

STRUCTURED, HIGHER-ORDER, HEXAHEDRAL MESH GENERATION OF VASCULAR NETWORKS

Domagoj Bošnjak¹

Thomas-Peter Fries²

*Institute of Structural Analysis
Graz University of Technology
Lessingstr. 25, 8010 Graz, Austria*

www.ifb.tugraz.at

¹ *bosnjak@tugraz.at*

² *fries@tugraz.at*

ABSTRACT

A method for the generation of structured, higher-order, hexahedral meshes is presented, with emphasis on vascular networks. Based on CT scans and segmentations, a topological skeleton and an implicit level-set surface description are extracted. From the skeleton, in fact, any wireframe model, a block structure is generated automatically, based on predefined prototypes. Mesh grading may be assigned in order to account for boundary layers or expected singularities. The mesh generator based on transfinite maps then utilises the block-structure, the grading information, and the implicit level-set surface description. The resulting meshes are suitable for fluid simulations, being of any order and resolution, hence, suitable in the vast context of the *hp*-FEM or other higher-order methods.

Keywords: mesh generation, block-structure, convolution surfaces, signed-distance function, transfinite maps, higher-order meshes

1. INTRODUCTION

Generating high-quality meshes of complex domains is a requirement for various numerical methods, additionally so because the mesh affects both the computational effort and the accuracy of the simulation. Unstructured meshing is generally more practical for arbitrary domains, however, there is only reduced control over mesh features such as grading and order of the elements. On the other hand, structured meshing enables direct control over the number of nodes and elements in the mesh, and the number of elements sharing a node. However, this comes at the price of usually simpler domains of interest to be meshed. With respect to element types, hexahedra are more suitable for resolution of *boundary layers* [1] in fluid flow simulations, i.e., thin layers of fluid in the vicinity of the domain boundary. Moreover, hexahedra are the better option when applying *multigrid methods* for the effi-

cient solution of systems of equations [2]. Hence, in this work we opt for structured, hexahedral meshing.

The established approaches to this type of structured hexahedral mesh generation [3, 4, 5, 6] usually focus on linear (hence, low-order) meshes, whereas we aim to generate higher-order meshes as well. The motivation lies in the growing popularity of higher-order methods, such as the well known *p*-FEM [7] (regardless of whether continuous or discontinuous Galerkin methods are employed), and the lack of (very) high order mesh generators [8].

Our approach is based on the *block-structured mesh generation* [9], where the domain is sub-divided into blocks, whose topology represents the topology of the domain. Each coarse block may then be meshed individually. The generation of the block structure depending on the domain of interest is a topic in itself [10, 11].

2. SURFACE DESCRIPTION

We outline two ways of representing the surface of the domain; though both result in a function whose zero-isosurface $\{\mathbf{x} \in \mathbb{R}^3 : f(\mathbf{x}) = 0\}$ represents the surface. The first approach consists of generating a signed-distance function directly from a triangulated surface, performed by established methods [12]. A *topological skeleton* (centerline) must be extracted as well, e.g., by mesh contraction [13]. Generating a suitable skeleton for a wide spectrum of domains may require user interaction, though the restrictions on the domain shape are more relaxed compared to the approach outlined next. In the second approach, which is employed by the authors of this work in [14], a *convolution surface* [15] description is extracted. This results in a smooth (at least C^1 -continuous) level-set function, based on a skeleton with radius information. The advantages of this approach are that it is fully automatic, in addition to inherent convolution surface benefits, such as smooth blending between vessels and easy topology control. However, a clear restriction is the assumption that each cross-section of the vessel (aside from, e.g., bifurcations) is exactly circular. Herein, we propose to move away from convolution surfaces, and instead utilise the first discussed approach, which is conceptionally simpler, yet relies on a faceted, hence, only C^0 -continuous surface description based on flat triangles.

3. BLOCK STRUCTURE GENERATION

We assume that a skeleton has been generated, alongside a signed-distance function, obtained from the initial surface representation by flat triangles (e.g., in the STL-format). We then generate the block structure from the skeleton, based on several prototype situations that occur in the domains we mesh. Fig. 1 illustrates the skeleton and the generated block structure for an iliac bifurcation.

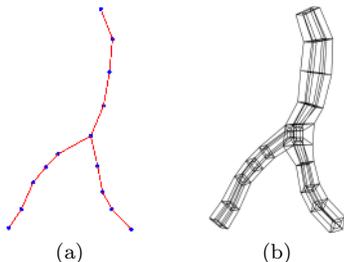


Figure 1: An example of a (a) skeleton of a patient-specific iliac bifurcation and (b) the generated block structure.

In the skeleton, we distinguish between junction and non-junction points. Around the latter, we gener-

ate a cross-section, as shown in Fig. 2a. The cross-sections are then connected to form blocks, as shown in Fig. 2b. Around the junction points, we generate cube-like structures shown in Fig. 2c, consisting of 7 blocks each. The sides of the junction structure are modified to be able to connect with a cross-section, as in Fig. 2d. The orientation is determined to minimize the angle to the junction branches. Several additional block structure prototypes are used, depending on the number of branches and the branching angles, but they are omitted here for brevity. It is worth noting that extending the overall method would only require adding new block structure prototypes, and the rest of the algorithm could remain unchanged.

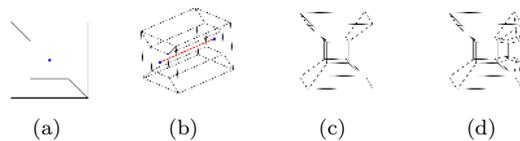


Figure 2: The block structure generation: (a) cross-section at a non-junction point, (b) the blocks formed by two cross-sections, (c) structure around a junction point and (d) modified junction structure to accommodate for a connection with a cross-section.

A crucial point here is the fact that the generated block structure does not have to be perfectly aligned to the surface. Given a signed-distance function, we employ the *closest point projection* to bring a given point to the surface of the domain. In the convolution surface approach from [14], the projection is achieved with a Newton scheme. Hence, the automatically generated block structure only needs to be a suitable first guess, with respect to the surface alignment. This bodes well in case we consider a more complex domain.

The user may assign mesh grading to the block structure to induce tailored refinement of the mesh. This may be done, e.g., to generate thinner layers of elements near the boundary, to account for boundary layers. Having generated the implicit surface description and the block structure, and potentially assigned mesh grading, we proceed to the mesh generation step.

4. MESH GENERATION

The mesh generator requires two input parameters, the number of elements per block, denoted by n_{el} , and the desired order of the mesh, denoted by p . From these two, we can compute the number of nodes per block edge $n_{edge} = n_{el} \cdot p + 1$, number of nodes per block face, $n_{face} = n_{edge}^2$, and total number of nodes in the interior of each block $n_{block} = n_{edge}^3$.

We give a brief overview of the transfinite maps [16],

in particular for quads, noting that a similar approach may be followed for hexas. For more details, we refer the reader to [14]. We begin with a reference quad, as in Fig. 3a, with an (r, s) coordinate system and the four bilinear shape functions N_i , $i = 1, \dots, 4$.

For each of the four edges, we define a *ramp function*, whose value along the given edge is 1:

$$\begin{aligned} R_1 &= N_1 + N_2, & R_2 &= N_2 + N_3, \\ R_3 &= N_3 + N_4, & R_4 &= N_4 + N_1. \end{aligned}$$

Given a point $\mathbf{r} = (r, s)^T$, eight other points are needed to define the map. The four points $\mathbf{x}_i^y \in \mathbb{R}^3$ are simply the vertices of the quad, and the four points $\mathbf{x}_i^e \in \mathbb{R}^3$ lying on the edges of the quad are naturally obtained as illustrated in Fig. 3a. Now, the definition of the transfinite map is given by

$$\mathbf{x}(\mathbf{r}) = \sum_{i=1}^4 R_i(\mathbf{r}) \cdot \mathbf{x}_i^e - \sum_{i=1}^4 N_i(\mathbf{r}) \cdot \mathbf{x}_i^y. \quad (1)$$

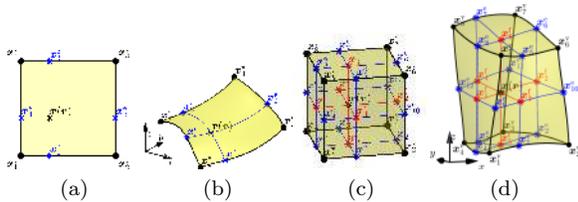


Figure 3: Transfinite maps: (a) points needed to get $\mathbf{x}(\mathbf{r})$, (b) switch to the Cartesian coordinate system (x, y, z) ; (c) and (d) the equivalent situation for hexas.

A schematic illustration of the approach for hexas is given in Figs. 3c and 3d.

Some edges and faces of the blocks may be associated with an implicitly defined surface. Hence, we go over each edge, place the n_{edge} points on it, then project the points onto the surface, as discussed in Sec. 3. Otherwise, no projection is required. Better start guesses for the projection may be obtained through *order elevation*. Interpreting the edge as a 1D linear finite element, we first raise the order of the edge to 2, yielding a new point, which we then project to the surface. Thus, points are placed on the quadratic element instead, to yield better start guesses. The process may be repeated up to a user-defined order.

After the edges, we generate points on faces of the block structure, via transfinite maps, as in Eq. 1. Thereafter, we consider the associated implicit geometry definition by projecting these points to the surface. Finally, transfinite maps for hexas are applied to generate the nodes in the interior of each block. After defining all nodal positions, a connectivity matrix has to be generated to relate the nodes to elements, finalizing the mesh generation.

5. EXAMPLES

We show examples of meshes coming from a patient-specific aorta, namely the bottom part starting from the iliac bifurcation. The meshes are generated based on convolution surfaces, but still differently than in [14], due to the incorporation of the level-set function into the mesh generator, thereby avoiding the in-between step of surface discretization.

We show the skeleton, the block structure, a linear mesh and a higher-order (cubic) mesh, in Fig. 4.

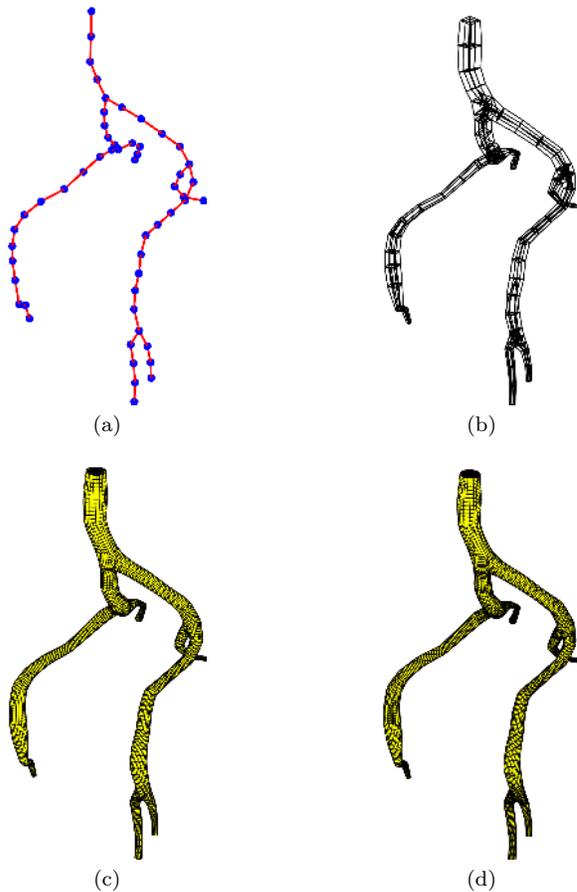


Figure 4: A patient-specific aorta, the bottom part starting from the iliac bifurcation at the top: (a) the skeleton, (b) the block structure with 373 blocks, (c) the linear mesh with 80568 elements and (d) the cubic (order 3) mesh with 80568 elements.

Additionally, we show mesh quality metrics, namely the *scaled Jacobian* and the *equiangular normalized skewness* [5], both of which are widely used metrics. The skewness is only computed for the linear mesh. Fig. 5 contains the histograms of the values of the aforementioned metrics.

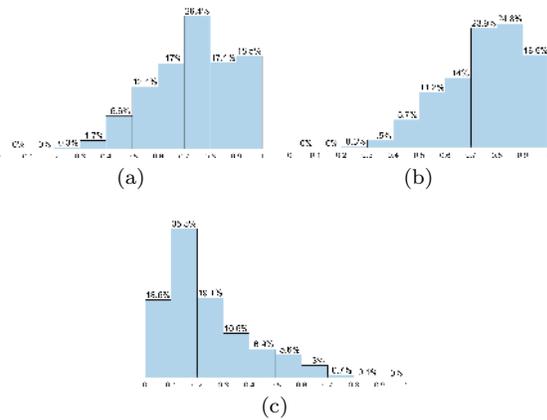


Figure 5: Mesh quality metrics for the meshes in Fig. 4: (a) the scaled Jacobian of the linear mesh, (b) the scaled Jacobian of the cubic mesh and (c) the equiangular normalized skewness for the linear mesh.

6. CONCLUSIONS

We propose a new method for pure hexahedral, block-structured mesh generation, based on an implicit surface description. The implicit level-set approach is successfully incorporated into the mesh generator. Good results were obtained with the convolution surface approach, motivating us to generalize the method to an approach which relies on the signed-distance function, whilst retaining a topological skeleton for block structure generation. This methodology would pave the way for meshing a wider class of domains, without the restriction of perfectly radial cross-sections as in [14].

References

- [1] Herbert T. “Secondary Instability of Boundary Layers.” *Annual Review of Fluid Mechanics*, vol. 20, no. 1, 487–526, 1988
- [2] Stüben K. “A review of algebraic multigrid.” *J. Comput. Appl. Math.*, vol. 128, no. 1, 281–309, 2001
- [3] Ghaffari M., Tangen K., Alaraj A., Du X., Charbel F., Linninger A. “Large-scale subject-specific cerebral arterial tree modeling using automated parametric mesh generation for blood flow simulation.” *Comput. Biol. Med.*, vol. 91, 2017
- [4] Antiga L., Ene-Iordache B., Caverni L., Cornalba G., Remuzzi A. “Geometric reconstruction for computational mesh generation of arterial bifurcations from CT angiography.” *Comput. Med. Imaging Graph.*, vol. 26, no. 4, 227–235, 2002
- [5] De Santis G., De Beule M., Van Canneyt K., Segers P., Verdonck P., Verhegghe B. “Full-hexahedral structured meshing for image-based computational vascular modeling.” *Med. Eng. Phys.*, vol. 33, 1318–1325, 2011
- [6] Vukicevic A., et al. “Three-dimensional reconstruction and NURBS-based structured meshing of coronary arteries from the conventional X-ray angiography projection images.” *Sci. Rep.*, vol. 8, 1711, 01 2018
- [7] Babuska I., Suri M. “The P and H-P Versions of the Finite Element Method, Basic Principles and Properties.” *SIAM Rev.*, vol. 36, no. 4, 578–632, 1994
- [8] Turner M. *High-order mesh generation for CFD solvers*. Dissertation, Imperial College London, 2017
- [9] Armstrong C., Fogg H., Tierney C., Robinson T. “Common Themes in Multi-block Structured Quad/Hex Mesh Generation.” *Procedia Eng.*, vol. 124, 70–82, 2015
- [10] Ali Z., Tyacke J., Tucker P., Shahpar S. “Block Topology Generation for Structured Multi-block Meshing with Hierarchical Geometry Handling.” *Procedia Eng.*, vol. 163, 212–224, 2016
- [11] Ali Z., Dhanasekaran P., Tucker P., Watson R., Shahpar S. “Optimal multi-block mesh generation for CFD.” *Intern. J. of Comp. Fluid Dynamics*, pp. 195–213, 2017
- [12] Guezlec A. “Meshsweeper”: dynamic point-to-polygonal mesh distance and applications.” *IEEE Trans. Vis. Comput. Graph.*, vol. 7, no. 1, 47–61, 2001
- [13] Au O., Tai C., Chu H., Cohen-Or D., Lee T. “Skeleton extraction by mesh contraction.” *ACM Trans. Graph.*, vol. 27, no. 3, 1–10, 2008
- [14] Bošnjak D., Pepe A., Schussnig R., Schmalstieg D., Fries T.P. “Higher-order block-structured hex meshing of tubular structures.”, 2023. Submitted
- [15] Fuentes Suárez A., Hubert E., Zanni C. “Anisotropic convolution surfaces.” *Comput. Graph.*, vol. 82, 106–116, 2019
- [16] Šolín P., Segeth K., Doležel I. *Higher-order finite element methods*. CRC Press, Boca Raton, FL, 2003