PATIENT-SPECIFIC MODELING AND SIMULATION OF BLOOD FLOW IN HUMAN ARTERIES: FROM MAGNETIC RESONANCE IMAGING TO COMPUTATIONAL FLUID DYNAMICS AND FLOW VISUALIZATION

Shanshan Luo

A THESIS

in

Applied Mathematics and Computational Science

Presented to the Faculties of the University of Pennsylvania

in

Partial Fulfillment of the Requirements for the

Degree of Master of Arts

2019

Supervisor of Thesis

Paris Perdikaris
Assistant Professor of Mechanical Engineering and Applied Mechanics

Graduate Group Chairperson

Pedro Ponte Castaneda
Raymond S. Markowitz Faculty Fellow and Professor
Professor of Mechanical Engineering and Applied Mechanics

ACKNOWLEDGEMENT

It was my first time to study in a foreign country, around 7500 miles away from my hometown in China. Life at UPenn is not easy for me, filled with inadaption, perplexation and stress. At first I wanted to finish my courses quickly to come back where I'm familiar with. However, my life began to change gradually and one of the reasons is taking course ENM360, taught by Professor Perdikaris. I learned basic theory of machine learning and Python code from clear and focused lecture given by Professor Perdikaris as well as his solid professional knowledge, which arouses my curiosity to know more about his research interest. When I searched Professor Perdikaris research experience, my eyes were caught by many fantastic visualization of blood simulation results. The story of my thesis began at that time.

The work of my thesis is more about how to use existing tools, which is different from my former learning experience focused on formulas. Instead of following steps in online tutorial and demo case directly, Professor Perdikaris suggests me to do on real case and learn from unexpected errors. When I'm stuck with problems for a long time, he doesn't blame me or show me solution but sharing references or hints to let me explore by myself. I know my thesis work is not his main present research interest, and sometimes when I come to his office I can see his tiredness from his eye, but he is still kind and patient with my disfluent English and slow progress. From Professor Perdikaris, one of things I learn is how to face failure in positive attitude, which is meaningful in my way of life.

I'm also grateful to Eileen Hwuang, who offers real medical data and much guidance in SimVascular software. Eileen explains biological knowledge of the real case contained in my thesis in an understandable way to the layman like me, which helps me finish my thesis very much.

Besides, I would like thank Professor Arratia and Professor Sinno for taking the time to be defense committee member and attend thesis defense.

During my two-year school life at Upenn, I'm fortunate to meet many amazing mentors and want to express my gratitude to them for letting me become a better one. I'd like to thank Dr Spyridon Bakas, who guided me in how to run Deepmedic on linux system to realize brain tumor segmentation. I also thank Professor Monique Guignard for teaching me some knowledge of graph theory and optimization. Many thanks also go to Professor Tony E.Smith for training me to write report formally and logically.

A special thank also goes to my best friend Zhen, for sharing weal and woe, my collaborator Siddhesh, for refreshing my knowledge in coding skills, and my classmate Lingxi, for accompanying me at first year and help me prepare for the prelim exam.

Lastly, I'm thankful to my family. Thanks to my parents for their love and unconditional support at any time. Thanks to them for giving me the chance to see the wider world.

ABSTRACT

Advances in imaging techniques and numerical methods enable patient-specific modeling and simulation of blood flow to become a powerful tool in clinical research of arterial diseases. In this thesis, we review a basic pipeline from medical image display, segmentation, solid model construction, mesh generation, boundary conditions assignment to blood flow simulation and results visualization. We mainly focus on the steps before simulation. Besides description of basic ideas and theory contained, we gives a brief tutorial on using the related software: SimVacuar, Gmsh and Netkar++. Applications of patient-specific models on mechanism of maternal-placental circulation will be discussed in the realization part.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

1. introduction

1.1. Background

Arterial disease, such as atherosclerosis[1] (a kind of arterial disease caused by fatty deposits on inner vessel walls), is the leading threat to human health. Atherosclerosis affects arteries supplying blood to brain, heart and other organs, which induces cardiovascular disease including stroke and coronary artery disease. In 2015, almost 15 million people in worldwide range were dead of such cardiovascular disease.[2] Both the causes and consequences[3] of arterial diseases involve abnormal circulation in blood flow. One important task of hemodynamics[4,5] is to study flowing blood and all the solid structures (such as arteries). Understanding the role played by hemodynamics in the development and progression of arterial disease is crucial in analysis and treatment of arterial diseases.

Amounts of qualitative study[6,7] shows the arterial disease is correlated to hemodynamics in various aspects. For example, Kwak *et.al.*[8] pointed out that blood vessels are exposed to various forces like mechanical stretch and shear stress. These strains and stresses induce the onset of atherosclerotic lesions. In addition, in the study of Morbiducci *et.al.*,[9] atherosclerotic lesion is predisposed to develop at districts with special geometric property such as bifurcations. These sites are disturbed flow regions and exerted to shear stress, providing evidence that hemodynamics are involved in initiation and progression of the atherosclerotic disease.

Besides the well-documented qualitative study, precise quantification of hemodynamics conditions is necessary to evaluate disease risk. Some invasive measurement techniques are useful in this regard. For instance, Chan *et.al.*[10] collect information from sensors placed at different body locations and estimate patient's arterial blood pressure to determine the relative risk of cardiovascular diseases. Invasive methods are accurate with reliable and continuous data and commonly used in clinical treatment.[11] However, these methods are sometimes expensive for special equipment and time-consuming, needs to be operated by trained clinicians. Worse still, methods like catheter insertion may cause complications such as lesion of nerves or vessels.[12] Invasive techniques are also sometimes too risky or impossible to be applied (e.g. measuring blood pressure in the uteroplacental arteries of pregnant women is dangerous.[58]).

For these weakness of *in vivo* measurements, non-invasive measurement techniques based on advanced medical imaging techniques[13, 14, 15, 16] such as computed tomography (CT), magnetic resonance imaging (MRI), ultrasonography, angiography, and elastography have been popular and used routinely in diagnose, since they can capture detailed anatomical structure, blood flow velocity and tissue mechanical properties.[17] Among these techniques, Doppler ultrasound[18] is utilized to measure blood flow velocity. Rather than only one-dimensional Doppler ultrasound, since late 1980s, two-dimensional phase contrast MRI (2D PC-MRI)[59] has become a routine method for assessment of regional blood flow in heart vessels. More recently, time-resolved PC-MRI (termed as '4D flow MRI') allows for the evaluation of complex blood flow patterns in 3D velocity field.[19]

1.2. Motivation

Although the above non-invasive imaging techniques are widely used in clinical treatment, they can't be regarded as reliable alternatives of *in vivo* measurements for inaccurate and not capable in direct measurement of important variables. For example, although MRI[20] can produce high-resolution images of the vessel bifurcation and capable of directing measure flow velocity, there still exists technique challenge in computing wall shear stress (WSS), a key factor in the development of atherosclerosis, because it requires extremely long scan times to obtain data at spatial and temporal resolutions. With the development of algorithms and computing resources, computational fluid dynamics (CFD) modeling,[21] combined with vascular imaging enables physicians to investigate indirectly measured parameters such as WSS and has been a powerful tool in blood flow analysis. Also, for the individual property of medical imaging and clinical data, CFD modeling is also patient-specific and enables individualized risk prediction and treatment plan.

The purpose of our thesis is to review a basic work flow in the application of CFD modeling, that is first using image-processing technique to construct vascular model on the given medical image data, which will be imported into a CFD package later for generating volumetric mesh and blood flow simulation.

The software package SimVascular provides a complete pipeline from medical image data segmentation to patient specific blood flow simulation and analysis. Gmsh is a free 3D finite element mesh generator with a built-in CAD engine and post-processor. Nektar++ is an open-source software framework designed to support development of high-performance scalable solvers for partial differential equations using the spectral/hp element method. Paraview is an open-source, multi-platform data analysis and visualization application. The goal of this thesis is to give a brief tutorial in patient-specific modeling and simulation of blood flow with several different software. SimVascular is used for generating vascular model from Magnetic Resonance Image (MRI). Then we import the geometry model into Gmsh for obtaining better mesh to accommodate specific physical structure and numeral computation convenience. With the boundary conditions we do simulation process in Nektar++ and visualize the output with Paraview for further analysis.

1.3. Open challenge

In medicine field, CFD models[21] are widely used by industrial medical device developers and academics, for the purpose of low-cost device prototyping, investigating arterial physiology and computing parameters not obtained directly. However, the models constructed in CFD modeling usually are complicated and time-consuming, which are not satisfied with clinicians' requirements of rapid and accurate results.

In addition, for CFD modeling application, model accuracy is related with clinical data quality. For example, medical images with low resolution increases difficulty in segmentation and thus the created model is less matching with the real geometry structure. Imaging, image-registration and segmentation algorithms are needed to be improved, although there is some progress in these algorithms in recent years.[22] With solid model, discretization process is also challenging, since it requires detailed knowledge of physiological structure to make mesh adapted to vessel structure. To obtain good simulation results, mesh density is supposed to be fine enough, which, however, induces great need of computing resource and large cost of computing time. Also in simulation step, the input flow waveform data is measured on medical image with assumption, which introduces errors. When setting patient-specific boundary conditions, it's not easy to produce reasonable rather than random simulation results.

1.4. Organization

The rest of our thesis is organized as follows. In section 2, we provide the basic pipeline for imaged-based modeling and preparation work for simulation. In section 3, a real case of constructing a vascular model for pregnant women is displayed. Conclusion and further discussions are summarized in section 4.

2. Methods

In this section, we introduce a pipeline for vascular modeling and simulation pipeline as shown in Figure 1. The first step of geometry modeling based on MRI image in SimVascular is to display image data. Next, create path approximated to center-line of vessel, along which, a set of 2D contours of vessels' lumen is generated and stitched together to create the geometry model. The next steps are mesh generation and define boundary conditions, prepared for blood flow simulation.

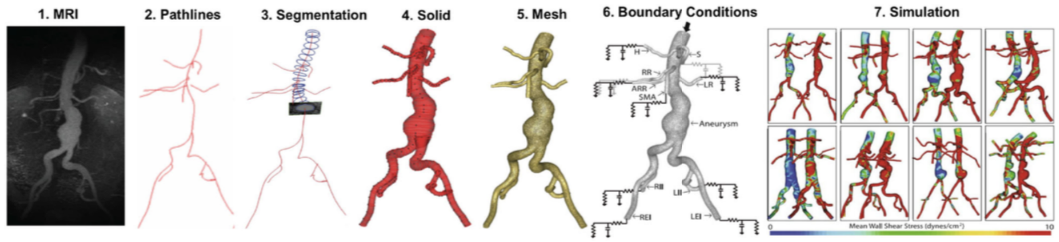


Figure 1: Pipeline for vascular modeling and simulation.[26]

This section is organized according to the steps in pipeline subsequently. In section 2.1, the graphical user interface (GUI) of SimVascular's visualizing tool is introduced. Section 2.2 then describes the how to generate a path from discrete points to a line with interpolation. Section 2.3 compares several segmentation techniques in SimVascular, especially give a brief review for the level set method in sections 2.3.1 and 2.3.2. With 2D segmented contours, section 2.4 explains lofting surface construction and enumerates surface smoothing techniques. Next, section 2.5 reviews role of mesh in simulation and common mesh generation methods such as Delaunay Triangulation. Finally, in section 2.6, the method for prescribing the analytic Womersley boundary profile is described.

2.1. Image Visualization

Typical medical image data is a set of intensity values defined on a 3D structured grid. It represents different physiology organs and tissues with different intensity. Visualization is the most common application for medical image data. In SimVascular, incooperating with VTK[23,24] and MTK libraries,[25] medical images such as magnetic resonance imaging (MRI) and computed tomography (CT) scans can be displayed with a multi-window tool in 2D/3D views as Figure 17.

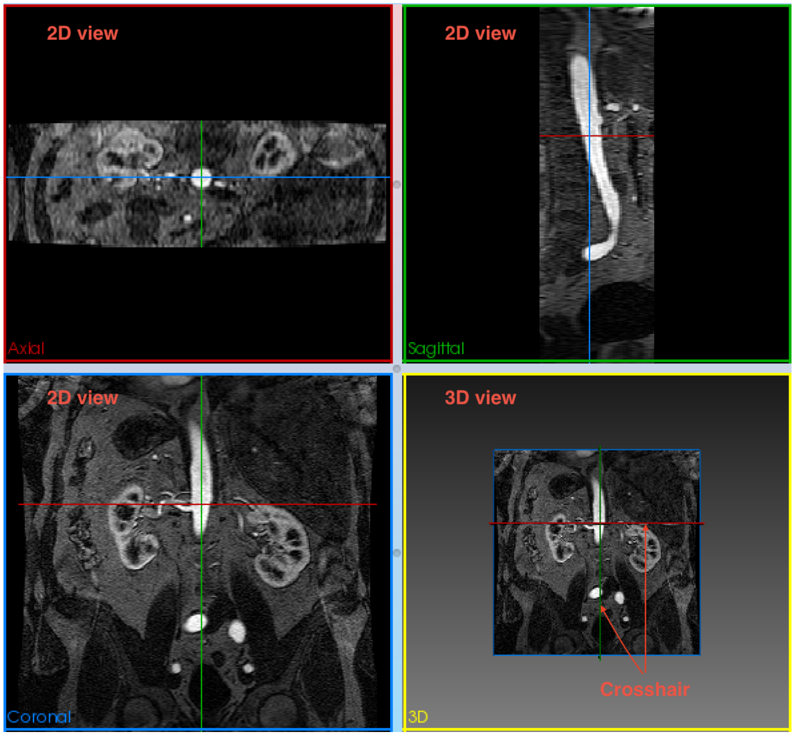


Figure 2: Display window in SimVascular.

There are three 2D views and one 3D view on the screen. The demo data used in Figure 17 is MRI of the aorta and the iliac bifurcation. In the Axial 2D view (upper left), the most lightening disk is the cross section of aorta. The following steps including path planning and image segmentation are based on these areas in 2D views.

2.2. Path Planning

In order to create 3D geometry vessel model, defining paths along vessel center-lines need to be done at first. Path construction includes two sets of ordered points as shown in Figure 3. One set consists of a small number "control points", used to calculate an interpolating line. The other set is made up of plenty of ”path points”, resampled from the calculated spine and the describe the path.

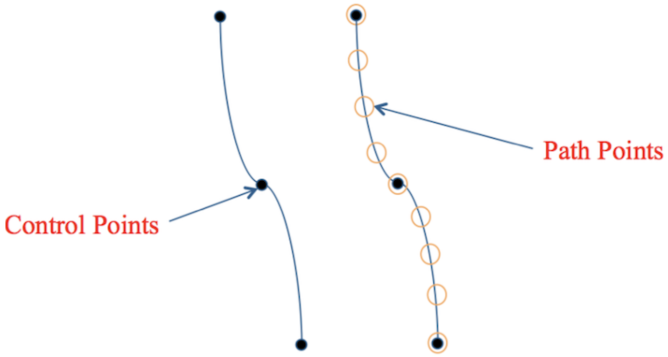


Figure 3: Two types of points in Path construction.[27]

With three 2D views in Figure 17 displayed simultaneously, we can approximately locate the center points for the vessel lumen and add them as control point. The detailed operation will be discussed in section 3.1.2.

After selecting a series of control points, a spline is created with interpolation. The choices of basis used in spline interpolation is various. Here we consider interpolation with cardinal basis (CBI)[29] and give a brief introduction on it in section 2.2.1.

After path is created, we can make modifications to the control points, moving and deleting them through the 2D windows, to ensure the good quality of path lines. Creating path lines is hard to automated and requires human time and expertise. The distance of control points is crucial and would affect a series of subsequent steps. When the distance between control points is too short, it would induce kinks in the path(Figure 4(a)). On the other hand, if the distance is too long, control points are not capable of catching the special feature of vessel shape such as bends or tortuosity accurately(Figure 4(b)). In addition, when the created path is jagged, we can apply Fourier smoothing on it. A quality path is significant for lofted 2D segmentation along pathline.

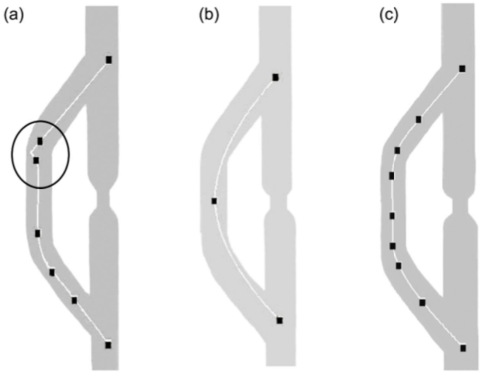


Figure 4: Impact of control points' distance on model.[28]

2.2.1 Interpolation with Cardinal Basis

Consider we have a set of distinct control points $S_n = \{x_i,\ i = 1, ..., n\}$ in the space $D \subset \mathbb{R}^3$ corresponding to MRI, with associated real values $\{f_i,\ i = 1, ..., n\}$, and a linear space $\phi(D)$, spanned by continuous basis function $g_i : D \to \mathbb{R}$, $(j = 1, ..., n)$. The multivariate linear interpolation problem at scattered data lines in finding $F \in \phi(D)$ such that

$$F(x_i) = \sum_{j=1}^{n} a_j g_j(x_i),\ (i = 1, ..., n). \tag{1}$$

The points x_i are the nodes and the f_i are the function values. Linear equations (1) are uniquely solvable if and only if the $n \times n$ interpolation matrix $A = g_j(x_i)$, $(i, j = 1, ..., n)$ is nonsingular. We use basis functions depend on the nodes themselves, which guarantees the property of unisolvence. CBI selecting continuous "cardinal functions" $\chi_j : D \to \mathbb{R}$, $(i = 1, ..., n)$, such that

$$\chi_j(x_i) = \delta_{ij}, \quad (i = 1, ..., n), \tag{2}$$

where δ_{ij} is the Kronecker delta operator. The interpolation function F can be rewritten as

$$F(x) = \sum_{j=1}^{n} f_j \chi_j(x). \tag{3}$$

The function (3) is the basic form of CBI. Next we use a simple and efficient procedure suggested by Cheney[30] to obtain CBIs.

Consider a real function α satisfying conditions as follows:

$$\alpha(x, y) > 0, \text{ if } x \neq y; \;\; \alpha(x, x) = 0; \; \forall x, y \in D. \tag{4}$$

Define the function χ_j by the equations

$$\chi_j(x) = \frac{\prod_{k=1,k\neq j}^{n} \alpha(x, x_k)}{\sum_{j=1}^{n} \prod_{k=1,k\neq j}^{n} \alpha(x, x_k)}, \tag{5}$$

Replace the $\chi_j(x)$ in 3 and simply it we can acquire

$$F(x) = \sum_{j=1}^{n} f_j \frac{1/\alpha(x, x_j)}{\sum_{j=1}^{n} 1/\alpha(x, x_j)}, \; (i = 1, ..., n) \tag{6}$$

For most application it is convenient to consider $\alpha(x, y)$ as a function of distance $d(x, y)$ defined on $D \in \mathbb{R}^3$, $i.e. \alpha(x, y) = \phi(d(x, y))$. In application of constructing path of vascular model, it's not proper to define $\alpha(x, y)$ as a function of distance. Other factors such as direction[31]also needed to be considered.

2.3. Image Segmentation

In the loft 2D segmentation approach, similar to path, the created contours also defined by two kinds of order points, control points and contour points. The contour points are calculated by the closed interpolation spline from control points. In application, user can translate or scale the contour by moving the control points. The location and shape of contour directly influence the quality solid model. Like the space of control points distributed in, the distance between ordered contour too short or too long would bring problems. Wang[32] explained that when two contours are too far, features of the geometry structure can't be represented well (Figure 5(c)). On the other hand, when their location are so close, they would intersect with each other. In this case, the generated geometry model is not physical since the wall of vessels are not likely to be folded and intersect with itself. Even if there doesn't occur intersection, the close space would make it difficult to satisfy the continuous constraints for surface smoothness.

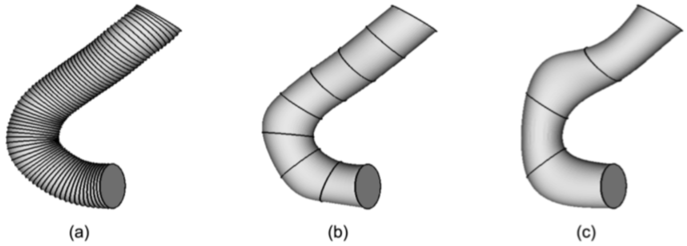


Figure 5: Impact of contours' space on the model.[28]

Not only improper contour space, but also low quality contour shape would cause distortion in geometry model. SimVascular offers several methods to create 2D contours, including thresholding,[36] level set method,[33, 34, 35] analytical methods, and manual methods. These methods are operated on a plane perpendicular to the path line. Threshold method is based on the image intensity values, which are assumed to be assigned centered at each pixel. A set of isocurves are created with bilinear interpolation function.[37] The contour is chosen from these isocurves. A circle specified radius is defined centered on the path. Then the contour of the lumen segmentation is supposed to be the smallest closed isocurve which completely encapsulates the circle. For the level set method, the basic idea is to set the initial contour, such as a small circle, and then grows in the direction of intensity gradient to find the location with sharpest intensity change. A brief review of level set method will be given later in sections 2.3.1 and 2.3.2. Analytical methods let users to obtain boundary of lumens with specific shape such as circle or ellipse, while manual methods enable users to get irregular contour shape by manually selecting points along the lumen, which will be automatically connected by a closed spline.

Among these methods, although threshold and level set methods can obtain contours close to lumen boundaries, more accurate then that generated by analytical methods, they focus more on intensity values or intensity gradient data then anatomy structure and result in

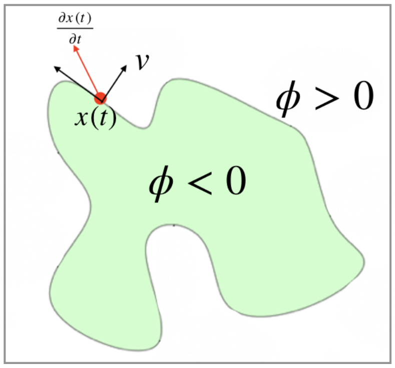


Figure 6: Mathematical representation in level set method.

jagged contours needed to smoothing processed. In addition, the contours lack of smoothness produces bad quality solid model and volume mesh. When our interest area is small and convoluted, these automated methods can't perform well as manually segmentation method. However, the contour quality generated by manually segmentation together with analytically method needed to estimate whether approximated to the real boundary. In general, users are supposed to choose segmentation method according to the feature of image with caution.

2.3.1 Level Set Method

Assume $x(t) = (x_1(t), x_2(t), ..., x_n(t)) \in \mathbb{R}^n$ to be the location of a point on the front, evolving over time. At any time t, every point $x(t)$ has no height, generating the formula:

$$\phi(x(t), t) = 0 \tag{7}$$

where $\phi(x(t), t) = 0$ can be any function as long as its zero root produces the contour. Take an example, ϕ is the distance from point $x(t)$ to the closet point on the contour. When $\phi(x(t), t)$ is positive(negative), point $x(t)$ is outside(inside) of the contour (see Figure 6). From equation (7), with the chain rules, we have

$$\frac{\partial \phi(x(t), t)}{\partial t} = 0$$

$$\frac{\partial \phi}{\partial x(t)} \frac{\partial x(t)}{\partial t} + \frac{\partial \phi}{t} \frac{t}{t} = 0$$

$$\frac{\partial \phi}{\partial x(t)} x_t + \phi_t = 0 \tag{8}$$

Recall that the gradient $\nabla \phi$ at point $x(t)$ is $\frac{\partial \phi}{\partial x}$. Notice that the gradient vectors at points on the contour are parallel to the plane. We can rewrite equation (8) as

$$\nabla \phi x_t + \phi_t = 0 \tag{9}$$

Since the velocity x_t is a vector, we can decompose it into two perpendicular subvectors on the plane with direction normal or tangent to the contour. Since the tangent subvector is orthogonal to $\nabla \phi$, their dot product is zero. This tangent subvelocity makes the contour spin but not change the shape. For the left normal subvelocity, parallel to $\nabla \phi$ governs the evolution of the contour, represented as vn, where v denotes the magnitude and $n = \frac{\nabla \phi}{|\nabla \phi|}$ is its direction. Hence the formula (9) is converted as

$$v \frac{\nabla \phi}{|\nabla \phi|} \nabla \phi + \phi_t = 0$$

$$\phi_t + v|\nabla \phi| = 0 \tag{10}$$

The implicit function (10) describes the motion of contour. Given initial state of ϕ at $t = 0$ and its (scalar) motion function $v(t)$, we can infer value of $\phi(x, t)$ at any point and time. The 2D contour is defined as the zero level set, the solution of $\phi = 0$.

Also we can get the surface curvature with ϕ. Take surface in 2D for an example, curvature of a planar contour is defined as the divergence of the normal vector,

$$\begin{aligned}
\kappa &= \nabla \cdot \frac{\nabla\phi}{|\nabla\phi|} \\
&= (\frac{\partial}{\partial x}, \frac{\partial}{\partial y})(\frac{\phi_x}{(\phi_x^2+\phi_y^2)^{1/2}}, \frac{\phi_y}{(\phi_x^2+\phi_y^2)^{1/2}}) \\
&= \frac{\phi_{xx}\phi_x^2 + \phi_{xx}\phi_y^2 - \phi_x^2\phi_{xx} - \phi_x\phi_y\phi_{yx} + \phi_{yy}\phi_y^2 + \phi_{yy}\phi_x^2 - \phi_y^2\phi_{yy} - \phi_y\phi_x\phi_{xy}}{(\phi_x^2+\phi_y^2)^{3/2}} \\
&= \frac{\phi_{xx}\phi_y^2 - 2\phi_x\phi_y\phi_{xy} + \phi_{yy}\phi_x^2}{(\phi_x^2+\phi_y^2)^{3/2}}
\end{aligned} \tag{11}$$

For surface in 3D, there are several kinds of curvature combined into level set method.[33,34]

2.3.2 Implementation Of Level Set

For the medical image with discrete pixels, the functions contained in the level set method need to be discretized. First, at some time point t^n, denote current values of ϕ as $\phi^n = \phi(t^n)$. After some time increment $\triangle t$, at $t^{n+1} = t^n + \triangle t$, we obtain updated values of ϕ and denote it as ϕ^{n+1}. When $\triangle t$ is sufficiently small, ϕ_t^n can be evaluated as

$$\phi_t^n = \frac{\phi^{n+1} - \phi^n}{\triangle t} \tag{12}$$

Next, consider one dimensional version of the equation (9) at point x and rewrite it as

$$\frac{\phi_i^{n+1} - \phi_i^n}{\triangle t} + u_i^n(\phi_x^n)_i = 0 \tag{13}$$

If $u_i^n > 0$, the values of ϕ moves from left to right, with first-order upwind scheme,[38,39] ϕ_x^n can be approximated by $D^-\phi = \frac{(\phi_i^n)-(\phi_{i-1}^n)}{\triangle x}$. Conversely, if $u_i^n < 0$, the values of ϕ moves from right to left, $D^+\phi = \frac{(\phi_{i+1}^n)-(\phi_i^n)}{\triangle x}$ is used to estimate ϕ_x^n. Notice that $u_i^n = u \cdot \frac{\phi_x}{|\phi_x|}$.

Hence the above two cases can be converted as

$$\phi_x^n = \begin{cases} D^-\phi & \text{if } u > 0,\ D^-\phi > 0 \\ D^+\phi & \text{if } u > 0,\ D^+\phi < 0 \\ D^+\phi & \text{if } u < 0,\ D^+\phi > 0 \\ D^-\phi & \text{if } u < 0,\ D^-\phi < 0 \end{cases} \tag{14}$$

$$|\phi_x^n| = \begin{cases} [\max(D^-\phi, 0)^2 + \min(D^+\phi, 0)^2)]^{1/2} = \bigtriangledown_i^+ & \text{if } u > 0 \\ [\max(D^+\phi, 0)^2 + \min(D^-\phi, 0)^2)]^{1/2} = \bigtriangledown_i^- & \text{if } u < 0 \end{cases} \tag{15}$$

Hence for the one dimension level set equation (13), the first order finite difference can be

$$\phi_i^{n+1} = \phi_i^n + \triangle t(\max(u, 0) \bigtriangledown_i^+ + \min(u, 0)\bigtriangledown_i^-) \tag{16}$$

Similarly for two dimensional version, we have the approximation

$$\phi_{ij}^{n+1} = \phi_{ij}^n + \triangle t(\max(v, 0) \bigtriangledown_{ij}^+ + \min(v, 0)\bigtriangledown_{ij}^-) \tag{17}$$

where

$$\bigtriangledown_{ij}^+ = [\max(D^{-x}\phi_{ij}^n, 0)^2 + \min(D^{+x}\phi_{ij}^n, 0)^2 + \max(D^{-y}\phi_{ij}^n, 0)^2 + \min(D^{+y}\phi_{ij}^n, 0)^2]^{1/2} \tag{18}$$

$$\bigtriangledown_{ij}^+ = [\max(D^{+x}\phi_{ij}^n, 0)^2 + \min(D^{-x}\phi_{ij}^n, 0)^2 + \max(D^{+y}\phi_{ij}^n, 0)^2 + \min(D^{-y}\phi_{ij}^n, 0)^2]^{1/2} \tag{19}$$

In addition, in order to create a relatively smooth contour, a penalized term of high curvature is added, that is

$$\phi_{ij}^{n+1} = \phi_{ij}^n + \triangle t(\max(v, 0) \bigtriangledown_{ij}^+ + \min(v, 0)\bigtriangledown_{ij}^-) + \triangle t[\kappa_{ij}] \tag{20}$$

2.4. Solid Modeling

With segmented 2D contours, a boundary representation (B-Rep) is created along the path by lofting the ordered 2D segmentation.

2.4.1 Lofting Surface

Here we introduce the general idea of surface lofting procedures. Our review starts with some basic conception including B-spline curves[41,42] and B-spline surface.[40]
A pth-degree B-spline curve $C(u)$ is defined by

$$C(u) = \sum_{i=0}^{n} N_{i,p}(u)\mathbf{p_i} \tag{21}$$

where $\mathbf{p_i}$ are control points, $N_{i,p}(u)$ are the pth-degree B-spline functions defined on the nonperiodic (and nonuniform) knot vectors $\mathbf{U} = \{u_0, u_1, ..., u_{n+p+1}\}$, which is

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u \in [u_i,\, u_{i+1}] \\ 0 & \text{otherwise} \end{cases}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \tag{22}$$

Now suppose we have a set of points $\{Q_k\}$, $k = 0, ..., n$ and want to obtain a pth-degree B-spline curve interpolated by these points. From equation (21), since the parameter value u is undetermined, we assign a set of $\bar{u}_k$ to every Q_k. Also we need to set up knot vectors $U = \{u_0, ..., u_m\}$ to construct the B-spline functions. Based on these set-up, our interpolation problem is converted to solving a $(n+1) \times (n+1)$ system of linear equation,

$$Q_k = C(\bar{u}_k) = \sum_{i=0}^{n} N_{i,p}(\bar{u}_k)\mathbf{p_i} \tag{23}$$

where the control points $\mathbf{p_i}$ are unknown.

For determining $\bar{u}_k$, three methods including equally spaced method, chord length method and centripetal method are used widely. Chord length method is the most widely used method. Let d denote the total chord length

$$d = \sum_{k=1}^{n} |Q_k - Q_{k-1}| \tag{24}$$

Then let $\bar{u}_0 = 0$, $\bar{u}_n = 1$, and

$$\bar{u}_k = \bar{u}_{k-1} + \frac{|Q_k - Q_{k-1}|}{d}, \quad k = 1, ..., n-1 \tag{25}$$

From,[40] the equally spaced method is not recommended, since it can produce wired shapes like loops when data points are not evenly spaced. Although the chord length method can generate a uniform parameterization of curve, the centripetal method performs better when the data turns sharply in the space.

Next knot vectors are selected with averaging method, that is,

$$\begin{aligned} &u_0 = u_1 = ... = u_p = 0 \qquad u_{m-p} = ... = u_m = 1 \\ &u_{j+p} = \frac{1}{p} \sum_{i=j}^{j+p-1} \bar{u}_i \qquad j = 1, ..., n-p \end{aligned} \tag{26}$$

Knot vectors defined with averaging method reflect the distribution of $\bar{u}_k$. With the above variables, we can compute $\mathbf{p_i}$ in equation (23).

A B-spline surface can be generated from a bidirectional net of $(n+1) \times (m+1)$ control points and two of knot vectors,

$$S(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,p}(u) N_{j,q}(v) \mathbf{p_{i,j}} \tag{27}$$

where u and v are two independent parameters and $\mathbf{p_{i,j}}$ are control points. $N_{i,p}(u)$ and $N_{j,q}(u)$ are pth-degree and qth-degree B-spline basis functions. The lofting method for free form surface modelling in current CAD systems is to construct a skinned surface $S(u,v)$ passing through a set of curves $C_0(u), C_1(u), ..., C_n(u)$ so that $S(u, v_i) = C_i(u)$ for $0 = v_0 <$

$v_1 < ... < v_n = 1$.

To implement the lofting method, we assume all curves have a common degree p and knot vector U. If not, we can adopt degree elevation and knot insertion algorithms.[40] As long as the v direction and degree p is chosen, we can calculate the parameters $\bar{u}_k$ with chord length method and knot vectors U with averaging method, which are used in interpolating B-spline curve for given data. Let $\mathbf{q_{ij}}$ be the i-th control point for j-th curve $C_j(u)$ and $\mathbf{p_{ij}}$ be the ij-th control point for the lofted surface $S(u, v)$. Then with the B-spline curve function on $C_j(u)$the parameterization, the points on the curves and lofted surface can be represented as

$$\begin{aligned} S(u, v_j) &= \sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,p}(u) N_{j,q}(v_j) \mathbf{p_{ij}} \\ &= \sum_{i=0}^{n} N_{i,p}(u) \sum_{j=0}^{m} N_{j,q}(v_j) \mathbf{p_{ij}} \\ &= \sum_{i=0}^{n} N_{i,p}(u) \mathbf{q_{ij}} \\ &= C_j(u) \end{aligned} \tag{28}$$

From the above equation (28), the control points $\mathbf{p_{ij}}$ can be calculated by solving the linear equations shown below:

$$\sum_{j=0}^{m} N_{j,q}(v_j) \mathbf{p_{ij}} = \mathbf{q_{ij}}$$

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ N_{0,q}(v_1) & N_{1,q}(v_1) & N_{2,q}(v_1) & \cdots & N_{n,q}(v_1) \\ N_{0,q}(v_2) & N_{1,q}(v_2) & N_{2,q}(v_2) & \cdots & N_{n,q}(v_2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \vdots & 1 \end{bmatrix} \begin{bmatrix} \mathbf{p_{i0}} \\ \mathbf{p_{i1}} \\ \mathbf{p_{i2}} \\ \vdots \\ \mathbf{p_{in}} \end{bmatrix} = \begin{bmatrix} \mathbf{q_{i0}} \\ \mathbf{q_{i1}} \\ \mathbf{q_{i2}} \\ \vdots \\ \mathbf{q_{in}} \end{bmatrix} \tag{29}$$

Therefore, with the solution of control points $\mathbf{p_{ij}}$, we get the loft surface $S(u, v)$.

2.4.2 Boolean Operation

Boolean operation is useful in vascular modeling with special shape such as branch. There are variety of Boolean implementation methods for B-Reps and can be roughly divided into four categories, volumetric methods, interval computation methods, exact arithmetic methods and approximate arithmetic methods.[43]

The Boolean operations between two object A and B include their union ($A \cup B$), intersection ($A \cap B$) and difference ($A \setminus B$ or $B \setminus A$). To get the Boolean operation result, we first find the intersection loops, where surface A crosses surface B. Then we separate these two objects into several portions by the intersection loops. Finally, we determine the appropriate combination of the separated sub-surfaces to obtain the Boolean results.

2.4.3 Surface Smoothing

For the surface obtained from Boolean operations, it may have sharp angles where two objects union, which is important to resolve as it can lead to bad meshes for CFD. So surface is an important step. Also, discrete B-rep surface may contain roughness since they are generated from discrete image data which analytic representation is not available. Then a series of surface manipulation operations, including smoothing, decimation and subdivision,[47] are needed to be done later. The main purpose of smoothing is to reduce the difference between two adjacent triangles in surface mesh. Decimation denotes the process of reducing the number of triangles in the surface mesh to obtain a simpler mesh or improve the mesh quality by removing triangles with bad quality. Oppositely, subdivision process is to increase the number of cells by splitting some of them into several smaller subcells. This process produces a finer mesh than the original one, containing more polygonal cells and influence the subsequent simulation process.

SimVascular contains smoothing methods such as Laplacian smoothing[44] and constrained smoothing.[?] A typical Laplacian smoothing method moves a point to a new location by average the points connected this specific point, that is,

$$\bar{x}_i = \frac{1}{n}\sum_{j=0}^{n} \bar{x}_j \tag{30}$$

where n is the number of the nodes sharing edge with node i, $\bar{x}_i$ is the new coordinate of node i and $\bar{x}_j$s are the coornidates of the adjacent nodes j. After iterations, the point moves closer to its neighbor nodes. Although it decreases roughness to, it shrinks the volume of whole geometry. The shrinkage problem causes inaccuracy in the blood simulation. To overcome the shrinkage problem, the constrained smoothing method[46] developed in work on watershed ridges minimizes the error between the original mesh and the Laplacian smoothed mesh. For each iteration (30), a constrained equation is added. The location of a smoothed point on the surface becomes the minimum of two equations in an optimization problem. The iteration becomes,

$$\begin{aligned}
&\bar{x}_i - (\bar{x}_{original} + w) = 0 \quad \text{where} \quad w = \|\bar{x}_i - \bar{x}_{original}\| * w_{user} \\
&\bar{x}_i - \frac{1}{n}\sum_{j=0}^{n} \bar{x}_j = 0
\end{aligned} \tag{31}$$

where w_{user} is a weight coefficient ranging from 0 to 1 to penalize the deviation between new coordinate and the old one. The matrix form of the equations (31) is

$$\underbrace{\begin{bmatrix}
1 & 0 & 0 & \cdots & \cdots & \cdots & 0 \\
0 & 1 & 0 & \ddots & & & \vdots \\
0 & 0 & 1 & \ddots & \ddots & & \vdots \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
\vdots & & \ddots & \ddots & 1 & 0 & 0 \\
\vdots & & & \ddots & 0 & 1 & 0 \\
0 & \cdots & \cdots & \cdots & 0 & 0 & 1
\end{bmatrix}}_{A_1}
\begin{bmatrix}
x_0^x \\ x_0^y \\ x_0^z \\ \vdots \\ x_m^x \\ x_m^y \\ x_m^z
\end{bmatrix}
=
\underbrace{\begin{bmatrix}
x_{original}^x + w_0 \\ x_{original}^y + w_0 \\ x_{original}^z + w_0 \\ \vdots \\ x_{original}^x + w_0 \\ x_{original}^y + w_0 \\ x_{original}^z + w_0
\end{bmatrix}}_{b_1} \tag{32}$$

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 & -\frac{1}{n} & \cdots & \cdots & \cdots \\ 0 & 1 & 0 & 0 & -\frac{1}{n} & \cdots & \cdots \\ 0 & 0 & 1 & 0 & 0 & -\frac{1}{n} & \cdots \end{bmatrix}}_{A_2} \begin{bmatrix} x_0^x \\ x_0^y \\ x_0^z \\ \vdots \\ x_m^x \\ x_m^y \\ x_m^z \end{bmatrix} = \underbrace{\begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}}_{b_2} \tag{33}$$

Here we assume each coordinates has 3 components and the total degree of free of this linear system is 3*m, m is the number of nodes in the mesh. In order to solve equation (32) and (33), we combine them together as $A = [A_1; A_2]$ and $b = [b_1; b_2]$. With conjugate gradient method, the problem of solving $Ax = b$ is converted as solving $A^T Ax = A^T b$.

2.5. Mesh Generation

When the solid model is created, it is discretized into small pieces with simple shape, such as triangles or quadrilaterals in two dimensions and tetrahedron or hexahedron in three dimensions. The discretized model is a "mesh" and the process of generating discretized model is "mesh generation". Meshes are widely used in many applications, including geography, cartography, computer graphics *et.al.*. In our thesis, we mainly briefly review meshes' important roles in numerical solution of differential equations in physical simulation. We start with the introduction of finite-element method to present the purpose of mesh.

2.5.1 Finite-Element Method

There are various methods to discrete approximation of partial differential equations. One popular method in is finite-element method. The general idea of this method is to replace the unknown function we seek with finite-dimensional approximations. Assume the partial differential equation as

$$\mathcal{L}u = f \tag{34}$$

where u is the unknown function of position and possibly also of time, $\mathcal{L}$ is the linear differential operator on the domain ω, and f is the given function we know. We see an approximate solution of (34) on some geometric domain Ω. The weak form is obtained by multiplying a test function and do integration on both sides of equation (34). That is for some test of test functions ψ_i, the solution $u(x)$ should satisfy

$$\int_\Omega (\mathcal{L}u)\psi_i = \int_\Omega f\psi_i \tag{35}$$

The finite-element method replaces the continuous function $u(x)$ with finite-dimensional approximation $\bar{u}(x) = \sum_{k=1}^{n} a_k\phi_k(x)$, where $\phi_k(x)$ are the basic functions, typically selected as lower-order polynomials. Then the results of $\mathcal{L}\phi_k$ can be calculated easily. Function f can also be approximated by basis functions ϕ_k can denoted as $\bar{f}$. The problem is converted as finding the values of a_k.

With finite-element method, the weak form is rewritten as

$$\begin{aligned} \int_\Omega (\mathcal{L}\bar{u})\psi_i &= \int_\Omega \bar{f}\psi_i = f_i \\ \int_\Omega (\mathcal{L}\bar{u})\psi_i &= \sum_{k=1}^{n} a_k \int_\Omega (\mathcal{L}\phi_k)\psi_i \end{aligned} \tag{36}$$

The efficiency of the method is related to the size of the set I_{ik}, consisting of different values of parameters x where both $\phi_k(x)$ and $\psi_i(x)$ are nonzero, which size is usually small. Then the computation is reduced to

$$A_{ik} = \int_\Omega (\mathcal{L}\phi_k)\psi_i = \sum_{e\in I_{ik}} \int_e (\mathcal{L}\phi_k)\psi_i \tag{37}$$

With computed f_i and A_{ik}, we can build a liner system from (36) to find the solution of unknown a_k and approximated $\bar{u}$. The mesh quality of domain Ω has a large influence on the accuracy and efficiency of the simulation based on a solution of the differential equations.[48]

2.5.2 Unstructured Mesh

The type of mesh can be roughly divided as three categories: structured mesh, unstructured mesh and hybrid mesh (see Figure 7). A structured mesh has vertices that are connected in a regular pattern, whereas a unstructured mesh shows irregularity. The hybrid mesh is a combination of small number of structured meshes and overall uninstructed pattern. Although structured meshes take advantages in requiring less memory for data storage and saving time for computation work in PDE solving, unstructured meshes are more flexible in fitting complicated geometry domains. Thus for arbitrary geometric models, unstructured meshes are more likely to be used. Numerous automated methods have been developed for unstructured mesh generation, including Delaunay triangulation,[52] constrained Delaunay triangulation, and quadtrees(2D)/octrees(3D) method. Here we give a brief introduction to Delaunay triangulation, one of the most common methods.

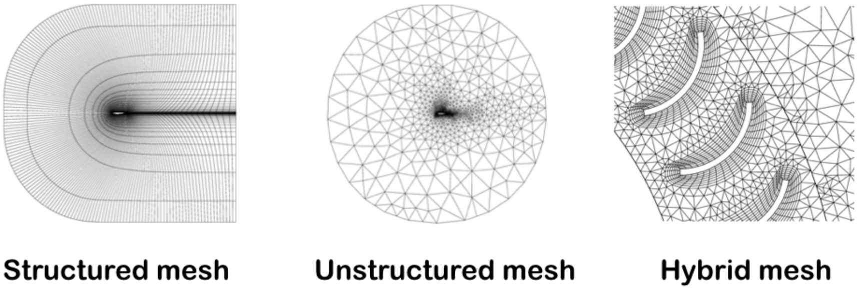


Figure 7: Three types of mesh.

2.5.3 Delaunay Triangulation

For the application of Delaunay Triangulation in two dimensions unconstruted mesh generation, it consists of two phases: placement of Steiner points, followed by triangulation. In the placement phase, Steiner points denote points added into the set of vertices of the input domain, which doesn't refer to a specific location. The placement phase first places Steiner points along the boundary of domain then places interior points. The number of

Steiner points on the boundary of domain Ω should be sufficient to make sure the resulting mesh accord to the original domain. One approach to place Steiner points advancing front method, adding points in successive layers from the boundary of domain. Notice that many advancing front methods can generate triangle meshes themselves rather than only Steiner points.The triangulation phase follows Delaunay triangulation property. That is given a set of points $S = \{s_1, s_2, ..., s_n\}$, the circumcircle of any triangle (s_i, s_j, s_k) doesn't enclose any other points in S. There are lots of Delaunay triangulation algorithms and we describe one of them called edge flipping algorithm, which starts from any triangulation of S and then optimizes each edge locally. Let e be any internal edge (not on the boundary) in the triangulation and Q_e be the triangulated quadrilateral generated by two triangles which share edge e. We say Q_e is reversed if each triangle's circumcircle contains the opposite vertex. If Q_e is reversed, then we change edge e for the another diagonal edge in Q_e. Algorithm 1 presents the pseudo code for the edge flipping algorithm.

Algorithm 1 Pseudo code for edge flipping algorithm

```
generate an initial triangulation of $S$
construct a list consisting of all the internal edges
while the list is not empty do
    remove the first edge $e$ in the list
    if quadrilateral $Q_e$ is reversed then
        flip it and add the outside internal edges of $Q_e$ to the list
    end if
end while
```

For Delaunny Triangulation in three dimensions, the above point placement menthods suffer from problem, that is the generated mesh might contain sliver tetrahedra with bad shape since the circumsphere of a sliver isn't much larger than sliver's edges. Thus some Delaunay mesh generation methods have extra postprocessing steps to remove slivers, which have bad effect in numerical simulation. Also the generation of using edge flipping algorithm directly to three dimensions has difficulties since not any tetrahedralization can be transformed into Delaunay triangulation by flips. However, there are other methods based on edge flipping idea, such as incremental Delaunay triangulation algorithms, work well. These algorithms use add one point in the triangulation, and split the tetrahedron which encloses the new point into four pieces by connecting new point to the four vertices. Then flip the edge to a new Delaunay triangulation. Usually the initial bounding tetrahedron consists of four dummy points which will be delete in the end together with their incident tetrahedra.

2.5.4 Mesh Quality

After finish meshing process, evaluation mesh quality is necessary since the quality of mesh influences the accuracy of results and running time.[48] For the exterior surface mesh, we can roughly judge it in a visual inspection. However, it's not practical to visualize and evaluate large number of interior tetrahedral mesh element. To overcome it, Geometric-based mesh indicators are used. Here we introduce two indicators: radius ratio and edge ratio. The radius ratio, suggested by Cavendish *et.al.*,[49] is the radio of radius of the maximum inscribed circle (sphere in 3D) over the radius of the circumscribe circle (sphere in 3D). The radius ratio of equilateral element shape is 2. The edge ratio is the ratio of the maximum length of the edge to the minimum length of the edge of the element. Edge ratio is always greater than or equal to 1 (equilateral element shape). Higher the value of edge ratio, the shape of the element is less regular with worse quality. These indicators can be visualized and inspected in Paraview software.

2.5.5 Mesh processing

In order obtain mesh with good quality, a series of mesh processing procedures, including remeshing and smoothing, are needed to be done. Remeshing is an important technique to reduce the complexity of a surface mesh or improve the quality of mesh for better performance in application. For example, as shown in Figure 8, the quality of surface mesh on the left side is unsatisfactory. Its elements are not isotropic, highly skewed, which would cause errors in simulation process. After remeshing, the right picture in Figure 8 improves mesh quality dramatically.

Besides remeshing, Smoothing techniques are also key to capture geometric information.

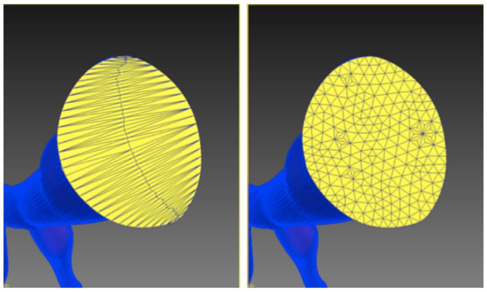


Figure 8: An example of Remesh tool in Simvascular.[51]

One of the smoothing techniques, which is based on polygon interpolation, named as Spherigon,[50] is capable in removing polygon aspect of rough polygonal meshes to smooth the shape. A brief description of this method in application of triangulation mesh is provided below. Consider having a triangle element T defined by the vertices P_i, $i = 1, 2, 3$ and each vertex is assigned with a normal direction N_i, $i = 1, 2, 3$. For each point P on the polygon surface T (see in Figure 9), its location is represented as $P = \sum_{i=1}^{3} r_i P_i$. The

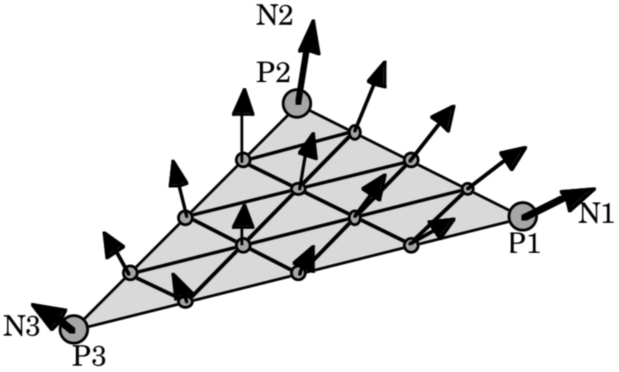


Figure 9: Interpolating positions and normals on the triangle. T^{50}

corresponding normal direction, also named as Phong normal, at this point is defined as $N = (\sum_{i=1}^{3} r_i N_i)_{Normalized}$. For each vertex P, the point Q projected on curved surface from P is created by moving P along its normal direction N. We then compute the displacement contribution related to each vertex P_i. For better explanation, we provide a geometric plot in Figure 10. Q_i is on the line of P with direction of N and also constructs a circular arc together with P_i. This circular arc is orthogonal to N_i at P_i and orthogonal

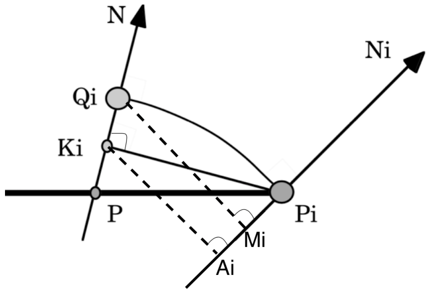


Figure 10: The construction of the vertex contribution for a given point.[50]

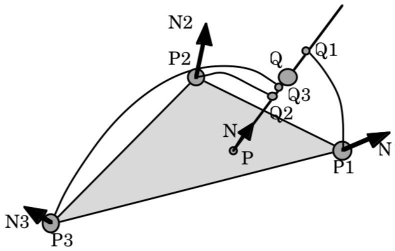


Figure 11: Interpolating the vertex contributions.[50]

to N at Q_i.

The first step is to project P_i onto N as K_i. Hence K_i is computed as

$$K_i = P + ((P_i - P) \bullet N) \tag{38}$$

To obtain Q_i, we project K_i and Q_i on to N_i as A_i and M_i. Because of symmetry, $|\overrightarrow{K_iQ_i}| = |\overrightarrow{M_iP_i}|$. Then $|\overrightarrow{A_iP_i}| = |\overrightarrow{K_iQ_i}| + \overrightarrow{K_iQ_i} \bullet N_i$. So Q_i is calculated with

$$Q_i = K_i + \frac{(P_i - K_i) \bullet N_i}{1 + N \bullet N_i} N \tag{39}$$

With all points Q_i in Figure 11, a point Q is generated by averaging these Q_i with weight normalized blending functions f_i, that is

$$Q = \sum_{i=1}^{3} f_i(r) Q_i, \quad r = (r_1, r_2, r_3) \tag{40}$$

With this algorithm, the interpolated point Q of any point P on the polygon T can be computed. The interpolation surface would be like Figure 12.

2.6. Boundary Conditions

With the generated unstructured mesh, blood flow simulations are performed using Nektar++.[54] Many parameters are needed to be calculated and reset before simulation. Before

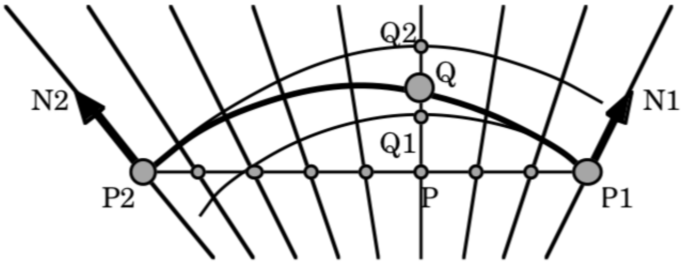


Figure 12: The interpolation surface.[50]

giving explanation in how to set them up, we give some physiologic information about the simulation problem in Figure 13.

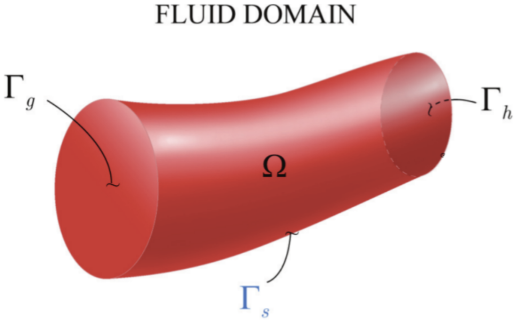


Figure 13: Structure of a simple cylindrical vessel.[56]

In spite of the complicate geometrically shape of a vascular model, its boundary layers can be classified into three groups as shown in Figure 13. Inlet (inflow boundary) Γ_g is the set of face(s) of the vascular model where a blood flow wave, detected from clinical treatments, is prescribed. Usually, the blood flow is regarded as pulsatile flow,[53] or a Womersley flow, since it has periodic variations. The wall boundary Γ_s refers to interface between vessel wall and the fluid domain Ω. In most cases, the wall is assumed to be rigid and a zero velocity condition is assigned to the wall layer. The other boundary layer is outlet (outflow boundary) Γ_h, where a pressure value is uniform prescribed.

2.6.1 Inflow boundary conditions

The boundary conditions are essential to obtain high quality simulation results. For the inlet boundary condition associated with velocities, Womersley velocity profile is defined for pulsatile flow in Nektar++ by using the following formation:

$$w(r,t) = A_0(1-(r/R)^2) + \sum_{n=1}^{N} \tilde{A}n[1 - \frac{J_0(i^{3/2}\alpha_n r/R)}{J_0(i^{3/2}\alpha_n}]e^{iw_n t} \tag{41}$$

where radius r is the radius, t is time and R is the radius of the inlet. J_0 is the Bessel function, and $\alpha_n = R\sqrt{2\pi n/T\nu}$ are the Womersley numbers with period T and kinematic viscosity ν. $\tilde{A_n}$ are the coefficients obtained by scaling Fourier coefficients A_n of the desired flow waveform in the following way:

$$\tilde{A_n} = 2A_n/[1 - \frac{1}{J_0(i^{3/2}\alpha)}] \tag{42}$$

2.6.2 Windkessel models

For the aortic outlet, a Windkessel[55] RCR boundary condition will be specified. In 1899, German physiologist Frank first derived the Windkessel model.

A Windkessel model (see Figure 14) mainly consists of three elements: a proximal resistance R modeling the viscous resistance of arterial vasculature, a capacitance C modeling the vessel compliance of downstream vasculature and a distal resistance R_d modeling the resistance of the capillaries and venous circulation. P_d is a downstream pressure varying as

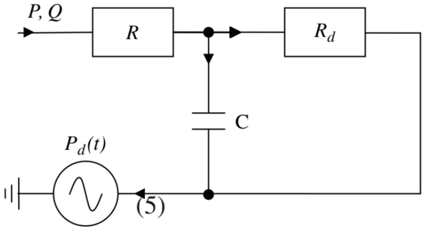


Figure 14: RCR Windkessel model electric analogue

a function of a time. With time varying, the pressure P from $t = 0$ can be calculated by

$$P(t) = [P(0) - RQ(0) - P_d(0)]e^{-t/\tau} + P_d(t) + RQ + \int_0^t \frac{e^{-(t-\tilde{t})/\tau}}{C} Q(\tilde{t}) d\tilde{t} \tag{43}$$

where $\tau = R_d C$. Three parameters R,R_d and C are tuned to let resulted pressure P accommodate with patient-specific physiologic conditions. The tuning is not a trivial task. The work of Chen and Quarteroni[57] is also interesting as they study the sensitivity of the predicted results on the R, C parameters.

3. Results

In this section, we give a brief tutorial in how to visualize MRI data, create a model in SimVascular, how to generate mesh in Gmsh, and how to write boundary condition file in Nektar++ for a realistic case.

3.1. Geometric Modeling

The geometry of vascular model we build consists of a descending aorta, left/right common iliac arteries, left/right external iliac arteries and left/right uterine arteries (UtAs), which is similar to that in the Figure 15. With the vascular model, we are interested in studying the uteroplacental (maternal-placental) circulation,[58] referring the maternal blood flow starts from UtAs, through arcuate arteries, radial arterials and spiral arteriesa and finally into placental intervillous space. The purpose of this study is trying to understand the biological mechanism of hypertensive pregnancy disorders (HPD) development, a most common medical problem occuring during pregnancy.

Pregnant women with HPD are at high risk of maternal and fetal morbidity and mortality. One typical clinical manifestation of HPD is having exceeding high blood pressure. Unfortunately, when this symptom is detected, the optimal opportunity for treatment is lost. Therefore, by understanding the initiation and development of this high blood pressure, medical experts hope to find biomarkers of HPD in early gestation that is predictive of adverse pregnancy outcomes and use it to develop effective clinical treatments. Most people consider this high blood pressure initiates in the spiral arteries and placental intervillous space. However, at present, there is no clinical method enables people to measure pressure in these place since the conduits are too small. Clinicians can only measure blood velocity in the UtAs, upstream from placenta, with Doppler ultrasound technique. Then if the velocity has features such as dicrotic notch of the velocity and high pulsatility (large amplitude) of the waveform, clinicians would infer high blood pressure in UtAs. The standard clinical method to measure arteries is to put a catheter pressure sensor into a patient. But this

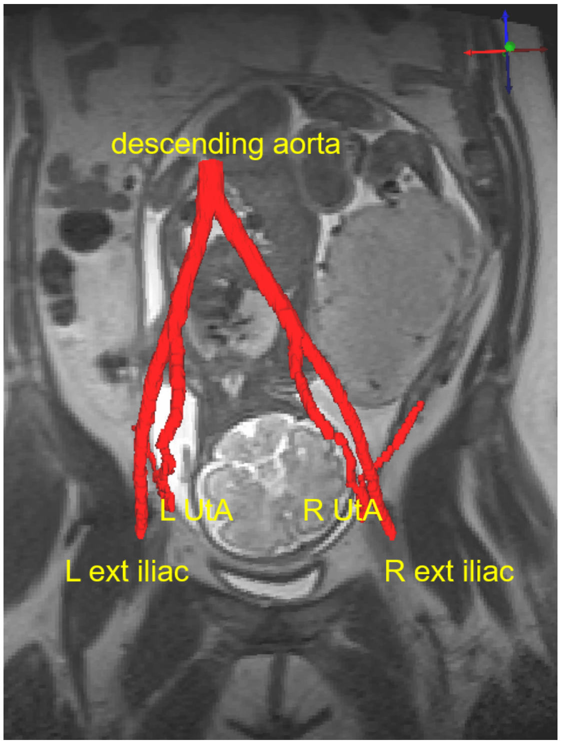


Figure 15: Geometry of vascular model.

method isn't suitable for a pregnant woman because of the risks of internal monitoring include maternal and fetal infection. Since the invasive method is unavailable, non-invasive method such as 4D flow MRI is used to measure the 3D velocity. With prior knowledge of Navier-Stokes equations, some assumptions and patient-specific inputs, pressure can be estimated with 4D flow MRI.

3.1.1 Image Data Visualization

Before displaying the medical image data, users are supposed to create a SimVascular (SV) project first. To create a project:

Menu: File → Create SV Project

Set project name and project source.

Click the button ’OK’.

Then the tree structure of the project would be displayed in **SV Data Manager** window.

Next, as shown in Figure 16, load the medical image data into the project by right clicking **Images** in **SV Data Manager** window and select **Add/Replace Image**.

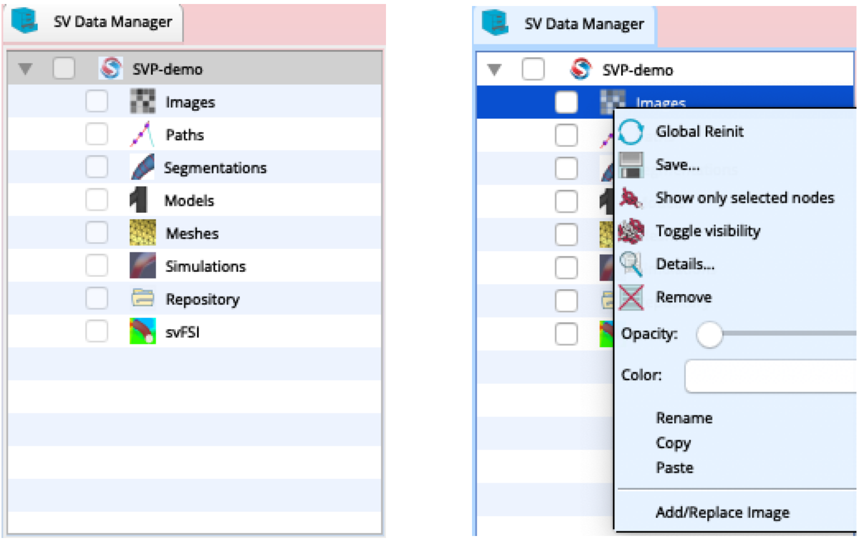


Figure 16: **SV Data Manager** window (left) and load image operation (right).

Then we obtain **Display** window as Figure 17. There are several common operations in adjust 2D/3D views in **Display**, which are widely used in path creation step. When place the mouse on a 2D view,

Move the crosshair by clicking left mouse button.

Zoom in (out) the 2D view by holding right mouse button and moving mouse ahead (back).

Change image slice in **Image Navigator** window by scrolling center mouse button.

We can also adjust the 3D view with the following operations:

Rotate the 3D volume with left mouse button.

Zoom in (out) the 3D view with right mouse button.

Translate the 3D volume with center mouse button (or 'shift'+left mouse button).

3.1.2 Path Creation

The geometry of the interested arteries has been introduce at the beginning of this section. The construction of vascular model starts with path creation explained in section**??**. We first construct the aorta, since it's the largest vessel in the image data we're interested in.

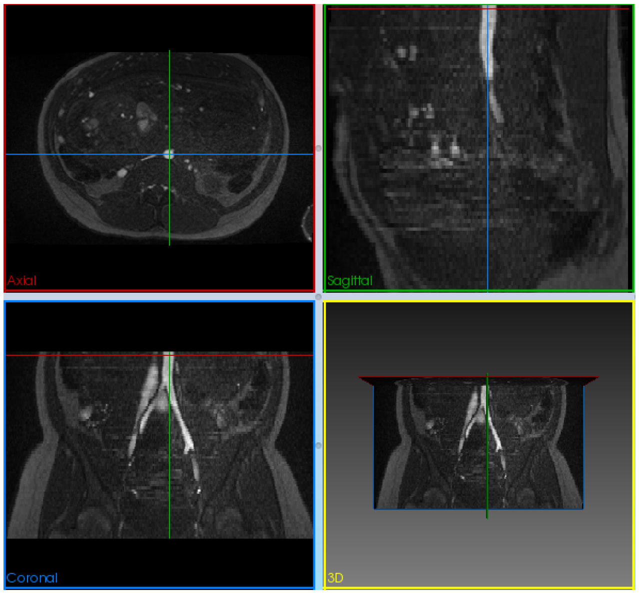


Figure 17: **Display** window of medical image data

To create a data node for storing information of aorta pathline:

Right click the **path** in the **SV Data Manager** window.

Select **Create Path** in the popup menu.

Path Name: aorta

Subdivison Type: **Spacing Based**

Click **Ok**.

Different **Subdivison Type** has different ways to set the number of path points. When choosing **Spacing Based** type, the **Spacing** s is animatedly filled with minimum image spacing. Assume l is the distance between two control points and the total number of control pointsis Nc, then the number of path points is Npa is calculated with $Npa = [l/s] * (Nc - 1) + 1$. Now a new data node **aorta** is added in the **SV Data Manager**. When click **aorta**, **SV Path Planning** tool pops up. To add a control point:

Image Navigator: Axial slider: 73 (the maximum value)

Place the cursor into the Axial 2D view and zoom it.

Move the cursor to the center of the aorta vesseS.

Click the button **Add** in **SV Path Planning**.

A control point is added into the aorta path and its location information appears in the

Control Point List. In the display view, the point is shown as red/blue point (square) when it is/isn't chosen. To prevent the point covering the display view, we can change the size of point:

Right Click **aorta** in **SV Data Manager**.

In the popup menu select **Point 2D Size**.

Give a smaller and proper size then click **OK**.

When a control point is created, it is important to check whether this point is located or approximated to the center of the vessel with three 2D views together. To move a control point,

Select the point in the **Control Point List**. The point in the 2D view becomes red.

Left press the cross center of the point and move it to a new place.

If it's not easy to select the center of the point, we can change the size of the point larger. We continue to decrease Axial slider value in **Image Navigator**. Then the display views also change correspondingly. We add a series control points center at the aorta lumen. When we meet the bifurcation area, where the aorta splits into two iliac arteries, we focus on the left iliac artery (see Figure 18). The distance of control points should be cared. As

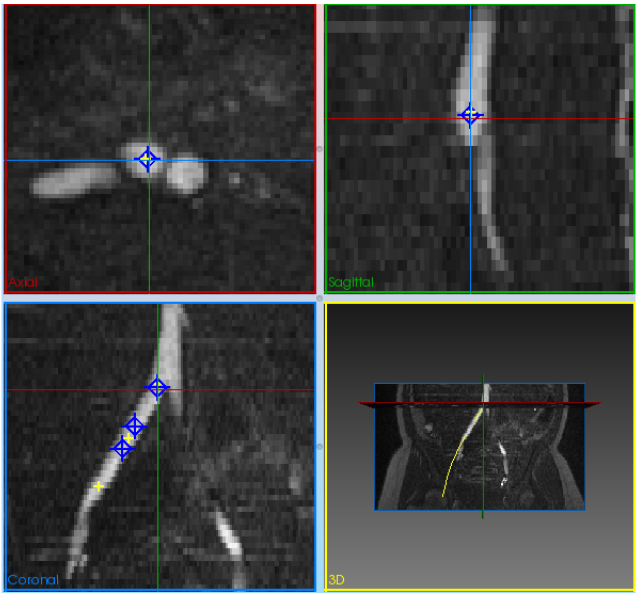
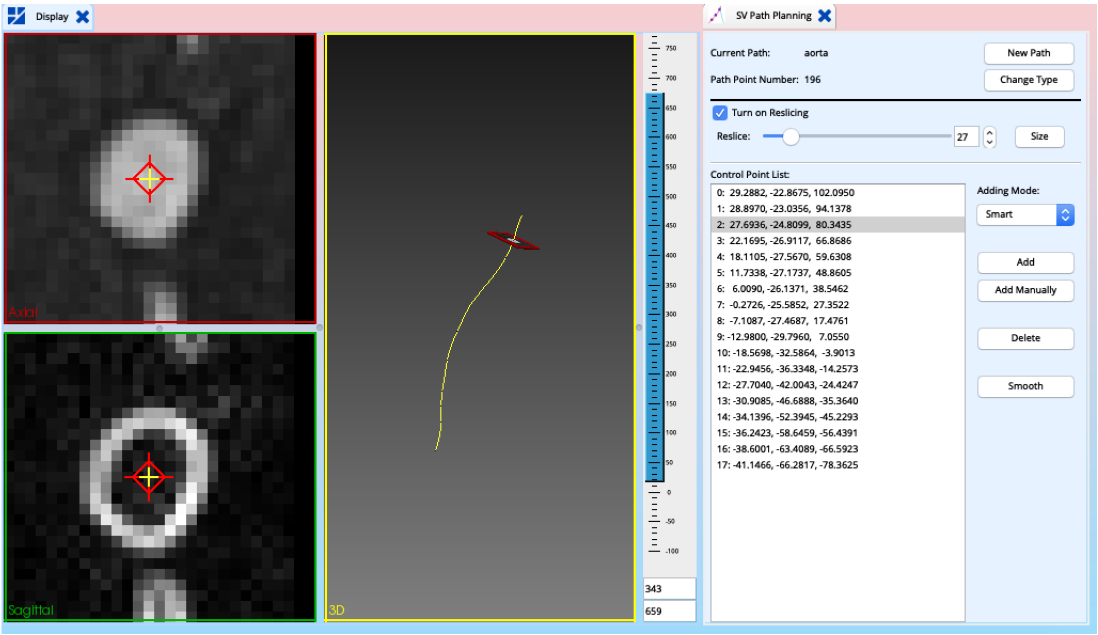


Figure 18: Path construction of aorta.

explanation in section 2.2, too short or too far distance has a bad effect on model. When the path through some vessel parts with specific shape (bifurcation, bend *et.al.*), additional control points can be added to capture the feature.

After a path is created, we can toggle the checkbox **Turn on Reslicing** (see Figure 19) in **SV Path Planning** to check the path quality with a series of slice perpendicular to the path. We can change the reslice position by scrolling the middle mouse button or change **Reslice** value in **SV Path Planning**. The reslice size can be changed by clicking **Size**, which is an important operation in segmentation step. The display window consists of three subwindows:



Current Path: aorta

Path Point Number: 196

☑ Turn on Reslicing

Reslice: 27

Control Point List:

0: 29.2882, -22.8675, 102.0950
1: 28.8970, -23.0356, 94.1378
2: 27.6936, -24.8099, 80.3435
3: 22.1695, -26.9117, 66.8686
4: 18.1105, -27.5670, 59.6308
5: 11.7338, -27.1737, 48.8605
6: 6.0090, -26.1371, 38.5462
7: -0.2726, -25.5852, 27.3522
8: -7.1087, -27.4687, 17.4761
9: -12.9800, -29.7960, 7.0550
10: -18.5698, -32.5864, -3.9013
11: -22.9456, -36.3348, -14.2573
12: -27.7040, -42.0043, -24.4247
13: -30.9085, -46.6888, -35.3640
14: -34.1396, -52.3945, -45.2293
15: -36.2423, -58.6459, -56.4391
16: -38.6001, -63.4089, -66.5923
17: -41.1466, -66.2817, -78.3625

Adding Mode: Smart

Figure 19: Reslicing view based on aorta path.

The right window (3D view) displays a path (yellow line) and a slice plane (outlined in a red square). The slice plane is called intensity probe.

The left top window (2D view) displays the image intensity of the intensity probe.

The left bottom window (2D view) displays the magnitude of the image intensity gradients of the intensity probe. The bright areas are the places with large intensity change. In this picture, the bright ring is the boundary of vessel.

We can move the location of control point in 2D views to improve the path quality. When there is a jag appearing on the path, click **Smooth** button to smooth the path.

When creating paths, let paths as long as enough to make sure they cover the entire distance of the vessels. For the vessels with tortuosity, make use of three 2D views to add control points carefully. For example, when creating path of left UtA (see Figure 20), the shape of vessel cross sections on **Axial** plane are not oval. With 3D view, instead of increasing **Axial** slider value as before, we continue to add controls points by decreasing **Sagittal** slider value (moving **Sagittal** plane to the left).

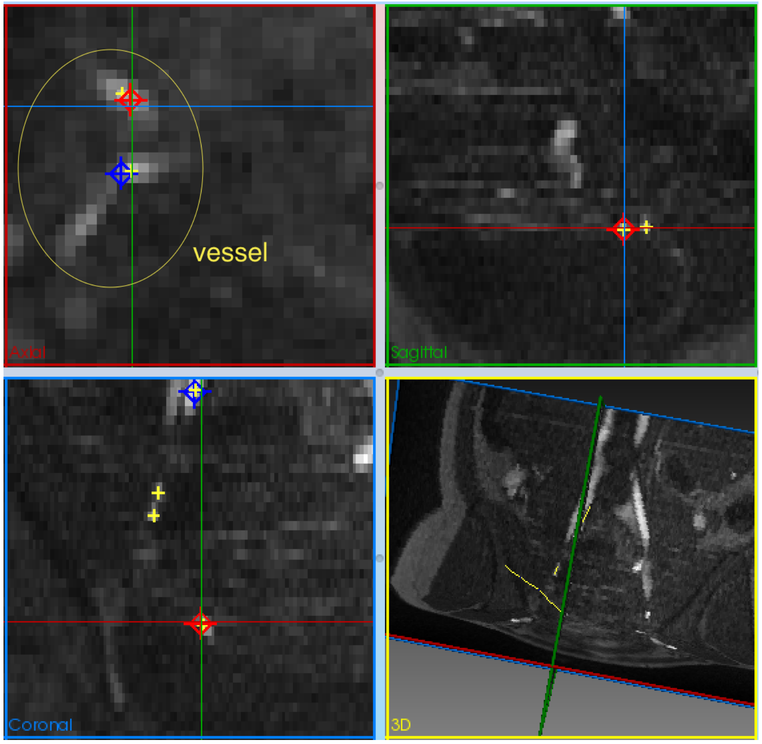


Figure 20: Display view of left UtA in SimVascular.

For the vessel branches, it's helpful to make paths have some overlap. The explanation can be seen in Figure 21. Lastly, let path close to the real centerline of the vessel, which benefits segmentation step. After processing, click **Save SV Projects** to save the path information.

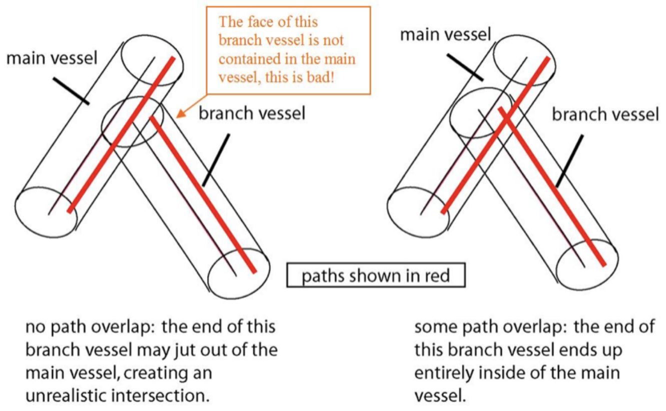

main vessel
The face of this branch vessel is not contained in the main vessel, this is bad!
branch vessel
main vessel
branch vessel
paths shown in red
no path overlap: the end of this branch vessel may jut out of the main vessel, creating an unrealistic intersection.
some path overlap: the end of this branch vessel ends up entirely inside of the main vessel.

Figure 21: Explanation of overlap paths in SimVascular user-guide.[27]

3.1.3 2D Segmentation

SimVascular provides lofted 2D segmentation and direct 3D segmentation method. We give description to the operations included in the former one, such as operations of threshold method and level set method introduced in section2.3.

We first add a data node for storage of a series of contours.

Right click **Segmentation** in **SV Data Manager**.

Select **Create Contour Group** in the popup menu.

Select path (aorta), set group name (aorta) and click **OK**.

Double click the new data node **aorta** in **Segmentation**, a toolbox **SV 2D Segmentation** comes out. The display view is almost similar as reslicing view in Figure 19. Before doing 2D segmentation on the display view, first set the size of display view large enough to contain all the lumen area. To realize threshold method,

Click **Threshold** button in **SV 2D Segmentation** tool window.

Two new panels show inside the tool window (above Contour List).

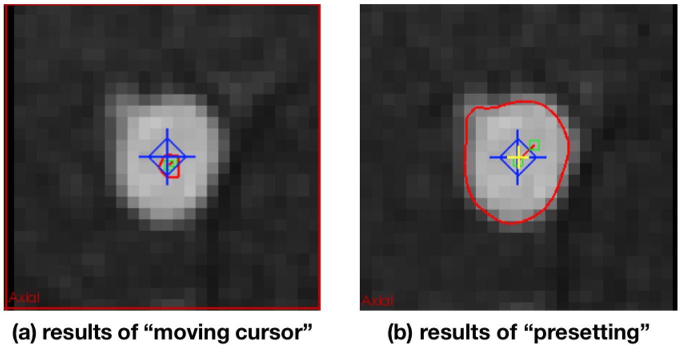
top panel:set a preset threshold value; bottom panel: postprocess the contour and realize batch segmentation. Toggle off the checkbox of **Preset**.

Move cursor to the 2D view window.

Press and hold the left mouse at the centre of the vessel.

Move the cursor down(up) to increase(decrease) the threshold value until the yellow contour is close to the boundary of lumen.

After releasing the mouse, the contour is constructed and becomes red. Sometimes the created contour is not satisfying such as in Figure 22(a). This may due to the low resolution and large variation of the intensities on the inside the lumen area. Thus we toggle on the checkbox of **Preset** to set the threshold value in advance. One quick way to estimate it roughly is to toggle off the checkbox **Segmentation** in **SV Data Manager**, then click the pixel on the boundary of the lumen to see its intensity (pixel value) on the bottom right corner of GUI.Here we preset the threshold value to be 200 and then adjust it to let contour match the edge. The finalized contour is presented in Figure 22(b) with two control points (green), which can be used for only for shifting and scaling. If we want change the shape of the contour, we can click **SplinePoly** in the tool window.to convert the contour into a closed spline with twelve control points. If the contour is jagged, we can use smoothing panel or click **Smooth** button in the tool window to postprocessing it.



(a) results of "moving cursor" **(b) results of "presetting"**

Figure 22: Contours with threshold method.

The threshold method relies on the user-specific threshold value, the generated contour is not always reliable. We turn to the usage of level set method. Level set method in SimVascular is initialized by a seed point (small disk) and then performs segmentation in two stages. In the first stage, the velocity is governed by exponentially decaying functions. In the second stage, the velocity is governed by edge attraction functions. More details are provided in.[32] When click **LevelSet**, a panel consists of parameters show up in the tool window.

Stage 1

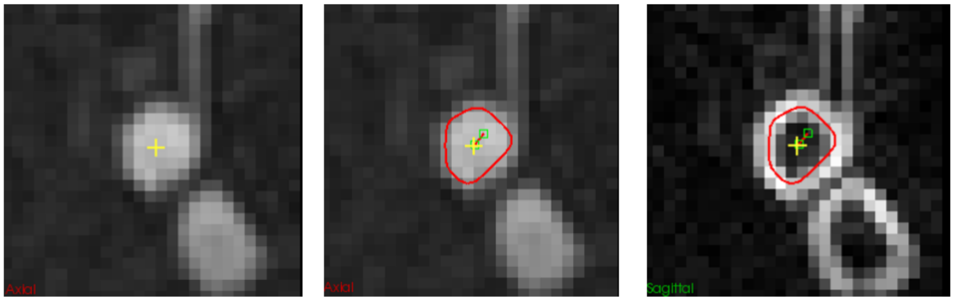
Seed radius: control the radius of the initial circle, which should be inside of the vessel lumen in the 2D view.

Kthr: equilibrium curvature value of the level set. The lower value induces a more smoothing curvature.

Blur Feat/Adv: control the image blur on the feature/advection image. When the image is noising, the segmentation accuracy can be improved by increasing it at cost of decreasing precision.

Stage 2

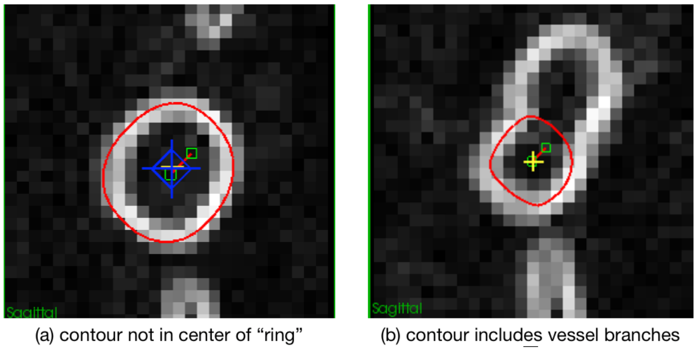
Kupp/Klow: define a range for curvature.



(a) original MR image slice (b) contour on image data (c) contour on intensity image

Figure 23: Contour with level set method.

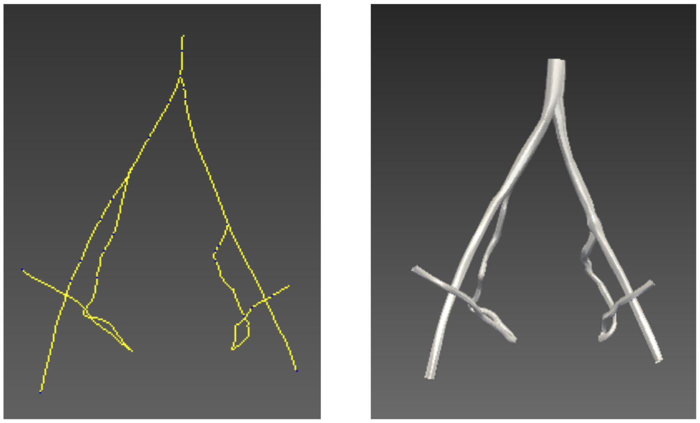
Similar postprocessing operations can also be used on the generated contours. An example of contour with level set method is given in Figure 23. Next we increase the **Reslice** slider in the tool window to move the reslicing plane along the path and create ordered contours. In order to improve the efficiency of segmentation work, We can use batch model. The batch model is to preset a series of locations, which are important to the vascular model, and generating 2D contours with the above automated segmentation techniques, which are needed to be checked later, at that locations simultaneously. When we go back to the output contours, we need to see whether these contours are in the center of the potential "ring" in the magnitude gradient image (see Figure 24(a)). If not, we can scale them. In addition, these contours shouldn't include branch vessel (see Figure 24(b)). Otherwise, the model would have an artificial geometry called lofting artifact. Before moving to the solid model creation step, we can preview the lofted surface by togging on the checkbox **Lofting Preview** in the tool window. If some part of the loft surface is not accordant to the lumen



(a) contour not in center of "ring" (b) contour includes vessel branches

Figure 24: Contours needed to be postprocessing generated with batch model.

edge, we can modify the contour group and the lofted surface would be updated. After path planning and segmentation step, we obtain a rough surface model in Figure 25.



(a) created paths (b) lofting surface

Figure 25: Created paths and lofting surface.

3.1.4 Model Creation

The model creation step can be divided into two phases. The first phase is to create a lofted solid model as explained in section2.4 for each branch. Then do Boolean operations on these individual brach model to get the blood flow domain. To create a polydata model from 2D segmentations:

> Right click **Models** in **SV Data Manager** and select **Create Model** in the popup menu.
>
> Model Type: PolyData
>
> Model name: demo
>
> Click **OK**.

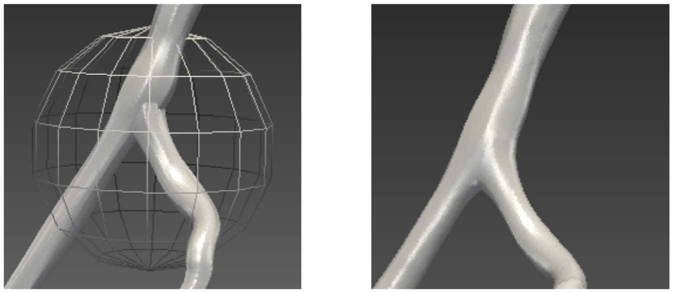
Then a new data node **demo** is under **Models**. Double click **demo** and see the tool window **SV Modeling**.

> Click **Create Model...** and toggle on checkboxs of contour group where the solid model is based.
>
> Number of Sampling Points: (optional for PolyData models)
>
> Use Uniform Lofting Parameters (Optional): if select yes, each group is used with same parameters to create the model.

All the faces together with their information are listed in **Face List**. We can edit their name, type (wall or cap), visibility , color and opacity in the tool view. To do Boolean operations:

> Select faces we are interested in.
>
> Combine them into one faces: Face Ops → Combine.
>
> Delete them: Face Ops → Delete.
>
> Fill holes created by deletion: Face Ops → Fill Holes w. IDs.

In our model, we have one inlet and four outlets, which are explained in section 2.6. After selecting cap faces as inlet or outlet, delete the remaining cap faces and fill holes. Then several new faces are created. Combine theses with all wall faces into a new wall face. Besides face operations, SimVascular also provides operations on solid modeling to improve the quality. Global operations are applied to the whole model while local operations are applied to part of the model such as faces, cells or faces. Usually, after Boolean operations, places where two surfaces union such as junctions of two branches have sharp angles and local smoothing operations are applied to these places (an example is presented in Figure 26). The detailed local operations are shown below:



(a) sphere of local operations on junction (b) junction after local smoothing process

Figure 26: Vessel junction before and after using local smoothing operation.

Local Ops → toggle on checkbox of **Show Sphere**

A sphere shows off. Resize and move it to include the area you're interested in.

Right press and move it to resize the sphere.

Left press and move it to change the sphere's location.

Select smoothing operations (**Laplacian Smooth** or **Constrain Smooth**).

After local operations, toggle of checkbox of **Show Sphere**.

Although global and local operations can let model more close to the real vessel, they may arise some problems. For example, after global smoothing operations, usually the planar inlet/outlet surfaces become curved. We need to trim the curved inlet/outlet surfaces. To cut a piece of model:

1) Trim → By Plane

Toggle on the checkbox of **Show Plane**.

A plane shows in the 3D-view window.

Move the plane manually to change the its location.

Or select the path and move the slider so that the plane at the point is perpendicular to the path.

Click **Cut Above** or **Cut Below** to trim the model.

2) Trim → By Box

Toggle on the checkbox of **Show Box**.

A box shows in the 3D-view window.

Similarly, move the box manually or select the path to change the its location and press box with right button to resize the box.

Click **Cut within Box** trim the model.

We give an example in Figure 27. After cutting operations, click **Fill Holes w. IDs** in

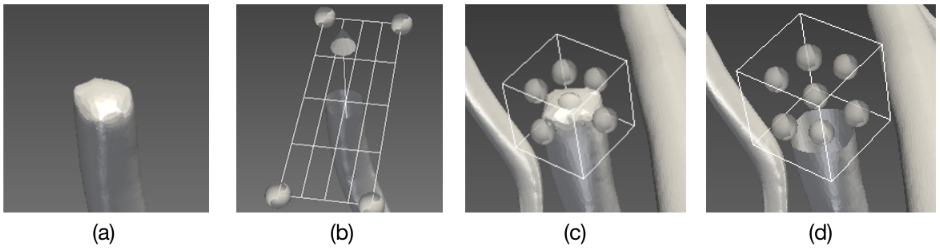


Figure 27: Curved cap surface (a) and cut it by plane with trim tool (b), curved cap surface (c) and cut it by box with box tool (d).

Global Ops to create new faces and set their types and names. Sometimes the centerline of the model is needed. It is obtained by clicking **Extract Centerlines**. Our find model in Figure 28is constructed by imaging six pregnant women, extracting their centerlines, computing an average centerline, and extruding the cylindrical shape.

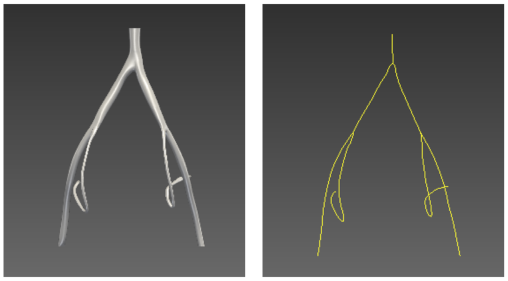


Figure 28: Solid model (left) and its centerline (right).

3.2. Meshing

With the solid model, we create a mesh using TetGen in Simvascular. We create a new data node below **Meshes** in **SV Data Manager** window. When **SV Meshing** tool window shows up, click **Estimate** to let SimVascular provide the global max edge size, which is about half of the radius of the smallest cap in the model. Then toggle on the checkbox of **Surface Meshing** and click **Run Mesher** to obtain surface mesh of the model.

To export the surface (the whole wall surface and individual faces) mesh to **vtp** (the Visualization Toolkit (VTK) format for surfaces) files, Right click the mesh node and click **Export Mesh-Complete** in the popup menu.

Select a directory and click **Choose**.

Then use VMTK software to convert **vtp** files into **stl** (a file format native to the stereolithography CAD software created by 3D Systems) files, which can be input in Gmsh software. The corresponding command is:

```
vmtksurfacewriter -ifile face.vtp -ofile face.stl
```

Here face.vtp is the file we want to convert. With the converted **stl** files, operations in Gmsh are described below:

Import multiple **stl** files into Gmsh software.

Add volume based on different surface meshes.

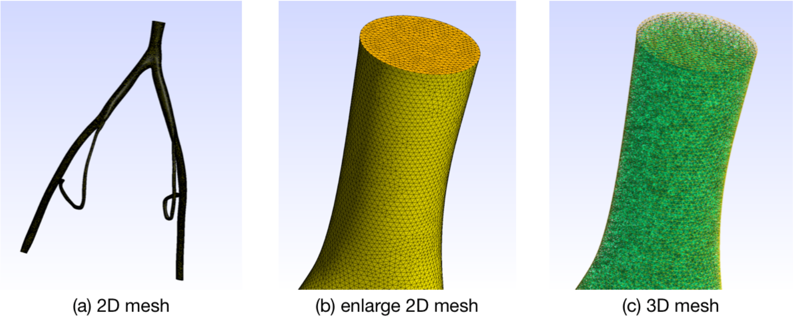
Build a physical group to assign different ID and name to each surface.

Save a final **stl** with physical elements.

Import the save **stl** file and construct 1D, 2D and 3D mesh.

Save mesh file as version *II* ascaii mesh file.

The 2D/3D mesh of the vascular model in Gmsh is presented in Figure 29. To run simulation, we need to convert the **msh** file to Nektar++ format with command:



(a) 2D mesh

(b) enlarge 2D mesh

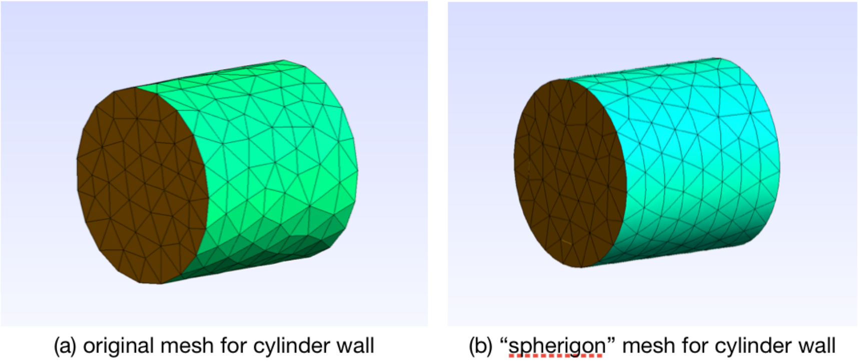
(c) 3D mesh

Figure 29: Mesh of the vascular model.

```
NekMesh input.msh input.xml:xml:uncompress
```

Nekmesh has smoothing operations such as spherigons, a kind of patch mentioned in section 2.5.5 for efficiently smoothing polygon surfaces with command:

```
Nekmesh -m spherigon:surf=1 MeshWithStraightEdges.xml MeshWithSpherigons.xml
```



(a) original mesh for cylinder wall

(b) “spherigon” mesh for cylinder wall

Figure 30: An example of Spherigon patch.

3.3. Set Boundary Condition

The import wave flow was measured in one patient using 2D phase contrast MRI,[59] by drawing a circle around the vessel in each time frame and calculating an average velocity and area over time. Because of the the temporal resolution of the MR imaging, the flow is presented with a set containing discrete points with spacing. It takes a while to collect one time frame so that the spacing can’t be avoided. In order to better understand the flow data and obtain reasonable interpolation, we extend flow data into several periods, interpolate it and apply Fourier transformation to the interpolated flow curve.

Non-dimensionalization is a technique which is partial or full removal of units from an equation involving physical quantities by a suitable substitution of variables. When using this technique in simulation computing, the problem is simplified and the number of free parameters are reduced. After substituting the length scale, flow velocity scale and time scale, we shift the flow data to be nonnegative and get interpolation in Figure 31. Then we calculate its Fourier coefficents and write them into input file. When setting inlet boundary condition file in Nektar++, center point location and radius of inlet surface are needed, which can be estimated in Gmsh by togging on checkbox of axes.

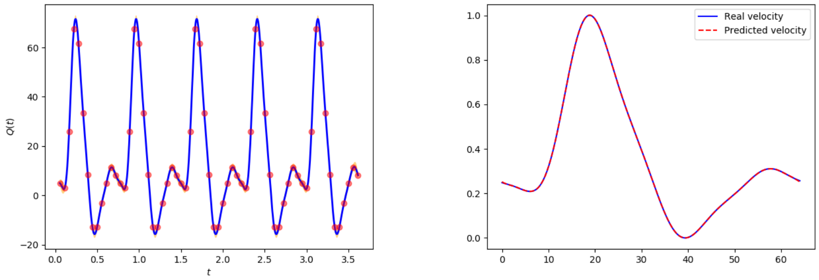


Figure 31: Real discrete flow data (left) and interpolated velocity curve (right).

3.4. Post-processing(Nektar++) and visualization(Paraview)

The output simulation results contain a series of binary (**chk** and **fld**) files. With utility **FieldConvert** in Nektar++, these **chk** files can be converted to **vtu**, which can be visulized in Paraview. The basic command is:

```
FieldConvert test.xml test.fld test.vtu
```

where **FiledConvert** is the executable associated to the utility **FieldConvert**, **test.xml** is the session file, **test.fld** is the file we want convert and **test.vtu** is the converted file. Notice that **test.xml** is the mesh file containing $< CONDITION >$ section. An example of visulization view in Paraview is given in Figure 32 (left). If we want to extract a boundary region and see velocity on it, the commands are:

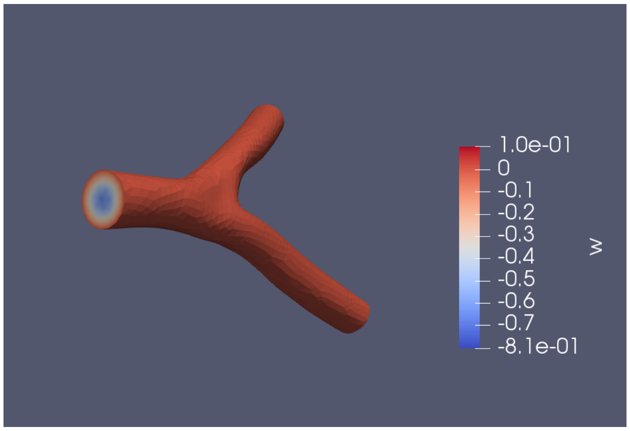


Figure 32: An example of Paraview view: velocity on vessel model in w direction.

```
Nekmesh -m extract:surf=3 test.xml test-b2.xml

FieldConvert -m extract:bnd=2:fldtoboundary=1 test.xml test.fld test-boundary.fld

FieldConvert test-b2.xml test-boundary_b2.fld test-boundary_b2.btu
```

The first command line is to extract the boundary region, where **surf=3** is its physical group ID in mesh file and **test-b2.xml** is the output region file. The second command line is to extract output data on the interested boundary region. The option **bnd** specifies which boundary region to extract and is corresponding to boundary ID in $< CONDITION >$ section. The extracted output is presented in Figure 32 (right). Sometimes we need to extract boundary region from multiple **chk** files. To realize it, we can write a bash file as follows:

```
#!/bin/sh
../../nektar++-4.4.1/build/dist/bin/Nekmesh -m extract:surf=3 geo.xml inlet/geo-b2.xml;
for ((i=0; i<65; i++)); do
    ../../nektar++-4.4.1/build/dist/bin/FieldConvert -m extract:bnd=2:fldtoboundary=1 geo.xml chk/
        scal_short_vessel_$i.chk inlet/fld/scal_short_vessel_$i-boundary.fld;

    ../../nektar++-4.4.1/build/dist/bin/FieldConvert inlet/geo-b2.xml inlet/fld/
        scal_short_vessel_$i-boundary_b2.fld inlet/vtu/scal_short_vessel_$i-boundary_b2.vtu;
done
```

4. Future work

In our thesis, we give brief introduction to the 2D segmentation tools using SimVascular. However, the whole process involves many manual work. In the path planning step, we need to manually choose control points on many slices. In 2D segmentation step, a series of parameters are needed to be set up if we want to use level set and threshold method. After solid model created, manually post-processing operations are also needed to be done. Thus the whole work flow is tedious and time-consuming, which is needed to be simplified and automatic. Far from now, it's challenging to construct vascular model automatically. However, we can make some steps automatic. For example, with the recent development of deep learning method like Deepmedic,[60,61] U-Net,[62] 2D segmentation can be realized automatically.

In addition to automating 2D segmentation work flow, we need to realize the limitation of this technique. When geometries of vessel have complicated structures, 2D segmentation method may not provide satisfactory results. In contrast, 3D segmentation methods which are not introduced in this thesis, can perform better. This should also be included in the future work.

Mesh generating is also an important task in CFD simulation. In this thesis, we use several different software(Gmsh, Simvascular, netfbb from Autodesk) for mesh generating. The reason why of this is because we can't find one software can do all the things for mesh generation. Also, the existing software we used in this thesis can not do the job perfectly. For example, when we apply curvature on vessel model combined by each components(inlet, outlet and wall) processed from other software but not as whole, the surface may not be smooth at the boundary of some components. Since many software we used are open source, we may fix those issue and provide better user experience in the future.

BIBLIOGRAPHY

[1] Insull W Jr. The pathology of atherosclerosis: plaque development and plaque responses to medical treatment. Am J Med. 2009 Jan;122(1 Suppl):S3-S14.

[2] George Thanassoulis, Mehdi Afshar. Atherosclerosis.`https://www.merckmanuals.com/home/heart-and-blood-vessel-disorders/atherosclerosis/atherosclerosis`.

[3] Jeffrey W. Olin, Brett A. Sealove. Peripheral Artery Disease: Current Insight Into the Disease and Its Diagnosis and Management. Mayo Clin Proc. 2010 Jul; 85(7): 678692.

[4] Timothy W. Secomb. Hemodynamics. Compr Physiol. 2016; 6(2): 9751003.

[5] McDonald DA. Blood Flow in Arteries. Second. London: Edward Arnold; 1974.

[6] Davies, P. F., M. Civelek, Y. Fang, and I. Fleming. The atherosusceptible endothelium: endothelial phenotypes in complex haemodynamic shear stress regions *in vivo*. *Cardiovasc. Res.* 99(2):315327, 2013.

[7] Nichols, W., M. ORourke, and C. Vlachopoulos. McDonalds Blood Flow in Arteries: Theoretical, Experi- mental and Clinical Principles. CRC Press, Boca Raton, 2011.

[8] Kwak, B.R., M. Bӓck, M.-L. Bochaton-Piallat, G. Caligiuri, M. J.A.P. Daemen, P. F. Davies, I. E. Hoefer, P. Holvoet, H. Jo, R. Krams, . Biomechanical factors in atherosclerosis: mechanisms and clinical implications. *Eur. Heart J.* 35(43):30133020, 2014.

[9] Morbiducci, U., A. M. Kok, B. R. Kwak, P. H. Stone, D. A. Steinman, J. J. Wentzel, et al. Atherosclerosis at arterial bifurcations: evidence for the role of haemodynamics and geometry. Thromb. Haemost. 115(3):484492, 2016.

[10] Chan, C., Poon, C., Wong, R. C., Zhang, Y.,A hybrid body sensor network for continuous and long-term measurement of arterial blood pressure. In: 2007 4th IEEE/EMBS International Summer School and Symposium on Medical Devices and Biosensors. IEEE, pp. 121123.

[11] Agnes S. Meidert, Bernd Saugel. Techniques for Non-Invasive Monitoring of Arterial Blood Pressure. Front Med(Lausanne) 2017;4:231. Published online 2018 Jan 8.

[12] Bernd Volker Scheer, Azriel Perel, Ulrich J Pfeiffer. Clinical revirew: Complications and risk factors of peripheral arterial catheters used for haemodynamics monitoring in anaesthesia and intensive care medicine. Crit Care.2002; 6(3):199-204. Published online 2002 Apr 18.

[13] Nieman, K., Oudkerk, M., Rensing, B. J., Ooijen, P., Munne, A., Geuns, R., and Feyter, P. J., 2001, Coronary Angiography With Multi-Slice Computed Tomography Lancet, 357(9256), pp. 599603.

[14] Elkins, C. J., and Alley, M. T., 2007, Magnetic Resonance Velocimetry: Applications of Magnetic Resonance Imaging in the Measurement of Fluid Motion, Exp. Fluids, 43(6), pp. 823858.

[15] Rosch, T., Braig, C., Gain, T., Feuerbach, S., Siewert, J. R., Schusdziarra, V., and Classen, M., 1992, Staging of Pancreatic and Ampullary Carcinoma by Endoscopic Ultrasonography: Comparison With Conventional Sonography, Computed Tomography, and Angiography, Gastroenterology, 102(1), pp. 823858.

[16] Sarvazyan, A., Hall, T. J., Urban, M. W., Fatemi, M., Aglyamov, S. R., and Garra, B. S., 2011, An Overview of ElastographyAn Emerging Branch of Medical Imaging, Curr. Med. Imaging Rev., 7(4), pp. 255282.

[17] Lan H., Updegrove A., Wilson N. M., Maher G. D., Shadden S. C., Marsden A. L. A Re-Engineered Software Interface and Workflow for the Open-Source SimVascular Cardiovascular Modeling Package. J Biomech Eng. 2018 Feb 1;140(2).

[18] Aaslid, R., Markwalder, T.-M., Nornes, H., 1982. Noninvasive transcranial doppler ultrasound recording of flow velocity in basal cerebral arteries. Journal of neurosurgery 57 (6), 769774.

[19] Stankovic, Z., Allen, B. D., Garcia, J., Jarvis, K. B., Markl, M. (2014). 4D flow imaging with MRI. Cardiovascular diagnosis and therapy, 4(2), 173192.

[20] Milner JS, Moore JA, Rutt BK, Steinman DA. Hemodynamics of human carotid artery bifurcations: computational studies with models reconstructed from magnetic resonance imaging of normal subjects. J Vasc Surg. 1998 Jul;28(1):143-56.

[21] Morris PD, Narracott A, von Tengg-Kobligk H, et al Computational fluid dynamics modelling in cardiovascular medicine Heart 2016;102:18-28.

[22] Barber DC, Hose DR. Automatic segmentation of medical images using image registration: diagnostic and simulation applications. J Med Eng Technol 2005;29:5363.

[23] Schroeder, Will; Martin, Ken; Lorensen, Bill (2006), The Visualization Toolkit (4th ed.), Kitware.

[24] Visualization Handbook, Academic Press, 2005.

[25] Christoph Hertzberg: A Framework for Sparse, Non-Linear Least Squares Problems on Manifolds, Diploma Thesis, University of Bremen, 2008.

[26] Updegrove, Adam, M. Wilson, Nathan, Merkow, Jameson, Lan, Hongzhi, Marsden, Alison, Shadden, Shawn. (2016). SimVascular: An Open Source Pipeline for Cardiovascular Simulation. Annals of Biomedical Engineering. Vol. 45.

[27] SimVascular tutorial in path planning. `http://simvascular.github.io/docsModelGuide.html#modelingPathPlanning`.

[28] Joy Paochi Ku. (2003) numerical and experimental investigations of blood flow with application to vascular bypass surgeries. Phd dissection of Standford University.

[29] Allasia, Giampietro. Cardinal basis interpolation on multivariate scattered data. Nonlinear Analysis Forum. 2001 Jan;Vol. 6.

[30] E. W. Cheney, Multivariate Approximation Theory: Selected Topics, CBMS-NSF Regional Conference Series in Applied Mathematics, Vol. 51, SIAM, Philadelphia, 1986.

[31] D. Shepard, A two-dimensional interpolation function for computer mapping of irregularly spaced data, Tech. Report ONR-15, Harvard Univ., Cambridge, Mass, March, 1968.

[32] K. C. Wang, ”Level Set Methods for Computational Prototyping with Application to Hemodynamic Modeling,” in Electrical Engineering. Stanford, CA: Stanford University, 2001.

[33] Osher, Stanley J., Fedkiw, Ronald P.(2002). Level Set Methods and Dynamic Implicit Surfaces. Springer-Verlag.

[34] Sethian, James A. (1999). Level Set Methods and Fast Marching Methods : Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science. Cambridge University Press.

[35] Osher, S.; Sethian, J. A. (1988), ”Fronts propagating with curvature-dependent speed: Algorithms based on HamiltonJacobi formulations” (PDF), J. Comput. Phys., 79 (1): 1249.

[36] Sokratis, Vavilis; Kavallieratou, Ergina (September 2011). ”A Tool for Tuning Binarization Techniques”. 2011 International Conference on Document Analysis and Recognition. IEEE: 15.

[37] Press, William H.; Teukolsky, Saul A.; Vetterling, William T.; Flannery, Brian P. (1992). Numerical recipes in C: the art of scientific computing (2nd ed.). New York, NY, USA: Cambridge University Press. pp. 123128.

[38] Courant, Richard; Isaacson, E; Rees, M. (1952). ”On the Solution of Nonlinear Hyperbolic Differential Equations by Finite Differences”. Comm. Pure Appl. Math. 5 (3): 243-255.

[39] Patankar, S. V. (1980). Numerical Heat Transfer and Fluid Flow. Taylor Francis.

[40] Piegl L, Tiller W. The NURBS book. 2nd ed. New York: Springer-Verlag;1997.

[41] Gary D. Knott (2000), Interpolating cubic splines. Springer. p15.

[42] Hartmut., Prautzsch, (2002). Bzier and B-Spline Techniques. Boehm, Wolfgang., Paluszny, Marco. Berlin, Heidelberg: Springer Berlin Heidelberg. p. 63.

[43] A. Tayebi, J. Gmez Prez, I. Gonzlez Diego and F. Ctedra. Boolean operations implementation over 3D parametric surfaces to be included in the geometrical module of an electromagnetic solver. Proceedings of the 5th European Conference on Antennas and Propagation (EUCAP), Rome, 2011, pp. 2137-2141.

[44] David A. Field. Laplacian smoothing and delaunay triangulations. Communications in Applied Numercial Methods, Vol. 4, 709-712,1988.

[45] A. Updegrove, N.M. Wilson, S.C. Shadden. Advances in Engineering Software 95, 16-27, 2016.

[46] Chen M., Hart J.C., Shadden S.C.Hierarchical watershed ridges for visualizing lagrangian coherent structures. Topological methods in data analysis and visualization, Springer, 2016.

[47] Joe D. Warren; Henrik Weimer (2002). Subdivision Methods for Geometric Design: A Constructive Approach. Morgan Kaufmann.

[48] Patrick M. Knupp. Remarks on Mesh Quality. 45th AIAA Aerospace Sciences Meeting and Exhibit, 7-10 January, 2007.

[49] James C. Cavendish, David A. Field, and William H. Frey. An Approach to Automatic Three Dimensional Finite Element Mesh Generation. International Journal for Numerical Methods in Engineering 21(2):329347, February 1985.

[50] Volino, Pascal Thalmann, Nadia. (1999). The SPHERIGON: A simple polygon patch for smoothing quickly your polygonal meshes. Proc. Computer Animation '98. 10.1109/CA.1998.681910.

[51] SimVascular tutorial in solid modeling. `http://simvascular.github.io/docsModelGuide.html#modelingSolidModelingPolyData`.

[52] Delaunay, Boris (1934). "Sur la sphre vide". Bulletin de l'Acadmie des Sciences de l'URSS, Classe des Sciences Mathmatiques et Naturelles. 6: 793800.

[53] Womersley, J.R. (March 1955). "Method for the calculation of velocity, rate of flow and viscous drag in arteries when the pressure gradient is known". J. Physiol. 127 (3): 553563.

[54] Cantwell, Chris D. . Nektar++: An open-source spectral/hp element framework. Computer Physics Communications 192 (2015): 205-219.

[55] Sagawa K, Lie RK, Schaefer J (March 1990). "Translation of Otto Frank's paper "Die Grundform des Arteriellen Pulses" Zeitschrift fr Biologie 37: 483-526 (1899)". Journal of Molecular and Cellular Cardiology. 22 (3): 2534.

[56] SimVascular tutorial in boundary conditions. `http://simvascular.github.io/docsFlowSolver.html#bcphysics`.

[57] Chen, P., Quarteroni, A., Rozza, G., 2013. Simulation-based uncertainty quantification of human arterial network hemodynamics. Interna- tional journal for numerical methods in biomedical engineering 29 (6), 698721.

[58] G.J. Burton, A.W. Woods, E. Jauniaux, J.C.P. Kingdom. Rheological and Physiological Consequences of Conversion of the Maternal Spiral Arteries for Uteroplacental Blood Flow during Human Pregnancy Placenta. 2009 Jun; 30(6): 473482.

[59] Lotz, J., Meier, C., Leppert, A., Galanski, M. Cardiovascular Flow Measurement with Phase-Contrast MR Imaging: Basic Facts and implementation. Radiographics 22(3), 651-671 (2002).

[60] Konstantinos Kamnitsas, Christian Ledig, Virginia F.J. Newcombe, Joanna P. Simpson, Andrew D. Kane, David K. Menon, Daniel Rueckert, and Ben Glocker, Efficient Multi-Scale 3D CNN with Fully Connected CRF for Accurate Brain Lesion Segmentation, Medical Image Analysis, 2016.

[61] Konstantinos Kamnitsas, Liang Chen, Christian Ledig, Daniel Rueckert, and Ben Glocker, Multi-Scale 3D CNNs for segmentation of brain Lesions in multi-modal MRI, in proceeding of ISLES challenge, MICCAI 2015.

[62] Ronneberger, Olaf; Fischer, Philipp; Brox, Thomas (2015). ”U-Net: Convolutional Networks for Biomedical Image Segmentation”. arXiv:1505.04597.