INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN BIOMEDICAL ENGINEERING
Int. J. Numer. Meth. Biomed. Engng. 2010; **00**:1–17
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/cnm

Cardiovascular and lung mesh generation based on centerlines

E. Marchandise[1]*, C. Geuzaine[2], J.F. Remacle[1]

[1] *Université catholique de Louvain, Institute of Mechanics, Materials and Civil Engineering (iMMC), Place du Levant 1, 1348 Louvain-la-Neuve, Belgium*
[2] *Université de Liège, Department of Electrical Engineering and Computer Science, Montefiore Institute B28, Grande Traverse 10, 4000 Liège, Belgium*

SUMMARY

We present a fully automatic procedure for the mesh generation of tubular geometries such as blood vessels or airways. The procedure is implemented in the open-source Gmsh software and relies on a centerline description of the input geometry. The is able to generate different type of meshes: isotropic tetrahedral meshes, anisotropic tetrahedral meshes and mixed hexahedral/tetrahedral meshes. Additionally a multiply layered arterial wall can be generated with a variable thickness. All the generated meshes rely on a mesh size field and a mesh metric that is based on centerline descriptions, namely the distance to the centerlines and a local reference system based on the tangent and normal directions to the centerlines. Different examples show that the proposed method is very efficient and robust and leads to high quality computational meshes. Copyright © 2010 John Wiley & Sons, Ltd.

Received . . .

KEY WORDS: Mesh Generation; Centerline; Patient-specific; Anisotropic mesh; Boundary Layer; Quad mesh; Hexahedral mesh

1. INTRODUCTION

Nowadays patient specific simulations are widely used in the cardiovascular or respiratory field in order to investigate either local hemodynamics or deposition fraction from inhaled respiratory aerosols. An important but time consuming step in the modeling process is the creation of a quality computational mesh based on the segmented geometry. This segmented geometry is a tubular geometry with possibly several bifurcations and that is most of the times an STL triangulation of low quality.

The pipeline leading from the segmented geometry to the computational mesh usually comprises: (i) the generation of a high quality surface mesh, (ii) the creation and meshing of planar regions for boundary conditions, (iii) the extrusion of the surface mesh in order to create the geometry and volume mesh of the solid wall, (iv) the volume mesh generation of the lumen. The latter step sometimes includes a boundary layer mesh in the vicinity of the lumen surface. Most known solutions to create a computational mesh still require to perform manually those different steps. If some of those steps (such as the tetrahedral meshing) are automated, other steps still require a large amount of manual processing.

In this paper, we present a fully automatic procedure for the mesh generation of tubular geometries. The procedure relies on a centerline description of the geometry that is computed

*Correspondence to: emilie.marchandise@uclouvain.be.

using the open source Vascular Modeling Toolkit (VMTK, www.vmtk.org). The new automatic meshing procedure is very efficient and robust and leads to high quality computational meshes. It is implemented inside the open source meshing software Gmsh [12] and examples on how to use it can be found on the Gmsh wiki (https://geuz.org/trac/gmsh).

The procedure is able to generate different type of meshes: isotropic tetrahedral meshes, anisotropic tetrahedral meshes or mixed hexahedral/tetrahedral meshes. Additionally a multiply layered arterial wall can be generated with a variable thickness. All the generated meshes rely on a mesh size field and a mesh metric that is based on centerline descriptions (distance to centerlines and local reference system based on the tangent and normal directions to the centerlines).

Our mixed hexahedral/tetrahedral meshes contain hexahedra in the extruded wall and tetrahedra in the vessel lumen. Pyramids are created as transition elements between tetrahedra and hexahedra [23]. Those meshes are based on a quadrangular surface mesh of the geometry. The presented algorithm of quadrangular mesh generation is an original approach that is a combination of a structured elliptic mesh generation approach [32] and an indirect approach [26] that uses distances in the L^∞ norm as a base for inserting new points and generates right triangles that are then recombined into quadrangles [27]. From the quadrangular surface mesh, we generate an extruded boundary layer mesh as well as an unstructured tetrahedral mesh of the lumen with pyramids as transition elements. We are currently also working on an optimal placement of the mesh vertices in the lumen for the recombination of the tetrahedra into hexahedra [35, 34, 20], which will allow to also generate hexahedral dominant meshes. It follows that the presented unstructured approach for hexahedral mesh generation is a general and quite fast compared with block-structured hexahedral mesh generation such as centerline-based sweeping methods [36, 7]. Indeed, such block-structured methods require that source and target have similar topology, which means that many templates for different tubular branching configurations (trifurcation or higher order branching) are needed.

Our centerline-based meshing algorithm is able to generate high quality tetrahedral and mixed meshes as well as adaptive tetrahedral meshes suitable for numerical methods, such as finite element or finite volume methods. As far as hexahedral meshes are concerned, such meshes are very attractive. It has indeed been shown by several authors that hexahedral meshes provide higher accuracy and reduce computational costs both for Finite Element Analysis (FEA) and Computational Fluid Dynamics (CFD) [7, 33, 16]. For example De Santis showed in [7] that when calculating the Wall Shear Stress (WSS) from the CFD solution, hexahedral meshes converge better and, for the same accuracy of the result, require less computational time when compared to tetrahedral or mixed tetrahedral/prismatic meshes. In a similar manner, Vinchurkar [33, 16] showed that structured or unstructured hexahedral meshes were more efficient (better grid convergence) than other types of meshes for predicting particle deposition in a bifurcating model of the respiratory tract because the mesh elements are aligned with the predominant direction of flow, thereby reducing numerical diffusion errors. In the same vein, anisotropic tetrahedral meshes aligned with the dominant direction of flow (i.e. the centerlines) are also very attractive as they reduce the number of elements and degrees of freedom, leading to significant computational savings for a given level of accuracy [30, 29, 4].

Different examples show that the proposed meshing approach is very efficient and robust and that the generated meshes are of high quality.

2. MATERIALS AND METHOD

In this section, we present the specific implementation of the pipeline of our automatic meshing approach based on centerlines. The starting point of our algorithm is the segmented triangulated surface of arbitrary quality[†], i.e a set of N mesh triangles $\mathcal{T} = \{T_1, ..., T_N\}$ and a set of M surface mesh vertices $\mathcal{X} = \{\mathbf{x}_1, ..., \mathbf{x}_M\}$.

[†]e.g a surface in STL format

2.1. *Generation of centerlines*

The centerlines $\mathcal{C}$ of a tubular geometry are the paths on the Voronoi diagram that minimize the integral of the radius of maximal inscribed spheres along the path. The centerlines are computed using the open source Vascular Modeling Toolkit [3, 2] (VMTK, `www.vmtk.org`) as follows[‡]:

```
vmtk vmtkcenterlines -seedselector openprofiles -ifile input.stl -ofile centerlines.vtk
```

The discrete centerlines (as given by VMTK) are a set of K consecutive line segments $\mathcal{C} = \{s_1, s_2, ..., s_K\}$ and L mesh vertices $\mathcal{X}_c = \{\mathbf{x}_{c_1}, ..., \mathbf{x}_{cL}\}$. Every line segment s_k is defined by a starting point $\mathbf{x}_{ci}$ and end point $\mathbf{x}_{cj}$ on the centerlines $s_k = \overline{\mathbf{x}_{ci}\mathbf{x}_{cj}}$. A new mesh field named *Centerline* has been implemented within Gmsh that takes as input such discrete centerlines $\mathcal{C}$. From the centerline field, two geometrical descriptors are computed and three different operators are defined.

Figure 1 shows a geometrical model of a bronchial airway tree together with the extracted centerlines. Both colors and thickness of the centerlines represent the vessel radius computed as described in the following paragraph.

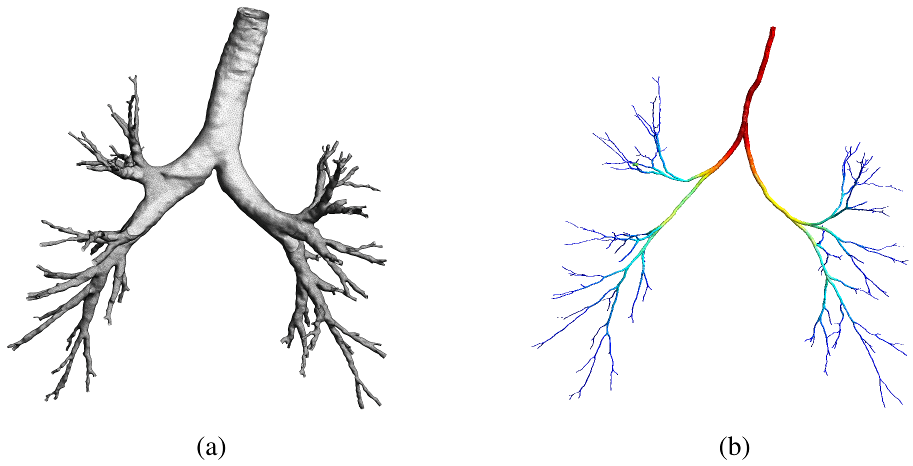


Figure 1. Geometric models of a patient-specific bronchial airway tree (a) together with the extracted centerlines (b). The colors and thickness of the centerlines represent the vessel radius (a centerline-based descriptor computed within Gmsh).

2.2. *Centerline-based descriptors in Gmsh*

Two geometrical descriptors are computed from the centerlines $\mathcal{C}$:

- the local radii,
- a triad axis.

The local radius $r(\mathbf{x}_c)$ of the vessel is the distance from a point on the centerline $\mathbf{x}_c$ to the tubular geometry. This local radius is computed efficiently by first storing all the mesh vertices of the tubular geometry in a search tree (Approximate Nearest Neighbor ANN) [21, 22] and by using the search tree to compute the closest point $\mathbf{x}_p \in \mathcal{X}$. The local radius $r(\mathbf{x}_c)$ is the distance between $\mathbf{x}_p$ and $\mathbf{x}_c$.

The triad axis is a local reference system for every point in the volume $\mathbf{x} \in \mathbb{R}^3$ that is based on the centerline field. The local axes are the abscissa (unit vector tangent to the centerlines), the normal (a vector perpendicular to the centerlines) and the binormal which can be calculated as a cross product

[‡]It should be noted that prior to computing the centerlines, flow extensions can be added with vmtk for the inlets and outlets of the tubular geometry (*inputExtended.stl*), so that the computed centerlines (*centerlines.vtk*) extend out from the inlets and outlets of the input tubular geometry (*input.stl*).

of the abscissa and normal: $(\mathbf{e}_t(\mathbf{x}), \mathbf{e}_n(\mathbf{x}), \mathbf{e}_{bn}(\mathbf{x}))$. The triad axis is computed as follows: for a point $\mathbf{x}$ in the volume to be remeshed (i) compute, using a fast search tree such as ANN, the two closest points on the centerlines $\mathbf{x}_{c1}$ and $\mathbf{x}_{c2}$ so that $\mathbf{e}_t = \mathbf{x}_{c2} - \mathbf{x}_{c1}$, (ii) compute the normal $\mathbf{e}_n = \mathbf{x} - \mathbf{x}_{c1}$, (ii) compute the binormal as the cross product of the abscissa and the normal. Figure 2 shows an example of the local radius $r(\mathbf{x}_c)$ for a point on the centerline and of a triad axis for a point $\mathbf{x}$ in the volume of an aortic arch. As will be explained in the next sections, the triad axis enables us to define an anisotropic mesh metric in order to produce anisotropic meshes.

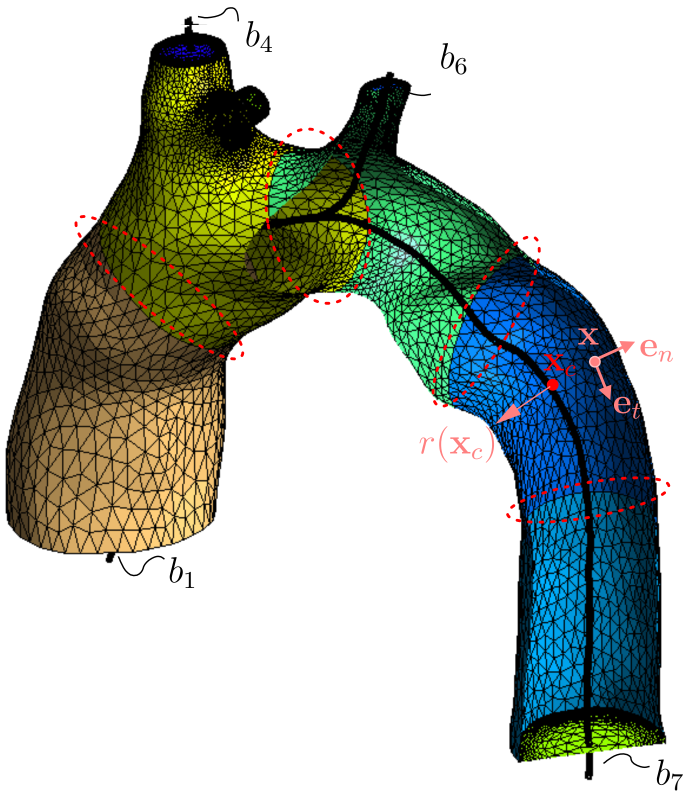


Figure 2. Centerline-based geometrical descriptors and operators used for the generation of an anisotropic mesh of a human aorta. The local radius $r(\mathbf{x}_c)$ is defined at a point on the centerlines $\mathbf{x}_c$ and the triad axis $(\mathbf{e}_t(\mathbf{x}), \mathbf{e}_n(\mathbf{x}), \mathbf{e}_{bn}(\mathbf{x}))$ is defined at a mesh point $\mathbf{x} \in \mathbb{R}^3$. The branched tree $\mathcal{B}$ defined from the centerlines is made of $W = 7$ branches. The initial surface mesh $\mathcal{T}$ has been cut by disks at 4 different locations (see red circles) being either the tree bifurcations or locations along the long branches (such as branch b_7).

2.3. *Centerline-based operators in Gmsh*

Besides the geometrical descriptors, three centerline-based operators are defined:

- a cut operator,
- a close-volume operator,
- a vessel wall model generation.

The cut operator is used for the remeshing of the initial triangulation. This centerline-based operator can cut the initial tubular geometry into different mesh patches S_j of moderate geometrical aspect ratio $\eta = H/D$, i.e the ratio between the height and the diameter of the tube. This cut operator is very useful for the remeshing of the tubular geometries that have a large aspect ratio η (see the 5 mesh patches of Figure 2 and the 17 mesh patches of Figure 4). This cut operator is important as our surface remeshing technique relies on a discrete conformal parametrization of the surface [17, 25, 18, 19], as we have shown that a necessary condition for avoiding indistinguishable coordinates in the parametric space is to ensure that the geometrical aspect ratio remains smaller than $\eta = 4$. The conformal parametrizing of a surface patch S_j is a mapping $\mathbf{u}(\mathbf{x})$ that transforms continuously the 3D surface S_j into a 2D surface S'_j embedded in $\mathbb{R}^2$ and that preserves (in a least square sense) the angles between the iso-u and iso-v lines.

$$\mathbf{x} \in \mathsf{S}_j \subset \mathbb{R}^3 \mapsto \mathbf{u}(\mathbf{x}) \in \mathsf{S}'_j \subset \mathbb{R}^2. \tag{1}$$

Figure 3 shows the computed conformal mapping for a triangulated surface of a tri-bifurcation (see

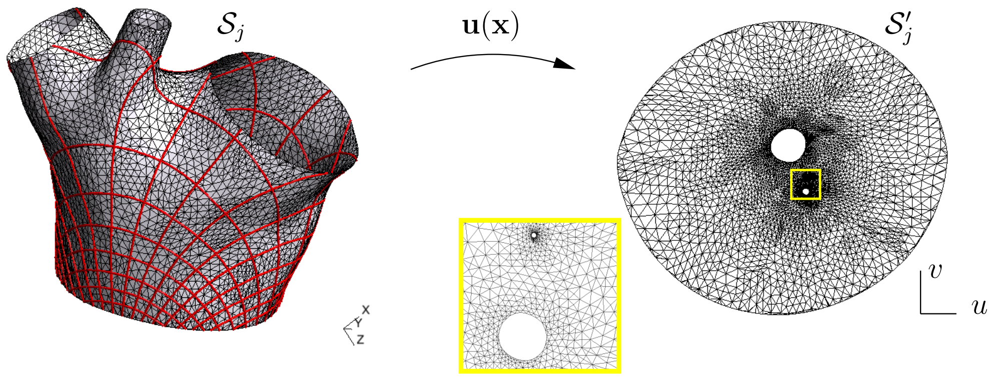
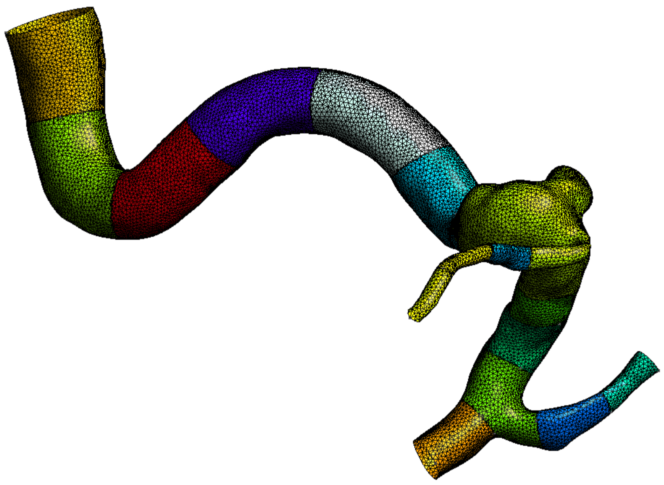


Figure 3. A conformal mapping $\mathbf{u}(\mathbf{x}) = \{u(\mathbf{x}), v(\mathbf{x})\}$ that transforms continuously a 3D surface $\mathsf{S}_j \in \mathbb{R}^3$ (a tri-bifurcation) into a 2D surface S'_j embedded in $\mathbb{R}^2$. The red lines on the 3D surface are the iso-u and iso-v lines.

[19] for more details about how to compute the conformal mapping). The initial surface patch S_j can then be remeshed in the parametric space using any 2D mesh generation procedure with a given variable isotropic mesh size field or an anisotropic mesh metric. The new mesh is then mapped back to the original surface.

The cut operator relies on a branched tree structure of the centerlines. A branched tree is a set of W branches $\mathcal{B} = \{b_1, ..., b_W\}$ that know their children branches (see the branched tree structure of the lung made of $W = 305$ branches in Figure 1b or the branched tree structure of the aorta made of $W = 7$ branches in Figure 2). The data structure of a branch is described in Algorithm 1. The variables of a branch b_j are its length l_j, the minimum and maximal local radius $r(\mathbf{x}_{cj})$ of all mesh vertices of the branch $\mathcal{X}_c^{b_j} = \{\mathbf{x}_{c1}, ..., \mathbf{x}_{cJ}\}$. Using the branched tree, the cut algorithm is defined in Algorithm 2. We loop over all the branches of the tree, and cut the mesh by a disk at all the tree bifurcations and at different lengths of the branch such that the maximal aspect ratio of a branch never exceeds $\eta = 4$. The *cutByDisk* function cuts the triangular input mesh $\mathcal{T} = \{T_1, ..., T_N\}$ by a disk. Every cut triangle T_i is divided into 3 sub-triangles. The disk is defined by a point $\mathbf{x}_c$ on the centerlines, the direction of the centerlines $\mathbf{d}(\mathbf{x}_c)$ and a radius R that is taken as a percentage of the local radius at that point: $R = 1.2\, r(\mathbf{x}_c)$. Figure 4 shows an example of an initial STL mesh $\mathcal{T}$ that has been cut by 16 disks using the centerline-based cut operator. The resulting mesh contains 17 different surface patches S_j than can be remeshed using the remeshing techniques based on parametrizations. Figure 4b shows the conformal parametrization of the white mesh patch.

The second operator is the close-volume operator. This operator creates planar faces at the inlet and outlet of the tubular geometry. The planar faces are defined by the mean plane of the mesh vertices $\mathbf{x} \in \mathcal{X}$ located at the boundaries of the input geometry. Those faces can then be subsequently meshed by the planar mesh generators.



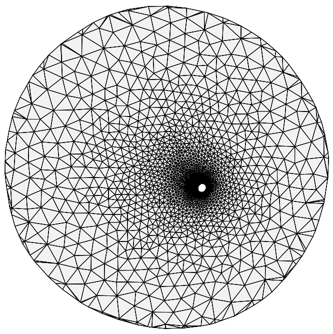


Figure 4. The cut operator based on the centerlines has cut the initial tubular geometry, a cerebral aneurysm, into 17 different mesh surfaces S_j (a). For the remeshing of those patches, a conformal parametrization $\mathbf{u}(\mathbf{x})$ is computed for every mesh patch. The parametrization is shown for the white mesh patch (b).

Algorithm 1 A Branch of a tree

```
struct Branch {
    double length;
    double minRad, maxRad;
    vector<line segments> lines;
    vector<Branch> children;
}
```

Algorithm 2 The cut-operator algorithm

```
for $i = 0$ to $W$ branches do
   b = branches(i)
   if b.children then
      point, direction, radius $\leftarrow$ b.lines.end (location of the bifurcation)
      cutByDisk (point, direction,radius)
   end if
   $\eta$ = b.length/(b.minRad+b.maxRad)
   if $\eta > 4$ then
      nbCut $= \eta/4$
      maxLength = b.length/nbCut
      for $j = 0$ to nbCut do
         point(j), direction(j), radius(j) $\leftarrow$ maxLength, b.lines
         cutByDisk(point(j), direction(j), radius(j))
      end for
   end if
end for
```

The last defined operator is a vessel wall model generation. In many cases, the entire wall surface is not available from image segmentation and needs to be reconstructed from the lumen wall using for example a quite realistic radius-dependent wall-thickness δ^W such as a percentage of the local radius of the tubular geometry. The geometry of the wall surface can then be obtained by extruding the lumen surface mesh in the outward direction with a radius-dependent wall thickness. The extrusion is performed in Gmsh using an advancing layer method [10, 6, 15] with a given number of layers. The method starts from a surface mesh on which a boundary layer must be grown. From each surface node $\mathbf{x} \in \mathcal{X}$ a direction is picked for placing the nodes of the boundary layer mesh. The direction is computed using an estimate to the surface normal at the node using Gouraud shading and the extrusion thickness δ is computed as a percent α of the vessel radius:

$$\delta^W(\mathbf{x}) = \alpha r\left(\mathbf{x}_c\left(\mathbf{x}\right)\right), \tag{2}$$

where $\mathbf{x}_c \in \mathcal{X}_c$ is the mesh vertex on the centerlines that is closest to $\mathbf{x}$. The nodes are connected to form layers of hexahedra or prisms that can be subsequently subdivided into tetrahedra. An example of vessel wall generation is shown in Figure 5.

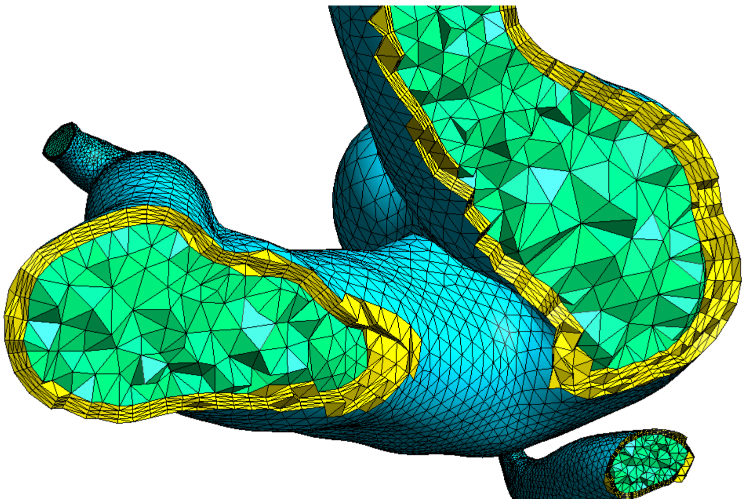


Figure 5. Vessel wall model generation for the geometry of cerebral aneurysm: cut-away view of the tetrahedral volume mesh of both the lumen (in green) and the vessel wall (in yellow). The vessel wall is built using a radius-dependent wall thickness $\delta^W(\mathbf{x})$ with $\alpha = 0.2$ and 4 layers.

2.4. *Defining mesh metrics for anisotropic meshes*

In order to create a 3D anisotropic computational mesh from an input geometry, we need to define mesh size fields h in respectively the tangent h_t, the normal h_n and the binormal h_{bn} direction to the centerlines. The following mesh metric $\mathcal{M}(\mathbf{x})$ can then computed:

$$\mathcal{M}(\mathbf{x}) = \mathbf{R}^T \begin{pmatrix} h_t^{-2} & 0 & 0 \\ 0 & h_n^{-2} & 0 \\ 0 & 0 & h_{bn}^{-2} \end{pmatrix} \mathbf{R}, \tag{3}$$

where $\mathbf{R} = (\mathbf{e}_t, \mathbf{e}_n, \mathbf{e}_{bn})$ is the triad axis computed from the centerline field. This mesh metric can then be used by the different 1D, 2D and 3D meshing algorithms.

There are several criteria than can be applied to assign the different mesh sizes at each point $\mathbf{x}$:

- If one is willing to build a CFD boundary layer mesh of thickness δ inside the lumen inside the lumen, the normal mesh size can be computed as:

$$h_n(\mathbf{x}) = h_n^0 + (r-1)d(\mathbf{x}), \tag{4}$$

  where $d(\mathbf{x})$ is the distance to the tubular geometry $\mathcal{T}$ that is again computed efficiently using an ANN [21, 22], $r > 1$ is the normal growth of the boundary layer, i.e. the ratio between two successive element sizes in the normal direction and h_n^0 is the normal size at the wall. This normal size can be for example a function of the vessel radius at the wall: $h_n^0 = \alpha r(\mathbf{x}_c)$.

- Some control on the geometrical accuracy of the mesh should be ensured. For that, mesh sizes h_t and h_{bn} should depend on the curvature of the surface in directions $\mathbf{e}_t$ and $\mathbf{e}_{bn}$. Note here that $\mathbf{e}_t$ and $\mathbf{e}_{bn}$ may not be the principal directions of curvature of the surface. Those two directions are chosen anyway for building the metric for reasons of stability and consistency. Typically, we choose for the mesh points on the wall:

$$h_t^0 = \frac{2\pi\rho_t^0}{N_p} \quad \text{and} \quad h_{bt}^0 = \frac{2\pi\rho_{bn}^0}{N_p} \tag{5}$$

with N_p a control parameter that is the number of mesh points per circumference and ρ_t and ρ_{bn} the radii of curvature of the surface in both directions $\mathbf{e}_t$ and $\mathbf{e}_{bn}$. Here, curvatures are computed on the discrete tubular surface using the method proposed by S. Rusinkiewicz [28]. When we leave the surface, the tangent mesh size is allowed to grow as follows:

$$h_t(\mathbf{x}) = h_t^0(\rho_0 + \sigma d(\mathbf{x}))/\rho^0, \tag{6}$$

where σ is the sign of the curvature.

A more subtle control on the tangent mesh size fields has however to be done in order to avoid large variations of mesh sizes. Let us consider a 2D geometrical domain Ω of boundary Γ with radius of curvature ρ^0 (see Figure 6). A boundary layer mesh of thickness δ is constructed with a prescribed "normal to the wall" mesh size h_n (Eq. (4)). The boundary layer thickness δ is sufficiently thin to $\rho \approx \rho_0$. We raise here the following question: is it possible to choose freely the "tangent to the wall" mesh size h_t? In most mesh generation procedures, a smoothing smoothing step is applied to the metric field in order to avoid large variation of mesh sizes. A smoothing ratio $\beta \geq 1$ is defined as the maximum ratio of two adjacent edge lengths. Let us demonstrate in 2D that mesh size $h_t(\mathbf{x})$ is indeed constrained by β, h_n and ρ^0 and should therefore not be given using (5) and (6). At point $\mathbf{x}_1$,

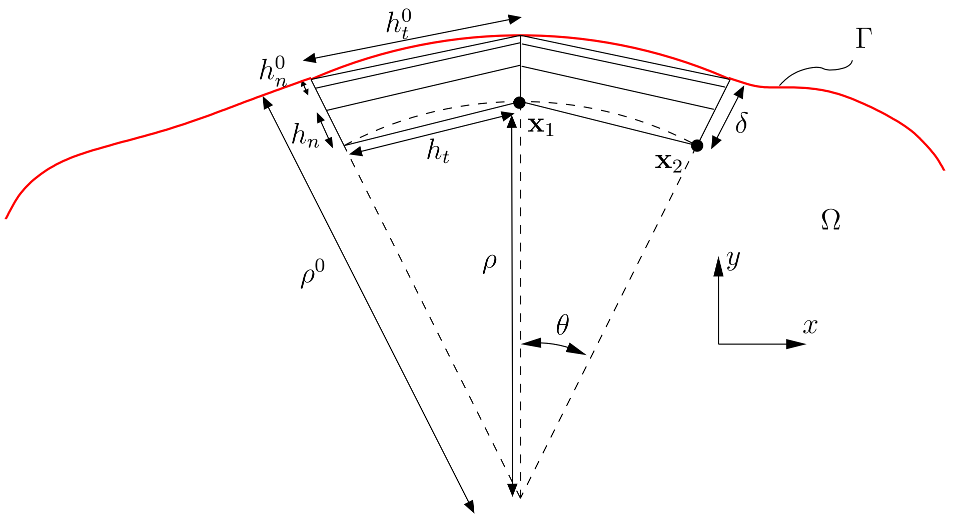


Figure 6. Defining the mesh sizes h_t and h_n on a 2D geometry of boundary Γ with radius of curvature ρ^0. The boundary layer thickness δ is sufficiently thin so that $\rho \approx \rho_0$ in the boundary layer.

the 2D metric field can be written as

$$\mathcal{M}(\mathbf{x}_1) = \begin{pmatrix} h_t^{-2} & 0 \\ 0 & h_n^{-2} \end{pmatrix}.$$

At point $\mathbf{x}_2$, the metric is rotated of an angle θ:

$$\begin{aligned}\mathcal{M}(\mathbf{x}_2) &= \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} h_t^{-2} & 0 \\ 0 & h_n^{-2} \end{pmatrix} \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \\ &= \frac{1}{h_n^2 h_t^2} \begin{pmatrix} h_n^2\cos^2\theta + h_t^2\sin^2\theta & (h_n^2 - h_t^2)\cos\theta\sin\theta \\ (h_n^2 - h_t^2)\cos\theta\sin\theta & h_t^2\cos^2\theta + h_n^2\sin^2\theta \end{pmatrix}. \end{aligned} \tag{7}$$

Assuming $\theta = h_t/\rho^0 \ll 1$, we have

$$\cos^2\theta \simeq 1 - \frac{h_t^2}{(\rho^0)^2} \quad \text{and} \quad \sin^2\theta \simeq \frac{h_t^2}{(\rho^0)^2}.$$

and the mesh size h_x in the x direction at point $\mathbf{x}_2$ is computed as

$$\frac{1}{h_x^2} = \frac{1}{h_n^2 h_t^2}(h_n^2\cos^2\theta + h_t^2\sin^2\theta) \simeq \frac{1}{h_t^2} + \frac{1}{(\rho^0)2}\left(\frac{h_t^2}{h_n^2} - 1\right).$$

It is now possible to correct mesh h_t in direction x at point $\mathbf{x}_1$ using the relation $h_t = \beta h_x$, i.e.,

$$\frac{\beta^2}{h_t^2} = \frac{1}{h_t^2} + \frac{1}{(\rho^0)^2}\left(\frac{h_t^2}{h_n^2} - 1\right). \tag{8}$$

Providing that $\frac{1}{h_t^2}$ is positive, (8) has the unique solution

$$\frac{1}{h_t^2} = \frac{1}{2(\rho^0)^2(\beta^2-1)}\left(\sqrt{1 + \frac{4(\rho^0)^2(\beta^2-1)}{h_n^2}} - 1\right). \tag{9}$$

Mesh size h_t is then an upper bound of the mesh size in the tangent direction: for example, if ρ is very large, then the tangent mesh size can be chosen arbitrarily. If $\beta = 1$, then equation (8) has the solution $h_t = h_n$ which means that no anisotropy is possible in a uniform mesh.

The final mesh metric $\mathcal{M}(\mathbf{x})$ is calculated as (3) with h_t, h_{bn} and h_n being computed as the minimum mesh size with respect to all criteria.

Using the anisotropic mesh metric (3), we can compute the length of a path given by a mesh edge $\mathbf{e} = \overline{\mathbf{x}_i\mathbf{x}_j}$ using the straight line parametrization $\mathbf{x}(t) = \mathbf{x}_i + t\,\overline{\mathbf{x}_i\mathbf{x}_j}$, $t \in [0,1]$:

$$l_{\mathcal{M}} = \int_0^1 \|\gamma(t)\| dt = \int_0^1 \sqrt{\mathbf{e}^T\mathcal{M}(\mathbf{x}(t))\mathbf{e}}\, dt \tag{10}$$

For the surface remeshing of the tubular geometry, we however do not remesh in the 3D space but remesh in the parametric space thanks to the conformal parametrization (see Figure 3). This enables us to use any available 2D meshers, and in particular the bi-dimensional anisotropic mesh generator BAMG [14] developed by F. Hecht and integrated within Gmsh. The length of the edge on the mesh patch S_j can then be computed from the length in the parametric space $\mathbf{e}'$ as follows: $\mathbf{e} = \mathbf{x}_{,\mathbf{u}}\mathbf{e}'$ so that the length of the edge with respect to the mesh metric can be computed as follows:

$$l_{\mathcal{M}_\mathbf{x}} = \int_0^1 \sqrt{\mathbf{e}'^T \underbrace{\left(\mathbf{x}_{,\mathbf{u}}^T\mathcal{M}(\mathbf{x})\mathbf{x}_{,\mathbf{u}}\right)}_{\mathcal{M}_{\mathbf{x}'}} \mathbf{e}'\, dt}, \tag{11}$$

where $\mathcal{M}(\mathbf{x})'$ is the mesh metric that needs to be given to the anisotropic bi-dimensional mesh generators. For our piecewise linear conformal mapping $\mathbf{u}(\mathbf{x})$, the derivatives $\mathbf{x}_{,\mathbf{u}}$ can be computed quite easily (see [25] for more details).

2.5. *Quadrangular and hexahedral mesh generation*

The first step for our hexahedral mesh generation algorithm is the generation of a quadrangular mesh of the initial geometry. The presented algorithm of quadrangular mesh generation is an original approach that combines a structured approach and an indirect unstructured approach. The mesh patches that are created by our centerline-based cut operator can be classified into two types of patches: annular patches and bifurcation patches. The annular mesh patches are then remeshed using the structured approach, while the bifurcation patches (tri-bifurcations or higher order bifurcations) are meshed with the indirect approach.

The structured meshes are build using an elliptic mesh generation method [32]. The elliptic grid generator relies on the conformal mapping $\mathbf{u}(\mathbf{x})$ (Eq. (1)) of the physical domain onto the parametric domain and on a second mapping that maps $\xi(\mathbf{u})$ the parametric domain onto a periodic rectangular computational domain (see top and mid Figs. in 7). At first, an initial algebraic structured grid is generated in the computational domain and mapped in the parametric domain. The elliptic smoother moves then the points in the parametric space by solving iteratively elliptic PDE's (see [32] for more details). Figure 7b) shows the final points (red points) in the parametric space as well as the resulting mapped mesh in the physical space.

The indirect way of producing quad meshes comprises three steps: (i) The triangulation is tailored with the aim of producing right triangles in the domain, using the infinity norm to compute distances in the meshing process [26], (ii) the triangles are recombined into quads using the well known Blossom algorithm of graph theory that computes the minimum cost perfect matching in a graph in polynomial time [27], (iii) local and global mesh cleanup operations are performed, such as Guy Bunin's one-defect remeshing operation [5] to reduce irregular nodes in the mesh.

Once the quadrangular surface mesh has been generated, we generate an extruded boundary layer mesh as well as an unstructured tetrahedral mesh of the lumen with pyramids as transition elements [23]. We are currently also working on an optimal placement of the mesh vertices in the lumen for the recombination of the tetrahedra into hexahedra [35, 34, 20], which will allow to also generate hexahedral dominant meshes..

3. EXAMPLES

In this section, we have run our new automatic meshing algorithm on different medical tubular geometries including cerebral aneurysm, carotid arteries and airways. Different types of computational meshes are generated: isotropic tetrahedral meshes, anisotropic tetrahedral meshes or mixed meshes. Some of the meshes include a vessel wall.

Timings as well as mesh qualities are given for the different meshes. The computations are performed on a MacBook Pro 2.66GHz 4GB RAM Intel Core i7.

3.1. *Isotropic tetrahedral computational meshes*

Let us first look at isotropic tetrahedral computational meshes. We define two quality criteria: one quality γ_T for the triangles T of the surface mesh [13] and one quality γ_τ for the tetrahedra τ of the volume [13, 11].

$$\gamma_T = \alpha \frac{\rho_{in}}{\rho_{out}} \tag{12}$$

$$\gamma_\tau = \frac{2\sqrt{6}\,\rho}{h}. \tag{13}$$

Here α is a constant, ρ_{in} is the radius of the inner circle/sphere of the mesh element, ρ_{out} is the radius of the circumscribed circle/sphere of the element and h is the length of the longest edge of the tetrahedron. With those definitions, the regular triangle/tetrahedron has $\gamma = 1$ and degenerated (zero surface/volume) mesh element has $\gamma = 0$.

Table I shows the mean triangle and tetrahedra quality ($\bar{\gamma}_T$, $\bar{\gamma}_\tau$), the minimum tetrahedral quality ($\gamma_\tau^{\min}$), the number of mesh elements # and the timings (in s) for the generation of different isotropic

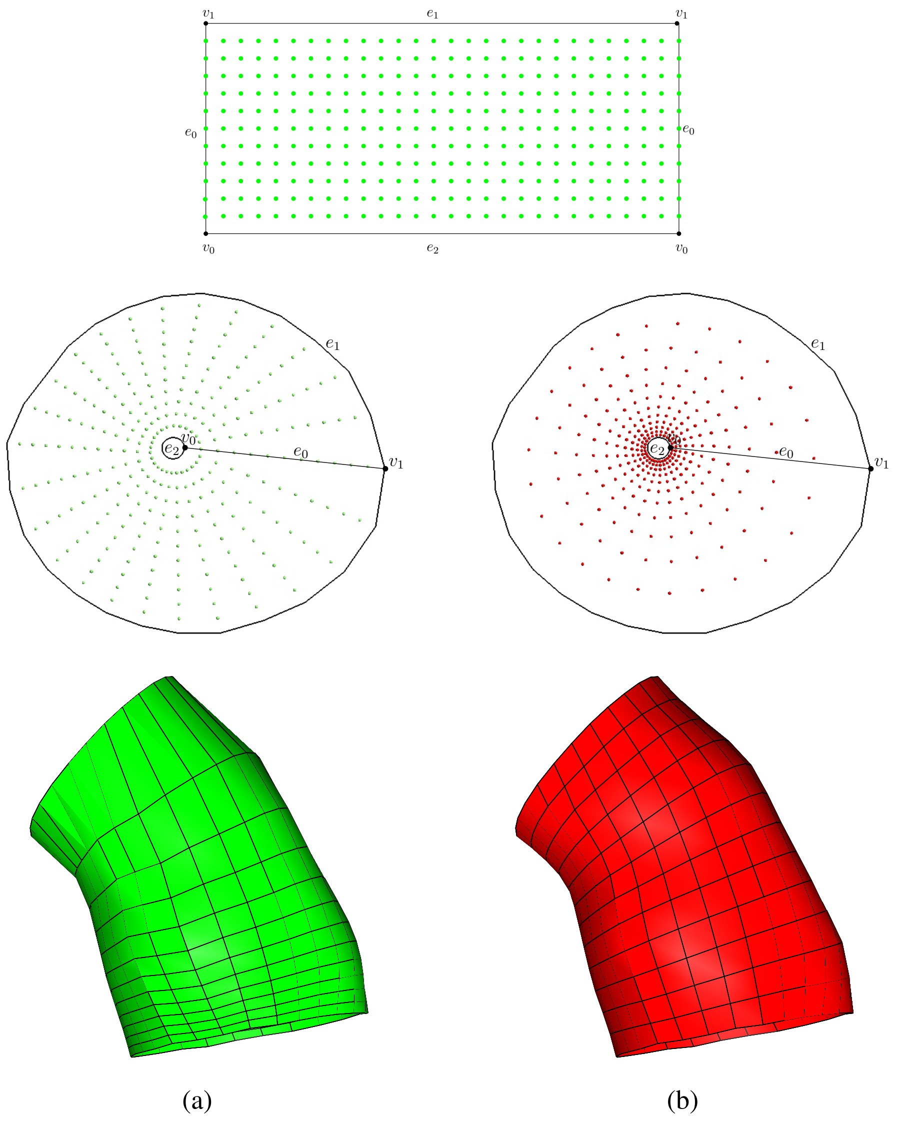


Figure 7. Elliptic grid generation of a tubular mesh patch (the mesh patch corresponds to the blue mesh patch of the aorta presented in Figure 2). a) Initial structured grid in computational domain (top), in parametric domain (center) and in physical domain (bottom), b) final mesh points and quad mesh obtained with the elliptic smoother in both parametric and physical domain.

tetrahedral meshes starting from tubular geometries (STL triangulations). The tubular geometries are the aorta shown in Figure 2, the cerebral aneurysm shown in Figure 4 and the airways in Figure 1. The meshes are obtained using the 2D Frontal-Delaunay[§] algorithm [24] and the 3D Delaunay algorithm. The timing for the computation of the centerline field (i.e the timing for the computation of the centerline-based descriptors and operators described in section 2) is given. For isotropic tetrahedral meshes, the only computed descriptor is the local radius and the two computed operators are the cut-operator and the close-volume operator. The timings are also given for the 2D and 3D mesh generation (including mesh optimization).

[§]It was indeed showed in [19] that optimal isotropic tetrahedral meshes (i.e with the highest mesh quality) can be obtained by combining a conformal parametrization with a Frontal mesher.

Geometry	STL		Surface mesh		Volume mesh			Time (s)	Time (s)	Time (s)
	$\#$	$\bar{\gamma}_T$	$\#$	$\bar{\gamma}_T$	$\#$	$\gamma_{\mathcal{T}}^{\min}$	$\bar{\gamma}_{\mathcal{T}}$	$\mathcal{C}$ field	2D mesh	3D mesh
Aorta	$4\ 10^3$	0.73	$12\ 10^3$	0.97	$58\ 10^3$	0.21	0.69	0.08	0.78	2.26
Aneurysm	$38\ 10^3$	0.94	$27\ 10^3$	0.97	$104\ 10^3$	0.19	0.65	3.30	3.60	4.20
Airways	$493\ 10^3$	0.87	$168\ 10^3$	0.93	$587\ 10^3$	0.06	0.68	410.10	25.90	35.11

Table I. Mean and minimum quality ($\bar{\gamma}_T$, $\bar{\gamma}_{\mathcal{T}}$, and $\gamma_{\mathcal{T}}^{\min}$), number of mesh elements $\#$ and timings (in s) for the generation of isotropic tetrahedral meshes starting from tubular geometries.

Figure 8 shows the generated isotropic tetrahedral mesh for the lung based on the centerline field. As can be seen, the mesh size is a function of the vessel radius, reducing therefore considerably the total number of mesh elements compared with a uniform tetrahedral mesh.

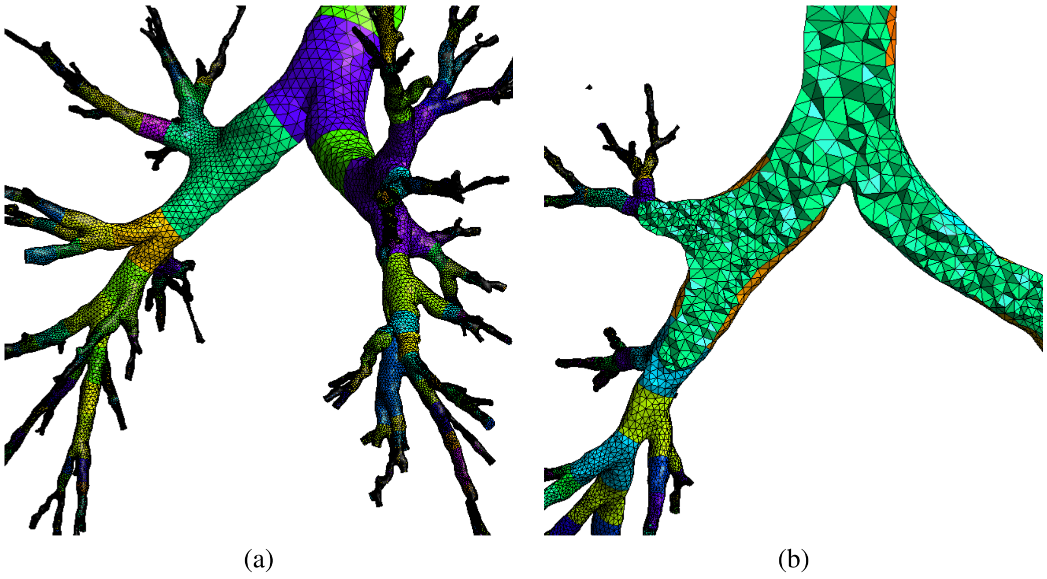


Figure 8. Isotropic tetrahedral mesh of the airways created using the centerline operators. The colors correspond to the different mesh patches that have been created by the cut operator.

3.2. *Mixed hexahedral/tetrahedral/pyramidal computational meshes*

We first define a quality measure for quadrilateral elements. Consider a quadrilateral element q and its the four internal angles $\alpha_k, \quad k = 1, 2, 3, 4$. We define the quality η_q as:

$$\eta_q = \max\left(1 - \frac{2}{\pi}\max_k\left(\left|\frac{\pi}{2} - \alpha_k\right|\right), 0\right). \tag{14}$$

This quality measure is $\eta = 1$ if the element is a perfect quadrilateral and is $\eta = 0$ if one of those angles is either ≤ 0 or $\geq \pi$. For the hexahedral mesh elements, we define the equi-skew angle mesh quality ζ_H as a normalized measure of skewness ranging from $\zeta_H = 1$ (best) to $\zeta_H = 0$ (worst) that depends on the angle formed between the faces's edges of each cell in the mesh ($\zeta_H = 1$ corresponds to a perfectly equiangular hexahedra) [7]:

$$\zeta_H = 1 - \max\left[\frac{\theta_{\max} - 90}{90}, \frac{90 - \theta_{\min}}{90}\right], \tag{15}$$

where $\theta_{\max}$ and $\theta_{\min}$ are the largest and smallest angle in the hexahedra.

Table II shows the minimum and quality (η_q^{min}, $\bar{\eta}_q$, ζ_H^{min}, and $\bar{\zeta}_H$), the number of mesh elements # and the timings (in s) for the generation of the mixed hexahedral/tetrahedral meshes starting from different tubular geometries. The parameters for the extruded hexahedral mesh are 4 layers of total thickness $\delta_W = 0.2r(\mathbf{x}_c)$. For the volume elements, we only show the statistics for the hexahedra of the mixed meshes. The timings presented in Table II are quite fast compared to timings for the generation of block-structured hexahedral meshes (timings of several hours in order to generate about 5000 tetrahedra in [7]).

Geometry	Quad Surface mesh			Hex Volume mesh			Time (s) 2D mesh	Time (s) 3D hex mesh	Time (s) 3D mixed mesh
	#	$\bar{\eta}_q^{min}$	$\bar{\eta}_q$	#	ζ_H^{min}	$\bar{\zeta}_H$			
Aorta	$3\ 10^3$	0.21	0.85	$11\ 10^3$	0.28	0.83	7.9	0.6	2.3
Carotid	$5\ 10^3$	0.22	0.91	$16\ 10^3$	0.33	0.86	4.3	0.3	2.5
Cerebral	$27\ 10^3$	0.19	0.89	$99\ 10^3$	0.28	0.85	29.2	2.5	20.0

Table II. Minimum and mean quality (η_q^{min}, $\bar{\eta}_q$, $\bar{\zeta}_H^{min}$, and ζ_H), number of mesh elements # and timings (in s) for the generation of mixed hexahedral/tetrahedral meshes starting from tubular geometries.

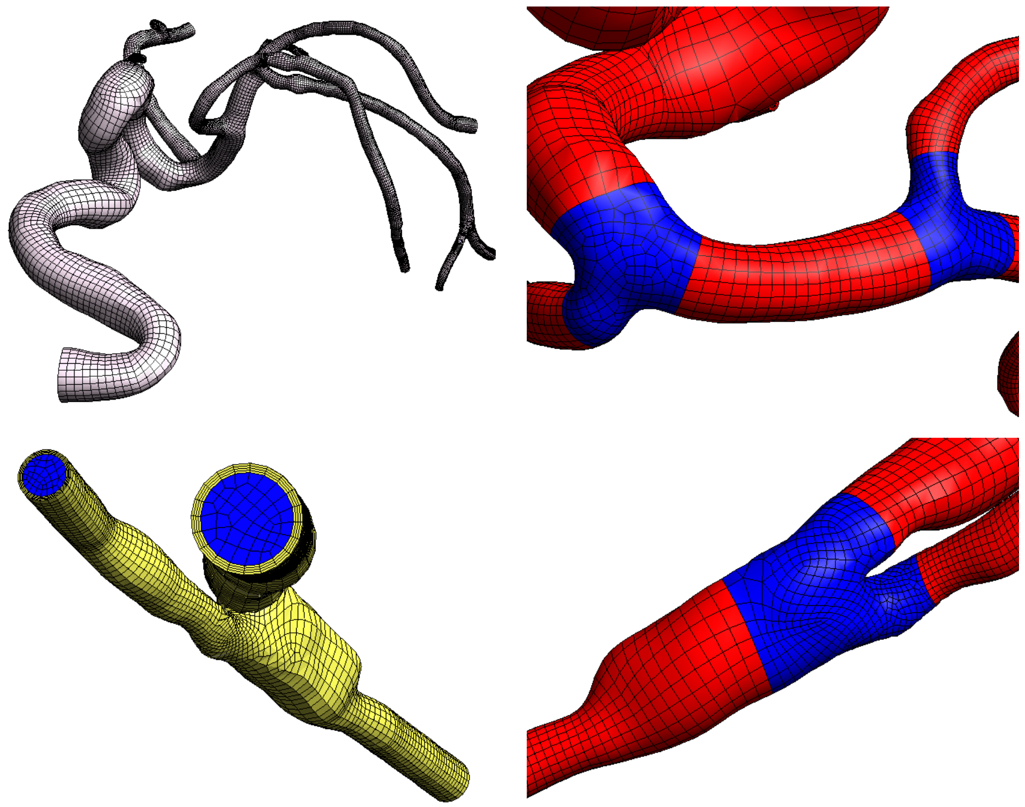


Figure 9. Quadrangular surface mesh of the cerebral arterial tree and hexahedral mesh of the arterial wall (in yellow) of the carotid bifurcation. The blue mesh patches (tri-bifurcations or higher-order bifurcations) on the right figures correspond to the quad meshes obtained using the indirect approach, while the red patches correspond to the direct quad approach.

Mixed three dimensional meshes can be generated that are composed of a mixture of tetrahedra, hexahedra and pyramids. Figure 10 show an image of a hybrid 3D mesh of the carotid.

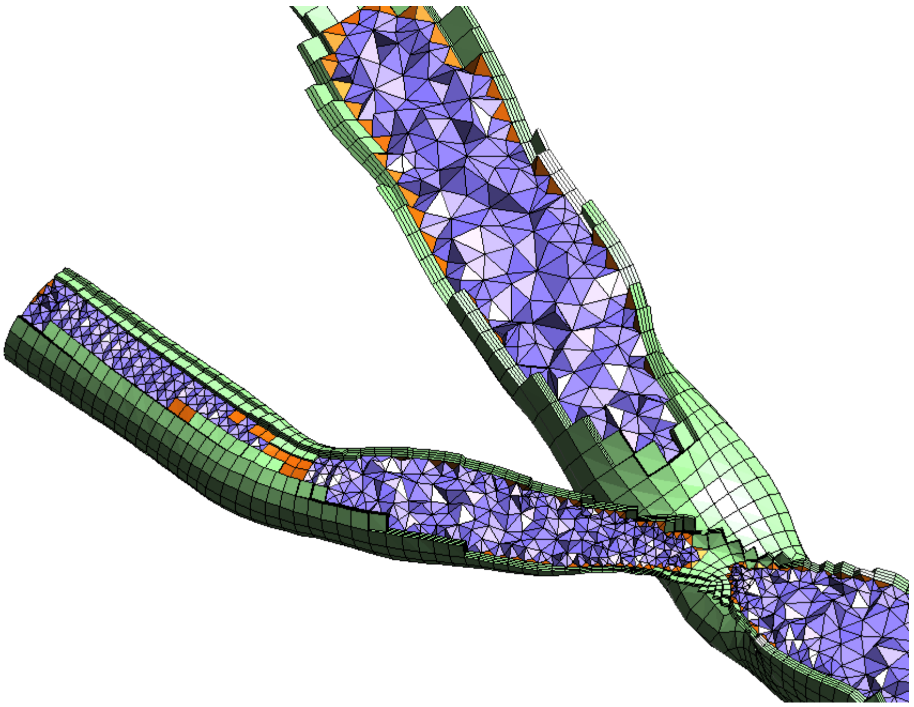


Figure 10. Mixed mesh of the carotid geometry. Hexahedra are in green, tetrahedra are in blue and pyramids are in orange. The mesh is composed of 58,046 tetrahedra, 16,500 hexahedra and 4,236 pyramids.

3.3. Anisotropic tetrahedral computational meshes

The anisotropic meshes are build using first a 2D and then a 3D anisotropic mesh procedure. From the initial triangulation, we first generate a new isotropic triangular surface mesh using a conformal mapping of the surface mesh (see Eq. (1)). Then, given the mapped surface mesh and the discrete metric tensor (11), the anisotropic surface mesh is generated in the parametric space by the Bidimensional Anisotropic Mesh Generator Bamg [14] integrated within Gmsh. Then, from this anisotropic surface mesh, an initial volume Delaunay-tetrahedral mesh is created using Tetgen [31] without adding any new points on the surface mesh during the tetrahedralisation. The initial tetrahedral mesh is then given as input to mmg3d [8, 9], a 3D Delaunay-based anisotropic mesh adaptation library. This library, also integrated in Gmsh, produces quasi-uniform meshes with respect to a metric tensor field using local mesh modifications and a Delaunay kernel to adapt the initial mesh.

Given the metric tensor $\mathcal{M}$ (Eq. (3)), we define the following criteria [9] that measures how well an anisotropic tetrahedron matches the metric specification, both in terms of size (edge lengths) and of shape (aspect ratio) :

$$Q_\tau = \frac{\left(\sum_{i=0}^{6} \mathbf{e}_i^T \bar{\mathcal{M}} \mathbf{e}_i\right)^3}{V_\tau \sqrt{det(\bar{\mathcal{M}})}}. \tag{16}$$

Here V_τ is the volume of the tetrahedron, $\bar{\mathcal{M}}$ is the average metric of the tetrahedron, and $\mathbf{e}_i$ are the edges of the tetrahedron. With this definition $Q_\tau \in [1, +\infty]$, with a high quality value indicating a degenerated tetrahedron. In numerical simulations, optimal meshes should have more than 90% of the tetrahedron with a quality measure better than 3, i.e $1 < Q_\tau < 3$ [9, 1].

In addition, we define an efficiency index τ that provides a single scalar value to evaluate how well the tetrahedral mesh complies with the metric requirements:

$$\tau = \exp\left(\frac{1}{n_e}\sum_{1<i<n_e} l_{\mathcal{M}}(\mathbf{e}_i)\right), \quad \begin{cases} l_{\mathcal{M}}(\mathbf{e}_i) = l_{\mathcal{M}}(\mathbf{e}_i) - 1 & \text{if } l_{\mathcal{M}}(\mathbf{e}_i) < 1 \\ l_{\mathcal{M}}(\mathbf{e}_i) = l_{\mathcal{M}}^{-1}(\mathbf{e}_i) - 1 & \text{otherwise} \end{cases} \tag{17}$$

where n_e denotes the total number of mesh edges and the edge length $l_{\mathcal{M}}(\mathbf{e}_i)$ is given by Eq. (10). Optimal meshes have an efficiency index τ close to the optimal value of one. In numerical simulations, a value of $\tau > 0.80$ is considered as an acceptable lower bound [9, 1].

Table III shows the quality of different anisotropic tetrahedral meshes together with meshing timings. We present the mean quality index $\bar{Q}_\tau$, the percent of elements that have a good quality measure $1 < Q_\tau < 3$, as well as the efficiency index τ. The geometries are STL triangulations of three different arteries trees: an abdominal aorta and a cerebral aneurysm described in table I as well as an arterial bypass¶. For the different examples, we have chosen a boundary layer thickness $\delta = 0.3r(\mathbf{x}_c)$, a mesh size normal to the wall $h_n^0 = \delta/20$, a boundary layer growth ratio $r = 1.3$ and a smoothing ratio $\beta = 1.5$.

Figure 11 shows the surface and a cut of the volume mesh of the anisotropic mesh of the geometry of the aneurysm presented in the left Figure 4 and Figure 12 shows the surface and a cut of the volume mesh of the anisotropic mesh of the geometry of the bypass.

Geometry	Volume mesh					Time (s)	
	#	$\bar{Q}_\tau$	$Q_\tau^{\min}$	$1 < Q_\tau < 3(\%)$	τ	2D mesh	3D mesh
Aorta	323 10^3	1.6	11	0.98	0.836	7.2	96
Aneurysm	477 10^3	1.5	17	0.96	0.821	17	145
Bypass	342 10^3	1.5	12	0.97	0.829	11	67

Table III. Mean and minimum quality ($\bar{Q}_\tau, Q_\tau^{min}$), efficiency index τ, number of mesh elements $\#$ and timings (in s) for the generation of anisotropic tetrahedral meshes starting from the tubular geometries.

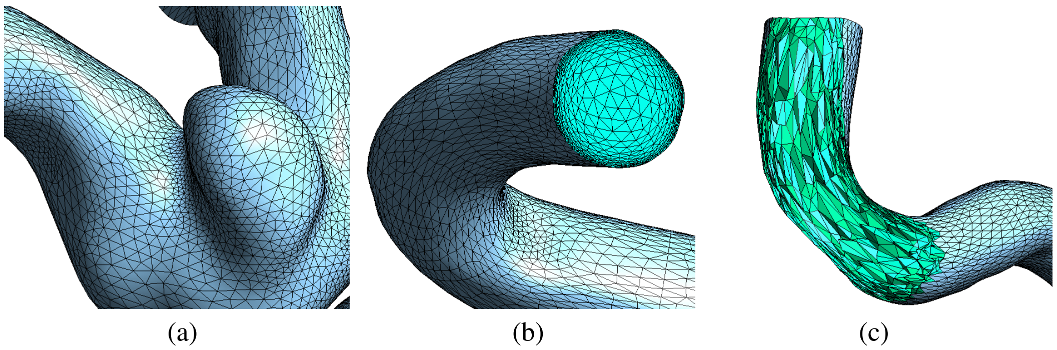


Figure 11. Anisotropic tetrahedral mesh of the aneurysm.

4. CONCLUSION

We have presented a new automatic meshing algorithm for the generation of computational meshes from a segmented tubular geometry. The proposed methodology is based on different centerline-based descriptors and operators.

¶This geometry has been downloaded from the simbios web site https://simtk.org/frs/download.php?file_id=183.

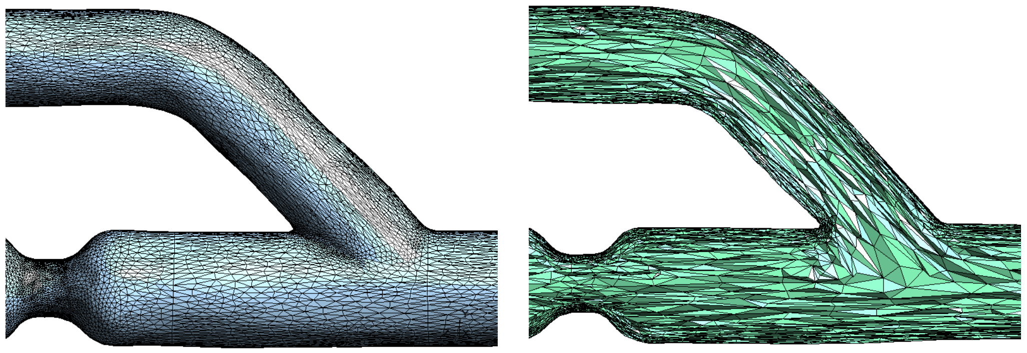


Figure 12. Magnified views of the anisotropic tetrahedral mesh of the bypass.

Different type of computational meshes can be generated with this method: isotropic tetrahedral meshes, anisotropic tetrahedral meshes or mixed hexahedral/tetrahedral meshes as well as boundary layer meshes for the lumen wall. The mesh size field is function of the centerline-based descriptor.

Besides the original centerline-based meshing algorithms, two important contributions are included in this paper:

- A subtle control on the tangent and normal mesh sizes for the generation of anisotropic CFD boundary layer meshes.
- A flexible and fast approach for hexahedral mesh generation. The proposed approach relies on the generation of a quadrangular surface mesh that combines a structured elliptic grid generator together with an indirect quadrangular meshing approach. Structured hexahedral meshes are then created for the vessel wall and connected to the tetrahedra of the lumen with pyramids.

We are currently working on several hexahedral meshing algorithms that will enable us to generate also dominant hexahedral meshes in a close future using the presented method for hexahedral mesh generation.

The presented automatic meshing algorithm is implemented in the open-source mesh generator Gmsh [13] and examples can be found on the Gmsh wiki[‖].

REFERENCES

1. F. Alauzet, L Xiabgrong, E. Seegyoung Seol, and M.S. Shephard. Parallel anisotropic 3d mesh adaptation by mesh modification. *Engineering with Computers*, 21:247–258, 2006.
2. L. Antiga, B. Ene-Iordache, and A. Remuzzi. Computational geometry for patient-specific reconstruction and meshing of blood vessels from mr and ct angiography. *IEEE TRANSACTIONS ON MEDICAL IMAGING*, 22(5), 2003.
3. Luca Antiga, Bogdan Ene-iordache, and Andrea Remuzzi. Centerline computation and geometric analysis of branching tubular surfaces with application to blood vessel modeling, 2003.
4. L. Botti, M. Piccinelli, B. Ene-Iordache, A. Remuzzi, and L. Antiga. An adaptive mesh refinement solver for large-scale simulation of biological flows. *Communications in Numerical Methods in Engineering*, 2009.
5. G. Bunin. Non-local topological clean-up. In Springer-Verlag, editor, *15th International Meshing Roundtable*, pages 3–20, Sep 2006.
6. S. Connell and ME. Braaten. Semistructured mesh generation for three-dimensional navier-stokes calculations. *AIAA Journal*, 33(6):1017–1024, 1995.
7. G De Santis, M De Beule, P Segers, P Verdonck, and B Verhegghe. Patient-specific computational haemodynamics: generation of structured and conformal hexahedral meshes from triangulated surfaces of vascular bifurcations. *Comput Methods Biomech Biomed Engin*, 14(9):797–802, Sep 2011.
8. C. Dobrzynski. Mmg3d 4.0: Anisotropic tetrahedral remesher/moving mesh generation, 2012. `http://www.math.u-bordeaux1.fr/~cdobrzyn/logiciels/mmg3d.php`.

[‖]Gmsh's wiki: `https://geuz.org/trac/gmsh` (username: gmsh, password: gmsh).

9. C. Dobrzynski and P.J. Frey. Anisotropic delaunay mesh adaptation for unsteady simulations. In *17th international Meshing Roundtable*, Springer, pages 177–194, Heidelberg, 2008.
10. Rao V. Garimella and Mark S. Shephard. Boundary layer mesh generation for viscous flow simulations. *International Journal for Numerical Methods in Engineering*, 49(1):193–218, 2000.
11. P.-L. George and P. Frey. *Mesh Generation*. Hermes, 2000.
12. C. Geuzaine and J.-F. Remacle. Gmsh: a finite element mesh generator with built-in pre- and post-processing facilities, 1996. http://www.geuz.org/gmsh/.
13. C. Geuzaine and J.-F. Remacle. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009.
14. F. Hecht. Bamg: Bidimensional anisotropic mesh generator, 2006. http://www.freefem.org/ff++.
15. Y. Kallinderis, A. Khawaja, and H. Mcmorris. Hybrid prismatic/tetrahedral grid generation for complex geometries. *AIAA Paper*, 34:93–0669, 1996.
16. P. Worth Longest and Samir Vinchurkar. Effects of mesh style and grid convergence on particle deposition in bifurcating airway models with comparisons to experimental data. *Medical Engineering & Physics*, 29:350–366, 2007.
17. E. Marchandise, G. Compère, M. Willemet, G. Bricteux, C. Geuzaine, and J.-F. Remacle. Quality meshing based on stl triangulations for biomedical simulations. *International Journal for Numerical Methods in Biomedical Engineering*, 83:876–889, 2010.
18. E. Marchandise, P. Crosetto, C. Geuzaine, J.-F. Remacle, and E. Sauvage. *Quality open source mesh generation for cardiovascular flow simulations*, volume accepted of *Springer Series on Modeling, Simulation and Applications*, chapter of modelling physiological flows. Springer-Verlag, Berlin Heidelberg, 2011.
19. E. Marchandise, J-F. Remacle, and C. Geuzaine. Optimal parametrizations for surface remeshing. *Engineering with Computers*, under review, 2012.
20. S. Meshkat and D. Talmor. Generating a mixed mesh of hexahedra, pentahedra and tetrahedra from an underlying tetrahedral mesh. *International Journal for Numerical Methods in Engineering*, 49(17-30), 2000.
21. D.M. Mount, N.S. Arya, R. Netanyahu, and Silverman. An optimal algorithm for Approximate Nearest Neighbor searching. *Journal of the ACM*, 45, 1998.
22. D.M. Mount and S. Arya. ANN library. http://www.cs.umd.edu/ mount/ANN/.
23. S.J. Owen, S.A. Cannan, and S. Saigal. Pyramid elements for maintaining tetrahedra to hexahedra conformability. In AMD, editor, *Trends in Unstructured Mesh Generation*, volume 220, pages 123–129. ASME, 1997.
24. S. Rebay. Efficient unstructured mesh generation by means of delaunay triangulation and bowyer-watson algorithm. *Journal of Computational Physics*, 106:25–138, 1993.
25. J.-F. Remacle, C. Geuzaine, G. Compère, and E. Marchandise. High quality surface meshing using harmonic maps. *International Journal for Numerical Methods in Engineering*, 83:403–425, 2010.
26. J-F. Remacle, F. Henrotte, T. Carrier-Baudouin, E. Bechet, E. Marchandise, C. Geuzaine, and T. Mouton. A frontal delaunay quad mesh generator using the l_∞ norm. *International Journal for Numerical Methods in Engineering*, 2011.
27. J.-F. Remacle, J. Lambrechts, B. Seny, E. Marchandise, A. Johnen, and C. Geuzaine. Blossom-quad: a non-uniform quadrilateral mesh generator using a minimum cost perfect matching algorithm. *International Journal for Numerical Methods in Engineering*, 2011. submitted.
28. S. Rusinkiewicz. Estimating curvatures and their derivatives on triangle meshes. In *Symposium on 3D Data Processing, Visualization, and Transmission*, September 2004.
29. O. Sahni, K.E. Jansen, and M.S. Shephard. Automated adaptive cardiovascular flow simulations. *Engineering with Computers*, 25(1):25–36, 2009.
30. O. Sahni, J. Mueller, K.E. Jansen, M.S. Shephard, and C.A. Taylor. Efficient anisotropic adaptive discretization of cardiovascular system. *Comp. Meth. Appl. Mech. Eng.*, 195:5634–5655, 2006.
31. H. Si. Tetgen a quality tetrahedral mesh generator and three-dimensional delaunay triangulator, 2004.
32. J. F. Thompson, B. K. Soni, and N. P. Weatherill, editors. *Handbook of Grid Generation*. CRC Press, 1998.
33. Samir Vinchurkar and P. Worth Longest. Evaluation of hexahedral, prismatic and hybrid mesh styles for simulating respiratory aerosol dynamics. *Computers & Fluids*, 37:317–331, 2008.
34. S. Yamakawa and K. Shimida. Fully-automated hex-dominant mesh generation with directionality control via packing rectangular solid cells. *International Journal for Numerical Methods in Engineering*, 57:2099–2129, 2003.
35. S. Yamakawa and K. Shimida. Increasing the number and volume of hexahedral and prism elements in a hex-dominant mesh by topological transformations. In *12th International Meshing Roundtable*, 2003.
36. Y. Zhang, Y. Bazilevs, S. Goswami, C.L. Bajaj, and T.J.R. Hughes. Patient-specific vascular nurbs modeling for isogeometric analysis of blood flow. *Comput Methods Appl Mech Eng*, 196(2943-2959), 2007.

ACKNOWLEDGEMENTS

This work has been partially supported by the Belgian Walloon Region under WIST grants ONELAB 1017086 and DOMHEX 1017074.

The authors like to thank Luca Antiga for his support with VMTK, for his constructive comments and the geometrical model of the cerebral arteries. The authors also like to thank the people from FluidDA n.v. for providing the geometrical model of human airways.