Published in final edited form as:

Biomech Model Mechanobiol. 2010 April ; 9(2): 187–201. doi:10.1007/s10237-009-0170-5.

Adaptive Generation of Multimaterial Grids from Imaging Data for Biomedical Lagrangian Fluid-Structure Simulations

J.P. Carson and **A.P. Kuprat**

Biological Monitoring & Modeling, Pacific Northwest National Laboratory, Richland, WA.
james.carson@pnl.gov

X. Jiao and **V. Dyedov**

Department of Applied Mathematics & Statistics, Stony Brook University, Stony Brook, NY.
jiao@ams.sunysb.edu, vladimir@ams.sunysb.edu

F. del Pin

G.A. Johnson

Center for In Vivo Microscopy, Duke Medical Center gjohnson@duke.edu

J.M. Guccione and **M.B. Ratcliffe**

UCSF Dept. of Surgery & San Francisco VA Medical Center, San Francisco, CA.
mark.ratcliffe@va.gov, julius.guccione@va.gov

D.R. Einstein

Biological Monitoring & Modeling, Pacific Northwest National Laboratory, Richland, WA.
daniel.einstein@pnl.gov

Abstract

Spatial discretization of complex imaging-derived fluid-solid geometries, such as the cardiac environment, is a critical but often overlooked challenge in biomechanical computations. This is particularly true in problems with Lagrangian interfaces, where the fluid and solid phases share a common interface geometrically. For simplicity and better accuracy, it is also highly desirable for the two phases to have a matching surface mesh at the interface between them. We outline a method for solving this problem, and illustrate the approach with a 3D fluid-solid mesh of the mouse heart. An MRI perfusion-fixed dataset of a mouse heart with 50 μ m isotropic resolution was semi-automatically segmented using a customized multimaterial connected-threshold approach that divided the volume into non-overlapping regions of blood, tissue and background. Subsequently, a multimaterial marching cubes algorithm was applied to the segmented data to produce two detailed, compatible isosurfaces, one for blood and one for tissue. Both isosurfaces were simultaneously smoothed with a multimaterial smoothing algorithm that exactly conserves the volume for each phase. Using these two isosurfaces, we developed and applied novel automated meshing algorithms to generate anisotropic hybrid meshes on arbitrary biological geometries with the number of layers and the desired element anisotropy for each phase as the only input parameters. Since our meshes adapt to the local feature sizes and include boundary layer prisms, they are more efficient and accurate than non-adaptive, isotropic meshes, and the fluid-structure interaction computations will tend to have relative error equilibrated over the whole mesh.

Keywords

multimaterial grid generation; micro MRI; mouse heart; fluid-structure interaction; volume-conserving smoothing

1 Introduction

The heart and its immediate environs are intimately linked with the movement of blood. From a mechanical point of view, for example, the resistance to blood flow through the cardiac valves feeds back into the ventricular afterload and contractility, and disturbances in the flow can lead to thrombus formation and coronary obstruction. On a more biological scale, shear stress and flow conditions at the interface are intimately linked to cell life/death and indeed gene expression. For example, it is well known that wall shear stresses regulate coronary endothelium and that inappropriate wall shear stresses can lead to build up of atherosclerotic plaque, e.g. [5,6,9,10,13]. In a similar vein, but less well known, wall shear stresses have been shown to regulate the phenotypes of valve endothelium and by extension valve interstitial cells on the ventricular and aortic sides of the aortic valve [2–4]. In fact, the connection between heart function and the movement and properties of blood is so intimate that the normal morphogenesis of the heart is dependent on proper blood dynamics [3,14,29,35], to the point where with interrupted flow the embryonic heart may develop three chambers rather than four [14].

Because the interface is so important from a biological point of view, it is highly desirable to treat the interface in a Lagrangian framework, where fluid shear stresses can be resolved more accurately. However, this poses two fundamental challenges, the first of which is that the discretization of each phase must be consistent at the interface despite each phase being driven by different physical considerations. For example, the boundary layer is a consideration on the fluid side of the interface but not on the solid side. In addition, the solid side may require only a few layers as a minimum accommodation for bending or lamination, while the fluid phase requires a dedicated grid for resolving the boundary layer, and a compatible fluid interior grid. Due to the complexity of the geometry, all algorithms must be fully automatic with few meaningful user-selectable parameters. The second fundamental challenge is that the grid in both fluid and solid phases may evolve separately. A canonical example is a closing — and opening — heart valve around which the fluid grid must change topology over the course of a simulation, while the grid requirements of the valve tissue are largely unchanged. Here we focus on the first of these challenges. We have addressed, in part, the second of these challenges elsewhere [12]; though work in that area is still ongoing. In short, the strategy is to add an arbitrary Lagrangian-Eulerian (ALE) term to the the Particle Finite Element Method (PFEM) equations of motion (cite), while preserving a Lagrangian interface by efficiently updating the mesh to avoid overly distorted or squeezed elements. Thus, rather than stopping the simulation to perform a remesh, the mesh is managed dynamically throughout the course of the simulation by invoking the usual operations of nodal smoothing and augmenting them with nodal point insertion and deletion as well as edge bisection to allow for topological changes in the mesh.

In this manuscript, we describe the underpinnings of our methods, namely multimaterial segmentation, multimaterial isosurface extraction, generation of the gradient-limited feature size, construction of layered tissue tetrahedra, generation of a prismatic boundary layer, and generation of a boundary constrained Delaunay fluid interior mesh. We propose new methods for generating quality, hybrid, Lagrangian fluid-solid grids for complex biomedical geometries from imaging data. The Lagrangian fluid-solid grid has its own set of challenges insofar as a fluid-solid grid is multi-material, consisting of — at least — fluid and solid phases. In addition, for simplicity and for reducing numerical error over the course of a simulation, we require that

the fluid and solid boundaries match exactly along their common interface. These challenges are present at every stage, including image segmentation, surface extraction, cleanup and grid generation. Our mesh-generation methods are by design automatic, meaning that they require no user intervention. They are also efficient in the sense that, in most cases, a computable grid may be obtained in few hours, given a proper segmentation. We focus on a perfusion fixed dataset of a mouse heart, though the methods are applicable to other biomedical geometries. Some of these ideas have been developed elsewhere, and are thus described in more general terms. Novel ideas presented herein are described in detail with pseudo code.

2 Methods

We describe our methods starting from a brief description of the acquisition of MRI data. Following that we describe our approach to multimaterial segmentation of the heart tissue and blood. We then describe our approach to multimaterial isosurface extraction. Finally, we highlight our newest contributions to fluid-solid grid generation.

2.1 MRI Dataset

An adult male C57BL6 mouse commercially obtained (Jackson Laboratory, Bar Harbor, ME) was actively stained and imaged as previously described in publication [21]. A 1:20 mixture of gadopentetate dimeglumine (Magnevist; Berlex Laboratories, Wayne, NJ) and 10% buffered formalin (Fisher Scientific, Fairlawn, NJ) were systemically perfused into the mouse through the left carotid artery and the right jugular vein. A 3D MR image ($512 \times 512 \times 2048$) was acquired at $50\mu\text{m}$ isotropic resolution with a modified MR apparatus (Signa; GE Medical Systems, Milwaukee, WI). Image reconstruction (Figure 1) was performed on a dedicated 64-bit reconstruction engine (Silicon Graphics, Mountain View, CA). A 3D image subset ($249 \times 216 \times 240$) encompassing the heart and nearby tissue was cropped from the whole volume as the focus of this work.

2.2 Segmentation

Voxel assignment to different materials for the 3D MRI dataset was performed using a semi-automated segmentation procedure consisting of four steps. Software for performing segmentation routines was written in Python (www.python.org) with the graphical user interface (GUI) implemented using Visual Python (www.vpython.org). The overall segmentation process is articulated in four steps.

2.2.1 Step 1: Initial assignment of voxels—Every voxel was first automatically assigned to one of four material types: the perfused blood material (M_b), the heart tissue material (M_h), the other fluid and non-cardiovascular tissue material (M_o), and the disputed material (M_d) that is yet to be determined. These assignments were based on voxel intensity, where $M_o \leq t_1 < M_h \leq t_2 < M_d \leq t_3 < M_b$ (Algorithm 1). For this dataset, the values of the thresholds separating the material types were determined empirically to be: $t_1 = 50$, $t_2 = 110$, and $t_3 = 150$ (Figure 2).

2.2.2 Step 2: Reassign disputed voxels—This step begins by automatically reassigning fully encapsulated disputed material M_d voxel groups to the surrounding material when the enclosing material is all of the same material type (blood M_b or heart M_h). This is accomplished by a depth-first search algorithm that follows the 26-adjacency connectivity of disputed material points and tracks the range of material types bounding the 3D pocket of disputed material. In cases where the disputed material is bound by more than one other material type, those pixels remained disputed material.

Next, disputed material M_d voxels representing coronary vasculature were automatically reassigned to blood material M_b using a “burrowing” technique whereby certain M_d voxels attached to M_b either directly or indirectly through other M_d voxels would be reassigned to M_b . This was implemented as a reassignment of disputed material M_d where the majority of 26-adjacent neighborhood voxels are heart material M_h . Since M_d voxels fully encapsulated by M_h have previously already been reassigned to M_h , any M_d voxels mostly surrounded by M_h must also touch M_b . Applying this logic enables the blood region represented by M_b to automatically penetrate into the tissue along the avenues represented by M_d , just as the coronary vasculature also permeates through the heart tissue.

To complete the reassignment of M_d voxels to M_b or M_h voxels, the remaining M_d voxels were highlighted by the GUI and visually evaluated to maximize smoothness and maintain proper anatomical topology. For example, M_d voxels representing chordae were assigned to be M_h . Based on these criteria, each M_d voxel was manually reassigned to be an M_b or M_h voxel.

The application of Step 2 is represented for a sample image and illustrated in Figure 3. The procedure is detailed in Algorithm 2.

2.2.3 Step 3: Reassign non-cardiovascular voxels—The non-cardiovascular tissues adjacent to the heart and vessel walls often appear at a similar intensity in the MRI images, thus making them distinguishable only by incorporating anatomical knowledge. A semi-automated masking process was implemented to facilitate the isolation of heart and blood from the remainder of the image (Algorithm 3). First, a user-generated mask indicating the cardiovascular region for the first image in the volume was created. Then, for each subsequent image, a new mask was generated from its preceding image by applying a warp function to the preceding mask. This warp function represents a transformation that deforms an image to match its subsequent image in the volume. The warp function was calculated using multiresolution elastic image registration [23] on two adjacent images in the volume, and then the function was applied to the mask of the first image to generate the mask of the second image. After generating a new mask, a minimal amount of manual augmentation to the mask was performed to address new features emerging from one image to the next.

2.2.4 Step 4: Validation of topology—Several topological criteria were iteratively enforced to ensure the segmentation would generate an anatomically accurate multimaterial mesh (Algorithm 4). First, blood material was forbidden from contacting any material other than heart material. This was accomplished by replacing all M_o voxels 26-adjacency connected to M_b with M_h voxels. Then, all same-material connections were checked to be 6-adjacency, that is face-to-face connectivity, in order to ensure that a Marching Cubes algorithm [28] would produce valid surfaces. Same material voxels connected by edges and not indirectly face connected by a mutual voxel were visually evaluated and the material reassigned based on the appropriate anatomical topology for that location. After the edge check, vertex-connected voxels lacking a mutual same-material voxel via a face and edge connection were similarly evaluated and reassigned. Finally, the number of connected material components was set to be exactly four. These are the heart tissue, the blood of the left heart, the blood of the right heart, and the rest of the volume (M_o). To accomplish this, all the same material connected components were detected, and the four most representative of the final components were identified. Then each component that is not one of the final four was visually inspected and either reassigned in material to be absorbed by its surrounding component, or neighboring face-connected voxels were reassigned in material to connect the component to one of the final four. To ensure that each step of validation did not reintroduce topological errors, the entire process was repeated until execution of all the steps resulted in no changes made to the material assignments.

2.3 Mesh Generation

Multimaterial imaging-based geometries, such as that represented by the mouse heart shown in Figure 4, represent clear challenges for generating any kind of quality computational grid, let alone one for Lagrangian FSI, where we require exact node matching at the interface. Our approach to this challenge is based on the gradient-limited feature size (GLFS) [25]. The GLFS serves three functions in our grid generation algorithms: 1) a field for tangential adaptation of the surface grid [25], 2) a metric for creating layered tetrahedra [25], and 3) a speed function for construction of a prismatic boundary layer [11] by application of the generalized Huygens' principle [17]. Details of these methods are beyond the present scope and were presented elsewhere [25,11]. Herein, we review some of the key points of those references as background to the novel refinements and enhancements that we will describe shortly. These novel contributions can be summarized as:

1. a modification of the functional relationship between the GLFS and the first principal curvature
2. a further adjustment of the GLFS to resolve thin tube-like structures of the heart
3. the method by which points are cast along normal rays to create quality layered tetrahedra
4. an explicit time-dependent computation of the GLFS during interior surface evolution for the generation of prismatic boundary layer elements

2.3.1 Gradient-Limited Feature Size—Following segmentation and topology validation, surfaces were extracted from the segmented mask with a variant of the marching cubes algorithm [34]. Let S be the oriented closed triangulated surface or network of surfaces, obtained in Section 2.2.4. For any point $\mathbf{x} \in S$, we define the “raw feature size” or “local diameter” $F[\mathbf{x}]$ as the length of the line segment formed by first shooting a ray from $\mathbf{x}$ in the direction of $\hat{\mathbf{n}}[\mathbf{x}]$, the inward normal at $\mathbf{x}$, and then truncating the ray at its first intersection with S . That is

$$F[\mathbf{x}] \equiv \min \{ \lambda > 0 | \mathbf{x} + \lambda \hat{\mathbf{n}}[\mathbf{x}] \in S \} \quad (1)$$

Since S is closed, the ray proceeding from $\mathbf{x}$ in the surface normal direction $\hat{\mathbf{n}}[\mathbf{x}]$ will intersect S at least once, and hence $F[\mathbf{x}]$ is well-defined. Similarly, we also perform an “outward” interrogation of the geometry to compute another raw feature size field $F_{out}[\mathbf{x}]$ using $\hat{\mathbf{n}}_{out}[\mathbf{x}] = -\hat{\mathbf{n}}_{in}[\mathbf{x}]$. This outwards value is finite in some areas (e.g., at concave parts of S). This raw feature size is bounded ($F_{out}[\mathbf{x}]$ may be unbounded), but it is sensitive to abrupt changes in the geometry. To address this, we first impose user-specified lower and upper bound to the feature size, denoted by L_{min} and L_{max} respectively.

In addition, $F_{out}[\mathbf{x}]$ must be modulated by surface curvature such that, at highly curved areas of the surface, $F_{out}[\mathbf{x}]$ is allowed to violate L_{min} . Computation of the the maximum principle curvature κ_1 is performed using an accurate and stable algorithm for piecewise linear surfaces [20]. One of the novel contributions, presented in this manuscript is to recognize the influence of the first principal curvature κ_1 and L_{min} and L_{max} must be integrated so that its properties as a surface metric for adaptation (Section 2.3.2) and as a propagator (Sections 2.3.3 & 2.3.4) are both coordinated and able to preserve thread-like structures such as the mitral valve cordae and the coronary vasculature. The algorithm for this coordination is given in Algorithm 5, wherein γ is a free parameter typically equal to unity.

Once $F[\mathbf{x}]$ has been modified to reflect this additional geometric information, it is gradient limited to render it relatively insensitive to abrupt changes in S [25]. First, we initialize $f[\mathbf{x}]$

to $F[\mathbf{x}]$. Given a bound G on the surface gradient of $f[\mathbf{x}]$, the algorithm places the directed edges that violate the gradient limit into a maxpriority queue, ranked by the key

$$f[x_1] - (f[x_2] + G \|x_1 - x_2\|) \quad (2)$$

which measures how much the gradient violates the gradient limit for a directed edge $\overrightarrow{x_1x_2}$ on S . Let $\overrightarrow{x_i x_j}$ be the directed edge with the largest measure in the queue. We relax $f[\mathbf{x}]$ to satisfy the gradient limit, recompute the measures for the incident edges of x_i , and update the priority queue accordingly. The process continues until the queue is empty. For computational efficiency, ray-triangle intersections are queried within an axis-aligned bounding box (AABB) tree [22,32] that contains at its leaf nodes the bounding box for each triangle. This algorithm has a complexity of $(N \log N)$, where N is the number of triangles in S . The same gradient-limiting is applied to $F_{out}[\mathbf{x}]$ to produce $f_{out}[\mathbf{x}]$. Figure 5 shows both $f[\mathbf{x}]$ and $f_{out}[\mathbf{x}]$ for the closed surfaces described in Section 2.2.4.

Having defined $f[\mathbf{x}]$ and $f_{out}[\mathbf{x}]$ as gradient-limited feature sizes, we now clarify that in the following presentation we will refer to $f[\mathbf{x}]$ as GLFS. We reiterate that it has a role in surface adaptation, interior point placement for creating layered tetrahedra, and as a speed function for generating prismatic layers. In contrast, $f_{out}[\mathbf{x}]$ has a role only in adaptation of surface meshes, where it is necessary to respect a minimum sampling frequency in order to provably recover a surface with Delaunay methods [1]. In a geometry as complex as the mouse heart, there are numerous instances where folds in the surface are such that $f_{out}[\mathbf{x}] \gg f[\mathbf{x}]$. In these cases, modulating $f[\mathbf{x}]$ with $f_{out}[\mathbf{x}]$ during surface adaption is necessary to preserve topology.

2.3.2 GLFS and Surface Adaption—Imaging based isosurfaces, such as those generated in Section 2.2.4, are highly irregular, noisy, and often over-resolved in some areas and conversely under-resolved in some others. To generate a volume mesh based on those isosurfaces would be wasteful of computational resources at best. At worse, error in the computed fields would be concentrated at the smaller scale, for example, in the coronary vasculature or around the valves. To address this, we modify S to produce a high-quality surface mesh S' by performing the operations of smoothing, refinement and de-refinement while limiting perturbations to a small fraction of a voxel. To assure scale-invariance, the operations are registered to the GLFS [25]. Specifically, we perform the operations of (1) Rivara edge bisection, (2) node merging, (3) edge swaps, (4) in order to create surface triangles whose edges are close to $c_f f[\mathbf{x}]$ for some constant c_f . Thus, where $c_f f[\mathbf{x}]$ is large, say across the myocardial wall, the surface triangles will be correspondingly large, and where $c_f f[\mathbf{x}]$ is small, say at the valves of coronary vasculature, the surface triangles will be correspondingly small. These operations have been presented in [11,25] and will not be repeated here.

Instead, we focus on a necessary further modification of the GLFS in the presence of tube-like structures such as the chordae tendinea and the coronary arteries. In these areas, modulating the GLFS with the first principle curvature, as described in Algorithm 5 and Equation 2, is not sufficient to guarantee a minimum number of elements circumferentially. This is particularly true in instances where these features are close to the resolution of the voxel.

For these cases, we introduce a novel curvature correction to $f[\mathbf{x}]$, such that the surface edge-length is sufficiently small to accommodate an angular spacing of NRING nodes around the circumference of the tube. This implies a surface edge length of

$$\frac{2\pi \times \text{tube radius}}{\text{NRING}} \quad (3)$$

This new edge-length may be less than the previous 'ray-shoot' edge length of $\frac{K}{M}f[\mathbf{x}]$, where K is desired volume-element anisotropy (defined as the ratio of edge-length to volume element depth), and M is the desired number of layers. This is especially true where the tube diameter is a single voxel wide, or equal to L_{min} . Thus, we dictate that the surface edge length should be as small as $\frac{2\pi}{\kappa_1 NRING}$. This is done by reducing $f[\mathbf{x}]$ if necessary. That is, we modify $f[\mathbf{x}]$ by

$$f[\mathbf{x}] \leftarrow \min\left(f_1[\mathbf{x}], \frac{K}{M} \frac{2\pi}{\kappa_1 NRING}\right) \quad (4)$$

where $f_1[\mathbf{x}]$ has been previously computed. We again perform a gradient-limiting operation on the resulting GLFS field (Equation 2) and then employ surface mesh manipulation operations

to establish an edge length of $c_t f[x] \equiv \frac{K}{M} f[x]$.

Remark 1 In this section we have specified a functional value for c_t , and the reader may wonder why this was not defined earlier. The reason is simply that c_t may be different for surface adaption and prism propagation (Section 2.3.4).

2.3.3 Layered Anisotropic Tetrahedra—As observed before, cardiac tissue is laminated. Thus, we desire the computational grid of the cardiac tissue to be similarly laminated. We achieve this by creating layers of tetrahedral elements on the interior of S' . Points are cast along “seeding rays” from each point $\mathbf{x}_i$ on the surface S' in the direction $\hat{n}[\mathbf{x}_i]$. Once these points have been cast, they are connected with a *layer-aware* Delaunay algorithm, presented in [25]. In this section, we present a modified point insertion method that produces measurably better results.

If M is the target number of layers across the cross-section of the geometry, then points

$\mathbf{x}_i^m, 0 \leq m \leq \frac{M}{2}$, are equally or ratio spaced between $\mathbf{x}_i^0 \equiv \mathbf{x}_i$ on the surface to $\mathbf{x}_i + \frac{1}{2}f[\mathbf{x}_i]\hat{n}[\mathbf{x}_i]$

Due to gradient-limiting, we may have $f[\mathbf{x}_i] < F[\mathbf{x}_i]$ and so extra ‘filler’ points $\mathbf{x}_i^{\frac{M}{2}+1}, \dots, \mathbf{x}_i^{M_i}$ are inserted further in towards the interior (using ratio spacing) between $\mathbf{x}_i + \frac{1}{2}f[\mathbf{x}_i]\hat{n}[\mathbf{x}_i]$ and $\mathbf{x}_i + \frac{1}{2}F[\mathbf{x}_i]\hat{n}[\mathbf{x}_i]$. The presence of these extra filler points guarantees points are distributed over the whole geometry, with possible overlap, and possible undesirable proximity to portions of the surface that are nearly grazed by the seeding rays. Consequently, a filtering operation eliminates “duplicate” points that lie within a fraction of L_{min} of each other. A *layer-aware* Delaunay algorithm connects these points with the restriction that the filler points are not inserted if Delaunay point insertion would connect them to any point $\mathbf{x}_i$ on the surface S' .

Tetrahedra that contain no interior points (points $\mathbf{x}_i^m, m \geq 1$) are removed. Finally, the tetrahedral grid is improved with layer-aware, edge-flipping operations and a “crushing algorithm”, which inserts nodes on the opposed diagonals of slivers and then merges the nodes together, thus eliminating them.

2.3.4 Prismatic Boundary Layer—Having generated a quality layered tetrahedral grid for the heart tissue, we wish to create several layers of prisms at the Lagrangian interface between tissue and blood. There are several reasons for this. First, prismatic boundary layers are desirable for viscous flows, which tend to have strong gradients at the boundary due to wall shear stress, an effect that – as we have already mentioned – has been shown to be extremely important in cardiac/cardiovascular biomechanics. In addition, prismatic layers have been shown to be effective in reducing errors in turbulence, particle deposition, and mass-transfer problems [33,27,7,24]. In biological fluid-structure interaction problems with a Lagrangian interface, prismatic boundary layers are also desirable near the interfaces, such as closing

cardiac valves, because the direction of impaction is locally normal to the boundary, so it would be aligned with axis of the prisms. This will not only provide more accurate results (since the larger gradients will be aligned with the mesh) but it is a desirable mesh property for any mesh movement algorithm in the framework of Arbitrary Lagrangian-Eulerian (ALE) formulations.

Our method advances a surface layer by solving the Lagrangian evolution equation,

$$\frac{\partial \mathbf{x}}{\partial t} = \varphi(\mathbf{x}, t) \hat{\mathbf{n}} \quad (5)$$

where t denotes time, φ denotes the normal velocity, and $\hat{\mathbf{n}}$ denotes unit surface normal. Generating a layer of prisms reduces to propagating the surface for some time interval by discretizing Equation (5). Traditional Lagrangian methods are prone to the problem of “swallowtails” in strongly concave regions with large curvatures. To avoid such issues, we apply the *face offsetting method* in [17], which is based on a geometric construction called the *generalized Huygens' principle*. Unlike traditional Lagrangian methods, which propagate vertices along some vertex normals, the face-offsetting method first propagates faces and then reconstructs the vertices using numerical techniques of least-squares approximation and eigenvalue analysis. The method also adapts the time step to prevent mesh folding. To achieve mesh quality of the prisms, we also developed a novel variational smoothing procedure that simultaneously improves the shapes of the base triangles and the orthogonality of the side edges, as described in [11].

We have described our general approach of prismatic boundary layer meshes in [11]. In our initial implementation, we used the GLFS of the surface as the speed function φ in (5), and propagated the surface by a desired percentage of the GLFS. For more complex geometries like the mouse heart, however, it is important for the speed function to adapt to the changing curvatures as the surface evolves. In the present paper, we enhanced our implementation to recompute the GLFS of the evolved surface periodically. This enhancement allows areas with large curvatures to slow down gradually, while allowing the remainder of the surface to propagate as far as possible. This enhancement significantly improved the robustness of our algorithm, and allowed the boundary-layer mesh to cover about 30% of the volume for the blood region of the mouse heart, nearly twice as much as our original method. The reevaluation of the GLFS introduces only moderate overhead to our method. For the mouse heart mesh, each reevaluation requires about 3.5 seconds on a moderate computer. After generating the prismatic boundary layer meshes, we tetrahedralize the interior with a boundary constrained Delaunay method implemented in the code TetGen [31].

3 Results

Figure 4 illustrates the highly detailed segmentation of tissue and blood, with the segmented blood volumes clearly separated into right (deoxygenated) and left (oxygenated) volumes. Figure 5 depicts the gradient limited feature sized field on exterior and interior surfaces of the reconstructed mouse heart. Note the detection of the coronary vasculature on the exterior surface (Figure 5A) and its revealed structure on the interior heart (Figure 5B). Figure 6 shows the generated layered tetrahedral mesh, which has 15,571,963 elements. Five prism layers, with 6,882,265 prisms in total, are shown in Figures 7 and 8. The Delaunay blood mesh, with 5,735,350 tetrahedra, is shown in Figure 8.

Mesh quality statistics for both tetrahedra and prisms are detailed in Figure 9. For tetrahedra, we report the *aspect ratio* which is proportional to the ratio of the inscribed radius to the length of the longest edge. For prisms, we report the so-called scaled aspect ratio (see [11]), whose definition is somewhat more involved. In effect, the scaled aspect ratio combines the measures

of triangle shapes and edge orthogonality. Both quality metrics vary between 0 and 1, where 1 is optimal. The mesh quality for all three regions is excellent. Boundary prisms are layered by design, and layered tetrahedra resolve tissue lamination at all scales, from the ventricular septum to the coronary vasculature.

4 Discussion and Conclusion

The role of computational continuum physics in cardiac research has expanded beyond an engineering analysis of devices or even idealized organs, to become a platform for discovery and integration. This thrust has been paralleled by a major research focus on biomedical image processing (www.itk.org, www.slicer.org, mipav.cit.nih.gov, rsbweb.nih.gov/ij). Not surprisingly, biomedical geometries are increasingly derived from imaging modalities such as magnetic resonance or computed tomography. Not only can a connection to imaging data provide critical geometric information — atherosclerosis for example is arguably a disease with a strong geometric component — but also critical physiologic data, such as diffusion anisotropy, genetic expression or even metabolism. Biomedical geometries, however, are complex and hierarchical. Despite this, little attention in the biomedical literature has been paid to the enormous challenges posed by geometric computation of biomedical geometries.

Efficient unstructured mesh generation of these geometries in a way that is tuned to the physics of the problem is challenging. Finite computational resources dictate that the grid must be adapted to the geometry in order to minimize both computational cost and discretization error. At the same time, many biophysics problems require that the grid be organized both by scale and by intrinsic properties. The wall of the heart, for example, is a laminated muscle consisting of three separate layers, each with a separate family of fiber and myocyte orientations. Thus, a grid of the heart muscle must be similarly layered. That same layering persists at the smaller scale of the heart geometry, where, for example the musculature transitions into the connective tissue of the cardiac valves; or at an even smaller scale where the heart muscle involutes to become a network of coronary arteries. These transitions of scale are of course mirrored in the blood. The lumen of a coronary artery, for example, may be several orders of magnitude smaller than the span across a ventricle, and thus there is a need to equilibrate error over a range of meaningful scales. These challenges of course are not limited to the heart.

However, the heart does pose some unique challenges. First, the effects of blood flow on cardiac mechanics and physiology are profound. From a biological viewpoint the interface is the zone of greatest activity and communication; from a medical point of view it is the site of the greatest vulnerability to disease. Thus, biomedical fluid-structure problems must aim to resolve continuum fields accurately at the interface. To address these challenges, we have presented the numerical framework to accomplish a Lagrangian treatment of tissue-blood interfaces [12]. Work in that area continues. Such an approach is greatly simplified if exact nodal compatibility exists between each region of the grid. Secondly, unlike arteries or veins in isolation, most of the heart structures cannot be approximated by a plane-stress membrane assumption. This implies the explicit discretization with volume elements such as tetrahedra, hexahedra and prisms. Quality discretization of complex geometries with hexahedra remains forbidding.

We have presented a method for computing high-quality scale-invariant grids from such imaging data, and have exemplified the approach with a demanding, perfusion fixed micro MRI dataset of the *in situ* mouse heart and its structure. The mesh quality achieved for this geometry is excellent by any measure. This quality owes to three factors: a careful and topologically correct segmentation to produce initial surfaces, the adoption of the gradient-limited feature size field, and variational optimization of marched prisms based on the face-offsetting method.

The concept of scale is directly embedded into all aspects of our approach by efficient computation of the gradient-limited feature size field. Were edge lengths based solely on some geometry-independent values, the number of elements required would increase by two or three orders of magnitude to capture the finest structures. In contrast, adoption of the GLFS field enables the efficient placement of mesh only where it is needed. Further savings are achieved by decoupling the mesh densities in the surface tangential and normal directions, though tangential density is locally isotropic. Adoption of the GLFS also enables adequate circumferential discretization of tubular regions of the geometry (Equation 4), and a finite boundary layer to resolve the stronger gradients that are found at the wall.

To explicate our method, we have presented detailed pseudo-code for novel aspects of our methods. Further details may be found in earlier references [11,25]. Our methods are applicable to a wide variety of multimaterial data sets and are particularly well-suited to segmentation of perfused tissues. The objective here is to enable others to reproduce our results on related data sets.

Despite the success of our approach, there remain at least four areas of potentially substantial improvement. First, the grid density tangential to the surface is isotropic. On structures such as the cordae or coronary arteries, where fluid flow is largely axial, it would be considerably more efficient to introduce anisotropy into the grid, such that the greater density were arranged in the direction with the highest curvature. Such an approach has been successfully applied to vascular trees in the past [30,16]. A promising approach that is compatible with our current methods is described in [18]. A second area of improvement would be to relax the requirement of constrained Delaunay tetrahedralization for the blood core. Discretization of this zone can be considered preliminary since adaption in our fluid solver is driven by an error metric, and thus discretization will evolve. Nevertheless, the initial quality of this interior grid is important. Although the current approach will typically generate high-quality hybrid meshes for accurate and efficient numerical simulations, the use of constrained Delaunay tetrahedralization can limit the quality of the tetrahedral core for some complex biological geometries. Thirdly, it must be noted that while the overall method does not depend on “good” surfaces, it does depend on “correct” surfaces. In most cases, surface adaptation and volume conserving smoothing improve the input surfaces sufficiently. Application of the latter is critical for mesh improvement, and has the advantage over conventional smoothing approaches that the volume conservation constraint limits drift. However, our smoothing algorithm does not guarantee against self-intersection, and therefore needs to be applied with care. With complex geometries, a robust volume-conserving smoothing with theoretical guarantees against self-intersection would be an important asset. Finally, prisms have inherent advantages with respect to tetrahedra in terms of orthogonality and their avoidance of numerical pitfalls, such as locking¹. Additionally, prisms can potentially reduce the element count in a given zone by a factor of 2–3. Thus, the face-offsetting method has additional potential as a discretization tool, if select nodes can be merged, collapsing prisms into a small number of pyramids and tetrahedra. Addressing these four areas of improvement is part of our future work.

Though computational geometry has received little attention in the biomedical literature, there are some notable exceptions. A number of references to discretization of biomedical geometries can be found in the numerical methods literature. We cite only a few here that have taken a comprehensive view of biomedical geometries that spans imaging and geometric analysis. For example, a method described by Labelle for “isosurface stuffing” [26], fills an isosurface with tetrahedral mesh with excellent quality statistics. Though this method has not been applied to biomedical geometries, that we are aware of, it is certainly applicable. While this approach

¹Modern linear tetrahedra have overcome historic issues with volumetric and shear locking in the large-deformation, nearly incompressible limit, e.g. see [8]

may be appropriate for some applications, it does not organize the grid by layer or scale and is generally unsuitable for fluid-flow simulations. A somewhat related approach that produces multiple material lattice-like tetrahedra on biological domains has been presented in [37]. This approach is similarly suited to a range of applications that fall outside cardiac fluid-structure interactions because the hexahedra are not aligned with the surface and there is no recognition of scale. A promising alternative is the isogeometric approach pioneered by Hughes et al. [15], and adapted for fluid-structure analysis of treed biomedical geometries by Zhang et al. [36]. This method has the dual advantages of discretizing volumes with quality hexahedra and being fully integrated with the numerical solver. However, it is unsuited to domains that cannot be reduced to a one-dimensional network of curves. Thus, it is well suited to arterial trees, but unsuited to the entire cardiac geometry.

In contrast, our method uses a gradient-limited feature size (GLFS) to guide the adaptation of the surface mesh, the generation of layered tetrahedra, the generation of the prismatic mesh at the boundary layer and the generation of tetrahedral mesh in the interior. This achieves scale-invariance such that even the smallest resolved coronary vessel is guaranteed to have a preset number of layers across its diameter. Furthermore, guided by the GLFS, we are able to laminate the cardiac tissue, seamlessly transitioning from myocardium, to cardiac valve to coronary lumen. In addition, this approach enables the automatic adaption of the boundary layer by scale. Our method is also automatic in the sense that the user must only specify a few meaningful parameters: number of layers, degree of anisotropy, growth ratio for the prisms, and minimum and maximum resolved feature sizes.

Acknowledgments

Research funded by the National Heart Lung and Blood Institute Awards 1R01HL073598-01A1 and 1R01HL084431-01A1, by the National Science Foundation under award number DMS-0809285, and by the U.S. Department of Energy under LDRD DE-AC05-76RL01830.

References

1. Amenta N, Bern M. Surface reconstruction by Voronoi filtering. *Discr. and Comp. Geom* 1999;22:481–504.
2. Butcher JT, Markwald RR. Valvulogenesis: the moving target. *Philos Trans R Soc Lond B Biol Sci* 2007;362(1484):1489–503. [PubMed: 17569640]
3. Butcher JT, Nerem RM. Valvular endothelial cells and the mechanoregulation of valvular pathology. *Philos Trans R Soc Lond B Biol Sci* 2007;362(1484):1445–57. [PubMed: 17569641]
4. Butcher JT, Tressel S, Johnson T, Turner D, Sorescu G, Jo H, Nerem RM. Transcriptional profiles of valvular and vascular endothelial cells reveal phenotypic differences: influence of shear stress. *Arterioscler Thromb Vasc Biol* 2006;26(1):69–77. [PubMed: 16293796]
5. Chen LC, Nadziejko C. Effects of subchronic exposures to concentrated ambient particles (caps) in mice. v. caps exacerbate aortic plaque development in hyperlipidemic mice. *Inhal Toxicol* 2005;17(4–5):217–24. [PubMed: 15804939]
6. Chien S, Li S, Shiu YT, Li YS. Molecular basis of mechanical modulation of endothelial cell migration. *Front Biosci* 2005;10:1985–2000. [PubMed: 15769679]
7. Chiou MC. Particle deposition from natural convection boundary layer flow onto an isothermal vertical cylinder. *Acta Mechanica* 1998;129:1619–6937.
8. Cisloiu R, Lovell M, Wang J. Astabilized mixed formulation for finite strain deformation for low-order tetrahedral solid elements. *Finite Elements in Analysis and Design* 2008;44:472–482.
9. Cummins PM, Cotter EJ, Cahill PA. Hemodynamic regulation of metalloproteinases within the vasculature. *Protein Pept Lett* 2004;11(5):433–42. [PubMed: 15544564]
10. Dardik A, Yamashita A, Aziz F, Asada H, Sumpio BE. Shear stress-stimulated endothelial cells induce smooth muscle cell chemotaxis via platelet-derived growth factor-bb and interleukin-1alpha. *J Vasc Surg* 2005;41(2):321–31. [PubMed: 15768016]

11. Dyedov V, Einstein DR, Jiao X, Kuprat AP, Carson JP, del Pin F. Variational generation of prismatic boundary-layer meshes. *Int. J. Numer. Meth. Engrg.* 2009 to appear.
12. Einstein DR, Pin FD, Kuprat AP, Jiao X, Carson JP, Kunzelman KS, Cochran RP, Guccione J, Ratcliffe M. Fluid-structure interactions of the mitral valve and left heart: Comprehensive strategies, past, present and future. *Communications in Numerical Methods in Engineering.* 2009 to appear.
13. Ganguli A, Persson L, Palmer IR, Evans I, Yang L, Smallwood R, Black R, Qwarnstrom EE. Distinct nf-kappab regulation by shear stress through ras-dependent ikappa-balpa oscillations: real-time analysis of flow-mediated activation in live cells. *Circ Res* 2005;96(6):626–34. [PubMed: 15731464]
14. Hove JR, Koster RW, Forouhar AS, Acevedo-Bolton G, Fraser SE, Gharib M. Intracardiac fluid forces are an essential epigenetic factor for embryonic cardiogenesis. *Nature* 2003;421(6919):172–7. [PubMed: 12520305]
15. Hughes TJR, Cottrell JA, Bazilevs Y. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry, and mesh refinement. *Computer Methods in Applied Mechanics and Engineering* 2005;194:4135–4195.
16. Jansen KE, Shephard MS, Beall MW. On anisotropic mesh generation and quality control in complex flow problems. 10th International Meshing Roundtable. 2001
17. Jiao X. Face offsetting: A unified approach for explicit moving interfaces. *J. Comput. Phys* 2007;220:612–625.
18. Jiao X, Colombi A, Ni X, Hart J. Anisotropic mesh adaptation for evolving triangulated surfaces. 15th International Meshing Roundtable. 2006
19. Jiao X, Einstein DR, Dyedov V, Carson JP. Automatic identification and truncation of boundary outlets in complex imaging-derived biomedical geometries. *Medical & Biological Engineering & Computing.* 2009 Submitted.
20. Jiao X, Zha H. Consistent computation of first- and second-order differential quantities for surface meshes. *ACM Solid and Physical Modeling Symposium.* 2008
21. Johnson GA, Cofer GP, Gewalt SL, Hedlund LW. Morphologic phenotyping with MR microscopy: the visible mouse. *Radiology* 2002;222(3):789–93. [PubMed: 11867802]
22. Khamayseh A, Hansen G. Use of the spatial kd-tree in computational physics applications. *Commun. Comput. Phys* 2007;2:545–576.
23. Kostelec P, Weaver J, D. H. Multiresolution elastic image registration. *Medical Physics* 1998;25(9): 1593–1604. [PubMed: 9775363]
24. Kroger C, Drossinos Y. Particle deposition in a turbulent boundary layer over a large particle size spectrum. *J. Aerosol Sci* 1997;28:631–632.
25. Kuprat AP, Einstein DR. An anisotropic scale-invariant unstructured mesh generator suitable for volumetric imaging data. *J. Comput. Phys* 2009;228:619–640. [PubMed: 19784397]
26. Labelle F, Shewchuk JR. Isosurface stuffing: Fast tetrahedral meshes with good dihedral angles. *ACM Transactions on Graphics (ACM SIGGRAPH Conference Proceedings)* 2007;26(3):1–57.
27. Longest WP. Comparison of blood particle deposition models for non-parallel flow domains. *J. Biomech* 2003;36(3):421–430. [PubMed: 12594990]
28. Lorensen WE, Cline HE. Marching cubes: A high resolution 3d surface construction algorithm. *Computer Graphics* 1987;21:163–169.
29. Mironov V, Visconti RP, Markwald RR. On the role of shear stress in cardiogenesis. *Endothelium* 2005;12(5–6):259–61. [PubMed: 16410225]
30. Remacle JF, Li X, Shephard MS, Flaherty JE. Anisotropic adaptive simulation of transient flows using discontinuous Galerkin methods. *Int. J. Num. Meth. Eng* 2005;62:899–923.
31. Si H. Adaptive tetrahedral mesh generation by constrained Delaunay refinement. *Int. J. Numer. Meth. Engrg.* 2008 <http://dx.doi.org/10.1002/nme.2318>.
32. Smits B. Efficiency issues for ray tracing. *SIGGRAPH '05: ACM SIGGRAPH 2005 Courses* 2005;6. <http://doi.acm.org/10.1145/1198555.1198745>.
33. Taylor CA, Hughes TJR, Zarins CK. Finite element modeling of blood flow in arteries. *Comput. Methods Appl. Mech. Engrg* 1998;158:155–196.
34. Treece GM, Prager RW, Gee AH. Regularised marching tetrahedra: improved iso-surface extraction. *Computers & Graphics* 1999;23(4):583–598.

35. Yashiro K, Shiratori H, Hamada H. Haemodynamics determined by a genetic programme govern asymmetric development of the aortic arch. *Nature* 2007;450(7167):285–8. [PubMed: 17994097]
36. Zhang Y, Bazilevs Y, Goswami S, Bajaj C, Hughes TJR. Patient-specific vascular nurbs modeling for isogeometric analysis of blood flow. *Computer Methods in Applied Mechanics and Engineering* 2007;196(29–30):2943–2959.
37. Zhang Y, Hughes TJR, Bajaj CL. An automatic 3d mesh generation method for domains with multiple materials. *Computer Methods in Applied Mechanics and Engineering*. 2009 to appear.

Fig. 1.
Sample axial slice (249×216) from the MRI dataset. Image has $50\mu m$ isotropic resolution

Fig. 2. Histogram of the pixel intensity across all images in the volume. Each voxel was assigned to four different material types based on empirically determined thresholds.

Fig. 3. Semi-automatic fuzzy determination of disputed voxels. The tissue is shown in dark grey; blood is shown in white, and disputed voxels are shown in light grey. The process of resolving the fuzzy set is shown from left to right.

Fig. 4. Multimaterial segmentation of the mouse heart. Initial example MR slice is shown in (A) with the cropped region highlighted by the box and shown segmented into blood (blue), tissue (red), and other (white) in (B). Various renderings of the segmentation (C-I) as visualized in MeVisLab (<http://www.mevislab.de/>).

Fig. 5. GLFS computed on tissue (A) and interior blood (B) surfaces. Note the different dynamic ranges. The coronary vasculature apparent in (B) is implicit in (A) where surface rays detect the involuted boundary of the arteries. Computation of $f[x]$ for these surfaces took 61 seconds and 63 seconds on a Mac Pro.

Fig. 6. Layered tetrahedral mesh of the cardiac tissue. (A) Apex-to-base cut-away of the entire mouse heart. (B) Left ventricular chords. Note the smooth transition of scales from chords to the myocardium. (C) Detail of the left atrium. (D) Tricuspid valve and septum. (E) Cut-away through the anterior and posterior leaflets. Arrows indicate locations of coronary tunnels. (F) View of the mitral valve from the left ventricle.

Fig. 7. Five layers of boundary prisms, propagated by a dynamically updated GLFS field. Note the adaption to scale and the preservation of orthogonality.

Fig. 8.

Prisms with enclosed Delaunay tetrahedra. Panel A shows a number of coronary arteries traveling through the myocardium (empty space in picture). The arrow indicates the termination of a coronary vessel. Since the capillaries are not resolved, correct termination with appropriate boundary conditions is a challenge that may be met with additional geometric analysis (see [19]). Panel B shows the blood surrounding a ventricular chord in the right ventricle. Panel C shows the entrance to the aorta. Tissue elements are not shown for clarity.

Fig. 9.
Mesh quality statistics for tissue, boundary layer and core blood.

Fig. 10. Tissue, boundary layer and core blood regions shown together. Panel A: cross section through atria and ventricle. Panel B: wall of the right ventricle and embedded coronary vessel. Panel C: detail of the coronary vessel. Panel D: superior vena cava.

Algorithm 1

Segment volume V into initial material types M_b, M_h, M_o, M_d

```

1:  initialize thresholds  $t_1, t_2, t_3$  for image volume  $V$  to be applied to segmentation (material mask)  $S$ 
2:  for each voxel  $v \in V$ 
3:    if  $V[v] \leq t_1$  then  $S[v] \leftarrow M_o$ 
4:    else if  $t_1 < V[v] \leq t_2$  then  $S[v] \leftarrow M_h$ 
5:    else if  $t_2 < V[v] \leq t_3$  then  $S[v] \leftarrow M_d$ 
6:    else if  $t_3 < V[v]$  then  $S[v] \leftarrow M_b$ 

```

Algorithm 2

Assign M_d to M_b or M_h

```

1:  [Assign fully encapsulated  $M_d$  voxel groups to  $M_b$  or  $M_h$ ]
2:   $Neighbors[v]$  is the set of neighboring voxels of  $v$  using 26-adjacency criterion (face, edge, or vertex)
3:  for each voxel  $v \in V$ 
4:    Initialize  $L_{visited}[v] \leftarrow FALSE$ 
5:  for each voxel  $v \in V$ 
6:    if not  $L_{visited}[v]$  then
7:      Initialize encapsulated  $M_d$  voxel group  $D \leftarrow \emptyset$ 
8:      Initialize voxel group boundary type blood  $T_b \leftarrow FALSE$ 
9:      Initialize voxel group boundary type heart  $T_h \leftarrow FALSE$ 
10:     call recursive_depth_first_search( $v, T_b, T_h, S, D, L_{visited}$ )
11:     if  $T_b = TRUE$  and  $T_h = FALSE$  then
12:       for each voxel  $d \in D$ 
13:          $S[d] \leftarrow M_b$ 
14:       else if  $T_b = FALSE$  and  $T_h = TRUE$  then
15:         for each voxel  $d \in D$ 
16:            $S[d] \leftarrow M_h$ 
17:
18:   Subroutine recursive_depth_first_search( $a, T_b, T_h, S, D, L_{visited}$ )
19:      $L_{visited}[a] \leftarrow TRUE$ 
20:     if  $S[a] = M_b$  then  $T_b \leftarrow TRUE$ 
21:     else if  $S[a] = M_h$  then  $T_h \leftarrow TRUE$ 
22:     else if  $S[a] = M_d$  then
23:       add  $a$  to group  $D$ 
24:       for each voxel  $n \in N[a]$ 
25:         call recursive_depth_first_search( $n, T_b, T_h, S, D, L_{visited}$ )
26:
27: [Assign coronary vessel  $M_d$  voxels to  $M_b$ ]
28: for each voxel  $v \in V$ 
29:   if  $S[v] = M_d$  then
30:     if  $|Neighbors[v] \cap S^{-1}[M_h]| > 13$  then  $S[v] \leftarrow M_b$ 
31:
32: [Assign remaining  $M_d$  to  $M_b$  or  $M_h$ ]
33: for each voxel  $v \in S^{-1}[M_d]$ 
34:   visually evaluate smoothness and anatomical topology
35:   manually assign  $S[v]$  to  $M_h$  or  $M_b$ 

```

Algorithm 3Reassign non-cardiovascular M_h and M_b voxels to M_o

```

1:   [Create and apply binary mask image  $B_1$  to first xy-plane  $P_1 \in V$ ]
2:   let  $v$  represent the voxel corresponding to the pixel  $\omega$ 
3:   for each pixel  $\omega \in P_1$ 
4:       if  $S[v] = M_o$  then  $B_1[\omega] \leftarrow 0$ 
5:       else if  $\omega$  is visually determined to be cardiovascular then  $B_1[\omega] \leftarrow 1$ 
6:       else  $B_1[\omega] \leftarrow 0$  and  $S[v] \leftarrow M_o$ 
7:
8:   [Create and apply binary mask image to successive  $P$ ]
9:   let  $\Phi_{j,k}$  represent a warp function that transforms input image  $P_j$  to match image  $P_k$ 
10:  for each xy-plane  $P_i \in V$ ,  $2 \leq i \leq \text{number of xy-planes in } V$ 
11:      generate  $\Phi_{i-1,i}$  via multiresolution elastic image registration (see [23])
12:       $B_i \leftarrow \Phi_{i-1,i}(B_{i-1})$ 
13:      visually compare  $B_i$  to  $P_i$  for new features and manually update  $B_i$  as appropriate
14:      for each pixel  $\omega \in P_i$ 
15:          if  $B_i[\omega] = 0$  then  $S[v] \leftarrow M_o$ 

```

Algorithm 4

Validate topology

```

1:  Faces[v] is the set of up to 6 face-connected neighboring voxels of v
2:  Edges[v] is the set of up to 12 edge-connected neighboring voxels of v
3:  Vertices[v] is the set of up to 8 vertex-connected neighboring voxels of v
4:  do until S does not change
5:    [Ensure  $M_b$  has no  $M_o$  neighbors]
6:    for each voxel  $v \in S^{-1}[M_b]$ 
7:      for each voxel  $n \in \text{Neighbors}[v]$ 
8:        if  $S[n] = M_o$  then  $S[n] \leftarrow M_h$ 
9:    [Ensure face-face connectivity for all materials]
10:   for each voxel  $v \in V$ 
11:     for each voxel  $e \in \text{Edges}[v]$ 
12:        $FF \leftarrow \text{Faces}[v] \cap \text{Faces}[e]$  [2 voxels]
13:       if  $S[v] = S[e]$  and  $S[v] \notin FF$  then
14:         visually evaluate anatomical topology
15:         manually assign one of  $FF \cup v \cup e$  to  $M_h$ ,  $M_b$ , or  $M_o$ 
16:       for each voxel  $r \in \text{Vertices}[v]$ 
17:          $EF \leftarrow \text{Edges}[v] \cap \text{Faces}[r]$  [3 voxels]
18:         if  $S[v] = S[r]$  and  $S[v] \notin EF$  then
19:           visually evaluate anatomical topology
20:           manually assign one of  $EF \cup v \cup e$  to  $M_h$ ,  $M_b$ , or  $M_o$ 
21:   [Ensure exactly four connected main material components]
22:   [heart tissue, blood left, blood right, other]
23:    $ncomponents = 0$ 
24:   for each voxel  $v \in V$ 
25:      $component[v] = 0$ 
26:   for each voxel  $v \in V$ 
27:     if  $component[v] = 0$  then
28:        $ncomponents = ncomponents + 1$ 
29:        $component[v] = ncomponents$ 
30:       stack  $K = \emptyset$ 
31:       put  $v$  on  $K$ 
32:       while  $K \neq \emptyset$ 
33:         pop  $w$  off  $K$ 
34:         for each voxel  $f \in \text{Faces}[w]$ 
35:           if  $component[f] = 0$  then
36:              $component[f] = ncomponents$ 
37:             put  $f$  on  $K$ 
38:   for each  $n$ ,  $1 \leq n \leq ncomponents$ 
39:      $C_n \leftarrow \{v | component[v] = n\}$ 
40:   determine which four of  $\{C_1, \dots, C_{ncomponents}\}$  represent the four main material

```

- 41: components: heart tissue, blood left, blood right, other
- 42: **for each** n , $1 \leq n \leq ncomponents$
- 43: **if** C_n is not a main main material component **then**
- 44: manually reassign voxels to M_h , M_b , or M_o as anatomically appropriate
- 45: either to connect C_n to a main material component
- 46: or to integrate C_n into its surrounding component
-

Algorithm 5Modulate $F[\mathbf{x}]$

```

1:  Inputs:  $L_{\min}, L_{\max}, F_{out}[\mathbf{x}]$ , first principal curvature ( $\kappa_1$ ), and a curvature parameter  $\gamma$ 
2:     $\forall x \in S$ 
3:     $F[\mathbf{x}] \leftarrow \min(L_{\max}, F[\mathbf{x}])$ 
4:     $A[\mathbf{x}] \leftarrow \max(L_{\min}, F[\mathbf{x}])$ 
5:     $B[\mathbf{x}] \leftarrow \frac{\gamma}{2} A[\mathbf{x}]$ 
6:     $C[\mathbf{x}] \leftarrow \max(\frac{\gamma}{\kappa_1}, B[\mathbf{x}])$ 
7:     $F[\mathbf{x}] \leftarrow \min(A[\mathbf{x}], C[\mathbf{x}])$ 

```