# Patient-Specific Vascular NURBS Modeling for Isogeometric Analysis of Blood Flow[*]

**Yongjie Zhang**[1], **Yuri Bazilevs**[1], **Samrat Goswami**[1], **Chandrajit L. Bajaj**[2], and **Thomas J.R. Hughes**[1]

Yongjie Zhang: jessica@ices.utexas.edu; Yuri Bazilevs: bazily@ices.utexas.edu; Samrat Goswami: samrat@ices.utexas.edu; Chandrajit L. Bajaj: bajaj@cs.utexas.edu; Thomas J.R. Hughes: hughes@ices.utexas.edu

[1] Institute for Computational Engineering and Sciences, The University of Texas at Austin, United States

[2] Department of Computer Sciences and Institute for Computational Engineering and Sciences, The University of Texas at Austin, United States

## Abstract

We describe an approach to construct hexahedral solid NURBS (Non-Uniform Rational B-Splines) meshes for patient-specific vascular geometric models from imaging data for use in isogeometric analysis. First, image processing techniques, such as contrast enhancement, filtering, classification, and segmentation, are used to improve the quality of the input imaging data. Then, lumenal surfaces are extracted by isocontouring the preprocessed data, followed by the extraction of vascular skeleton via Voronoi and Delaunay diagrams. Next, the skeleton-based sweeping method is used to construct hexahedral control meshes. Templates are designed for various branching configurations to decompose the geometry into mapped meshable patches. Each patch is then meshed using one-to-one sweeping techniques, and boundary vertices are projected to the lumenal surface. Finally, hexahedral solid NURBS are constructed and used in isogeometric analysis of blood flow. Piecewise linear hexahedral meshes can also be obtained using this approach. Examples of patient-specific arterial models are presented.

### Keywords

Patient-specific vascular models; hexahedral mesh; skeleton-based sweeping; NURBS; isogeometric analysis; blood flow

## 1 Introduction

Recently, patient-specific modeling was proposed as a new paradigm in simulation-based medical planning. Physicians, using computational tools, construct and evaluate combined anatomical/physiological models to predict the outcome of alternative treatment plans for an individual patient. A comprehensive framework has been developed to enable the conduct of computational vascular research [1,2]. Blood flow simulations provide physicians with physical data to help them devise treatment plans.

Isogeometric analysis is a new computational technique that improves on and generalizes the standard finite element method. It was first introduced in [3], and expanded on in [4]. In an effort to instantiate the concept of isogeometric analysis, an analysis framework based on

NURBS was built. Mathematical theory of this NURBS-based approach was put forth in [5]. NURBS is not the only possible basis for isogeometric analysis but it is certainly the most highly developed and widely utilized. For an introductory text on NURBS, see Rogers [6]. A more advanced treatment of the subject is given in Piegl and Tiller [7]. Other geometric modeling techniques that have potential as a basis for isogeometric analysis include: A-patches [8], T-splines [9], and subdivision [10]. These warrant further investigation.

Figure 1 shows one such model, obtained from patient-specific imaging data. We have designed a set of procedures which allows us to create solid NURBS vascular models directly from patient-specific data. We have named this process the vascular modeling pipeline, which can be divided into four main steps:

1.  Preprocessing – in scanned Computed Tomography (CT) or Magnetic Resonance Imaging (MRI) data, the intensity contrast may not be clear enough, noise exists, and sometimes the blood vessel boundary is blurred. Therefore, we use image processing techniques to improve the quality of CT/MRI data, such as contrast enhancement, filtering, classification, and segmentation.

2.  Path Extraction – The goal is to find arterial pathes. Vascular surface models can be constructed from the preprocessed imaging data via isocontouring. The skeleton is then extracted from the surface model using Voronoi and Delaunay diagrams. This skeletonization scheme is suitable for noisy input and creates one-dimensional clean skeletons for blood vessels.

3.  Control Mesh Construction – a skeleton-based sweeping method is developed to construct hexahedral NURBS control meshes by sweeping a templated quad mesh of a circle along the arterial path. Templates for various branching configurations are presented which decompose the geometry into mapped meshable patches using the extracted skeleton. Each patch can be meshed using one-to-one sweeping techniques. Some nodes in the control mesh lie on the surface, and some do not. We project nodes lying on the surface to the vascular surface. The blood vessel wall can be built by radially extending the surface outside 10%-15% of the distance to the center line.

4.  NURBS Construction and Isogeometric Analysis – after generating hexahedral control meshes, we construct solid NURBS geometric models and employ isogeometric analysis to simulate blood flow. Piecewise linear hex meshes can also be obtained. Three numerical examples, coronary, thoracic and abdominal arteries, are presented.

The remainder of this paper is organized as follows: Section 2 reviews related previous work. Section 3 describes the meshing pipeline and preprocessing for our geometric modeling approach. Section 4 talks about solid NURBS construction and isogeometric analysis. Section 5 explains the skeleton-based sweeping method and decomposition templates for various branching configurations. Section 6 presents three numerical examples. Section 7 draws conclusions and outlines planned future work.

## 2 Previous Work

### Sweeping Method

Sweeping, or Inline $2\frac{1}{2}-D$ meshing, is one of the most robust techniques to generate semi-structured hexahedral meshes. One-to-one sweeping requires that the *source* and *target* surfaces have similar topology. The source surface is meshed with quadrilaterals [11], which are swept through the volume using *linking* surfaces as a guide [12].

However, few geometries satisfy the topological constraints required by one-to-one sweeping. In the CUBIT project [13] at Sandia National Labs, a lot of research has been done to automatically recognize features and decompose geometry into mapped meshable areas or volumes. Various many-to-one and many-to-many sweeping methods have been developed [14,15,16,17]. Care should also be taken in locating internal nodes during the sweeping process [18,19].

### Medial Axis-based Mesh Generation

Medial axis is the locus of points that are minimally equidistant from at least two points on the geometry's boundary. The medial axis transform provides an alternative representation of geometric models that has many useful properties for analysis modeling [20]. Applications include decomposition of general solids into subregions for mapped meshing, identification of slender regions for dimension reduction and recognition of small features for suppression. The medial surface subdivision technique [21] decomposes the volume into map-meshable regions, which are then filled with hex elements using templates.

Medial axis has been used to construct hexahedral meshes for CAD objects. The skeleton-based modeling methods were developed for solid models [22]. Quadros et al. used a skeleton technique to control finite element mesh size [23]. Besides other unstructured mesh generation methods [24,25,26], a skeleton-based subdivision method has also been used in biomedical applications, such as a below-knee residual limb and external prosthetic socket [27], and bifurcation geometry in vascular flow simulation [28]. However, trifurcations and more complex branchings also exist in the human artery tree. Therefore, decomposition templates for arbitrary branching configurations are desirable and are constructed in this paper.

### NURBS in Mesh Generation and Analysis

As the most highly developed and widely utilized technique, NURBS [6,7,29] has evolved into an essential tool for a semi-analytical representation of geometric entities. Sometimes NURBS solid models are taken as input for finite element mesh generation [30]. Anderson et al. proposed a fast generation of NURBS surfaces from polygonal mesh models of human anatomy [31]. An enhanced algorithm was developed for NURBS evaluation and utilization in grid generation [32]. In isogeometric analysis [3], NURBS basis functions are used to construct the exact geometry, as well as the corresponding solution space.

## 3 Meshing Pipeline and Preprocessing

The input images are often of poor quality which makes it difficult to generate quality meshes for regions of interest. To circumvent this problem we pass the raw imaging data through a preprocessing pipeline where the image quality is improved by enhancing the contrast, filtering noise, classifying, and segmenting regions of various materials. The surface model is then extracted from the processed imaging data, and the vessel path is obtained after skeletonizing the volume bounded by the surface. First we modify the geometry by removing unnecessary components, then extract the skeleton. The generated path can also be edited according to simulations, e.g., adding a path for a Left Ventricle Assist Device (LVAD) in the thoracic aorta model (Figure 12). A skeleton-based sweeping method is then used to generate hexahedral control meshes for solid NURBS construction and isogeometric analysis. Figure 2 shows the meshing pipeline. The preprocessing step of our skeleton-based meshing approach is described below, including image processing, isocontouring and geometry editing, and path extraction.

### Image Processing

We choose a fast localized method for image contrast enhancement [33]. The basic idea is to design an adaptive transfer function for each individual voxel based on the intensities in a

suitable local neighborhood. A bilateral pre-filtering coupled with an evolution driven anisotropic geometric diffusion PDE (partial differential equation) [34] is utilized to remove noise. Sometimes we need to classify the voxels into several groups, each of which corresponds to a different type of material. We choose an approach which relies on identification of the contours by membership of seed points which are located by the gradient vector diffusion [35]. A variant of the fast marching method is adopted [36] to segment the imaging data to find the clear boundary of each voxel group belonging to a certain category.

### Isocontouring and Geometry Editing

There are two main isocontouring methods from imaging data: Primal Contouring (or Marching Cubes [37]) and Dual Contouring [38]. In this application we choose Dual Contouring to extract the isosurface, because it tends to generate meshes with better aspect ratios. We then modify the model to suit our particular application. This can be done in various ways, for example, by removing unnecessary components, adding necessary components which are not constructed from imaging data, denoising the surface, etc. After getting the vessel path, we can edit it according to simulation requirements. For example, we can add a path for the left ventricle assist device (LVAD) in the thoracic aorta model (Figure 12).

### Path Extraction

The vertex set of the extracted and possibly repaired geometry is then used to create an interior path lying in the middle of the blood vessels. We define a squared distance function which assigns to any point $x \in \mathbb{R}^3$, the minimum square distance to the vertex set. We further compute the index 1 and index 2 saddle points of this distance function and compute the unstable manifold of these two types of critical points. The identification of the critical points along with their indices and the computation of the unstable manifold are done efficiently via the Voronoi and its dual Delaunay diagram of the point set. The details of this method can be found in [39]. We adopt this method of path generation because it has several advantages which are useful for the patient specific modeling of blood vessels. One advantage is that it can handle noisy input gracefully. Often the noise present in the data is not fully eliminated after the preprocessing stage. In the path generation step we employ another stage of filtering which helps to construct a clean skeletal path for the extracted geometry. Secondly, the extracted geometry may have flat regions where it is not straight forward to obtain a linear path. Fortunately our starring scheme, as described in [39], eliminates these spurious features and create the one-dimensional path. The results of this path generation step are shown on various datasets (Figures 1, 11, 12).

## 4 Solid NURBS Construction and Isogeometric Analysis

In a NURBS-based isogeometric analysis a physical domain in $\mathbb{R}^3$ is defined as a union of patches. A patch, denoted by $\Omega$, is an image under a NURBS mapping of a parametric domain $(0,1)^3$

$$\Omega = \{\mathbf{x} = (x, y, z) \in \mathbb{R}^3 | \mathbf{x} = \mathbf{F}(\xi, \eta, \zeta), 0 < \xi, \eta, \zeta < 1\},$$

(1)

where

$$\mathbf{F}(\xi, \eta, \zeta) = \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{l} R_{i,j,k}^{p,q,r}(\xi, \eta, \zeta)\, \mathbf{C}_{i,j,k},$$

(2)

$$R_{i,j,k}^{p,q,r} = \frac{N_{i,p}(\xi) M_{j,q}(\eta) L_{k,r}(\zeta) w_{i,j,k}}{\sum_{\hat{i}=1}^{n} \sum_{\hat{j}=1}^{m} \sum_{\hat{k}=1}^{l} N_{\hat{i},p}(\xi) M_{\hat{j},q}(\eta) L_{\hat{k},r}(\zeta) w_{\widehat{i,j,k}}}.$$

(3)

In the above, $R_{i,j,k}^{p,q,r}$ ($\xi, \eta, \zeta,$)'s are the rational basis functions, and $C_{i,j,k}$'s $\in \mathbb{R}^3$ are the control points. In the definition of the rational basis, $N_{i,p}(\xi)$'s, $M_{j,q}(\eta)$'s, and $L_{k,r}(\zeta)$'s, are the univariate B-spline basis functions of polynomial degree $p$, $q$, and $r$; $w_{i,j,k}$'s, strictly positive, are the weights.

In isogeometric analysis the geometry generation step involves construction of a control mesh, which is a piecewise multi-linear interpolation of control points, and the corresponding rational basis functions. The initial mesh encapsulates the 'exact geometry' and, in fact, defines it parametrically.

For the purposes of analysis, the isoparametric concept is invoked (see Hughes [40]). The basis for the solution space in the physical domain is defined through a push forward of the rational basis functions defined in (2) (see [5] for details). Coefficients of the basis functions, defining the solution fields in question (e.g., displacement, velocity, etc.), are called control variables.

As a consequence of the parametric definition of the 'exact' geometry at the coarsest level of discretization, mesh refinement can be performed automatically without further communication with the original description. This is an enormous benefit. There are NURBS analogues of finite element $h$- and $p$-refinement, and there is also a variant of $p$-refinement, which is termed $k$-refinement, in which the continuity of functions is systematically increased. This seems to have no analogue in traditional finite element analysis but is a feature shared by some meshless methods. For the details of the refinement algorithms see [3].

The isogeometric approach is fundamentally higher-order. For example, in order to represent circles, cylinders and spheres, rational polynomials of at least quadratic order are necessary. The generation of refined NURBS bases of all orders is facilitated by simple recursion relationships. The versatility and power of recursive NURBS basis representations are truly remarkable. Equation systems generated by NURBS tend to be more homogeneous than those generated by higher-order finite elements and this may have some benefit in equation solving strategies. NURBS satisfy a 'variation diminishing' property. For example, they give monotone fits to discontinuous control data and become smoother as order is increased, unlike Lagrange interpolation polynomials which oscillate more violently as order is increased. NURBS of all orders are non-negative pointwise. This means that every entry of the NURBS mass matrix is non-negative. These properties are not attained in finite element analysis. On the other hand, NURBS are not interpolatory. They are fit to nets of control points and control variables. This aspect is less transparent to deal with than the corresponding finite element concepts of interpolated nodal points and nodal variables but somewhat similar to the situation for meshless methods. There are many robust algorithms to create very complex geometries with NURBS.

## 5 The Skeleton-based Sweeping Method

Blood vessels are tubular objects, therefore we choose the sweeping method to construct hexahedral control meshes for NURBS-based isogeometric analysis.

### 5.1 Sweeping Method

In the sweeping method, a templated quadrilateral mesh of a circle is projected onto each cross-section of the tube, then corresponding vertices in adjacent cross-sections are connected to form a hexahedral mesh. A hexahedral NURBS control mesh should satisfy the following four requirements:

1. Any two cross-sections can not intersect with each other.

2. Each cross-section should be perpendicular to the path line.

3. In the intersection region of several branches, each cross-section should remain perpendicular to the vessel surface.

4. In order to achieve a $G^1$-continuous surface, the boundary vertex shared by two patches in the control mesh should be collinear with its two neighbors along the axial direction, and the boundary vertex shared by three or more patches should be coplanar with all of its neighboring boundary vertices. This is because, for a so-called open knot vector, a NURBS curve is tangent to the control polygon at the first and the last control nodes.

We choose to parameterize the template cross-section as follows. One parametric direction is associated with a closed circular curve, while another parametric direction is associated with a radial coordinate. Rational quadratic basis is used to define the circular curve with a control polygon given by the linear interpolation of the green and blue points shown in Figure 3. For the template shown in Figure 3a, the control polygon is a square consisting of 8 control nodes, while in Figure 3b, it is an octagon. Note that the circular cross-section is unchanged geometrically and parametrically as more control points are chosen for its representation. The green control points lie on the circle, while the blue control points do not. This is due to the fact that the rational basis is interpolatory at the green points and is not interpolatory at the blue points. Also note that each interpolatory control point has two neighboring non-interpolatory points that are collinear with it. This construction guarantees the resultant circular curve to be $G^1$-continuous. Later, when we discuss data fitting, it is only the interpolatory points that get projected onto the true surface. The non-interpolatory points are adjusted to preserve the collinearity in order to obtain a $G^1$-continuous cross-section.

In the process of sweeping, we translate the cross-section template to the selected locations on the path, and rotate it to make its normal vector pointing in the direction tangent to the path as shown in Figure 4. This gives the third parametric direction for the solid NURBS representation. The hexahedral control mesh is constructed by connecting the corresponding control nodes in adjacent cross-sections. Piecewise linear hexahedral meshes can also be generated at the same time by projecting all boundary vertices to the vessel surface, or by interpolating the elements of the solid NURBS geometry.

### 5.2 Branching Templates

One-to-one sweeping requires that the source and target surfaces have similar topology. Generally, arterial models do not satisfy this requirement, therefore we need to decompose arterial networks into mapped meshable regions. In this section, we will discuss various decomposition templates for different branching configurations. An *n*-**branching** is formed when *n* branches join together, where $n \geq 3$. When $n = 3$, it is a bifurcation; when $n = 4$, it is a trifurcation; when $n > 4$, we call this situation **higher order branching**.

In the human vascular system, most branchings are bifurcations. However, trifurcations or higher order branchings also exist. For example, there are several trifurcations in the coronary arteries (Figure 11) and the abdominal aorta (Figure 1). In the following, we will discuss decomposition templates for all possible branching configurations.

**Bifurcation—**For every intersection, a so-called master arterial branch is chosen. Typically, it is an artery with the largest diameter. Suppose the master branch consists of two sub-branches (Branch 1 and Branch 2), and the slave branch is Branch 3, as shown in Figure 5a. The axes of Branch 1, 2 and 3 are Axis 1, 2 and 3 respectively (Axis 1 and Axis 2 may not be collinear). There is one basic case, shown in Figure 5, and all bifurcations can be decomposed into three map-meshable regions by a variant of this basic template.

Figure 5 shows the path, the constructed hexahedral control mesh, the solid NURBS mesh, and the piecewise linear hexahedral meshes of the bifurcation template. The bifurcation geometry is decomposed into three patches: the master branch contains two patches (red and green), and the slave branch has one patch (yellow). Here we choose Level-1-template (Figure 3a) for each cross-section, as the master and slave branches have similar diameters. The bifurcation template also works for finer cross-sections.

When the master branch and the slave branch have different diameters, the control nodes of some cross-sections are distributed unevenly in order to generate better intersection regions. Figure 6 shows two control meshes and their corresponding solid NURBS meshes. The master branch control polygon is deformed from a square to a trapezoid so as to accommodate a slave branch with a smaller diameter. Note that the NURBS basis changes accordingly so as to preserve the circular cross-section, and the quality of the intersection geometry is improved as can be seen in Figure 6 and Figure 7, where the axes of the master and the slave branches are non-orthogonal, or non-coplanar. Although deforming the control polygon of the master branch gives better results as compared to the non-deformed case, for the intersection of branches with high diameter ratios we advocate the use of a finer template for the master branch, such as a Level-2-template or a Level-3-template.

**Trifurcation—**Trifurcation has one master branch and two slave branches. According to the position of slave branches relative to the master branch, we classify all possible trifurcations to fall into five irreducible cases. All other trifurcations can be decomposed into map-meshable regions by extending the five basic decomposition templates.

**Case 1:** The two slave branches are distributed along the peripheral direction of the master branch, and they are in opposite relative to the master branch (the angle between them is around 180°). The same cross-section template can be used for the master and slave branches.

**Case 2:** The two slave branches are distributed along the peripheral direction, and the angle between them is arbitrary. Finer cross-section template is chosen for the master branch.

**Case 3:** The two slave branches are distributed along the axial direction of the master branch, and they intersect with each other.

**Case 4:** The two slave branches are distributed along the axial direction of the master branch, and they do not intersect with each other. This situation degenerates into two bifurcations.

**Case 5:** The two slave branches do not intersect with the master branch at the same point, but they intersect with each other. In this situation, two bifurcations merge into one trifurcation.

If a Level-1-template is selected as the cross-section of the master branch, then there are at most two slave branches along the peripheral direction as shown in Figure 8 (Case 1). If the

two slave branches are not opposite relative to the master branch, or the two slave branches have different diameters from the master branch, then a Level-1-template is not suitable, and we need to choose finer cross-section templates, such as a Level-2-template or a Level-3-template. Similarly, if a Level-2-template is selected as the cross-section of the master branch, then there can be at most four slave branches along the peripheral direction. A Level-3-template allows at most eight slave branches along the peripheral direction. In Case 2 of Figure 8, the two slave branches are distributed along the peripheral direction and they are not opposite, therefore we choose the finer cross-section template for the master branch (Level-2-template), while the slave branch may have coarser cross-sections (Level-1-template).

Case 3 and Case 4 have the same path, but Case 4 degenerates into two bifurcations because its two slave branches do not intersect with each other even though their axes intersect. There is another special situation (Case 5) where two slave branches do not intersect with the master branch at the same intersection point in the skeleton, but the two intersection points are very close and the two slave branches intersect with each other. This situation contains two bifurcations in the skeleton, but it should be considered as one trifurcation. Therefore, when we choose branching configurations, both the path and the vessel size should be considered.

**Remark:** In $n$-branching, $n$ should be decided not only by the path, but also by the diameter of each slave branch. In other words, if neighboring slave branches intersect with each other, then it is $n$-branching. Otherwise, it degenerates into several $m$-branchings, where $m < n$. On the other hand, several $m$-branchings may merge into one $n$-branching if its slave branchings intersect with each other.

**Higher order branching—**Here we discuss three basic templates for $n$-branching when $n > 4$. Relative to the master branch, there are only two directions to arrange slave branches, the peripheral and axial directions of the master branch. All other $n$-branching configurations can be obtained by combining the three basic ones.

**Case 1:** There are three or more slave branches distributed along the peripheral direction of the master branch. Figure 9 shows one example of four slave branches along the peripheral direction. Level-2-template is selected for the master branch. If there are more than four slave branches, the master branch needs to have a finer cross-section. The cross-section template of slave branches can be coarser.

**Case 2:** There are three or more slave branches distributed along the axial direction of the master branch. Neighboring slave branches intersect with each other.

**Case 3:** There are three or more slave branches distributed along the axial direction of the master branch. Slave branches do not intersect with each other. $n$-branching degenerates into several $m$-branchings ($m < n$).

Several lower order branchings may merge into a higher order one, for example, one bifurcation and one trifurcation can merge into a 5-branching. Case 1, Case 2 and Case 3 can be combined together to form more complex configurations. Figure 9 shows one example of 7-branching. It has four slave branches along the peripheral direction and two slave branches along the axis of the master branch.

### 5.3 Data Fitting

After the sweeping step, each circular cross-section needs to be projected onto the vessel surface as shown in Figure 10. First the interpolatory control points (green points) are moved in the radial direction to the true surface. Then, the non-interpolatory points (blue points) are

placed at the intersection of the lines tangent to the true surface passing through the two neighboring interpolatory points.

There are situations when the tangent lines do not intersect inside the fan region defined in Figure 10b, or do not even intersect with each other when they are parallel. This may occur when the cross-section template is not sufficiently fine to capture features of the true surface, or when the true surface is noisy. This situation will also result in an overlap in the geometry. In order to avoid overlap, we force the non-interpolatory point to stay inside the fan region (the sector between two radial rays) by placing it at the midpoint (indicated by the magenta color) of the segment connecting the two interpolatory points. Finally, the location of the interpolatory points is changed so as to preserve $G^1$-continuity of the surface.

After projecting each cross-section to the vessel surface, we construct hexahedral control meshes and generate solid NURBS for patient-specific vascular models. The geometric error can be reduced by choosing a finer template for each cross-section.

### 5.4 Implications for Analysis of Blood Flow in Arteries

The proposed construction of a NURBS solid mesh for isogeometric analysis has the following implications:

1. Parametric definition of the NURBS mesh allows one to refine the boundary layer region near the arterial wall in order to accurately capture flow features.

2. In the case of a flow in a straight circular pipe driven by a constant pressure gradient, NURBS basis of quadratic order gives rise to a point-wise exact solution to the incompressible Navier-Stokes equation system. This also has implications on the overall accuracy of the approach.

3. The choice of the parameterization of the cross-section template gives rise to a singularity in the geometrical mapping at the center. This singularity does not seem to affect the accuracy of the computational results. Other parameterizations of the circular cross-section, containing multiple patches, are also possible.

## 6 Numerical Examples

In this section we present applications of the meshing pipeline to three patient-specific vascular models: a model of a portion of the coronary tree, a model of the thoracic aorta, and a model of the abdominal aorta. Isogeometric analysis is then used to compute blood flow in the models. In all cases, time-dependent, viscous, incompressible Navier-Stokes equations were used as the blood model. The fluid density and dynamic viscosity were chosen to be representative of blood flow. The first example makes use of the Casson model for the dynamic viscosity while in other examples viscosity was set to a constant value. All models are subjected to a time-periodic inflow boundary condition, which simulates the input from a beating heart. The arterial wall is assumed rigid in the first example. Examples two and three present fluid-structure interaction calculations in which the wall is assumed to be elastic (see Bazilevs *et al.* [2] for the details of the mathematical formulation). The rigid wall simulation was performed on a single processor, while the elastic wall simulations were done in parallel.

### A model of a portion of the coronary tree

Data for this model was obtained from CT Angiography imaging data of a healthy male, over 55 years of age. Large motions of the heart, as it supplies blood to the circulatory system, decreases the quality of the imaging data, and makes construction of patient-specific coronary models a challenging task. Nevertheless, we managed to extract a portion of the coronary tree for the purposes of creating an analysis suitable model. Results of the isocontouring algorithm

are shown in Figure 11a. Figures 11b-11d show the path, the control mesh, and the solid NURBS model of the arterial segment. The model was used to study drug delivery processes in arteries. The drug concentration in the blood is modeled as a passive scalar governed by an unsteady advection-diffusion equation. Figure 11e shows the isosurface of the drug concentration at 50% colored by the blood velocity magnitude, revealing that the flow is unsteady, and has many complex features.

### Thoracic aorta model

Data for this model was obtained from CT Angiography imaging data of a healthy male over-30 volunteer. A patient-specific model of the thoracic aorta was constructed by running through the meshing pipeline. An extra branch, representing a left ventricular assist device (LVAD), was added to the arterial model. Evaluation of LVADs, as well as other electromechanical devices used to support proper blood circulation, is of great interest to the cardiovascular community. The path, the control mesh, and the solid NURBS model are shown in Figures12a-12c. Figure 12d shows a result of the fluid-structure interaction simulation. Note that the inlet and the three smaller outlet branches were extended for the purposes of analysis.

### Abdominal aorta

Data for this model was obtained from 64-slice CT angiography of a healthy male over 55 years of age. Various stages of the meshing pipeline are illustrated in Figure 1a-1f. Figure 1g shows a result of the fluid-structure simulation. A computational study using a truncated geometrical model of this aorta was performed in [2]. We used 85 seconds for path extraction and 8 seconds for control mesh construction on a 64-bit dual-AMD 2GHz linux system, and 20 seconds for solid NURBS generation on an Intel 3GHz linux system.

## 7 Conclusions and Future Work

We have developed a four-stage process to construct analysis suitable geometric models from patient-specific vascular imaging data with a goal of using them in isogeometric analysis of blood flow in arteries. We have focused on the NURBS modeling, and did not treat other geometrical modeling technologies, such as A-patches, T-splines, and subdivision. We would like to investigate these techniques in the future.

We have successfully applied our method to three patient-specific examples, which involve a model of a part of the coronary arterial tree, a thoracic aorta model, and an abdominal aorta model. As part of the future work, we would like to apply the techniques described here to modeling and analysis of the human heart.

## Acknowledgments

## References

1. Taylor CA, Hughes TJ, Zarins CK. Finite element modeling of blood flow in arteries. Computer Methods in Applied Mechanics and Engineering 1998;158:155–196.

2. Bazilevs Y, Calo V, Zhang Y, Hughes TJ. Isogeometric Fluid-Structure Interaction Analysis with Applications to Arterial Blood Flow. Computational Mechanics. 2006

3. Hughes TJ, 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.

4. Cottrell J, Reali A, Bazilevs Y, Hughes T. Isogeometric analysis of structural vibrations. Computer Methods in Applied Mechanics and Engineering. 2005 In press.

5. Bazilevs Y, da Veiga LB, Cottrell J, Hughes T, Sangalli G. Isogeometric analysis: Approximation, stability and error estimates for h-refined meshes. Mathematical Models and Methods in Applied Sciences. 2006 Submitted, available as ICES report 06-04.

6. Rogers, DF. An Introduction to NURBS With Historical Perspective. Academic Press; San Diego, CA: 2001.

7. Piegl, L.; Tiller, W. The NURBS Book (Monographs in Visual Communication). 2nd. Springer-Verlag; New York: 1997.

8. Bajaj C, Chen J, Xu G. Modeling with Cubic A-Patches. ACM Transactions on Graphics 1995;14:103–133.

9. Sederberg TW, Cardon DL, Finnigan GT, North NS, Zheng J, Lyche T. T-Spline Simplification and Local Refinement. ACM Transactions on Graphics (TOG), SIGGRAPH 2004;23:276–283.

10. Cirak F, Scott MJ, Antonsson EK, Ortiz M, Schröder P. Integrated modeling, finite-element analysis, and engineering design for thin-shell structures using subdivision. Computer-Aided Design 2002;34:137–148.

11. Blacker T. A New Approach to Automated Quadrilateral Mesh Generation. Int J Numer Meth Engng 1991;32:811–847.

12. Cook WA, Oakes WR. Mapping methods for generating three-dimensional meshes. Computers in Mechanical Engineering 1982:67–72.

13. CUBIT Mesh Generation Toolkit. Web site: http://sass1693.sandia.gov/cubit

14. Blacker T. The Cooper Tool. 5th International Meshing Roundtable 1996:13–29.

15. Shepherd J, Mitchell S, Knupp P, White D. Methods for MultiSweep Automation. 9th International Meshing Roundtable 2000:77–87.

16. Knupp P. Next-Generation Sweep Tool: A Method for Generating All-Hex Meshes on Two-And-One-Half Dimensional Geometries. 7th International Meshing Roundtable 1998:505–513.

17. White D, Saigal S, Owen S. Automatic Decomposition of Multi-Sweep Volumes. Engineering With Computers 2004;20:222–236.

18. Staten M, Canaan S, Owen S. BMSweep: Locating Interior Nodes During Sweeping. 7th International Meshing Roundtable 1998:7–18.

19. Scott M, Earp M, Benzley S. Adaptive Sweeping Techniques. 14th International Meshing Roundtable 2005:417–432.

20. Armstrong C, Robinson D, McKeag R, Li T, Bridgett S, Donaghy R, McGleenan C. Medials for Meshing and More. 4th Int Meshing Roundtable 1995:277–288.

21. Price MA, Armstrong CG, Sabin MA. Hexahedral Mesh Generation by Medial Surface Subdivision: I. Solids with Convex Edges. Int J Numer Meth Engng 1995;38:3335–3359.

22. Storti D, Turkiyyah G, Ganter M, Lim C, Stal D. Skeleton-based modeling operations on solids. ACM Symposium Solid Modeling Applications 1997:141–154.

23. Quadros WR, Owen SJ, Brewer M, Shimada K. Finite Element Mesh Sizing for Surfaces Using Skeleton. 13th International Meshing Roundtable 2004:389–400.

24. Zhang Y, Bajaj C, Sohn BS. 3D Finite Element Meshing from Imaging Data. The special issue of Computer Methods in Applied Mechanics and Engineering (CMAME) on Unstructured Mesh Generation 2005;194(4849):5083–5106.

25. Zhang Y, Bajaj C. Adaptive and Quality Quadrilateral/Hexahedral Meshing from Volumetric Data. Computer Methods in Applied Mechanics and Engineering (CMAME) 2006;195(912):942–960.

26. Ito Y, Shum PC, Shih A, Soni B, Nakahashi K. Robust generation of high-quality unstructured meshes on realistic biomedical geometry. Int J Numer Meth Engng 2006;65:943–973.

27. Zachariah SG, Sanders JE, Turkiyyah GM. Automated Hexahedral Mesh Generation from Biomedical Image Data:Applications in Limb Prosthetics. IEEE Transactions on Rehabilitation Engineering 1996;4(2):91–102. [PubMed: 8798076]

28. Verma CS, Fischer PF, Lee SE, Loth F. An All-Hex Meshing Strategy for Bifurcation Geometries in Vascular Flow Simulation. 14th International Meshing Roundtable 2005:363–375.

29. Thompson, JF.; Soni, BK.; Weatherill, NP. Grid Generation. CRC Press; LLC: 1999.

30. Gursoy HN. Tetrahedral Finite Element Mesh Generation from NURBS Solid Models. Engineering with Computers 1996;12(19):211–223.

31. Anderson CW, Crawford-Hines S. Fast Generation of NURBS Surfaces from Polygonal Mesh Models of Human Anatomy. Technical Report CS-99-101, Colorado State University. 2000

32. Yu TY, Soni BK. NURBS Evaluation and Utilization for Grid Generation. 5th International Conference on Numerical Grid Generation in Computational Field Simulations 1996:323–332.

33. Yu Z, Bajaj C. A Fast and Adaptive Algorithm for Image Contrast Enhancement. IEEE International Conference on Image Processing (ICIP'04) 2004;2:1001–1004.

34. Bajaj C, Wu Q, Xu G. Level Set Based Volumetric Anisotropic Diffusion. ICES Technical Report 301, the Univ of Texas at Austin. 2003

35. Tomasi C, Madcuchi R. Bilateral filtering for gray and color images. IEEE International Conference on Computer Vision 1998:839.

36. Yu Z, Bajaj C. Image Segementation using Gradient Vector Diffusion and Region Merging. 16th International Conference on Pattern Recognition 2002;2:941–944.

37. Lorensen W, Cline H. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. SIGGRAPH 1987:163–169.

38. Ju T, Losasso F, Schaefer S, Warren J. Dual Contouring of Hermite Data. SIGGRAPH 2002:339–346.

39. Goswami S, Dey TK, Bajaj CL. Identifying Flat and Tubular Regions of a Shape by Unstable Manifolds. 11th ACM Sympos Solid and Physical Modeling, to appear. 2006

40. Hughes, TJR. The Finite Element Method: Linear Static and Dynamic Finite Element Analysis. Dover Publications; Mineola, NY: 2000.
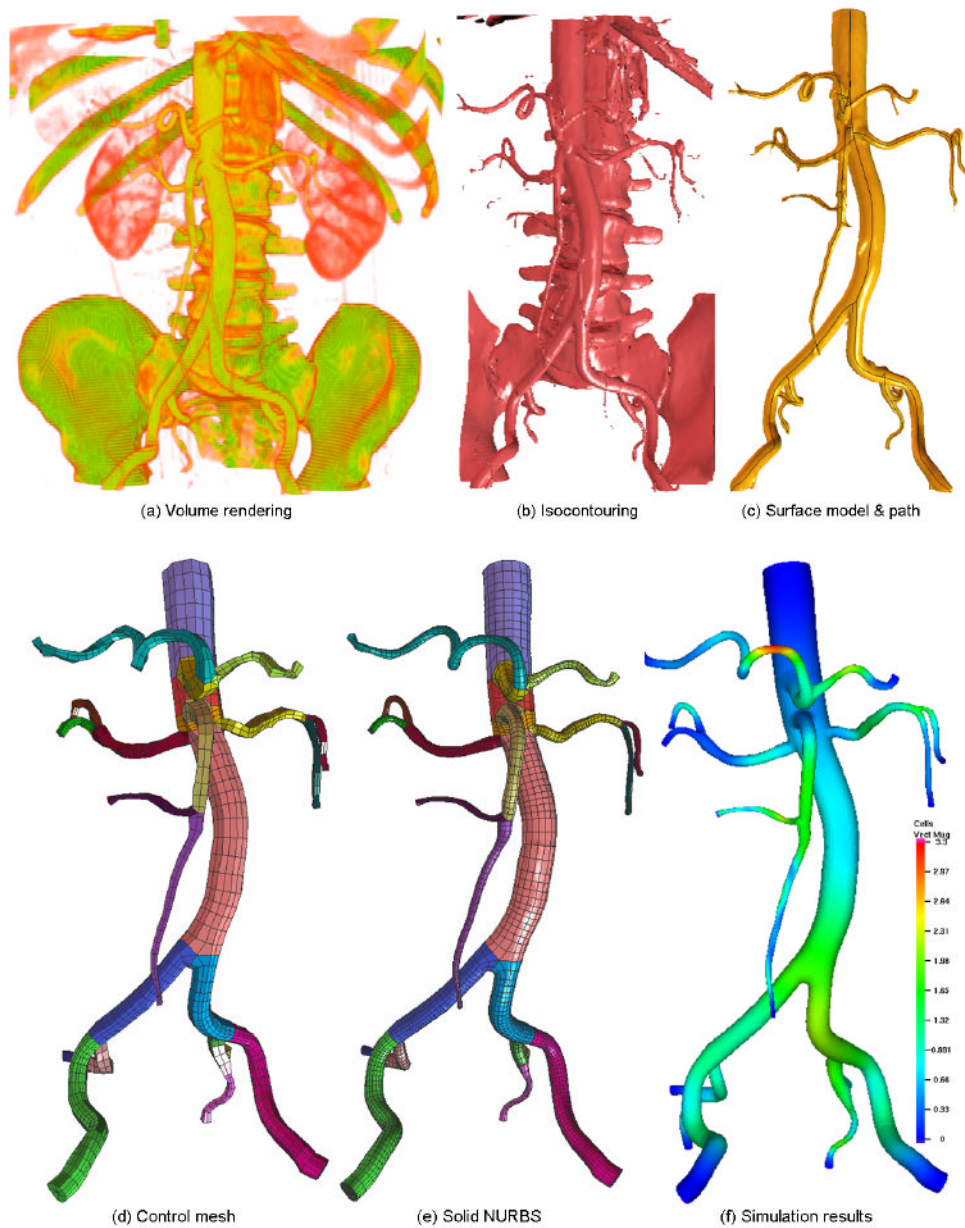
(a) Volume rendering    (b) Isocontouring    (c) Surface model & path

(d) Control mesh    (e) Solid NURBS    (f) Simulation results

**Fig. 1.**
The abdominal aorta model is divided into 26 patches, and each color represents one different patch. (a) - volume rendering result; (b) - isocontouring result; (c) - surface model and its path after removing unnecessary components; (d) - control mesh; (e) - solid NURBS mesh after refinement (73,314 elements); (f) - fluid-structure interaction simulation results: contours of the arterial wall velocity (cm/s) during late systole plotted on the current configuration. Only major branches are kept in (d-f).
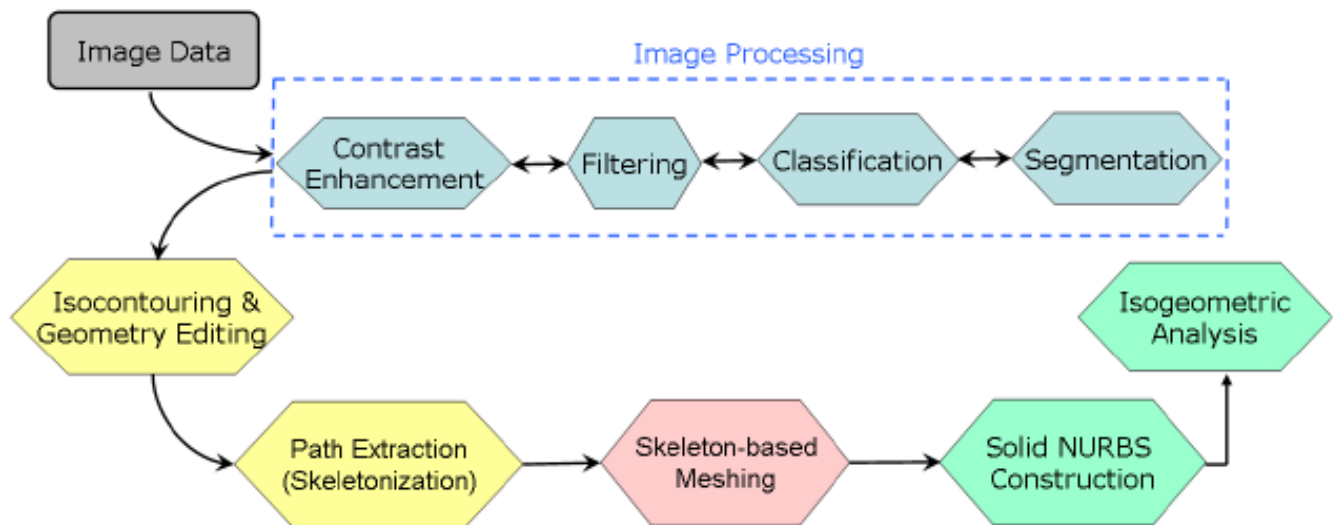
**Fig. 2.**
A schematic diagram of the meshing pipeline. Preprocessing includes three modules: image processing, isocontouring and geometry editing, and path extraction.
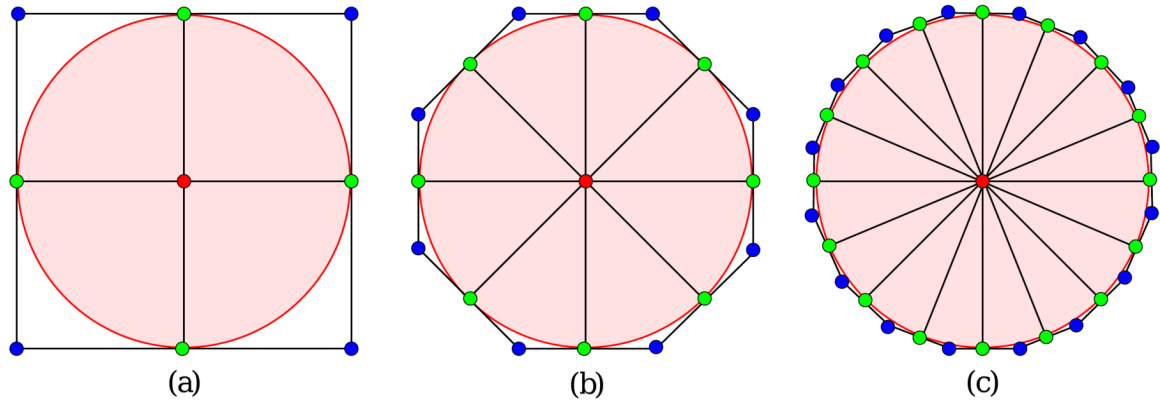
**Fig. 3.**
Multi-resolution templates for cross-sections. (a) Level-1-template (9 control nodes); (b) Level-2-template (17 control nodes); (c) Level-3-template (25 control nodes). Red points are circle centers, green points are interpolatory control nodes on the circle, and blue points are non-interpolatory control nodes defined as the intersection point of two tangent lines at its two neighboring green points.

(a) Path     (b) The sweeping method
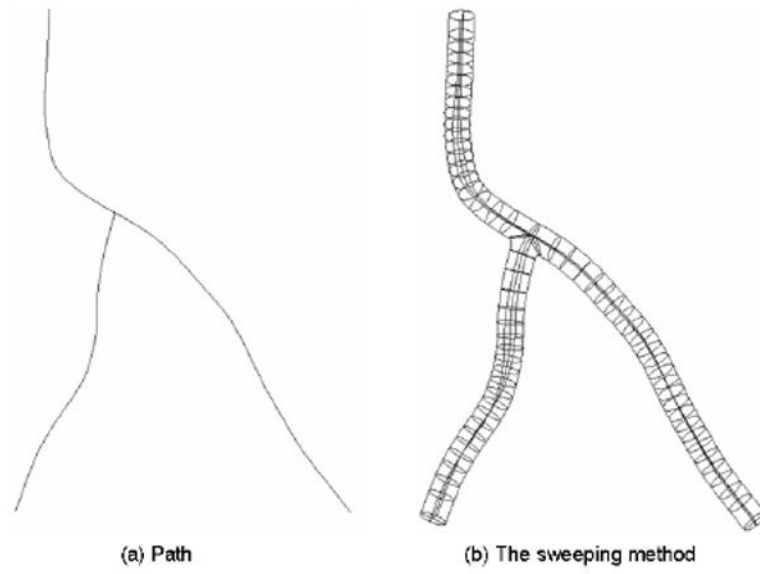
**Fig. 4.**
The skeleton-based sweeping method. (a) - a blood vessel skeleton; (b) - a templated circle is translated and rotated to each cross-section. A bifurcation is shown.
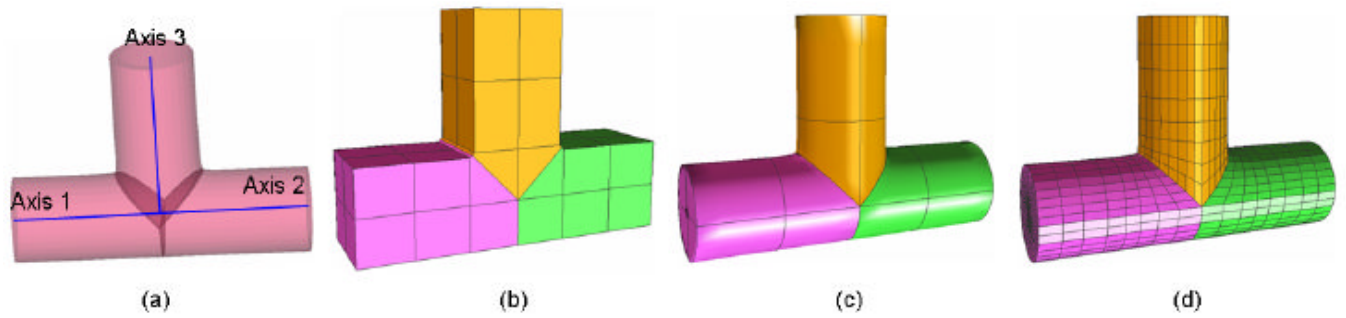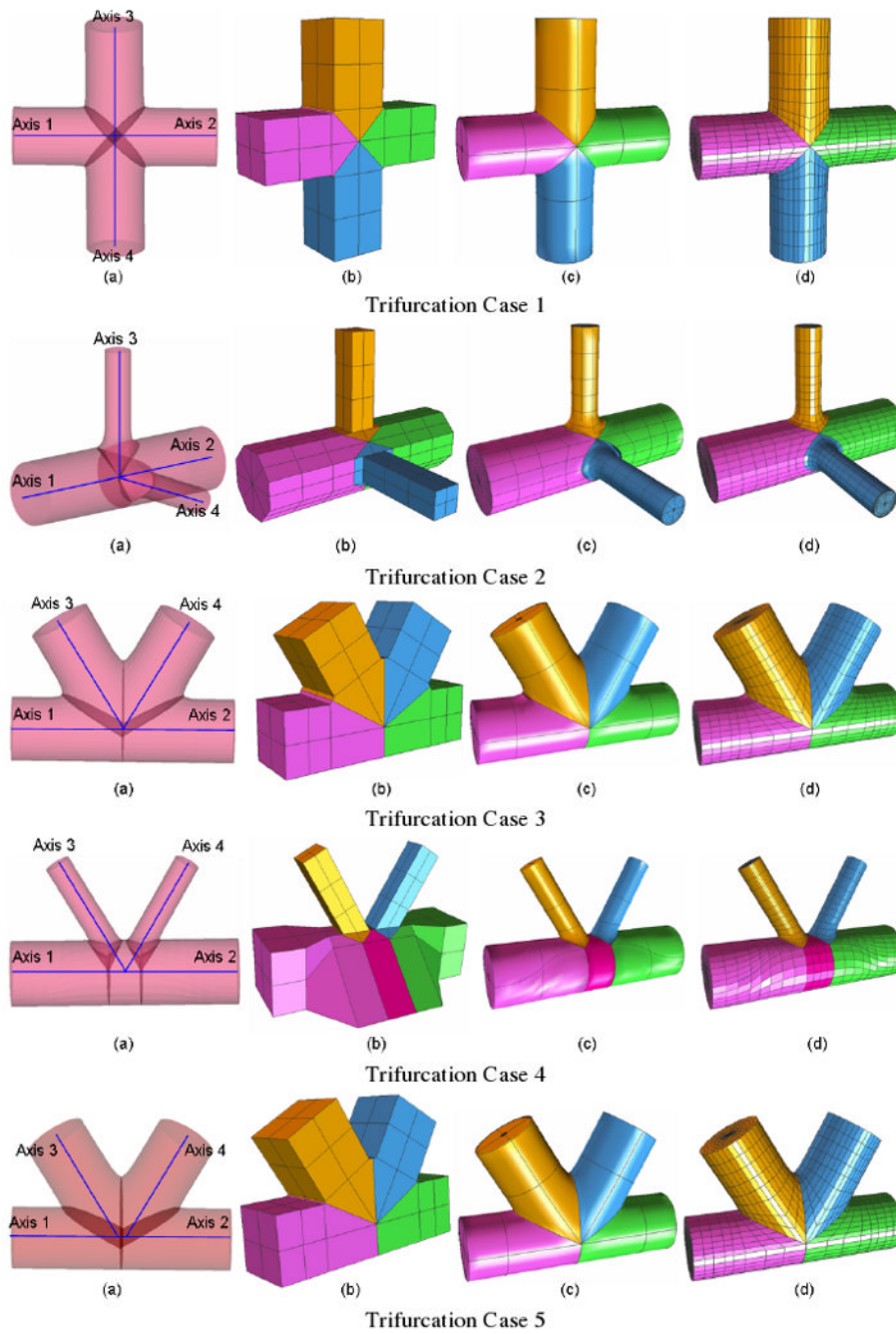
**Fig. 5.**
The bifurcation decomposition template. (a) - path; (b) - control mesh; (c) - solid NURBS; (d) - a piecewise linear hex mesh. The bifurcation geometry is decomposed into 3 patches, and each patch is rendered with a different color.
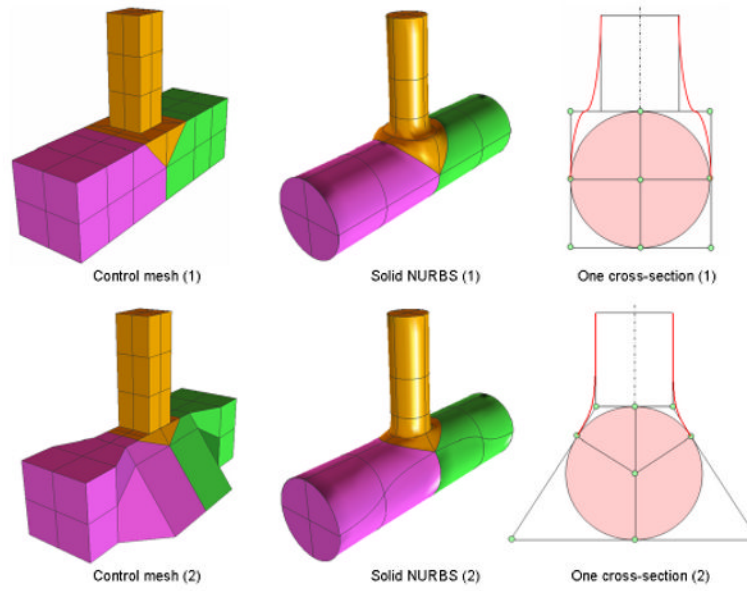
**Fig. 6.**
Comparison of two meshes for the situation when the master branch and the slave branch have different diameters. Control nodes on cross-sections are distributed evenly in Mesh (1) (the top row), and unevenly in Mesh (2) (the bottom row). The red curves in the right two pictures are transition curves.
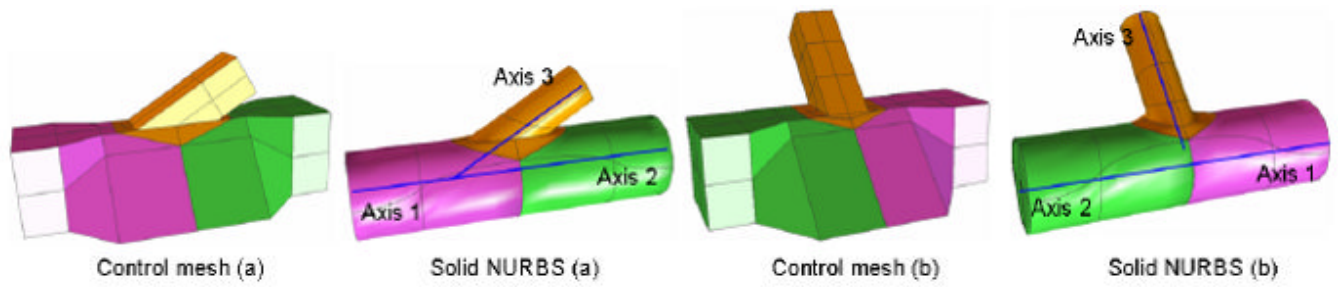
**Fig. 7.**
Control mesh and solid NURBS (a) - the axes of the master and slave branches are not perpendicular to each other; Control mesh and solid NURBS (b) - the axes of the master and slave branches are not coplanar.

**Fig. 8.**
The trifurcation decomposition templates of Case 1-4. (a) - path; (b) - hex control mesh; (c) - solid NURBS; (d) - piecewise linear hex mesh. The Trifurcation geometry is decomposed into 4 patches (Case 1, 2, 3, 5) or 5 patches (Case 4). Each patch is rendered with a different color.
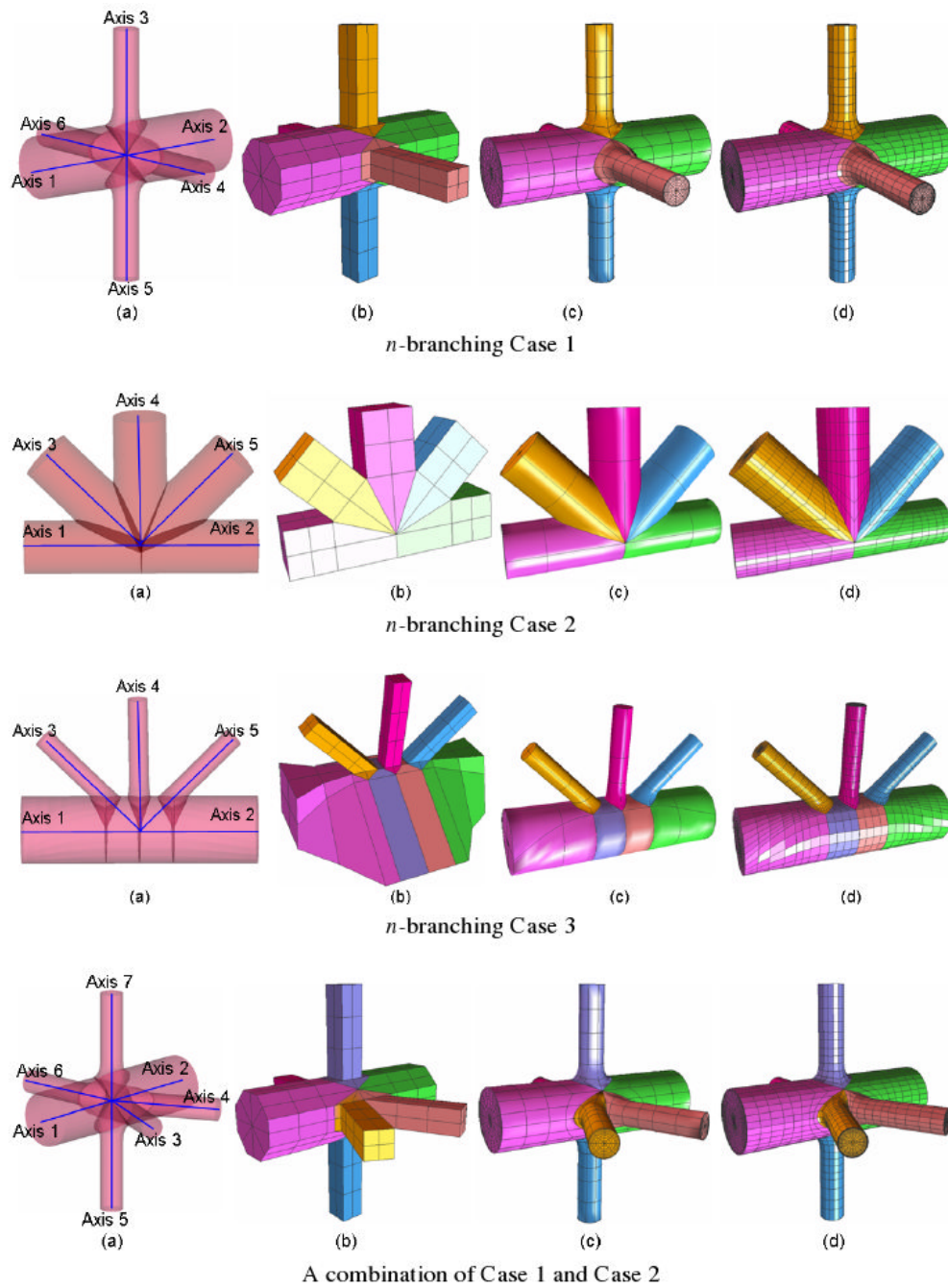
**Fig. 9.**
The *n*-branching templates of Case 1-3 and a combination of Case 1-2. (a) - path; (b) -control mesh; (c) - solid NURBS; (d) - piecewise linear hex mesh. The Trifurcation geometry is decomposed into 6 patches (Case 1), 5 patches (Case 2) or 7 patches (Case 3, a combination of Case 1 and Case 2). Each patch is rendered with a different color.
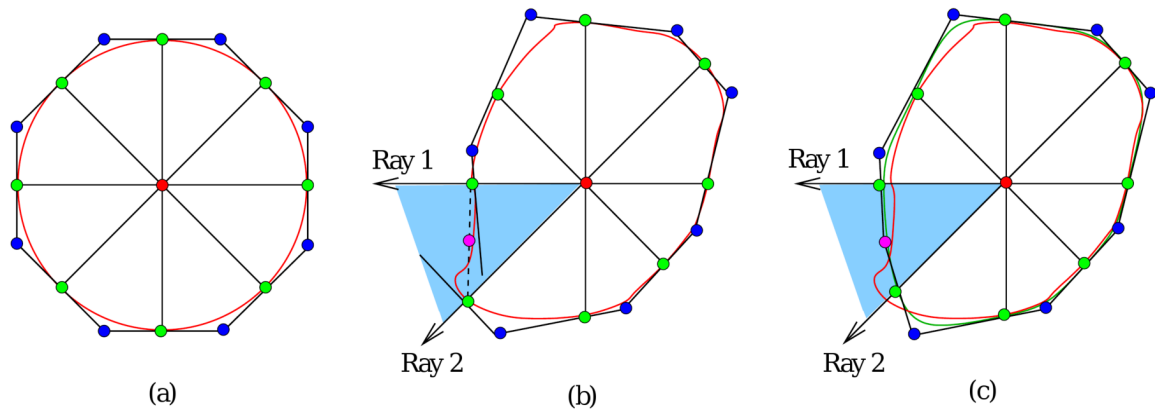
**Fig. 10.**
One cross-section template is projected to the vessel surface. (a) Level-2-template for one circular cross-section; (b) The red curve is the vessel curve. In the blue fan region, the two tangent lines do not intersect with each other, and the magenta point is the calculated control node; (c) The two neighboring green control nodes are adjusted. The green curve is the constructed spline curve. Green control nodes are interpolatory points lying on the vessel surface, and blue points are non-interpolatory.

(a) Isocontouring results (two view angles)
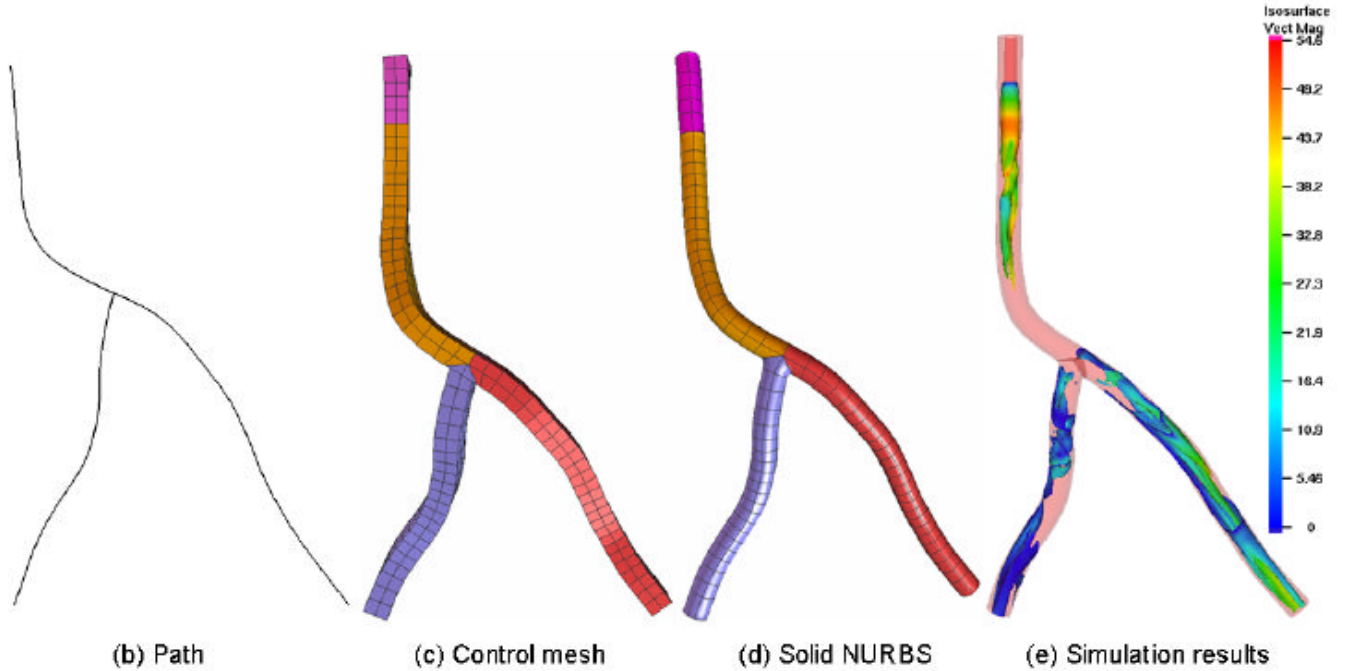
(b) Path  (c) Control mesh  (d) Solid NURBS  (e) Simulation results

**Fig. 11.**
Coronary artery. (a) - isocontouring results (two different view angles); (b) - path; (c) - control mesh; (d) - solid NURBS model (20,824 elements); (e) - rigid wall simulation results: isosurface of the drug concentration at 50% colored by the blood velocity magnitude (cm/s).
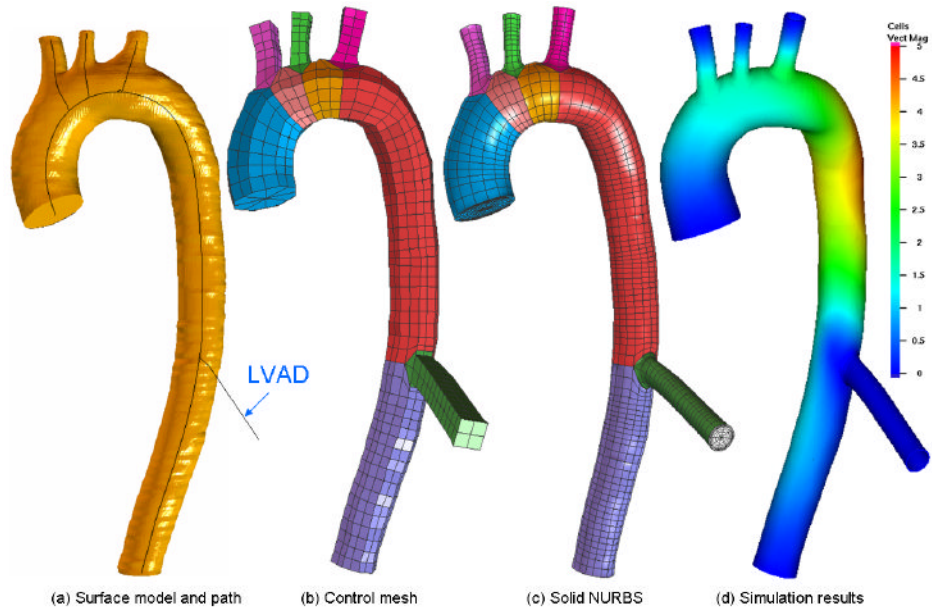
**Fig. 12.**
Thoracic aorta. (a) - surface model and the path, a LVAD is inserted; (b) - control mesh; (c) - solid NURBS (41,526 elements); (d) - fluid-structure interaction simulation results: contours of the arterial wall velocity (cm/s) during late diastole plotted on the current configuration.