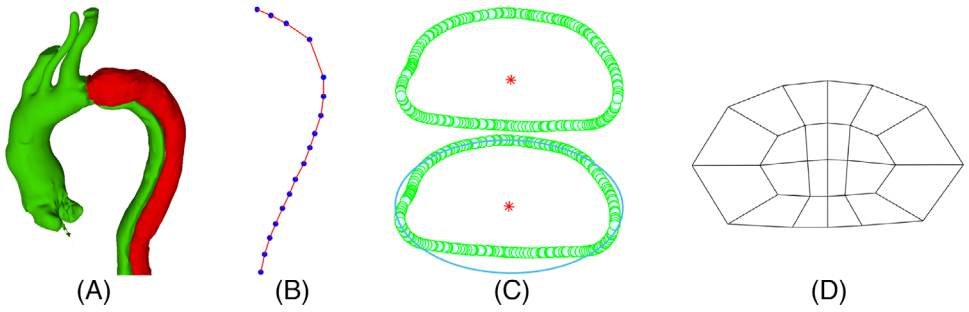

FIGURE 4 Generation of the cross-sections and blocks along a non-circular lumen of a dissected aorta: (A) the surface mesh of an example of aortic dissection, with the true lumen in green and the false lumen in red, (B) the skeleton of the true lumen in the immediate vicinity of the false lumen, (C) an example of a cross-section around a skeletal point in the true lumen, with its minimum-volume bounding ellipse and (D) a cross-section for block-structure generation.

circular. However, in the examples we encountered, they follow somewhat of an ellipsoidal or kidney-shaped pattern, and they may even be non-convex.

Given a non-junction point X, the plane for the cross-section is determined as described earlier. Thereafter, the point X is projected to every surface triangle that intersects the cross-section plane, to form a set of points on the boundary of the lumen, as in Figure 4C. Now we find the *minimum-volume bounding ellipse* of those points. This algorithm seeks to generate an ellipse that contains all points of interest with the minimum volume, noting that this approach may be used in any dimension. Since it may be formulated as a convex optimization problem, we can solve it via established methods, in this case utilising the implementation in Mostagh [11]. Figure 4C showcases a cross-section and the associated minimum-volume (in this case, area) bounding ellipse. Next, we observe that we can use the two axes of the ellipse to automatically build a scheme for the cross-section, such as Figure 4D. Following the construction of the cross-sections, blocks are formed between them, yielding the false lumen block-structure. For the part of the true lumen that is the immediate vicinity of the false lumen, the same procedure is applied. The accompanying block-structures and meshes are in the examples section. Now, everything is ready for the mesh generation step.

The mesh generator employs transfinite maps to compute the proper (inner) node positions, whereas the implicit surface representation is used to equip the block-structure with the geometrical details of the domain. Two user-defined parameters are required, the order of the mesh, denoted by p, and the number of elements per block, denoted by n_{el}, which corresponds to the refinement level (spatial resolution) of the mesh. From these two parameters, we compute the relevant values for the number of nodes per block edge $n_{\text{edge}} = n_{\text{el}} \cdot p + 1$, number of nodes per block face, $n_{\text{face}} = n_{\text{edge}}^2$ and the number of nodes in each block interior $n_{\text{block}} = n_{\text{edge}}^3$.

For completeness, we provide a summary of the transfinite maps [12] for quadrilaterals, noting that the procedure for hexahedra is quite similar. A reference quad (Figure 5A) equipped with an (r, s) coordinate system is associated with four (bilinear) shape functions, denoted by N_i, $i \in 1, \dots, 4$. From the shape functions, we construct a *ramp function* for each edge as

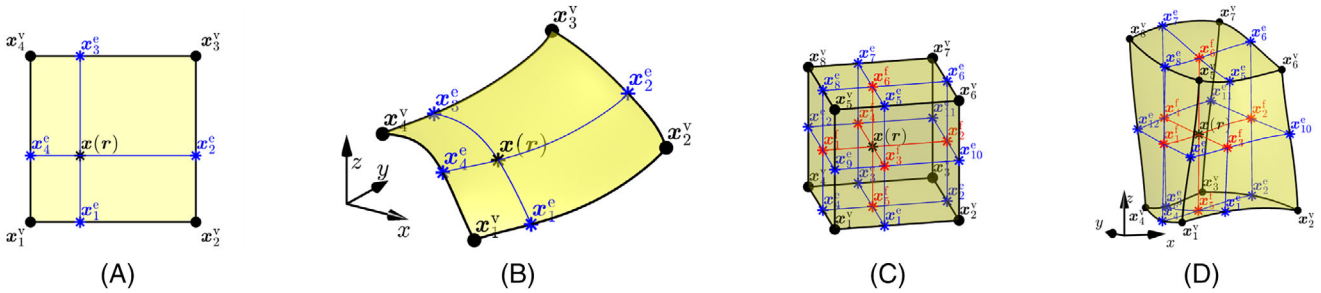

FIGURE 5 Transfinite maps: (A) points needed to get $\boldsymbol{x}(\boldsymbol{r})$, (B) switch to the Cartesian coordinate system (x, y, z); (C) and (D) the equivalent situation for hexahedra. The figure is taken from Bošnjak et al. [7], fig. 16 and fig. 17.

$$R_1 = N_1 + N_2, \quad R_2 = N_2 + N_3,$$
$$R_3 = N_3 + N_4, \quad R_4 = N_4 + N_1.$$

Note that each ramp function equals exactly 1 along the corresponding edge. Let us fix the point $\boldsymbol{rrrr} = (r, s)^{\mathrm{T}}$. We define $\boldsymbol{x}_i^{\mathrm{v}} \in \mathbb{R}^3$ as the vertices of the quadrilateral, and $\boldsymbol{x}_i^{\mathrm{e}} \in \mathbb{R}^3$ as the edge points illustrated in Figure 5A. Given those values, the transfinite map is defined as

$$\boldsymbol{x}(\boldsymbol{r}) = \sum_{i=1}^{4} R_i(\boldsymbol{r}) \cdot \boldsymbol{x}_i^{\mathrm{e}} - \sum_{i=1}^{4} N_i(\boldsymbol{r}) \cdot \boldsymbol{x}_i^{\mathrm{v}}. \tag{1}$$

As mentioned, the process of defining the transfinite maps for hexahedra is quite similar, thus, it is only illustrated in Figure 5C. We refer the interested reader to Solín et al. [12] for more details on transfinite maps, and the associated topics.

Alongside transfinite maps, a method for projecting a point to the surface of the domain is required. If the surface is represented with a SDF, a simple closest point projection may be applied. The evaluation of the function inherently requires the computation of the closest projection on the surface. In the case of convolution surfaces, one may apply a Newton-type iteration algorithm, made possible by the fact that the resulting function is at least C^1. Thereby, before the mesh generation step, all of the surface points of the block-structure are projected onto the surface of the domain.

The mesh generation starts by placing n_{edge} equidistant points on every edge of the block-structure. If the given edge lies on the surface of the domain, we project the edge points onto the surface, noting that only the inner edge nodes need to be projected. To amend potential problems that may occur if the projections are far from their original points, we employ *order elevation*. An edge of the block-structure may be viewed as a one-dimensional linear finite element. It is then converted to a quadratic element, by adding an additional point, which is then projected to the surface. In contrast to the original approach, where points are placed onto the original edge (linear element), they are now placed on the quadratic element, yielding better start guesses for the projection method. If necessary, the process may be repeated, by increasing the order of the edge.

Based on the points generated on the edges of the block-structure, transfinite maps are applied to generate n_{face} points on each block-structure face. Similarly, faces on the surface require the projection of the inner nodes to the surface. In the penultimate step, the transfinite maps for hexahedra are applied to yield the n_{block} points in each block. Finally, we define a connectivity matrix based on the generated points, thereby generating a sub-mesh in every block which is, in turn, combined to the final mesh in the block-structure. It should be noted that the mesh generation step does not depend on this specific setup. Given a block-structure and an accompanying implicit surface representation, the same procedure may be applied. This is particularly relevant as it implies that developing new block-structure schemes does not negatively impact the remainder of the algorithm.

3 | RESULTS

In this section, we show examples of the meshes generated with our method, along with appropriate mesh quality metrics. Figure 6 showcases a part of a patient-specific aorta, following our pipeline. Specifically, we show meshes of different

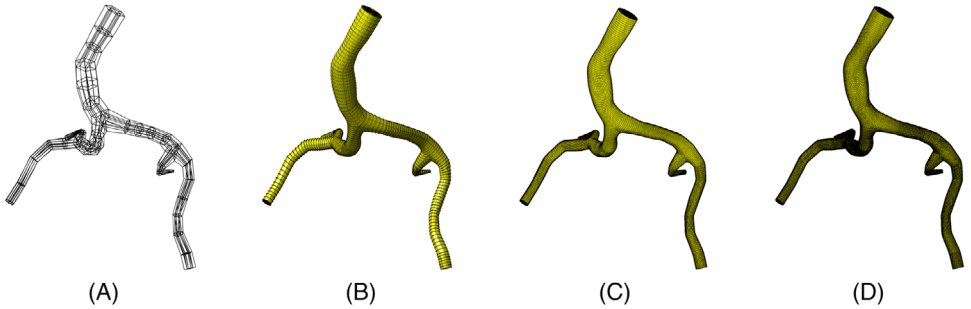

FIGURE 6 Resulting meshes that correspond to the skeleton and implicit surface shown in Figure 1: (A) the block-structure (272 blocks), (B) a sixth-order coarse mesh (272 elements), (C) a linear (first-order) refined mesh (470 016 elements) and (D) a cubic (third-order) refined mesh (470 016 elements).

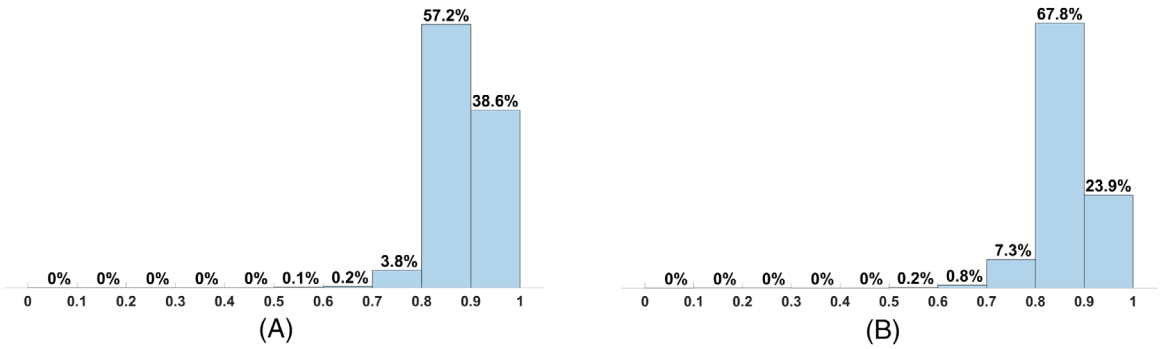

FIGURE 7 The histograms of the scaled Jacobian metric values for the meshes in Figures 6C and 6D. Values closer to 1 indicate elements of higher quality.

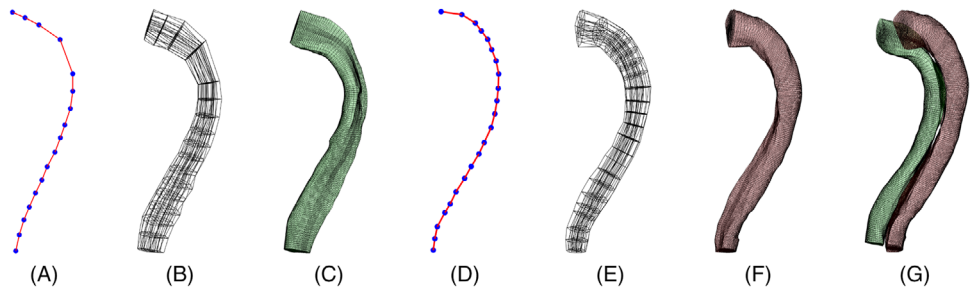

FIGURE 8 An example of hexahedral meshing of a part of a dissected aorta, namely the parts of the lumina with non-circular cross-sections: (A) the skeleton of the true lumen, (B) the block-structure of the true lumen (320 blocks), (C) the linear mesh of the true lumen (163 840 elements) and (D) the skeleton of the false lumen, (E) the block-structure of the false lumen (320 blocks), (F) the linear mesh of the false lumen (215 040 elements) and (G) the two lumina meshes plotted together for better visibility, in terms of the full dissected aorta from Figure 4.

orders and spatial resolutions. As mentioned in the previous section, the blocks are valid linear elements themselves, thus we observe that the coarsest mesh we are able to produce with our approach has the same number of elements as the number of blocks in the block-structure.

Thereafter, Figure 7 shows the standardly used *scaled Jacobian* mesh quality metric for the two refined meshes presented in Figure 6C and 6D. For a given element, the value may be in $\langle -\infty, 1]$, where a negative value indicates an invalid element, leading to an invalid and unusable mesh, and a value of 1 indicates an ideal element.

Observing the results, we see that both meshes exhibit very high quality, as most of the elements belong to the final two intervals. We do not reasonably expect for all of the elements to be ideal, thus, the results are extremely satisfactory. For both cases, the median value of the scaled Jacobian was around 0.89.

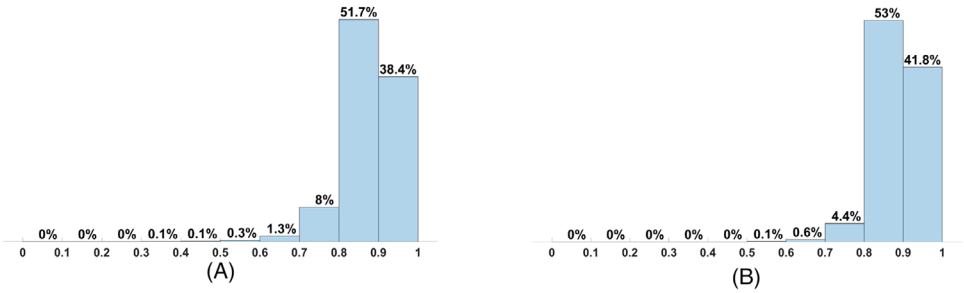

FIGURE 9 The scaled Jacobian metric for the meshes of the false and true lumen shown in Figures 8C and 8F.

The next example shows the mesh of the part of the dissected aorta with non-circular cross-sections, separately for the true and the false lumen. As mentioned, the cross-sections are ellipsoidal and potentially non-convex. Figure 8 illustrates the skeletons, the block-structures and the linear meshes. The relevant scaled Jacobian histograms are shown in Figure 9. As the cross-sections are more complex compared to the healthy blood vessel case, we see an increase in the number of blocks in the block-structure in both lumina, relative to the number of points in the skeleton.

4 | CONCLUSIONS

We have presented an automatic approach for the generation of structured, higher-order and fully hexahedral meshes of vascular networks. They offer parametric control of anisotropy, and may be very coarse if desired. The meshes are generated in MATLAB and may be exported in the well-known *.msh* format. The values of the widely used scaled Jacobian mesh metric are an indicator of very high mesh quality. Additionally, we presented relevant steps towards structured meshing of dissected aortas, whereby the additional effort required to automatically generate appropriate block-structures is left for future work.

ACKNOWLEDGMENTS

The authors gratefully acknowledge Graz University of Technology, Austria for the financial support of the LEAD-project: Mechanics, Modelling and Simulation of Aortic Dissection.

The open access publishing was enabled by the funding of Graz University of Technology.

ORCID

Thomas-Peter Fries https://orcid.org/0000-0003-1210-1557

REFERENCES

1. Herbert, T. (1988). Secondary instability of boundary layers. *Annual Review of Fluid Mechanics*, *20*(1), 487–526.
2. Ghaffari, M., Tangen, K., Alaraj A., Du, X., Charbel, F., & Linninger, A. (2017). Large-scale subject-specific cerebral arterial tree modeling using automated parametric mesh generation for blood flow simulation. *Computers in Biology and Medicine*, *91*, 353–365.
3. Livesu, M., Muntoni, A., Puppo, E., & Scateni, R. (2016). Skeleton-driven adaptive hexahedral meshing of tubular shapes. *Computer Graphics Forum*, *35*(7), 237–246.
4. Pietroni, N., Campen, M., Sheffer, A., Cherchi, G., Bommes, D., Gao, X., Scateni, R., Ledoux, F., Remacle, J., & Livesu, M. (2022). Hex-mesh generation and processing: A survey. *ACM Transactions on Graphics*, *42*(2), 1–44.
5. Schussnig, R., Pacheco, D., Kaltenbacher, M., & Fries, T. P. (2022). Semi-implicit fluid–structure interaction in biomedical applications. *Computer Methods in Applied Mechanics and Engineering*, *400*, 115489.
6. Au, O., Tai, C., Chu, H., Cohen-Or, D., & Lee, T. (2008). Skeleton extraction by mesh contraction. *ACM Transactions on Graphics*, *27*(3), 1–10.
7. Bošnjak, D., Pepe, A., Schussnig, R., Schmalstieg, D., & Fries, T. (2023). Higher-order block-structured hex meshing of tubular structures. *Engineering with Computers*, (05), 1–21.
8. Bloomenthal, J., Bajaj, C., Blinn, J., Cani-Gascuel, M., Rockwood, A., Wyvill, B., & Wyvill, G. (1997). *Introduction to implicit surfaces*. Morgan Kaufmann.
9. Fuentes Suárez, A., Hubert, E., & Zanni, C. (2019). Anisotropic convolution surfaces. *Computers & Graphics*, *82*, 106–116.
10. Armstrong, C., Fogg, H., Tierney, C., & Robinson, T. (2015). Common themes in multi-block structured quad/hex mesh generation. *Procedia Engineering*, *124*, 70–82.

11. Mostagh, N. (2023). Minimum volume enclosing ellipsoid, https://www.researchgate.net/publication/254980367_Minimum_volume_enclosing_ellipsoid.
12. Šolín, P., Segeth, K., & Doležel, I. (2003). *Higher-order finite element methods*. CRC Press.

How to cite this article: Bošnjak, D., & Fries, T.-P. (2023). Block-structured mesh generation from implicit geometries for cardiovascular applications. *Proceedings in Applied Mathematics and Mechanics*, *23,* e202300023. https://doi.org/10.1002/pamm.202300023