THESIS

EFFECTS OF STRUCTURE ON FLOW MECHANICS IN THE HUMAN LEFT VENTRICLE AND RESPIRATORY TRACT

Submitted by

Brandon L. Moore

Department of Mechanical Engineering

In partial fulfillment of the requirements

For the Degree of Master of Science

Colorado State University

Fort Collins, Colorado

Summer 2011

Master's Committee:

Advisor: Lakshmi Prasad Dasi

Christopher Orton
Hiroshi Sakurai

ABSTRACT

EFFECTS OF STRUCTURE OF FLOW MECHANICS IN THE HUMAN LEFT VENTRICLE AND RESPIRATORY TRACT

Cardiac and respiratory dysfunctions represent a large portion of healthcare problems in the United States. Many of these problems are caused by abnormal flow mechanics due to altered anatomical structure. This structure in the human body is very complex and ranges over many different scales. At relatively small scales, one facet that is still not well understood is the role of trabeculae on the biomechanics of the left ventricle. Similarly, large-scale airflow through the respiratory tract has not been fully investigated as a function of age or mechanical ventilation. This research has revealed some of the flow patterns caused by these different scale structures. Fractal geometry was used to help characterize the inner surface of the left ventricle at different times during the cardiac cycle. The fractal dimension of the ventricle was determined using a custom box-counting algorithm developed in MATLAB, and it was shown that trabeculae do indeed play a role in the biomechanics of heart pumping. Computational fluid dynamics (CFD) was also run on the respiratory tracts of three different patients to determine airflow effects due to age and intubation. Three dimensional models were constructed from computed tomography (CT) scans and simulations were run in ANSYS Fluent. Results of the study were validated through grid and time step sensitivity studies

as well as comparison to previous studies. It was shown that flow mechanics in the airways of children change with age as well as with the introduction of an intubation tube.

ACKNOWLEDGEMENTS

There are a number of people who I would like to thank for their support and contributions. My advisor, Dr. L. Prasad Dasi, has been an outstanding mentor, for which I am extremely grateful. I would also like to thank my committee members Dr. Hiroshi Sakurai and Dr. Christopher Orton. In particular, I appreciate the technical support and expert advice I have received from Dr. Sakurai over the past two semesters.

I am grateful to the Mechanical Engineering Department for supporting me through a Graduate Teaching Assistantship. This position provided both financial support as well as invaluable experience.

For supplying our lab with clinical data, I would like to thank collaborators Dr. John Oshinski and Dr. Kristen Lloyd Baugnon, both of Emory University in Atlanta, GA. Without their help none of this research would have been possible.

The students in the Cardiovascular and Biofluid Mechanics Laboratory have all contributed in some way to this thesis. I appreciate the knowledge and expertise that everyone has shared with me. It has been a pleasure working with this lab group.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Cardiovascular and respiratory dysfunctions represent a large percentage of healthcare problems in the United States. In fact, over 80 million Americans have some form of cardiovascular disease while around 24 million have shown evidence of impaired lung function. In many cases symptoms are caused by altered anatomy and can go unnoticed for years. Since structure impacts fluid mechanics, an understanding of this relationship could help lead to earlier detection of problems.

Anatomical structure in the human body is quite complex, and this complexity ranges over many different scales. At relatively small scales, one facet that is still not well understood is the role of trabeculae on the biomechanics of the left ventricle. Similarly, large-scale airflow through the respiratory tract has not been fully investigated as a function of age or intubation. This research aims to elucidate the flow patterns caused by these different scale structures.

The overarching hypothesis of this research is that small scale and large scale structure play a role in the biomechanics of the left ventricle and respiratory tract, respectively. The following are specific aims that test this hypothesis:

- Part 1 – To understand how the geometrical structure of trabeculae may influence the biomechanics of the left ventricle

- Part 2 – To investigate the dependence of airflow dynamics on age and presence of mechanical ventilation

In this work, a novel fractal geometry based analytical technique was developed to investigate the dynamic behavior of trabeculae in the left ventricle. It was found that these structures do indeed play a role in the biomechanics of heart pumping. Through the use of fractal geometry, it was discovered that trabeculae start to disappear during systole, expelling blood as the gaps between them are closed. Also, a patient-specific modeling framework was developed to perform computational fluid dynamics (CFD) analysis on the respiratory tract. Results suggest that flow mechanics are dependent on airway structure, which changes significantly with a child's age and is altered due to the presence of an intubation tube. Specifically, the airways of older children have a much larger average cross-sectional area, which allows for lower pressure gradients and velocities. Also, mechanical ventilation through an intubation tube causes non-physiologic flow that leads to high velocity and wall shear stress in the lower trachea.

A brief background on the anatomical structures that were tested and the tools that were used is presented in Chapter 2; specific methodologies that were implemented are explained in Chapter 3; and results are presented and discussed in Chapter 4, along with a summary of research and an explanation of future work.

2. BACKGROUND

The following is a brief overview of the objects and theories studied in this research. Since there were two distinct anatomical structures tested, along with the use of a complex geometric analytical technique, this chapter will be divided into three sections. Section 2.1 covers the structure and function of the respiratory airways and gives some background on computer modeling of internal flow. Section 2.2 explains cardiac biomechanics, specifically the small scale structure of the left ventricle. Lastly, Section 2.3 is an introduction to the theory and methods comprising fractal geometry.

2.1 Airway

2.1.1 Anatomy and Physiology

In order for the human body to function, its tissues must constantly receive oxygen and release carbon dioxide. While this process happens via capillaries in nearly all regions of the body, these two gases must ultimately be exchanged with the outside air through the nose and mouth. Since the sites of gas exchange (alveoli) between the ambient air and the cardiovascular system are located centrally in the body, there must be a system to transport air through the body. This system is known as the conducting airways and is a key element in the respiratory system.

The conducting airways serve a number of purposes. First of all, outside air is often much colder than the body's core temperature and, therefore, must be warmed before it

reaches the lungs. Usually this air is very dry and must also be humidified (Rouadi, Baroody et al. 1999). In addition, ambient air can contain a number of foreign substances capable of harming the body (Sauret, Halson et al. 2002), and these are ideally removed by features of the conducting airways. One more purpose of the conducting airways is to keep food and other ingested substances from being aspirated.

There are two main subcategories of conducting airways: the upper airways (Figure 2.1) and the lower airways (Figure 2.2). The upper airways are composed of the nose, nasal cavity, pharynx, and larynx, while the trachea and bronchioles comprise the lower airways. Following is a brief explanation of how air travels from the ambient environment to the alveoli of the lungs.

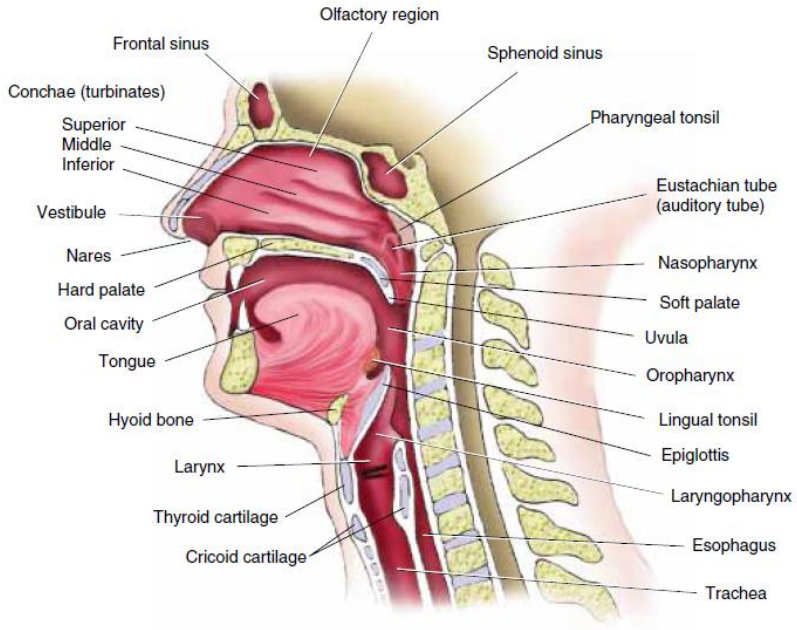


Figure 2.1: Diagram of the upper airways (Cotes, Chinn et al. 2009)

First, air enters the nostrils and immediately fills the nasal vestibule, a locally-enlarged region filled with hair follicles. These follicles are the first line of defense in filtering out any unwanted particles. Next, the air travels into the nasal cavity and is split into multiple streams by the superior, middle, and inferior conchae (turbinates). The purpose of this flow separation is to achieve maximum contact with the walls of the nasal cavity, which serve to warm and humidify the air (Churchill, Shackelford et al. 2004). Connected to the nasal cavity are the paranasal sinuses, which do not take part in airway conduction but are useful due to their production of mucus and ability to resonate sound (Sundberg, Birch et al. 2007).

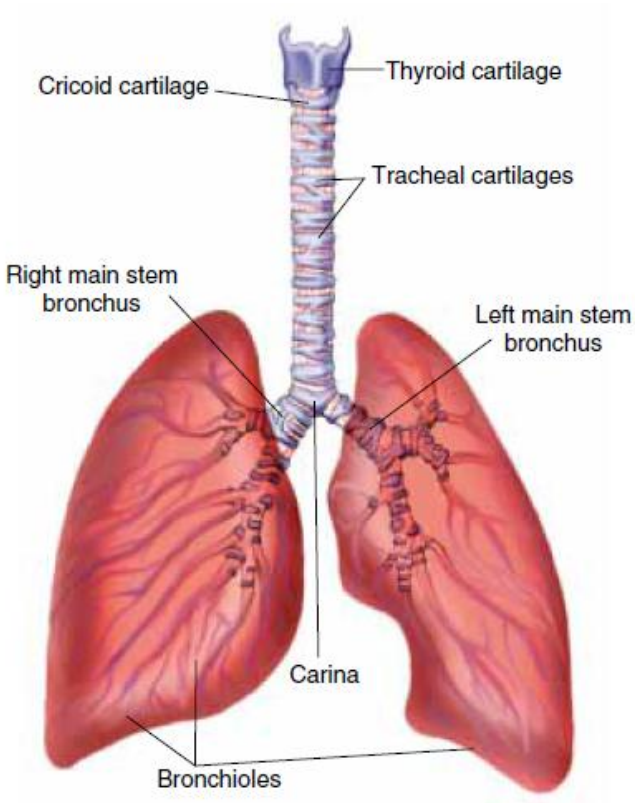


Figure 2.2: Diagram of the lower airways (Cotes, Chinn et al. 2009)

After leaving the nasal cavity, air enters the nasopharynx, one of three parts of the pharynx. Before proceeding along the conducting pathway, it should be noted that the oral cavity is generally considered merely an accessory part of the upper airways since it does not primarily function as a conducting passageway. However, it is anatomically relevant as it is connected to the pharynx at a region known as the oropharynx. This is the next step after the nasopharynx and precedes airflow to the laryngopharynx, where

the esophagus begins. Lying along the anterior wall of the laryngopharynx is the epiglottis, a flap of cartilage that forms a closure over the trachea when swallowing. The epiglottis is technically part of the larynx – the last organ of the upper airways. Aside from conducting air into the trachea, the larynx also aids in sound production and is often colloquially referred to as the "voicebox".

After passing through the larynx, air enters the lower airways. The first part of the lower airways is the trachea, a large cartilaginous tube lined with mucus and cilia. Any harmful particles still in the conducting airways can be trapped by this mucus and delivered superiorly by the cilia, where they are coughed out of the body or swallowed into the esophagus (where powerful stomach acids destroy them). The trachea branches into two bronchi, which then branch themselves. Each new level of branching is termed a generation, and the cartilaginous bronchi exist until about the 9^{th} generation. At this point, the conducting vessels are called bronchioles and are not composed of any cartilage. Finally, the terminal bronchioles exist from about generations 16 to 19, where they connect with the alveoli.

2.1.2 Ventilation

Ventilation consists of two phases: inhalation and exhalation. Inhalation is driven by the action of the diaphragm, a sheet-like muscle underneath the lungs that closes off the thoracic cavity from the abdominal cavity. As the diaphragm contracts it moves downward, thus increasing the volume of the thoracic cavity. This increase in volume causes a pressure drop in the lungs, which draws in air from the atmosphere. As air fills the lungs, they generate a reaction force due to the elastic nature of the parenchyma.

Upon relaxation of the diaphragm this elastic force pushes air out of the lungs and the thoracic cavity returns to its original volume. This process is known as expiration.

Pressure inside the lungs of an average adult is normally within 3 mmHg of atmospheric at resting ventilation rates. This corresponds to a tidal volume of 0.5 L in the average person. Tidal volume is the amount of air that is inhaled (and therefore exhaled) during one breath and is generally between 7 and 9 ml/kg ideal body weight. However, an average adult has a total lung volume much greater than this (around 6 L), and can use up to about 5.5 L of that at a time. Figure 2.3 shows the various measures used to quantify lung performance. In addition, the conducting airways contain approximately 0.15 L of anatomic dead space. This means that any inhaled breath will contain 0.15 L of stale air that has already exchanged its oxygen in the lungs.

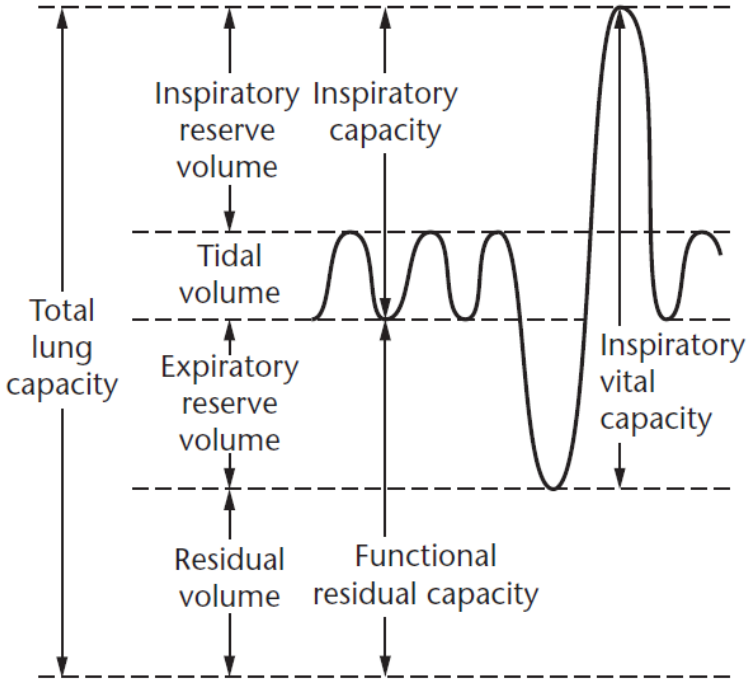


Figure 2.3: Diagram of different measures of lung volume (Cotes, Chinn et al. 2009)

In certain situations an individual may need assistance from an external source for proper ventilation. This often occurs during surgery when a person has undergone anesthesia. In such a case a positive pressure mechanical ventilator will generally be used. This device, in conjunction with an intubation tube, creates a positive pressure

gradient which forces air into the lungs and expands the thoracic cavity. Most mechanical ventilators merely shut off at a certain pressure and allow the elastic recoil of the lungs and chest to expel the air. Great care must be taken not to exceed a certain pressure or the lungs could rupture.

2.1.3 Internal flow

Internal flow, often referred to in an engineering context as pipe or duct flow, is a broad and important topic that applies to a range of situations, from oil pipelines and air conditioning ducts to blood vessels and airways inside the body (from here on the general term “pipe” will be used to describe any of these vessels). Whatever the application, there are certain properties that are common to nearly every flow. Perhaps the most fundamental of these properties is the pressure gradient. This is the driving force that must be present in order for a fluid to flow.

Depending on the magnitude of the pressure gradient, along with other factors such as fluid type and pipe geometry, the resulting flow will appear one of three ways: laminar, transitional, or turbulent (Figure 2.4). Laminar flow is characterized by smaller pressure gradients, and therefore lower flow rates, and appears very streamlined and predictable. Turbulent flow, on the other hand, contains random fluctuations in its flow and local properties are much less predictable. Transitional flow appears as intermittent bursts of laminar and turbulent flow. From a mathematical point of view, laminar flow is generally modeled with one velocity component whereas turbulent flow is modeled with three components.

While there are many factors affecting the type of flow, the most important of these can be summed up into one dimensionless parameter - the Reynolds number. As shown in equation 1,

$$\mathrm{Re} = \frac{\rho v L}{\mu} \tag{1}$$

the Reynold's is a function of fluid velocity (v), density (ρ), and viscosity (μ) as well as pipe geometry (L). While there are other factors that can affect the type of flow, this single correlation is by far the most descriptive and is often used as the sole predictor of

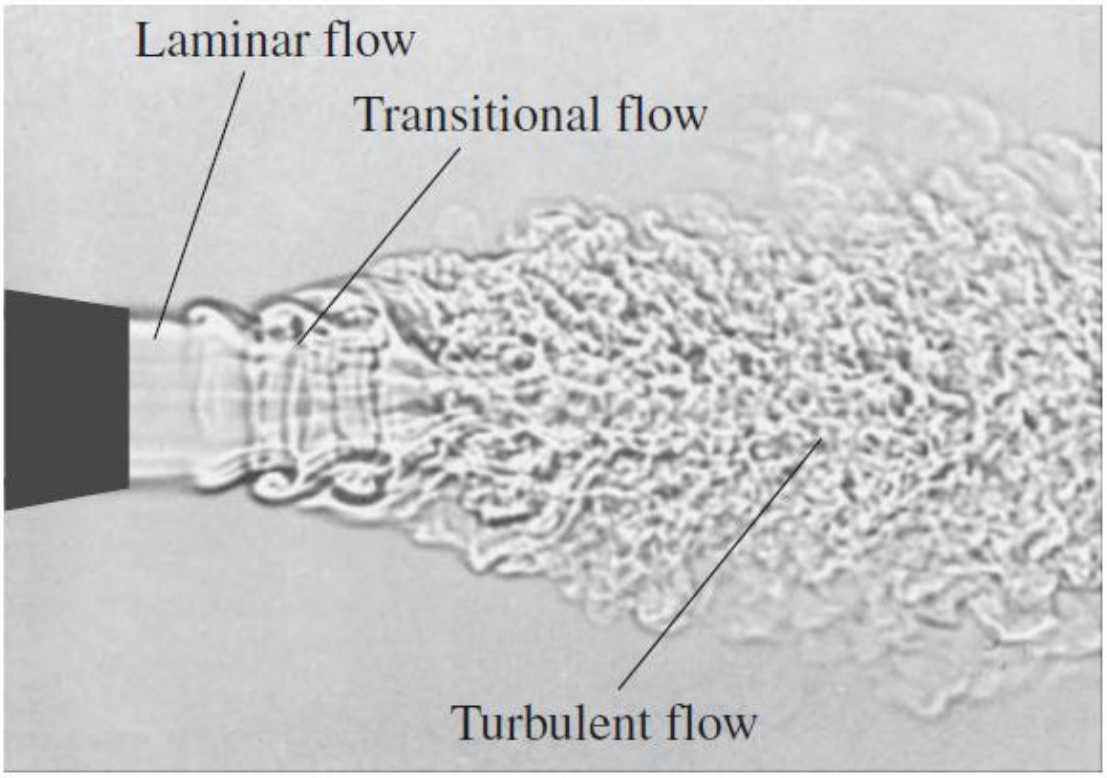


Figure 2.4: Image of a subsonic open jet depicting laminar, transitional, and turbulent flow (White 2011)

flow type. In most engineering applications, a Reynolds number below 2300 is assumed to be laminar. Transitional flow is present above this value and up to about 10,000, above which turbulent flow dominates.

2.1.4 Computational Modeling

The approach to solving computational fluid dynamics problems is based on three fundamental laws of physics: conservation of mass, conservation of energy, and Newton's second law. From these principles stem the equations for continuity, energy, and momentum, which all must be solved in order to determine the state of a fluid. This method is much more complex than modeling a solid, because a fluid can behave differently, and often unpredictably, across space and time, whereas a solid always retains the same overall shape.

There are a few basic steps to solving any CFD application. First, one must create the geometry for the problem. This geometry must then be divided into a grid, or mesh (Figure 2.5), of small simple geometries so that solutions to partial differential equations can be approximated across them. After this, a model must be chosen based on what type of flow is expected to occur. Boundary conditions are also imposed, generally at the inlet(s) and outlet(s) of the model, as well as solution methods and controls. Finally, the computation is run and post processing can be done to evaluate results (Wendt and Anderson 2009).

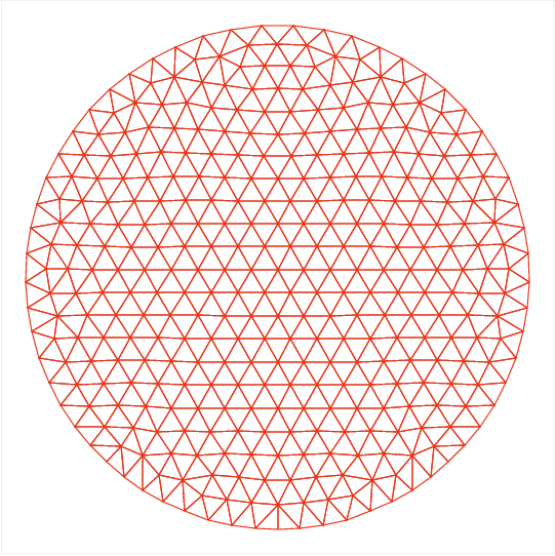


Figure 2.5: Image of a meshed circle divided into a grid of triangles (Burkardt)

Depending on the type of flow that is expected to occur, one has a number of model options from which to choose. If the Reynolds number will be sufficiently low, then a laminar model can be utilized. However, if higher Reynolds numbers, and therefore turbulent flow, are expected to occur, then a turbulence model must be implemented. There are a number of these models that are aimed at predicting the effects of turbulence. Their use is much less computationally demanding than Direct Numerical Simulation (DNS), which models all scales of turbulence, thus requiring a very fine grid and small time steps. Large Eddy Simulation (LES) also models flow features directly, but neglects the smallest scale ones in order to save computation time (Zhang, Zhai et al. 2007).

Among the most prevalent turbulence models are the Spalart-Allmaras Model, k-ω Model, k-ε Model, and Reynolds Stress Model (RSM). The Spalart-Allmaras Model is perhaps the simplest turbulence model as it uses only one equation and is good for boundary layer dominated flows. The k- ω Model is also a good selection for wall-bounded flows and takes into account compressibility and shear flow spreading. There are two equations used based on the turbulent kinetic energy (k) and the specific dissipation rate (ω). Perhaps the most widely-used turbulence model is the k- ε Model, which is due to it is range of applicability and relative computational efficiency. This model is based on equations for turbulence kinetic energy (k) and dissipation rate of turbulent energy (ε). The last model mentioned, RSM, is most likely to yield the best results for complex flows, but this comes at great computational expense since it uses seven equations(Ansys 2010). A problem should be analyzed based on known parameters, such as geometry, fluid type, and boundary conditions, before a turbulence model is selected.

2.2 Heart

2.2.1 Anatomy and Physiology

It is estimated that there are over 80 million people in the United States with some form of cardiovascular disease (Iaizzo 2009; AHA 2011). This number could be greatly reduced through better understanding of cardiac biomechanics. Studies have been conducted to examine many different facets of this concept, from the overall shape of the heart (Adhyapak and Parachuri) all the way down to the function of its individual muscle fibers (ter Keurs, Shinozaki et al. 2008; Grosberg, Gharib et al. 2009).

To understand the mechanics of heart pumping, it is best to first gain some background knowledge on the anatomy and physiology of the cardiovascular system. Lying at the center of this system is the heart, which is responsible for pumping blood through the countless number of vessels that span the human body. These vessels range in size and number, from the large but relatively sparse arteries to the microscopic but numerous capillaries. It is estimated that an adult human body contains 25,000 miles of capillaries, each with a diameter of approximately 5-10 micrometers. Such an intricate web of capillaries is necessary since these thin-walled vessels are the sites of continuous oxygen exchange between blood and body tissues. However, this offers a substantial flow resistance, which must be overcome by a large pressure difference (Equation 2). This pressure difference is generated solely by the heart.

$$Q = \frac{\Delta P}{R} \tag{2}$$

The human heart consists of four chambers: the left and right atria and the left and right ventricles (Figure 2.6). Blood passing through the right atrium enters the right ventricle and is then delivered to the lungs via the pulmonary arteries. Here the blood exchanges

carbon dioxide for oxygen with the alveoli of the lungs. Once oxygenated, blood returns to the heart and enters the left atrium. When the pressure inside the left atrium exceeds the pressure inside the left ventricle, the mitral valve opens and blood fills the left ventricle. From here, blood is pumped through the aorta, which branches off into smaller arteries, and ultimately ends up in capillaries throughout the body.

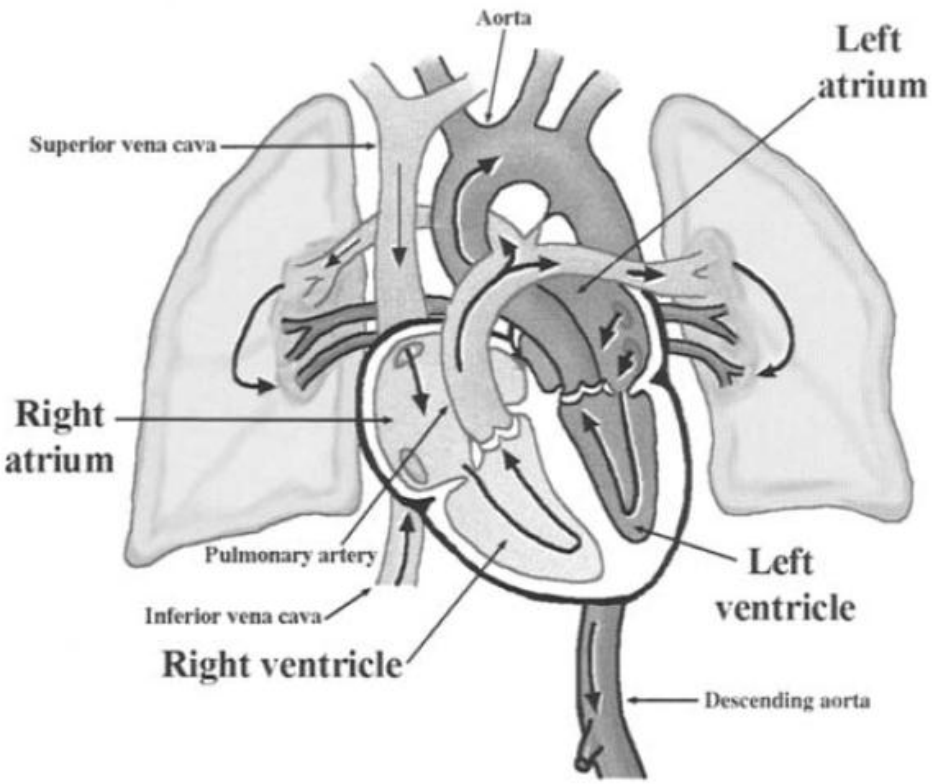


Figure 2.6: Sketch of the four chambers of the heart and the path of blood flow through them (Iaizzo 2009)

Since the left ventricle is responsible for pumping blood to all regions of the body, it must reach higher pressures (100-140 mmHg) than the right ventricle (15-33 mmHg), which pumps blood a much shorter distance to the lungs only. This difference is evident from a macroscopic inspection of both ventricles, as the walls of the left are much thicker

overall than those of the right. In addition, the shape of the left ventricle is optimized for generating high pressures. Often modeled as an ellipsoid or a prolate spheroid, the ventricle's longer axis stretches from its base to its apex. While such a stretched-out shape allows for higher pressures, it is also advantageous for directing blood flow from its inlet (mitral valve) to its outlet (aortic valve).

Although the myocardium of the left ventricle is generally thicker than that of the right, it is not of a uniform thickness. This is because different geometries require more tension to achieve the same amount of pressure. For example, the ventricle wall is very thick near its base (proximal to valves) and rather thin at its apex, which is due in large part to the radius of curvature at these two locations. The law of Laplace (Equation 3) states that more tension is required in a flatter region, or a region with a larger radius of curvature, which is why the base wall requires more tension, and is therefore thicker (i.e. more muscle mass) than the more sharply curved apex (Burton 1957).

$$\Delta P = \gamma \left(\frac{1}{R_1} + \frac{1}{R_2} \right) \tag{3}$$

2.2.2 Left Ventricular Geometry

As mentioned earlier, the left ventricle is often modeled as some type of ellipsoid. While it has been shown that a more spherical shape can be more energy efficient (Burch, Ray et al. 1952), this geometry can also lead to backflow through the mitral valve (mitral regurgitation) (Yokusoglu, Uzun et al. 2005). An inspection of the left ventricular anatomy (depicted in Figure 6, above) shows that the mitral and aortic valves are in close proximity. Therefore, the specific shape of the ventricle is necessary to facilitate proper blood flow from inlet to outlet.

In the very early stages of embryonic development, the myocardium of the human heart consists of a sponge-like network of interwoven fibers. Around weeks 5-8, these fibers are “compacted” into a more solid, continuous structure (Weiford, Subbarao et al. 2004) This compaction process forms the basic layout of cardiac fibers and gives the heart chambers their characteristically rough inner surfaces. The left ventricle is generally the roughest of these chambers as it contains the largest amount of trabeculae (Figure 2.7), which are fingerlike projections of cardiac muscle that protrude into the ventricle and are the result of the compaction process. During the cardiac cycle, these muscle bundles change in frequency and prominence. While they are normal features of the human heart, there is still very little known about the physiologic function of trabeculae. These cardiac features will be discussed more later in the context of their role in volume change of the left ventricle.

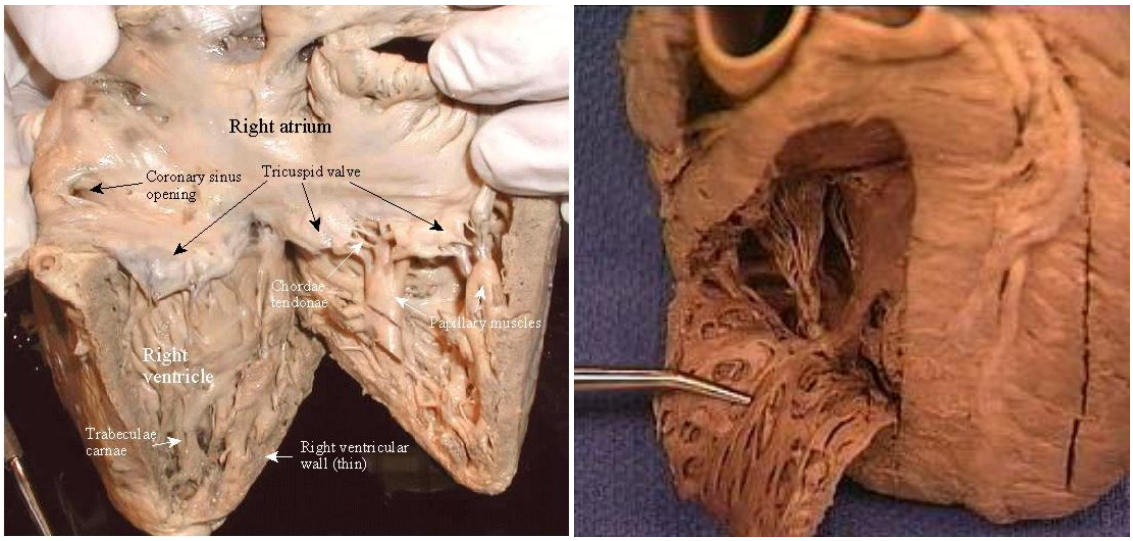


Figure 2.7: Interior views of human hearts show trabeculae along the walls of the ventricular chambers. Images taken from (Juliet)(left) and (Martino)(right)

Another interesting geometric characteristic of the left ventricle is the orientation of its muscle fibers. These fibers are oriented neither longitudinally (aligned with the long

axis) nor latitudinally (aligned with the short axis), but rather in a helical pattern. It has been suggested that this pattern is advantageous for minimizing energy expenditure during contraction. Grosberg et al used analytical models to test this hypothesis and concluded that a helical fiber pattern does indeed require less energy expenditure for the same ejection fraction than a latitudinally arranged fiber pattern (Grosberg, Gharib et al. 2009).

2.2.3 Cardiac Biomechanics

Cardiac muscle fibers are also very adaptable, as they must be in order to power such a robust pumping mechanism. In accordance with the Frank-Starling Law (Schneider, Shimayoshi et al. 2006) (Figure 2.8), the tension developed in myocardial fibers is proportional to the amount that they are stretched. So if the amount of blood entering the ventricle is greater than average, its muscle fibers will be able to generate more force in contraction to expel this increased volume. It has also been demonstrated that an increase in frequency of contraction can cause an increase in contractile force (Janssen). Both of these phenomena can be largely attributed to sarcomere properties, and both aid in balancing the amount of arterial supply and venous return from and to the heart.

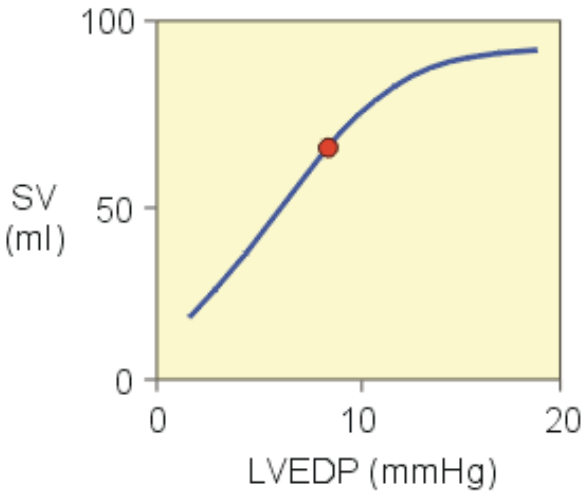


Figure 2.8: Frank-Starling mechanism illustrating the relationship between stroke volume (SV) and left ventricular end-diastolic volume (LVEDP) (Klabunde)

While there are many factors that regulate pumping efficiency of the heart, they can mostly be summarized into two key mechanical phases: filling (diastole) and ejection (systole). In order for the heart to pump effectively, it must eject blood forcefully and refill fully. Both of these events are governed by sarcomere mechanics as well as the inner geometry of the ventricle. In addition, any change in frequency of contraction must be compensated by a change in frequency of filling. This has been demonstrated to indeed occur (Janssen) and is yet another example of how the heart maintains a balance of blood flow through the body.

2.3 Fractals

The left ventricular endocardium is a very complex surface. Fractal geometry is used as a tool to characterize and analyze the biomechanical significance of the trabeculae comprising this surface. Therefore, a brief background of fractal theory is given.

2.3.1 Conceptual Introduction

Since the concept of fractal geometry is somewhat abstract, it is best to start this explanation with a simple visualization. First, imagine a blank piece of paper sitting on a table. Classical geometry would describe this piece of paper as two-dimensional, since its thickness is effectively zero. Now, imagine this piece of paper is crumpled into a ball. The resulting object now appears three-dimensional. When did this change in dimension occur? The answer is that there is no single point of dimension change. In fact, the paper is constantly changing its dimension value throughout the crumpling process. Fractal geometry can be used to quantitatively describe this changing object as well as many other complex geometric phenomena.

Many of the ideas behind fractal geometry have been around for centuries, but the term “fractal” was coined by Benoit Mandelbrot in the 1970’s and has gained much popularity since. This term is used to describe complex objects that cannot be explained by classical geometry. Stemming from the Latin word frangere meaning “to break”, fractals are often fragmented objects with irregular features (Mandelbrot 1982). Examples of these objects can be found in nature or in mathematically-created curves and surfaces.

One of the most popular natural objects that displays fractal features is the coastline of Great Britain. In Mandelbrot's paper, How Long Is the Coast of Britain: Statistical Self-Similarity and Fractional Dimension (Mandelbr.B 1967), he concludes that length is an arbitrary measure for such a situation. The total length of the coastline is merely a function of the incremental scale used to measure it. Rather, coastlines should be classified according to their geometric complexity, or "space-filling" capacity. For example the jagged coastline of Britain, with its many inlets and peninsulas, should be classified differently than the relatively smooth coastline of South Africa.

2.3.2 Technical Description

While exploring the question of Britain's coastal length, Mandelbrot initially used three measures: total length (L), scale or "ruler" length (λ), and number of rulers (N) (Equation 4).

$$L = N\lambda \tag{4}$$

As shown in Figure 2.9, a varying ruler length causes a change in the number of rulers as well as a change in the total length. This is because shorter rulers are able to capture more detailed features of the coast. In fact, as the ruler length decreases, the length of the coast increases – eventually to infinity – as ruler length tends to zero.

Since coastal length is dependent on ruler length and is therefore an arbitrary measure, there must be another parameter to effectively describe these features. Building on the work of Richardson (Richardson 1961), Mandelbrot realized that there is often a correlating parameter that appears when examining a range of ruler lengths.

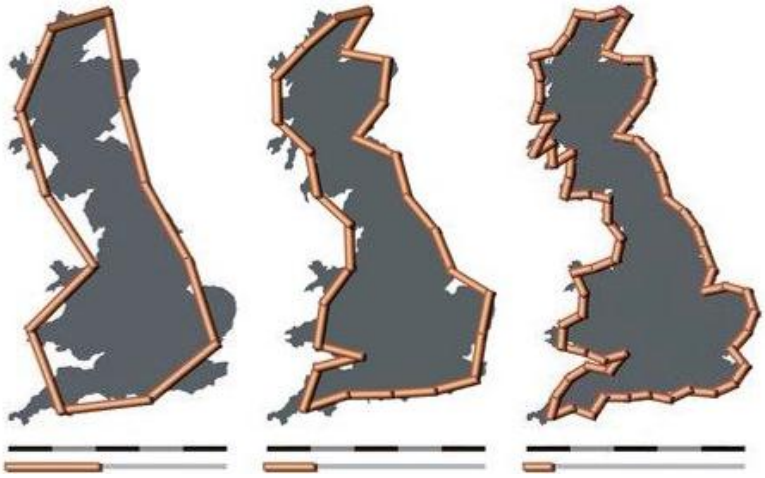


Figure 2.9: Illustration of the effect of ruler length on total length measurement of a coastline (in this case Britain). The image on the left shows a measurement using a very long ruler, which is unable of capturing fine features of the coastline. As the ruler length decreases, more features are captured and thus the measured coastal length increases (Wikipedia).

That parameter (D) appears in a power law function relating ruler length and total coastal length (Equation 5).

$$\mathrm{L}(\lambda) = \lambda \mathrm{N}(\lambda) \sim \lambda^{1-\mathrm{D}} \tag{5}$$

For some objects, the parameter D is the same for all values of λ. In such a case, it is a convenient measure for describing the geometric complexity of the object. Since D is often a fractional value, Mandelbrot termed it the “fractal dimension”.

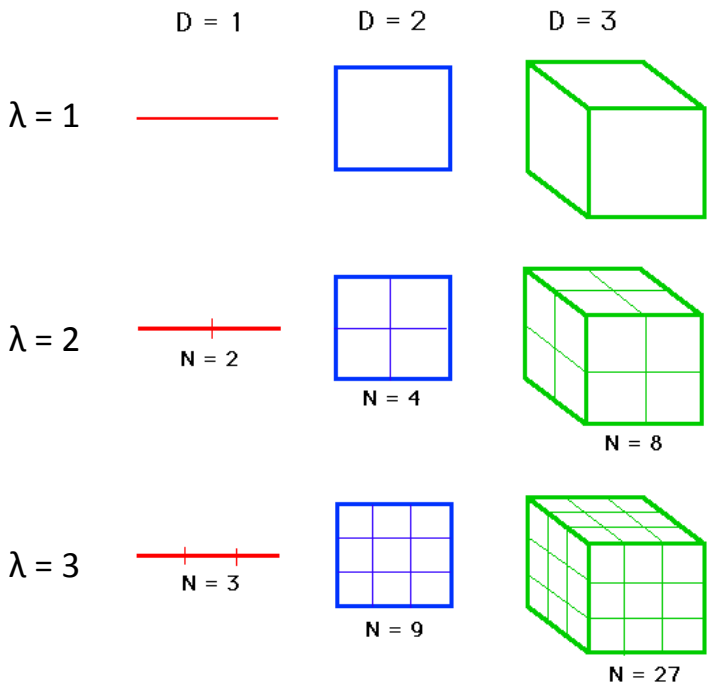
There are other objects, however, that cannot be described by a single fractal dimension value since they display different characteristics at different scales. Realizing this, Catrakis (Catrakis 2000) suggested a hierarchy of geometric complexity:

- Level 1: complexity only at a single scale – Euclidean geometry

- Level 2: complexity is the same at all scales – power law behavior (self-similar fractal)
- Level 3: complexity varies with scale – scale-dependent geometry (multi-fractal)

2.3.3 Fractal Dimension and Scale

Objects that display complexity at a single scale (Level 1 complexity) are merely common geometric objects with no fractal behavior. Level 2 complexity, however, is the trademark of many mathematically-created fractals. Figure 2.10 shows how a few different simple objects can be divided into self-similar parts. Note that the dimension value stays the same for each object regardless of the scale that is used.



$$N = \lambda^D$$

Figure 2.10: Images showing relationship between scale (λ), number of objects (N), and dimension (D) for three simple self-similar objects (Vanderbilt)

These objects are said to be self-similar since they are divided into smaller parts that are similar (in the geometric sense of the word) to the whole object. With this knowledge, many other shapes have been created that are also self-similar, but have non-integer dimension values. An example of one of these fractals is the Koch snowflake (Figure 2.11), which is composed of three contiguous Koch curves. This fractal is created by starting with an "initiator" and then performing an operation on it called the "generator". Here, the initiator is merely a line segment and the generator involves removing the middle third of that line segment and adding two more line segments, the length of each is also one third of the initiator. This creates four new line segments, which all become initiators and the generator acts on them, and the process iterates indefinitely. Since each generator creates 4 new line segments, each of length 1/3 of the initiator, the dimension of the Koch snowflake is $\mathrm{D} = \frac{\log(\mathrm{N})}{\log(\lambda)} = \frac{\log(4)}{\log(3)} = 1.26$. Similarly, the Sierpinski Triangle (Figure 2.12) starts with a solid triangle as the initiator, and then the generator removes a triangle from it, creating 3 new triangles of side length 1/2 of the original. Again, this process repeats indefinitely, yielding a dimension value of $\mathrm{D} = \frac{\log(\mathrm{N})}{\log(\lambda)} = \frac{\log(3)}{\log(2)} = 1.585$. Initiators and generators are used to create a number of other fractals with known dimension values.

The highest level of geometric complexity proposed by Catrakis is that of varying complexity at different scales, represented by multi-fractals. Almost all fractal objects found in nature are multi-fractals, such as Britain's coastline. For example, a bird's-eye-view of this coast from an airplane may appear much differently than it does if one is walking along the shore. However, even these types of objects often exhibit what is

known as statistical self-similarity. That is, for sufficiently long sections of coastline, there is a statistical similarity between features across a range of scales.

2.3.4 Applications of Fractal Theory

Fractal geometry has been used in a variety of applications to help describe or classify complex objects. For example, Zubair and Catrakis (Zubair and Catrakis 2009) used fractals to describe turbulent scalar interfaces while Iftekharuddin (Iftekharuddin, Jia et al. 2000) studied the benefits of fractals for detection of brain tumors via magnetic resonance imaging (MRI). In many cases such as these, there must be a means for determining the fractal dimension of a shape. While there are many methods available to do this, one of the most common is known as the box-counting algorithm.

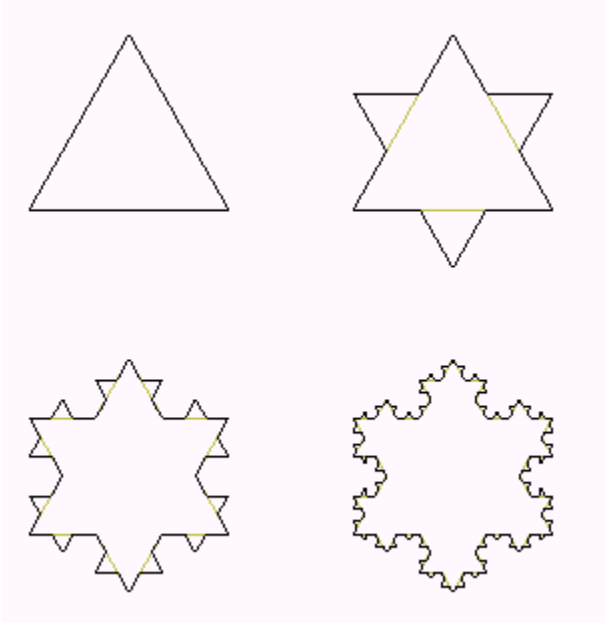


Figure 2.11: The first four iterations of the Koch Snowflake. At each iteration, a triangular protrusion is added to each line segment, and the middle third of each line segment is removed (knowledgerush)

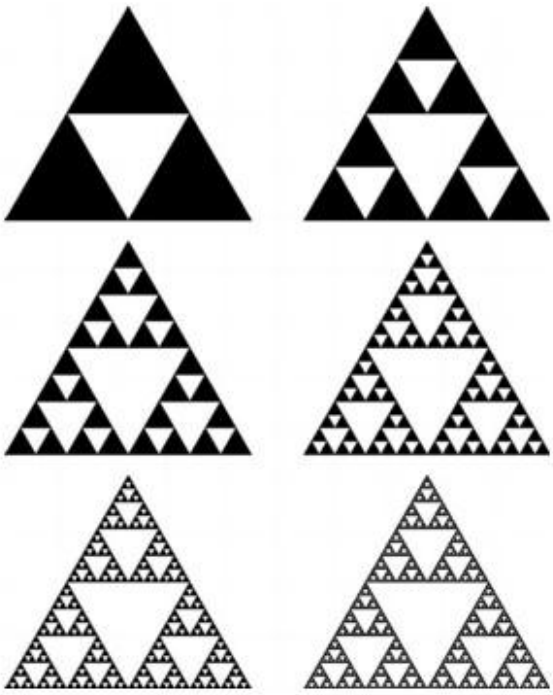


Figure 2.12: The first six iterations of the Sierpinski Triangle. At each iteration, a triangle (white) is removed from each existing triangle (black) (mathaware)

The box-counting algorithm, as its name suggests, involves dividing a space into a grid of boxes and then counting the number of boxes that contain at least some part of the (fractal) object of interest. If an object is embedded in two dimensions, meaning that it is constrained to a single plane, then the grid of "boxes" is technically a grid of squares. By employing a range of different grid resolutions, one can determine the fractal dimension of an object from equation 6. Figure 2.13 is a graphical demonstration of how this algorithm works.

$$\mathrm{D} = -\frac{\mathrm{dlog(N)}}{\mathrm{dlog}(\lambda)} \tag{6}$$

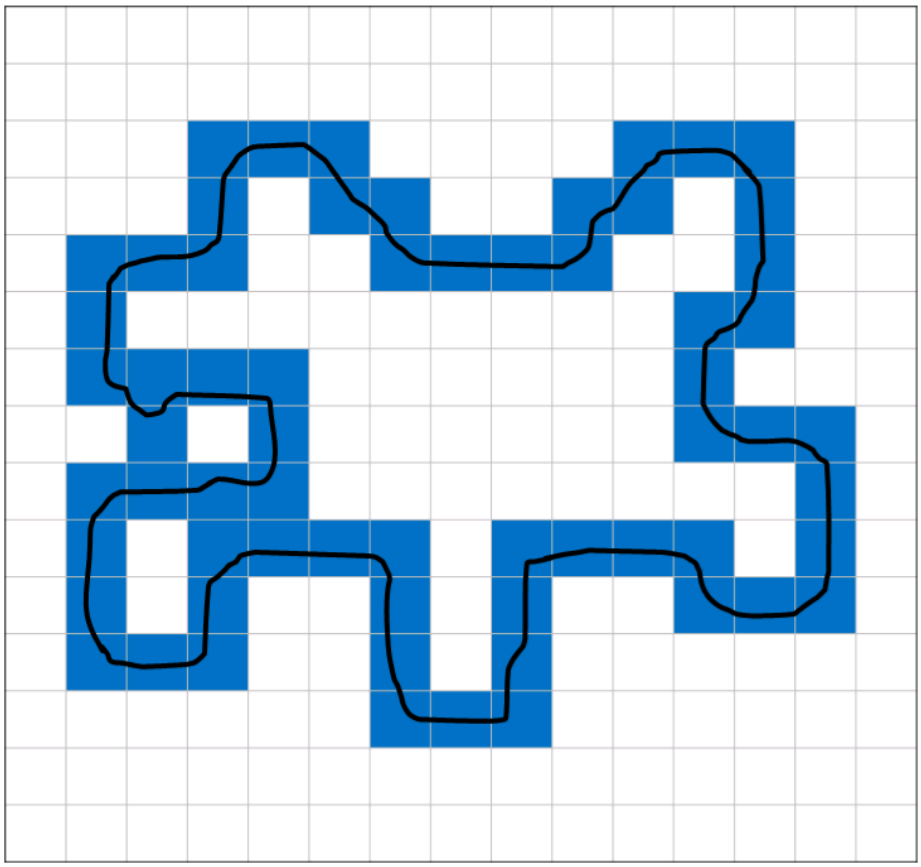


Figure 2.13: Illustration of the box-counting algorithm. The space containing a fractal object (dark black line) is divided into a grid of squares of side length λ. The boxes containing some part of the fractal object are shaded in blue (N=number of blue boxes). Many different values of λ must be used in order to determine the fractal dimension.

2.4 Objectives

The objectives of this thesis are:

- To extract and mesh a three dimensional model of human upper airways for three different patients
- To elucidate fluid mechanics phenomena present in each airway model and validate results through grid and time step convergence tests as well as comparison to previously published studies
- To develop a custom box-counting algorithm that can determine the fractal dimension of any object in three dimensional space and validate this algorithm through the use of mathematically created fractals with known dimension values
- To determine the fractal dimension and volume of the interior surface of the human left ventricle at ten different time steps throughout the cardiac cycle

3. METHODS

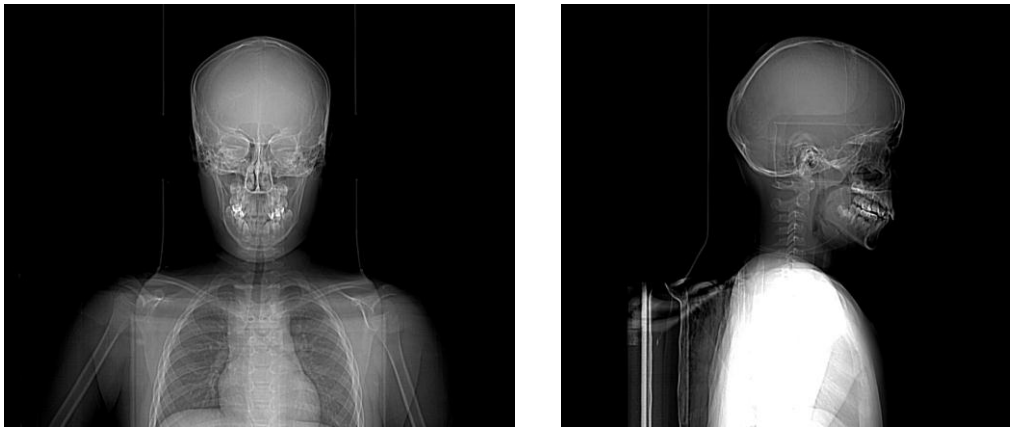
The methodologies used in this research will be explained first for research conducted on the respiratory airways and then for the study of left ventricular geometry. For the airways, 3D models were generated for three different patients. These models were meshed and then used for CFD simulations. Accuracy of simulations was verified based on grid and time step sensitivity studies and 2D slices were extracted at various anatomical locations for post processing. In the study of the left ventricle, more 3D models were created for a single patient during different times of the cardiac cycle. A custom box-counting algorithm was developed and applied to these models in order to determine fractal dimension of the ventricular surface.

3.1 Airway Methods

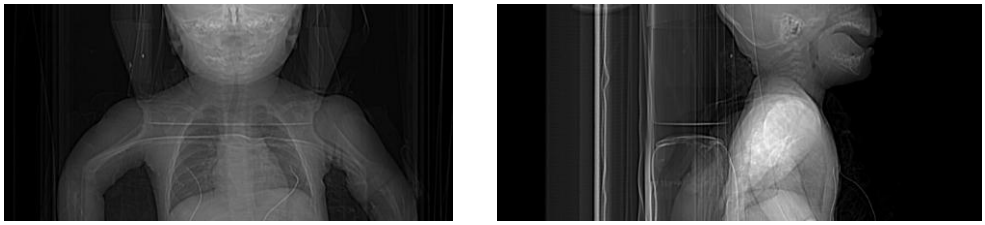
3.1.1 Geometry Construction

Human Computed Tomography (CT) data was acquired from collaborator Dr. Kristen Baugnon in the Radiology Department at Emory University, Atlanta GA. The data acquisition protocol was approved by the Institutional Review Board of Emory University. Three dimensional finite element models were extracted from this CT data for three different patients: an 8-year-old child (Patient A), a 7-month-old infant (Patient B), and a 3-year-old child with an intubation child (Patient C). Figure 3.1 shows the scout images for each patient. All imaging was conducted at Children's Healthcare of Atlanta using

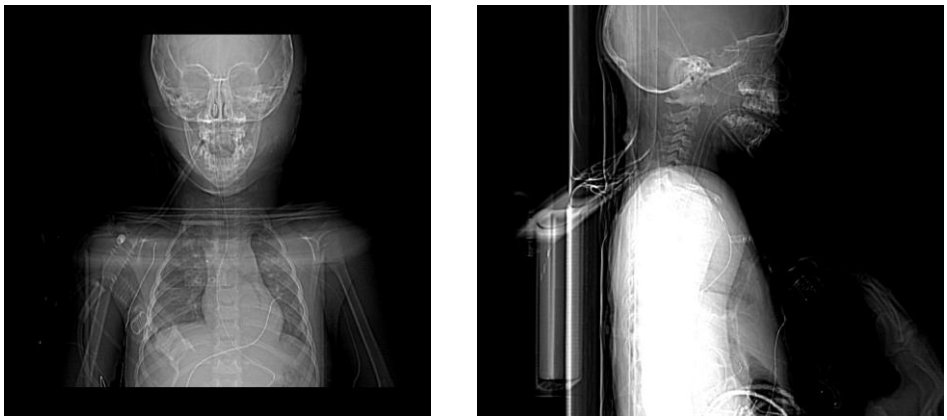
GE Medical Systems LightSpeed VCT scanners with varying amounts of Optiray contrast applied via intravenous therapy (IV). The slice thickness in each case was 0.625 mm. Table 1 is a more detailed summary of the imaging parameters that were used for each patient.



(a)



(b)



(c)

Figure 3.1: Scout images for Patients A (a), B (b), and C (c)

Table 1: CT imaging parameters

Patient	Gender	Age	Weight (lb)	Image Resolution/Pixel Spacing (x mm, y mm)	Acquisition Year	Device Manufacturer	Institution	Contrast Agent
A	M	8y	77	0.46875, 0.46875	2009	GE Medical Systems	CHOA Egleston	76 CC Optiray
B	M	7m	---	0.351562, 0.351562	2008	GE Medical Systems	CHOA at SR	14 CC Optiray
C	M	3y	20	0.488281, 0.488281	2009	GE Medical Systems	CHOA Egleston	40 CC Optiray

Patient (cont.)	Slice Thickness (mm)	Exposure Time (ms)	Rows/Columns	Image Presentation Group	Software Versions	Finite Meta Elements Group Length	Device Model
A	0.625	400	512/512	182	07MW18.4	194	LightSpeed VCT
B	0.625	500	512/512	182	07MW18.4	194	LightSpeed VCT
C	0.625	400	512/512	182	07MW18.4	Variant	LightSpeed VCT

The software Mimics, by Materialise NV (Materialise), was used to generate 3D models from the CT scans. Initially, images were segmented using a thresholding technique to include all portions of the scans that were occupied by air. Unwanted material, such as air in the sinuses, oral cavity, and outside the body, was then cropped out (Figure 3.2). The mouth and nasal sinuses were not included because they do not conduct air during normal respiration. Previous models (Wang, Liu et al. 2009) were consulted to verify proper geometry and local thresholding was applied to add or remove features. Region growing was utilized to ensure a continuous model and wrapping and smoothing tools eliminated any small holes or sharp points from the final geometry. Figure 3.3 shows the final 3D models for each patient studied.

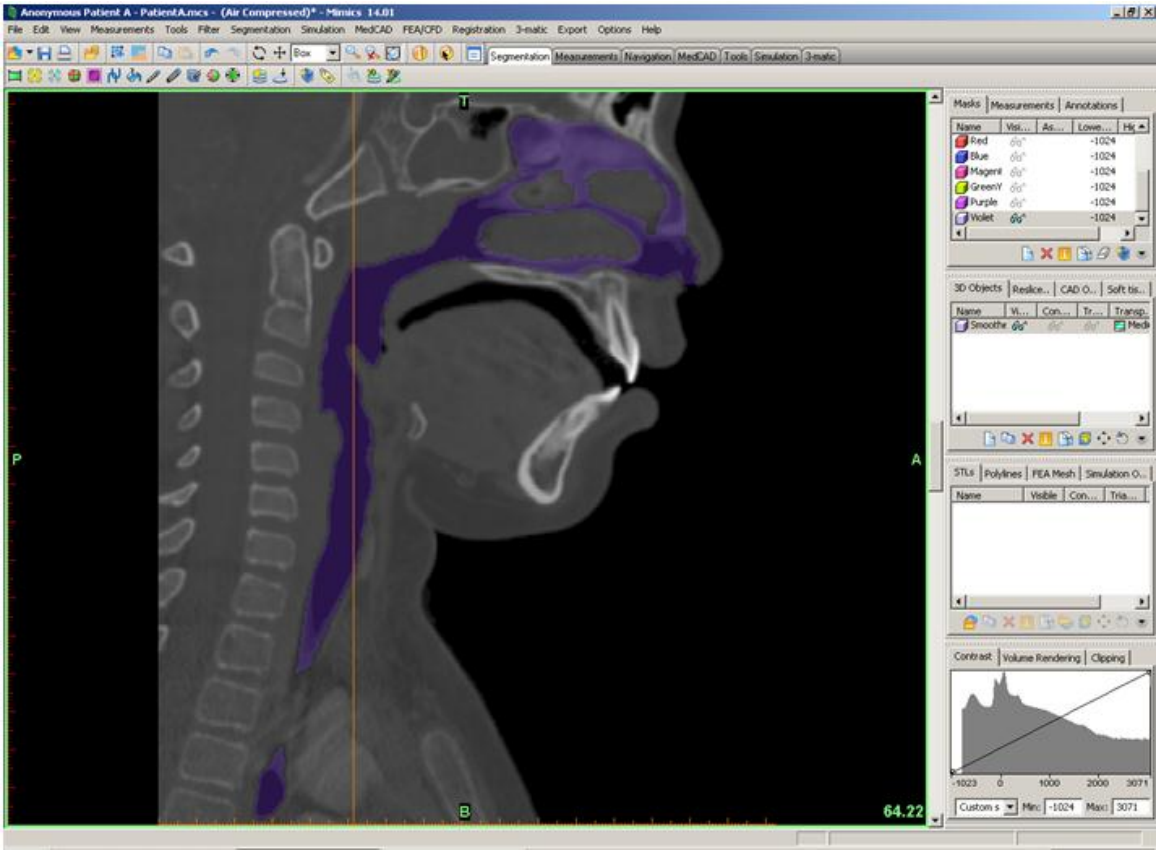
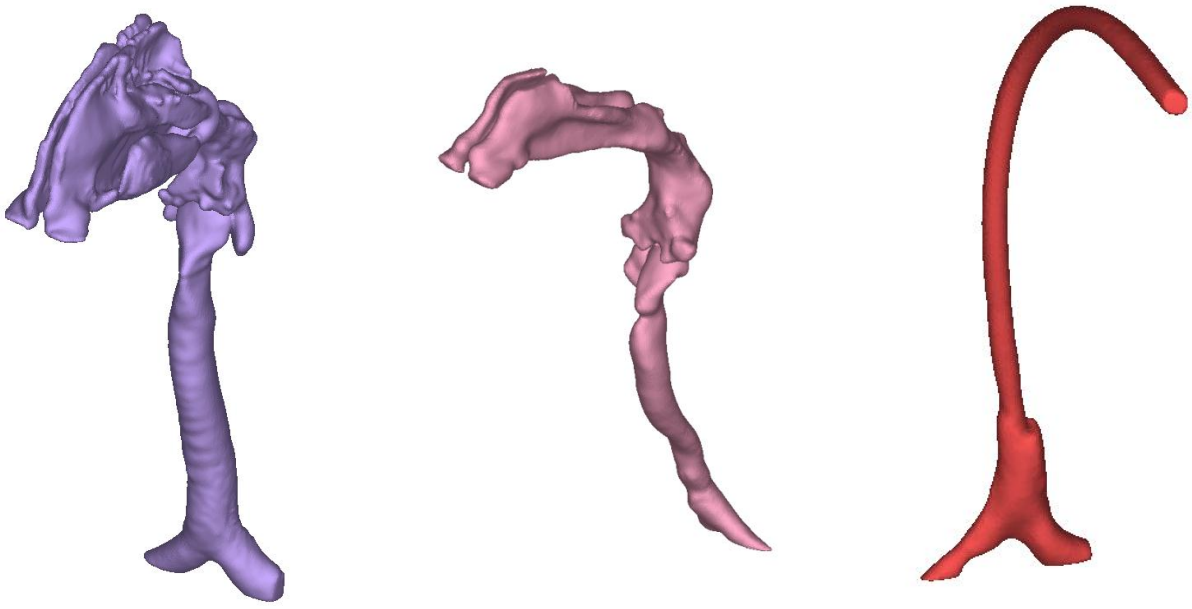


Figure 3.2: Screenshot of the Mimics user interface. Thresholding was used to segment out black and very dark grey areas while cropping and region growing eliminated unwanted features (e.g. air outside the body, mouth, sinuses)



(a) (b) (c)

Figure 3.3: Images of 3D models for (a) Patient A (child), (b) Patient B (infant), and (c) Patient C (child with intubation tube). *Not to scale

3.1.2 Mesh Generation

An initial surface mesh was created for each model using 3-matic, a CAD/meshing software package available with Mimics. These surface meshes were imported into ANSYS TGrid, which was used to create tetrahedral volume meshes. The meshes created for each patient are shown in Figure 3.4. Elements in the unstructured tetrahedral grid were generated from a surface mesh consisting of elements with an aspect ratio of no less than 0.4. From this a volume mesh was automatically created and then refined for better quality elements. Table 2 shows the number of elements comprising each model and the quality of their worst elements.

Table 2: Mesh information for all patients

Patient	Number of Cells	Minimum Orthogonal Quality
A	224787	0.16
B	87039	0.14
C	37495	0.20

Figure 3.4: Surface mesh images for Patients A (left), B (middle), and C (right)

Orthogonal quality of an element is defined using three vectors:

a) A vector from the cell centroid to the centroid of a face
b) The normal vector of a face
c) A vector from the cell centroid to the centroid of an adjacent cell, which shares the face used in b)

The normalized dot product is taken between a) and c) as well as between b) and c) for each face and the lowest value is given as the orthogonal quality of the element. Values range from 0 to 1, where those near zero are of poor quality and values near one are of good quality.

3.1.3 CFD Simulations

One of the first steps in setting up a CFD simulation is to determine what kind of model should be used. If turbulent flow is expected, there are a number of options that can be chosen. A decision should be made based largely on the expected Reynold's numbers as well as the geometry of the model. In this case, the standard k-ε model was chosen in order to provide an accurate solution without excessive computational expense (Ansys 2010). This model is good for turbulent flows driven by relatively small pressure gradients and has been used extensively due to its robust performance. Enhanced wall treatment was also used to capture the near wall flow at narrow regions.

Boundary conditions are another important aspect of CFD setup. They are generally chosen based on empirical data and previously known flow mechanics. Since airflow in the respiratory system is driven by a pressure gradient due to movement of the

diaphragm, a sinusoidal pressure outlet condition (Figure 3.5) was assigned at the tracheal bifurcation (which was the most proximal position to the lungs in these models). Similarly, a zero gauge pressure (atmospheric) inlet was assigned at the nostrils to mimic actual conditions (Figure 3.6). A study by Xi et al. (Xi, Si et al.) was used to determine the magnitude of pressure for Patients A and B. Equation 7 was used to calculate the pressure drop for Patient C so that the mass flow rate was equal to an average value for a 3-year-old child.

$$\Delta P = f\frac{L}{D}\frac{\rho V^2}{2} \tag{7}$$

Additional constraints on the model included a no-slip condition (zero velocity) at the walls and an assumption of incompressible air since pressure gradients were expected to be relatively small. A turbulent intensity of 1% was prescribed at both inlet and outlet for each case as well as a hydraulic diameter of 0.01 m at all inlets and 0.02 m at all outlets. Gravity was expected to have negligible effects so it was not included.

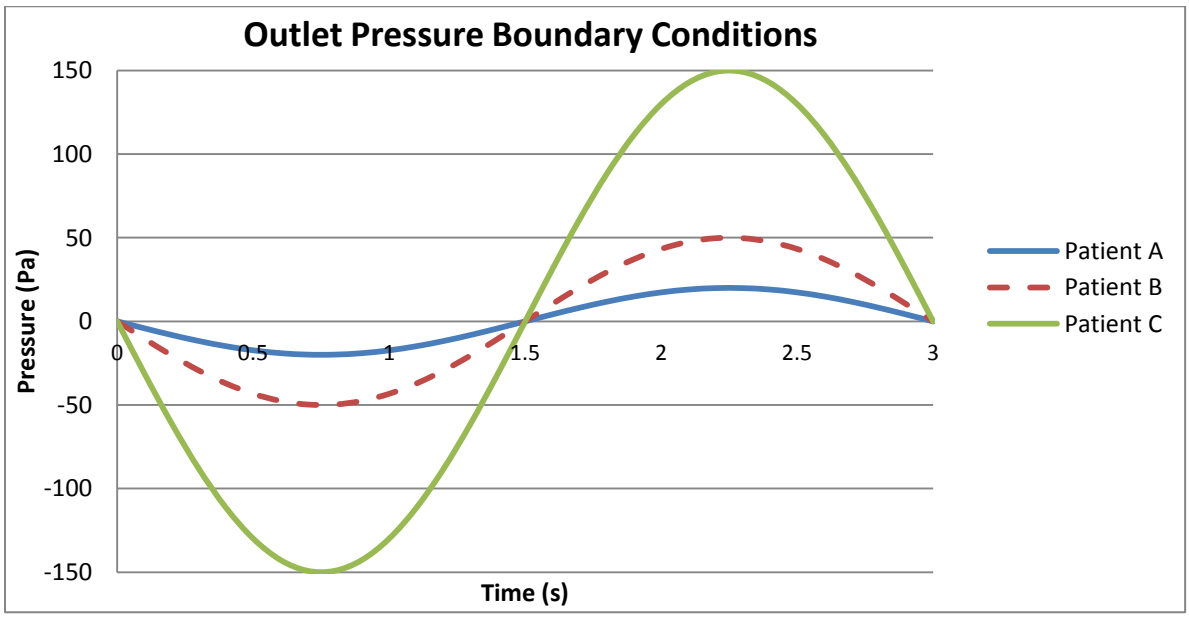


Figure 3.5: Prescribed pressure at tracheal bifurcation (outlet) over time. Pressure magnitude for Patient A = 20 Pa, Patient B = 50 Pa, and Patient C = 150 Pa. Pressure for Patient C is much higher due to the relatively narrow width of the intubation tube.

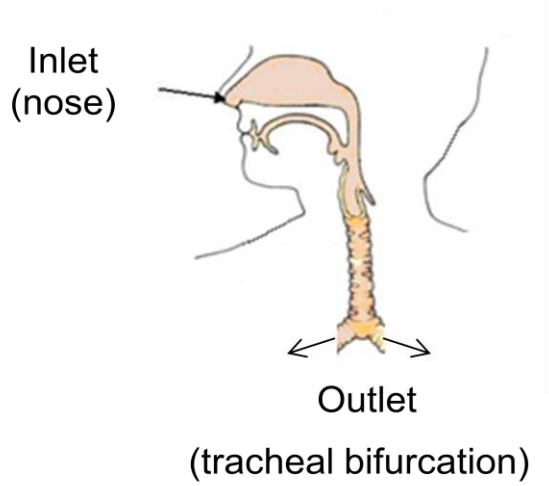


Figure 3.6: A zero gauge pressure inlet was prescribed at the nose and a sinusoidal pressure outlet was prescribed at the tracheal bifurcation

3.1.4 Sensitivity Studies

In order to help verify the accuracy of simulations, the maximum Reynolds number was calculated for the trachea/intubation tube of each patient. As shown in Table 3, these values are all within the same range. In addition, time step (Table 4) and grid (Table 5) sensitivity studies were run for each model. For each test, a constant pressure outlet condition equal to that at maximum inspiration was imposed and the simulation was run for 0.4 s. Results shown correspond to the end of the 0.4 seconds. It can be seen that the largest difference in peak velocity between any two trials on the same model was about 2.6%. Since this is a very small difference, it was determined that the initial grid and time step sizes were adequate.

Table 3: Maximum Reynolds number calculations for all patients

Patient	Tracheal/Tube Diameter (cm)	Maximum Velocity (m/s)	Reynolds Number
A	1.1	3.94	2677
B	0.7	6.63	2867
C	0.4	8.89	2197

Table 4: A time step sensitivity study was conducted to determine if the time step was small enough to capture the flow. The largest difference was just over 2%

Patient	Time Step Size (s)	Max Velocity (m/s)	Percent Difference
A	0.01	3.940	-----
	0.001	3.952	0.33
B	0.01	6.630	-----
	0.001	6.641	0.17
C	0.01	8.891	-----
	0.001	9.074	2.05

Table 5: A grid sensitivity study was conducted to determine if the mesh resolution was adequate. Results show that the greatest difference in peak velocity between any two grid resolutions was just over 2.5%

Patient	Number of Cells	Max Velocity (m/s)	Percent Difference
A	87039	6.630	-----
	107423	6.621	0.12
	131510	6.615	0.23
B	224787	3.940	-----
	306540	4.036	2.6
	397484	4.011	1.8
C	37495	8.891	-----
	59048	8.876	0.17
	75813	8.831	0.67

3.1.5 Post Processing

Results were exported from ANSYS Fluent every 0.05s and then imported into Tecplot 360 for post processing. In order to visualize the flow at different locations, two dimensional slices were extracted from each of the models (Figure 3.7). This was done using arbitrary planes that cut through the regions of interest, which included the tracheal bifurcation, larynx, and nasopharynx. Since Patient C was intubated, only a slice at the bifurcation was taken since flow in the tube was not of interest. Arrows are shown to represent the viewing angle for each set of slices. The three main parameters of interest were pressure, velocity, and wall shear stress. Contour plots were created for pressure and wall shear stress and vector plots colored by magnitude were created for velocity. Animations were also recorded for all of inspiration and expiration and can be found in the appendix.

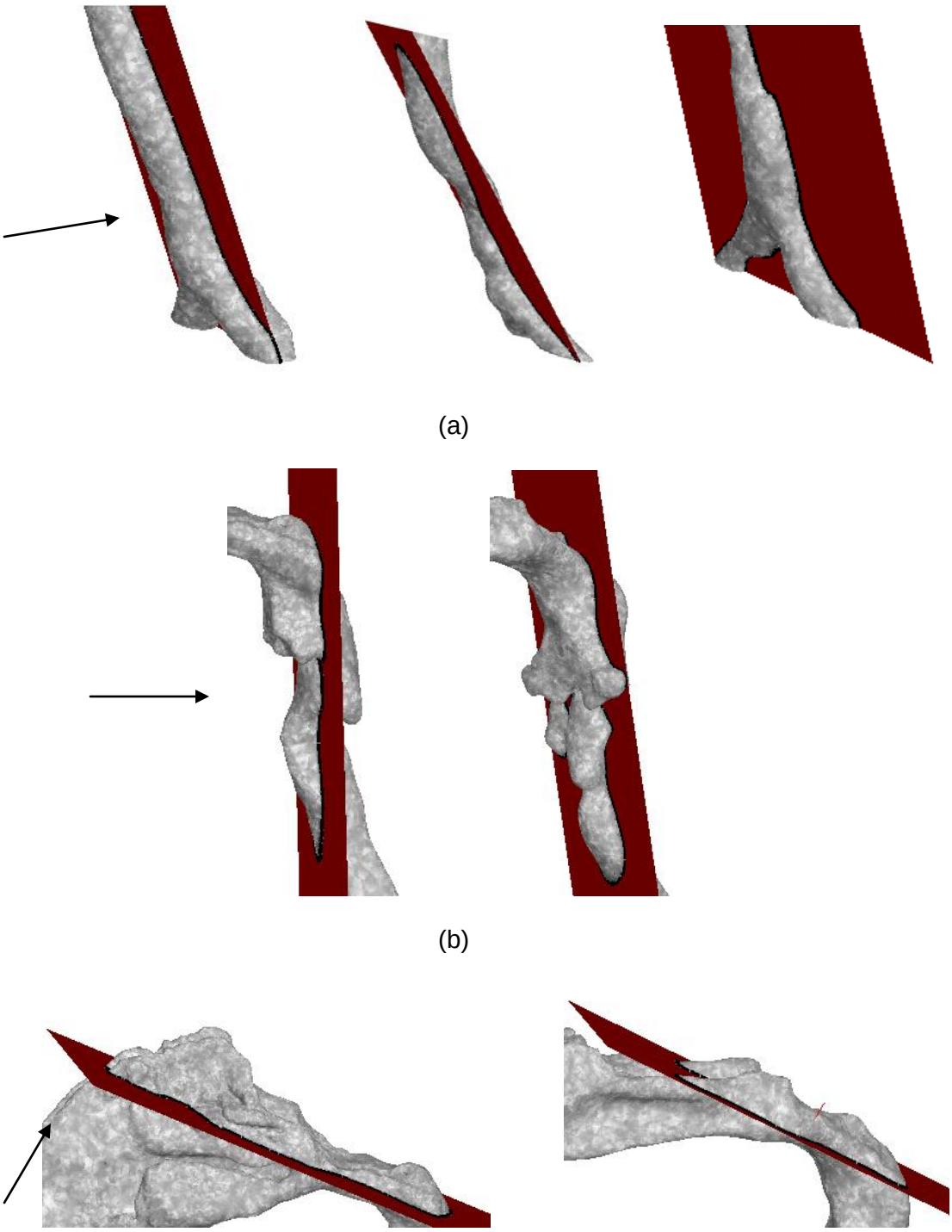


Figure 3.7: Planar slices were extracted at three anatomical locations: a) bifurcation – for Patients A (left), B (middle), and C (right); b) larynx – for Patients A (left) and B (right); and c) nasopharynx – for Patients A (left) and B (right). Arrows indicate the viewing orientation.

3.2 Heart Methods

3.2.1 Model Generation

Human CT data was acquired from collaborator Dr. John Oshinski in the Radiology Department at Emory University, Atlanta GA. The data acquisition protocol was approved by the Institutional Review Board of Emory University. An adult male patient 51 years old was recruited and CT was performed per coronary protocol to image the heart. A Siemens Definition scanner acquired slices every 2 mm at a pixel resolution of 0.7695 mm (in x and y). The whole ventricle was scanned at ten evenly spaced time steps during one cardiac cycle. Since the patient had a heart rate of 91 beats per minute, this means that time steps were 66 ms apart.

Mimics software was again used to extract the 3D geometry of interest, in this case the interior surface of the left ventricle. A thresholding technique was implemented to capture the surface of blood, which appears white in Figure 3.8 due to a contrast agent. Similar protocol to that in the airway study was followed to generate the 3D model, including cropping, region growing, and wrap and smooth functions. In this case, however, a surface mesh (Figure 3.9) was created rather than a volume mesh.

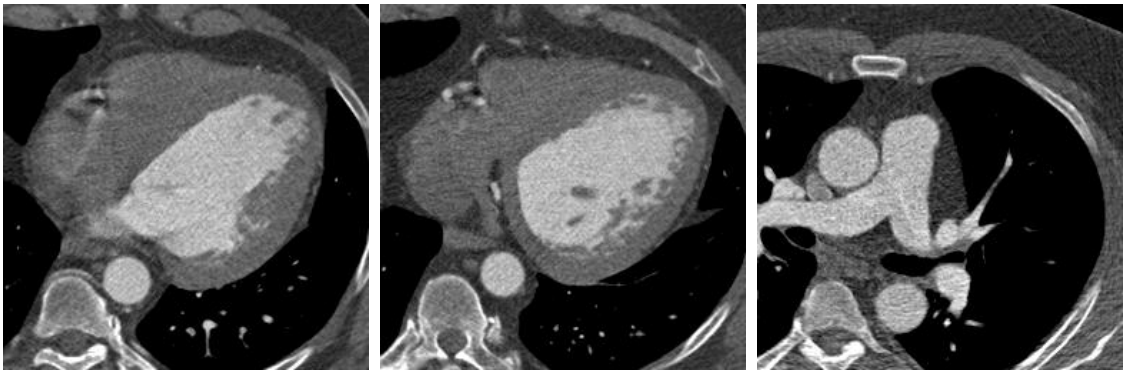


Figure 3.8: CT scan images of heart at different locations. Contrast shows blood as a very light shade.

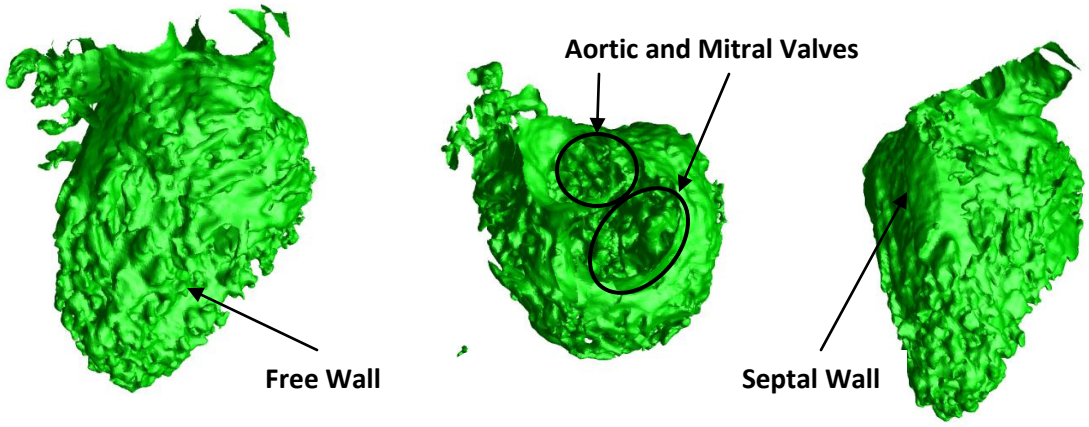


Figure 3.9: 3D Images of left ventricle created in Mimics. Different views show the various features.

3.2.2 Box-counting Algorithm

While there exist many methods for determining fractal dimension, the box-counting algorithm is one of the most widely-used due to its robust nature and ease of implementation. A custom algorithm was developed and executed in MATLAB to determine the dimension of the inner surface of the left ventricle at different times during the pumping process. Tecplot 360 was also used to aid in visualization of the geometrically complex surface.

One of the first steps in developing a box-counting algorithm is to determine an appropriate bounding box size for the fractal object of interest. This bounding box must then be divided into a grid of smaller boxes of side length λ, which is referred to as scale. In this case, boxes were always assigned coordinates of integer values such that there were two coordinate systems: a real coordinate system and box coordinate system. The box system was used to merely divide space into finite-sized pieces. For example

all points with x-coordinates between zero and λ were assigned to box 1 in the x-direction. All points with x-coordinates between λ and 2 λ were assigned to box 2 in the x-direction, and so forth (Figure 3.10). Using this system, all points on the fractal object were assigned to a box in the x-, y-, and z-direction.

After systematically assigning all points on the fractal object to boxes (with some points sharing the same box), the boxes containing one or more points were summed. This yielded an N value, which was recorded along with the corresponding λ value. The scale (λ) was then changed and another count of the object-containing boxes (N) was conducted. After a large range of scales was tested, the fractal dimension was computed for each of these scales. Note that the fractal dimension is determined by taking a derivative, so the dimension at a certain scale is actually the dimension between two (or more) scales that are close together.

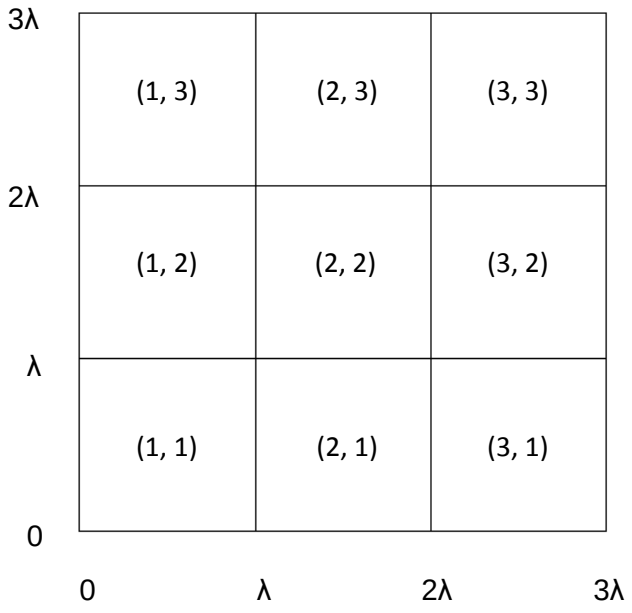


Figure 3.10: Illustration of how boxes are labeled based on λ. A box is occupied if any part of the fractal object lies within it.

3.2.2.1 Volume Calculation

Because this study concerns the mechanics of pumping, it was necessary to calculate the volume of the ventricle as well as its fractal dimension. Therefore, the box-counting algorithm was modified to count the number of boxes that would fit inside an object in addition to the number that resided on the surface of the object. To do this, an approach of filling the object from the inside-out was taken. A box was placed near the geometric center of the object and then each adjacent box was checked to see if it contained part of the surface. If a box did contain part of the surface, it was marked as such and no further action was taken. However, if a box did not contain part of the surface, it was marked as a "volume box" and then all of its adjacent boxes were checked. This process continued until there were no boxes left to be checked. Figure 3.11 shows how this algorithm works for a two dimensional grid.

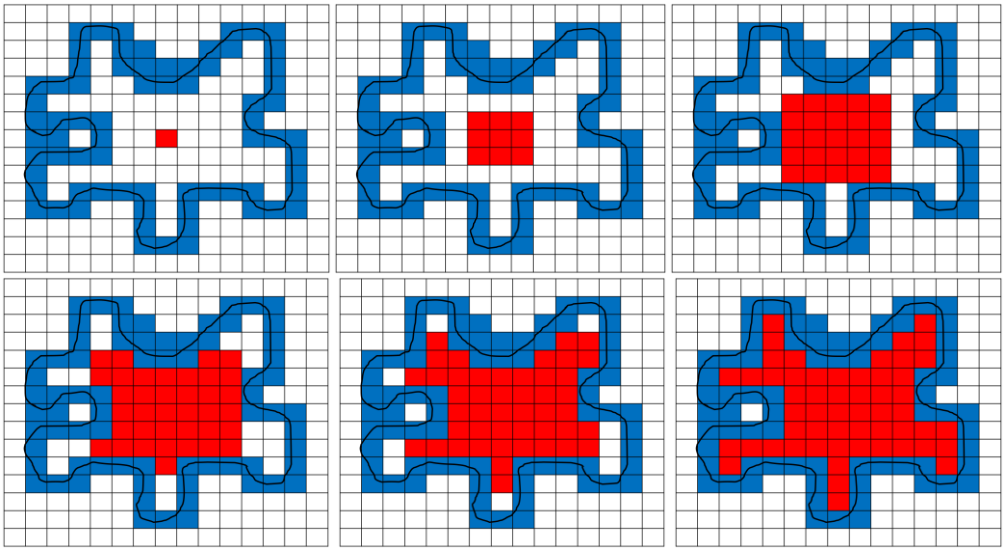


Figure 3.11: The bounding box around a fractal object (solid black line) is divided into a grid. Boxes touching the object are shaded in blue and boxes inside the object are shaded in red. All other boxes are shaded white.

There were three different possible box assignments:

1) Empty – shaded in white, designated as “0”

2) Surface – shaded in blue, designated as “1”

3) Volume – shaded in red, designated as “2”

3.2.2.2 Validation of Algorithm

The custom box-counting program that was written was validated through the use of mathematically-created fractals with known dimension values. This was done for a curve embedded in two dimensions – the Koch Snowflake (Figure 2.11, above) – and a surface embedded in three dimensions – the quadratic Koch surface. The true dimension of the Koch snowflake is 1.26 (explained in Background section) and the true dimension of the Koch surface is 2.33 ($\mathrm{D} = \frac{\log(\mathrm{N})}{\log(\lambda)} = \frac{\log(13)}{\log(3)} = 2.33$). A graphical representation of the first three iterations of the Koch surface is shown in Figure 3.12.

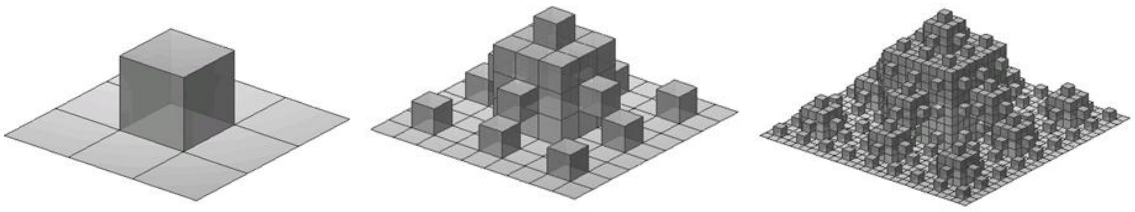


Figure 3.12: First three iterations of the Koch surface. At each iteration, a box is added to the middle of each square surface, thus creating 13 new square surfaces of 1/3 the previous length.

Since the box-counting algorithm can only be applied to a physical object and does not take into account mathematical formulation, it must be tested on each fractal for a significant amount of iterations. As the number of iterations increases, the fractal

appears to take on more of its “true” form, however rendering the exact object is not possible since both of these fractals are iterated indefinitely. In addition, each iteration causes an exponential increase in the complexity of the object, so it is very computationally expensive to generate fractals with large numbers of iterations. This is especially true for the Koch surface, since it requires many more data points to fully describe it than the Koch curve does for the same number of iterations. Because of these limiting factors, the custom box-counting algorithm was tested on the Koch curve up to six iterations (Figure 3.13) and on the Koch surface up to four iterations (Figure 3.14).

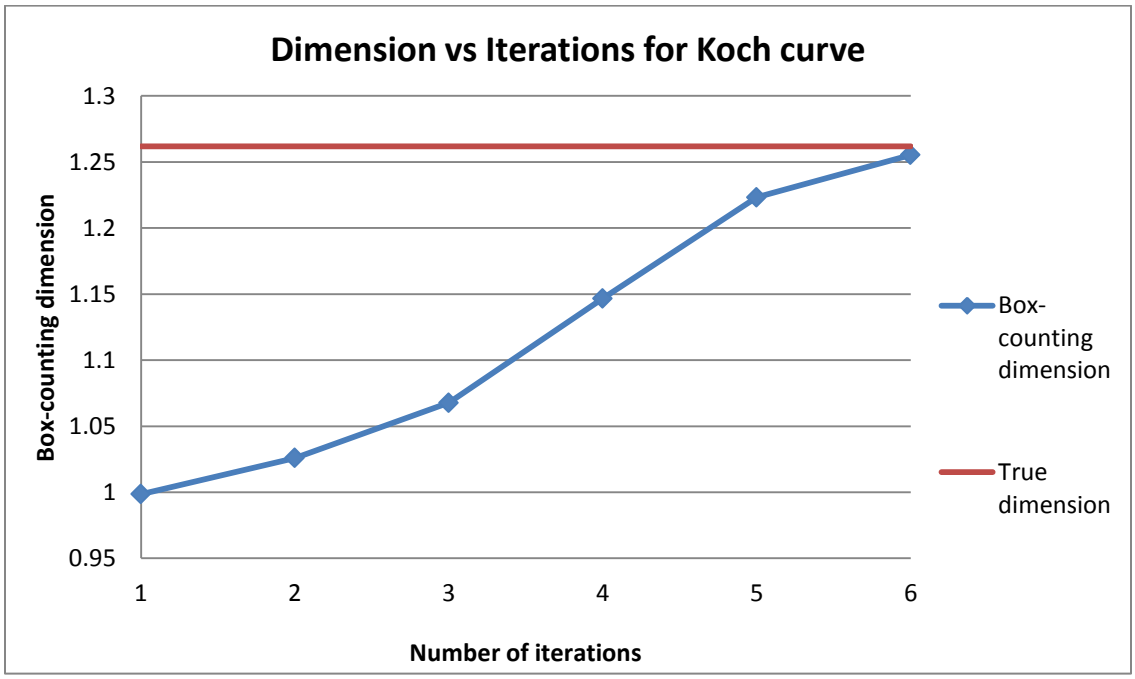


Figure 3.13: Box-counting algorithm was tested on the Koch curve to determine its validity for objects embedded in two dimensions. As expected, the box-counting dimension approaches the true dimension as the number of iterations increases.

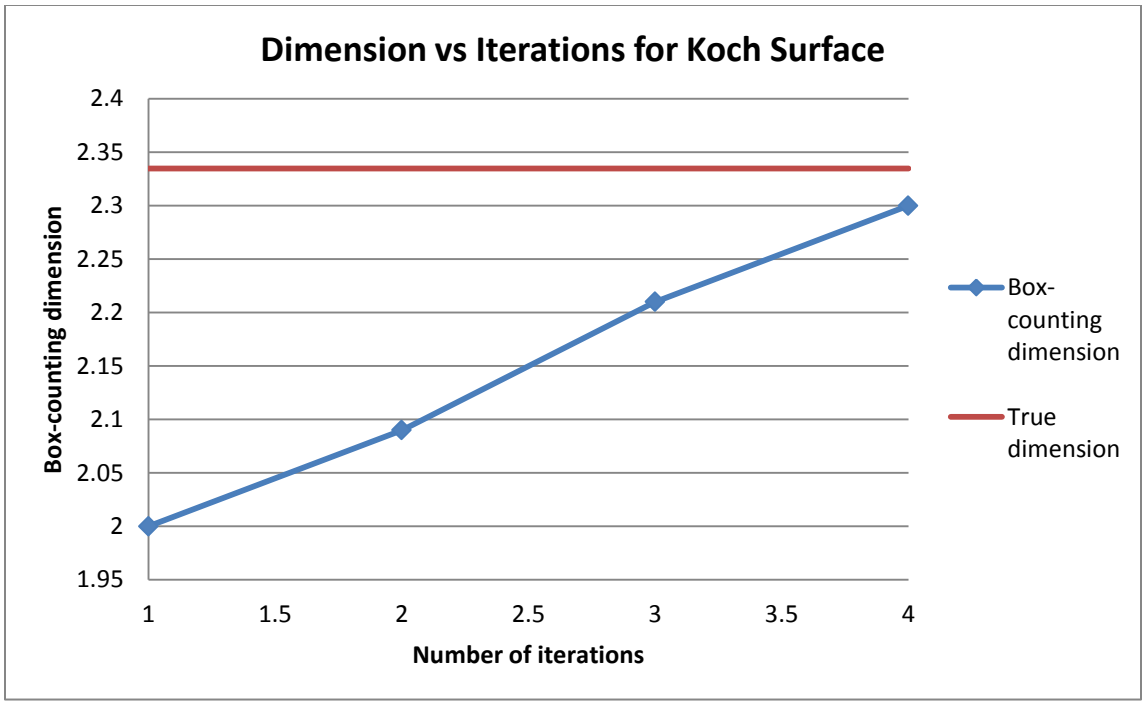


Figure 3.14: Box-counting algorithm was tested on the Koch surface to determine its validity for objects embedded in three dimensions, such as the left ventricular surface. Less iterations were used than on the Koch curve due to computational limitations of fractal generation, but the true dimension is still approached after four iterations.

4. RESULTS AND DISCUSSION

Results will be presented and simultaneously discussed for the airway simulations first and then for the fractal analysis of the left ventricle. However, additional discussion is included at the end of each section to demonstrate validation and explore broad parameters. Such parameters include the effects of age and intubation on airflow in the respiratory tract as well as implications of change in geometry on volume change and biomechanics of the left ventricle.

4.1 Airway Results and Discussion

Based on observations of results at many different locations spanning the entire airway model, three regions of interest were chosen. These regions – the tracheal bifurcation, larynx, and nasopharynx – all demonstrate unique flow phenomena which are largely based on their structure. Moreover, 2D fields are presented as they are more suited for direct comparisons.

4.1.1 Bifurcation

Results are shown at the tracheal bifurcation of all patients, with 2D slices taken along the axis of the trachea (as shown in Figure 3.7 in the Methods section). Contour and vector plots are shown for pressure (Figure 4.1), velocity (Figure 4.2), and wall shear stress (Figure 4.3) during times of peak inhalation (a) and exhalation (b). The bifurcation displayed a wide range of results, largely due to the different pressure gradients

necessary to drive the flow but also due to variations in structure. It can be seen that Patient A has a very regular structure. The diameter of the trachea is fairly constant and the branching of the bronchi is very symmetric. Patient B, on the other hand, is not as developed and therefore the airway in this case shows much more irregularity. This led to the asymmetric pressure gradient that can be seen for Patient B, as well as higher velocities and max shear stresses than Patient A.

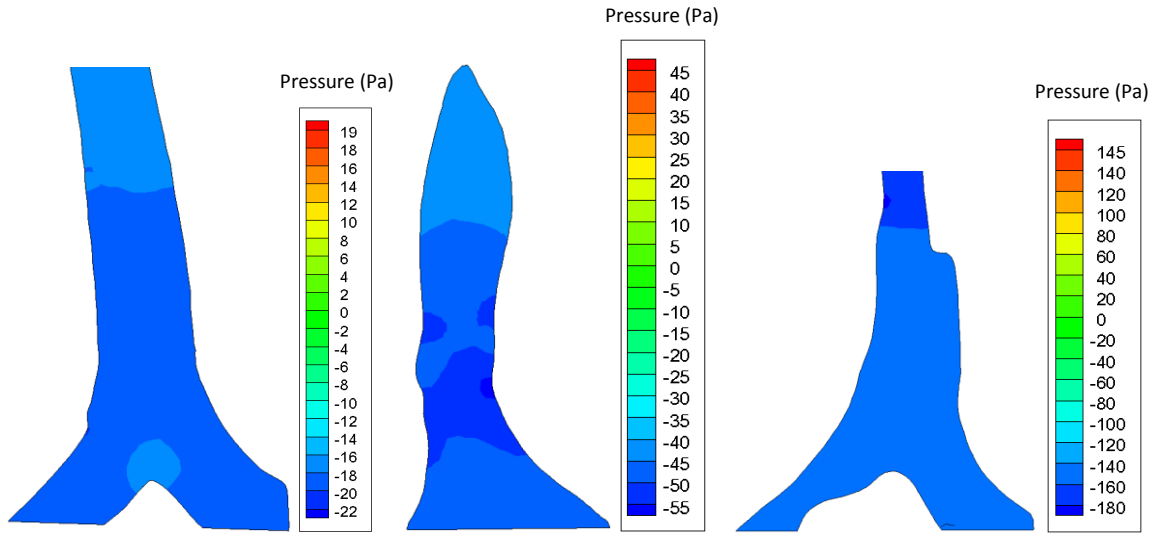
Another difference in structure is caused by the presence of the intubation tube in Patient C. This tube terminates just above the bifurcation which causes a sharp change in diameter. Because of this, velocities are much lower outside of the tube but there is relatively high wall shear stress at the bifurcation during inspiration. Likewise, high wall shear stress occurs on the bronchi during expiration due to the large pressure gradient needed to drive the flow through a narrowed conduit. The highest stresses occur along the tube but there appears to be a small region of increased stress on the trachea proximal to the entrance of the tube during expiration.

Outlet pressures were set to values of -20, -50, and -150 Pa for Patients A, B, and C, respectively. It can be seen in Figure 4.1a that the minimum pressure dropped slightly below these values for Patients B and C due to airway geometry. The maximum pressures during expiration (Figure 4.1b), however, do not exceed the values specified for the outlet boundary condition.

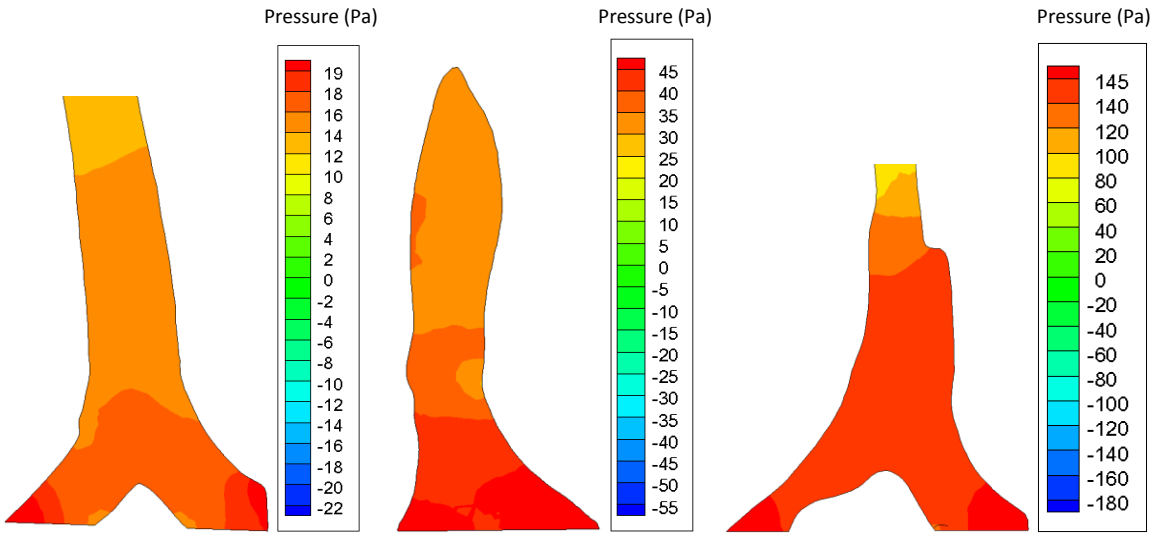
In Figure 4.2, velocity is basically constant along the length of the trachea for Patient A, with a maximum value of about 2.4 m/s at the center of the airway. Patients B and C, on the other hand, display small areas of high velocity because of locally narrow structure. A jet of air just superior to the bifurcation of Patient B has a maximum velocity of about

4.25 m/s while a similar jet leaving the intubation tube of Patient C has a maximum velocity of over 7 m/s.

Wall shear stress is noticeably larger at the bifurcation during inspiration (Figure 4.3a) than it is during expiration (Figure 4.3b). This is due to the previously mentioned jet of air formed by the trachea. As expected, Patient C experiences higher shear stresses at this location than Patient A because of the relative velocity magnitudes for these two cases, with maximum values of 0.18 and 0.36 Pa for Patients A and C, respectively. Shear stresses are much lower on the bifurcation and approximately the same on the walls of the trachea during expiration as compared to inspiration. Maximum values for Patients A, B, and C are approximately 0.22, 0.55, 0.38, respectively.

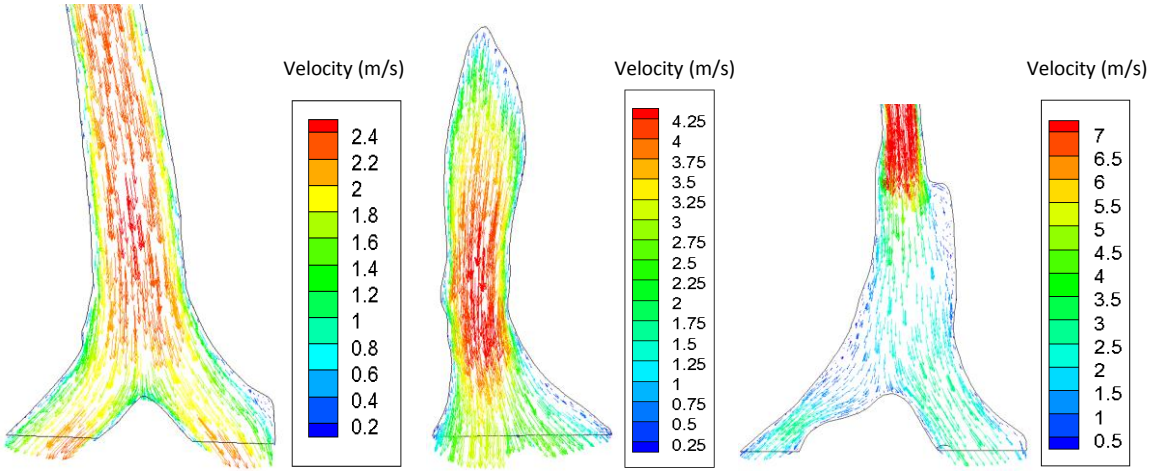


(a)

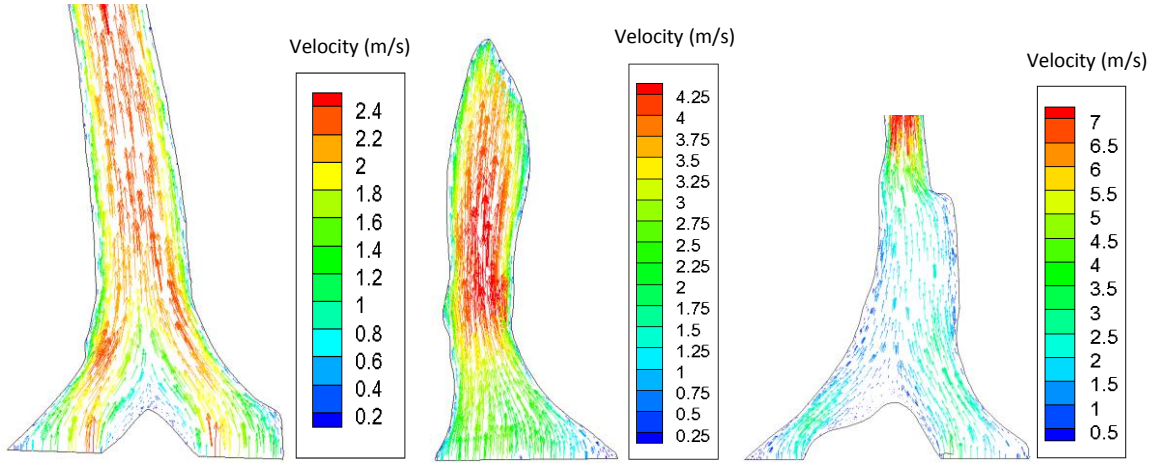


(b)

Figure 4.1: Contours of static pressure at tracheal bifurcation during a) peak inspiration and b) peak expiration for Patients A (left), B (middle), and C (right)

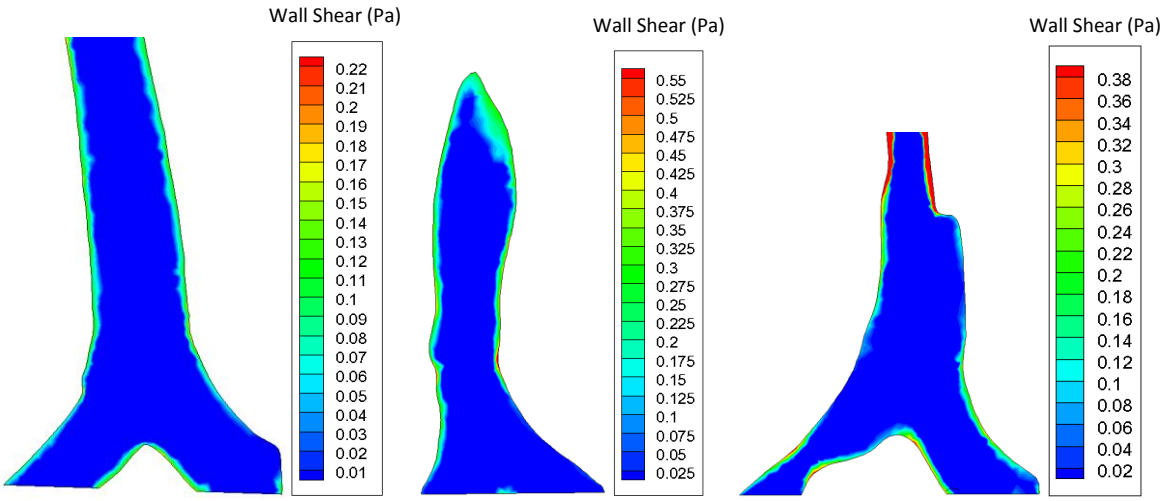


(a)

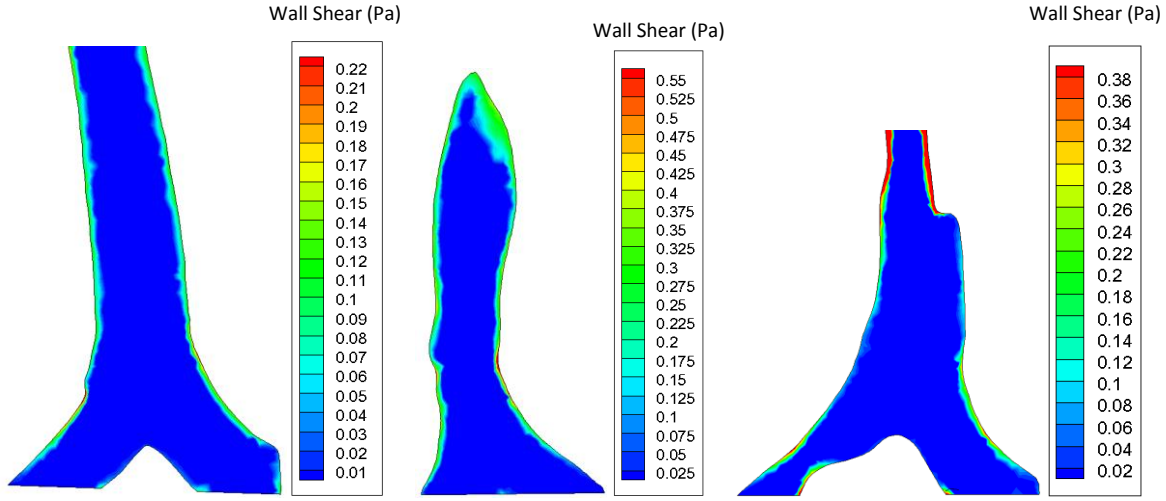


(b)

Figure 4.2: Velocity vectors colored by magnitude at tracheal bifurcation during a) inspiration and b) expiration for Patients A (left), B (middle), and C (right)



(a)



(b)

Figure 4.3: Contours of wall shear stress magnitude at tracheal bifurcation for a) inspiration and b) expiration for Patients A (left), B (middle), and C (right)

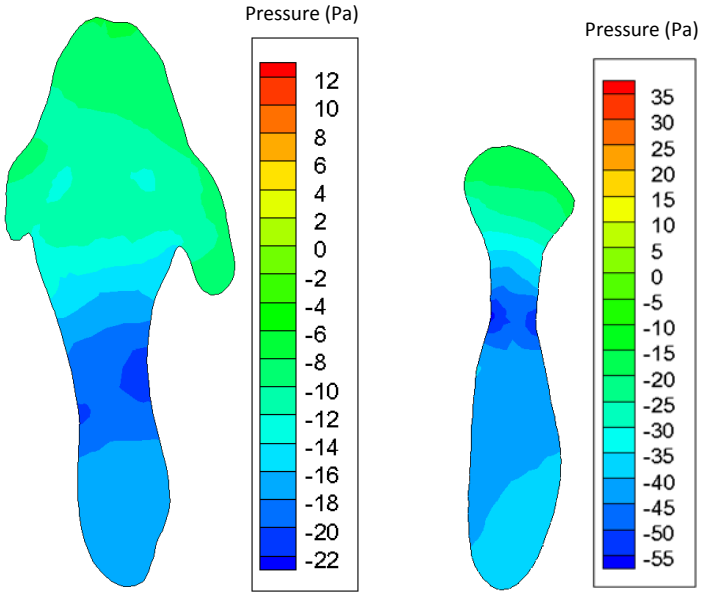
4.1.2 Larynx

Slices were extracted from regions containing the larynxes of Patients A and B. Pressure (Figure 4.4), velocity (Figure 4.5), and wall shear stress (Figure 4.6) are all plotted for these regions. The most noticeable difference between the larynxes of these two patients is the size (diameter). Patient A has a much larger diameter larynx that is approximately a constant size, while Patient B has a much smaller larynx that narrows at a point. The overall size is partially responsible for determining the needed pressure gradient. The narrowing that occurs for Patient B causes high local velocity and wall shear stress during both inspiration and expiration. Additionally, there is a much larger pressure difference between the highest and lowest pressures recorded for Patient B.

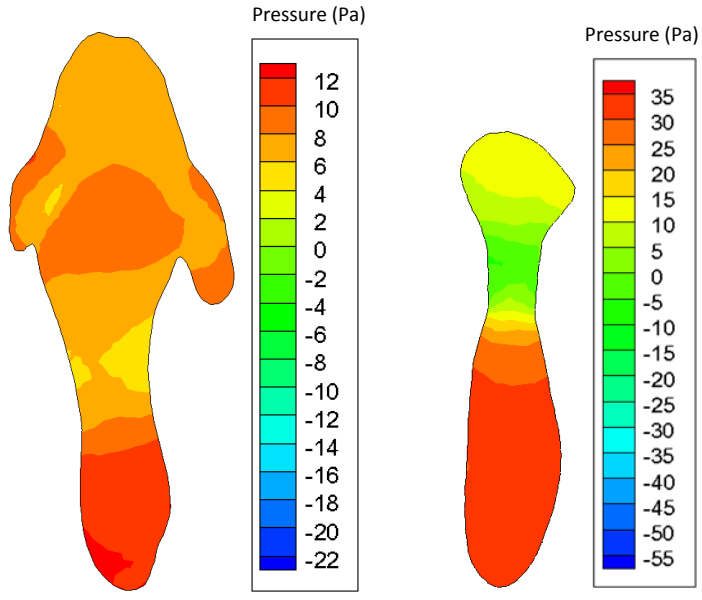
Differences in inhalation and exhalation are minor but are still present. For both patients, there is a greater pressure drop during inspiration than expiration. However, Patient B appears to experience slightly higher overall velocities through the larynx during expiration than inspiration. Wall shear stresses are very similar for each of the two phases.

Pressure along the larynx is consistently a minimum at the narrowest point of this region. During inspiration (Figure 4.4a) pressure drops to about -22 Pa in the larynx of Patient A and around -55 Pa for that of Patient B. Similarly during expiration (Figure 4.4b), pressure reaches a minimum of 6 Pa for Patient A and 0 Pa for Patient B. Again as a result of the narrower larynx in Patient B, velocity is higher for this patient than for Patient A. Maximum velocities during inspiration (Figure 4.5a) for child and infant are approximately 3.6 and 7.0 m/s, respectively. The peak value is similar during expiration (Figure 4.5b) for Patient A but slightly higher for Patient B, at about 7.5 m/s. Wall shear

stress varies little between inspiration (Figure 4.6a) and expiration (Figure 4.6b), but peak values are much higher for Patient B (1 Pa) than for Patient A (0.28 Pa).

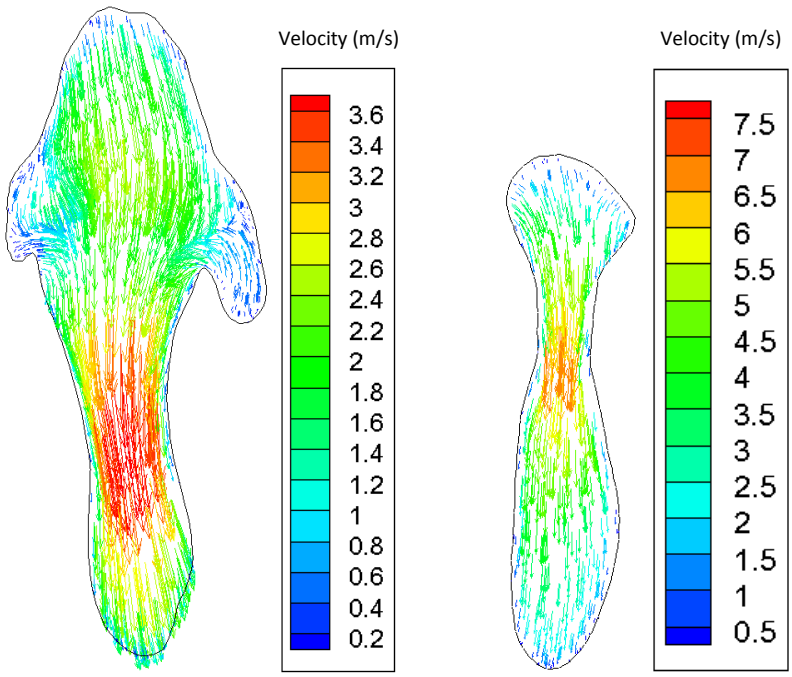


(a)

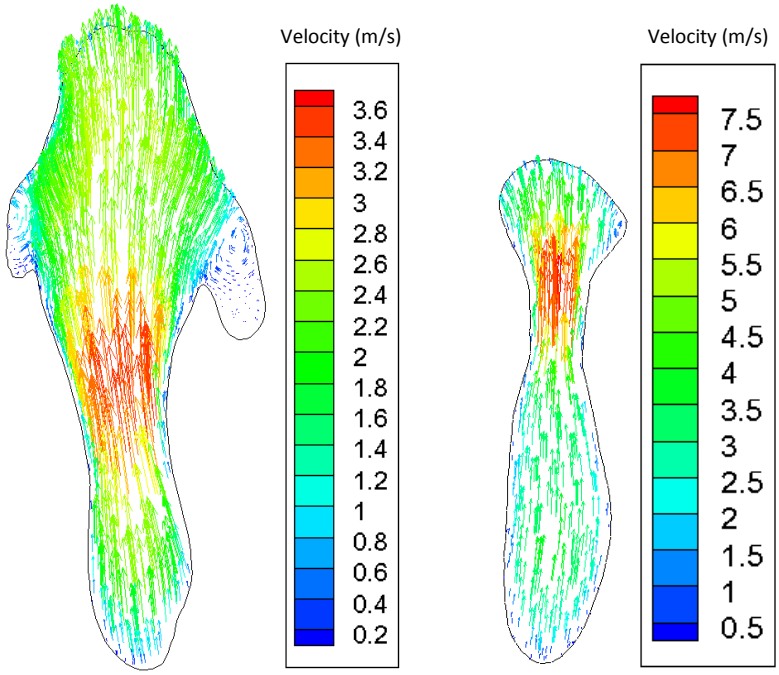


(b)

Figure 4.4: Contours of static pressure at the larynx during a) inspiration and b) expiration for Patients A (left) and B (right)

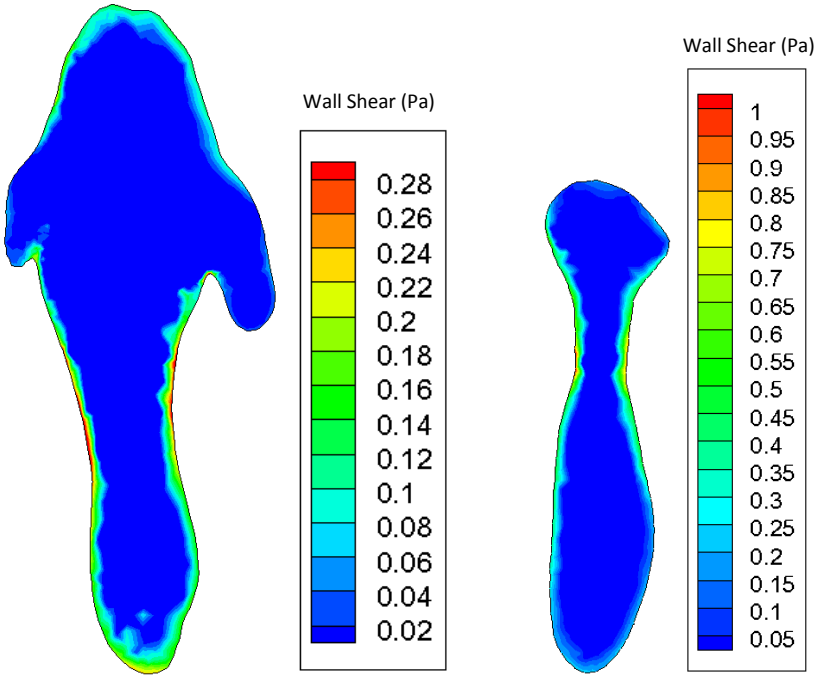


(a)

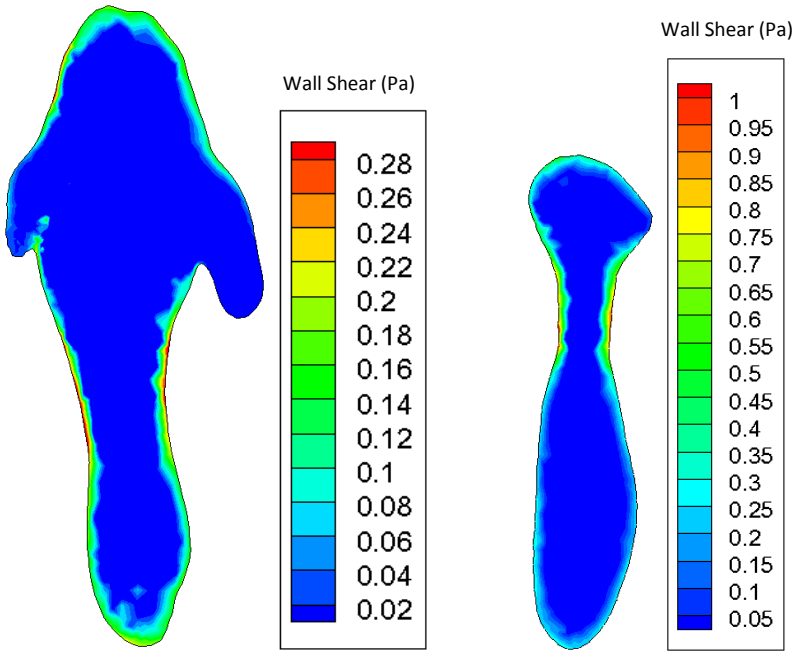


(b)

Figure 4.5: Velocity vectors colored by magnitude at the larynx during a) inspiration and b) expiration for Patients A (left) and B (right)



(a)



(b)

Figure 4.6: Contours of wall shear stress magnitude at larynx for a) inspiration and b) expiration for Patients A (left) and B (right)

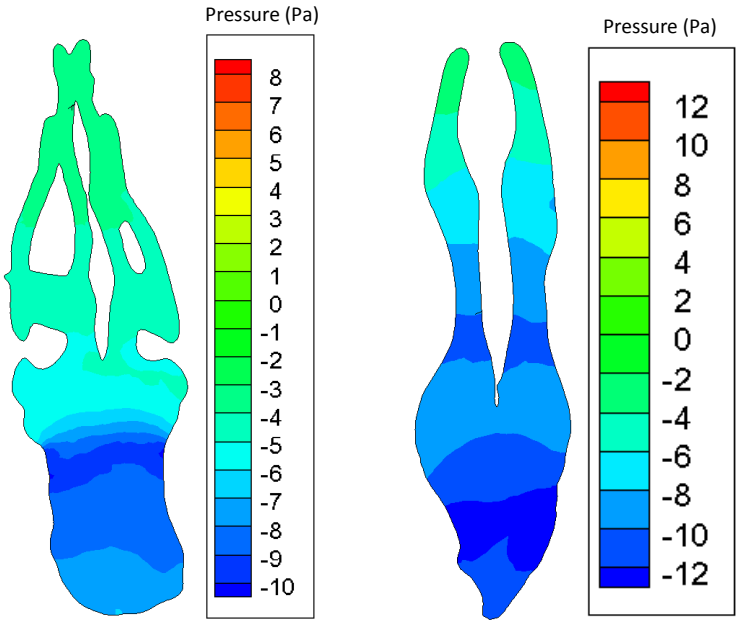
4.1.3 Nasopharynx

The last region of interest that was examined is the nasopharynx. Once again velocity (Figure 4.7), pressure (Figure 4.8), and wall shear stress (Figure 4.9) are parameters of interest. Slices taken along the nasopharynx also capture portions of the nasal cavity, so the anatomical structure can be seen at both of these locations. The geometry of the two nasal cavities is quite different due to the age discrepancy between the patients. Patient B is younger and therefore has not developed the features that are present in Patient A.

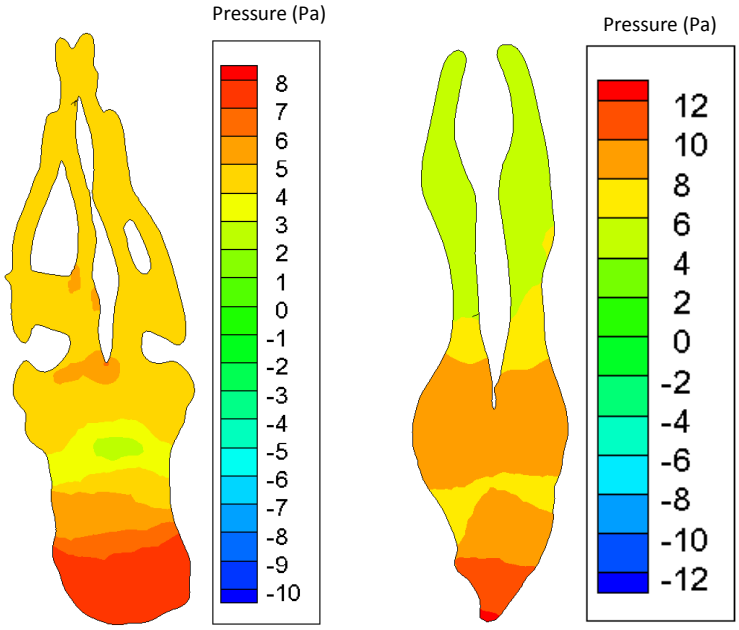
In both cases, the pressure gradient along the pharynx and nasal cavity is relatively small, yielding small velocities and wall shear stresses. The main difference between the two patients is caused by the structure of the nasal cavities. Velocity is distributed fairly evenly throughout the turbinates of Patient A while there is a region of high velocity in the inferior nasal cavity of Patient B. Because of this, only the younger patient experiences significant wall shear stress along the walls of the nasal cavity.

Figure 4.7a shows how the pressure distribution differs between the two patients during inspiration. There is a sharp pressure drop in the nasopharynx of Patient A, while this parameter decreases gradually from the nasal cavity to nasopharynx in Patient B. However, minimum pressure is about the same for each case, with a slightly lower value of -12 Pa for Patient B as opposed to -10 Pa for Patient A. Maximum pressures during expiration (Figure 4.7b) are around 8 and 12 Pa in the pharynx of Patients A and B, respectively. Maximum velocity is around 2.5 m/s in both patients during inspiration (Figure 4.8a) and expiration (Figure 4.8b). Patient B experiences higher max shear stresses (about 0.28 Pa) during both inspiration (Figure 4.9a) and expiration (Figure

4.9b) than Patient A (about 0.24 Pa). Additionally, these stresses are present on a much larger area for Patient A than Patient B.

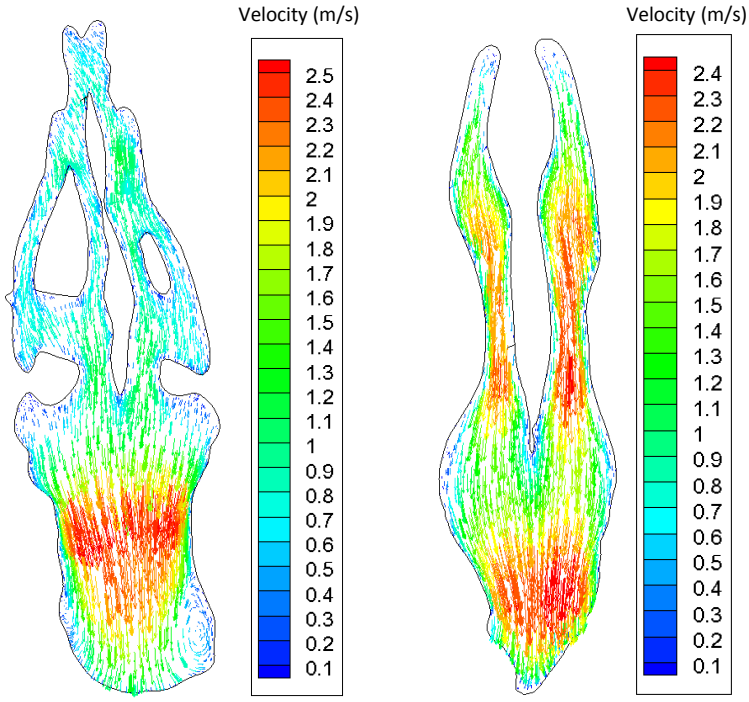


(a)

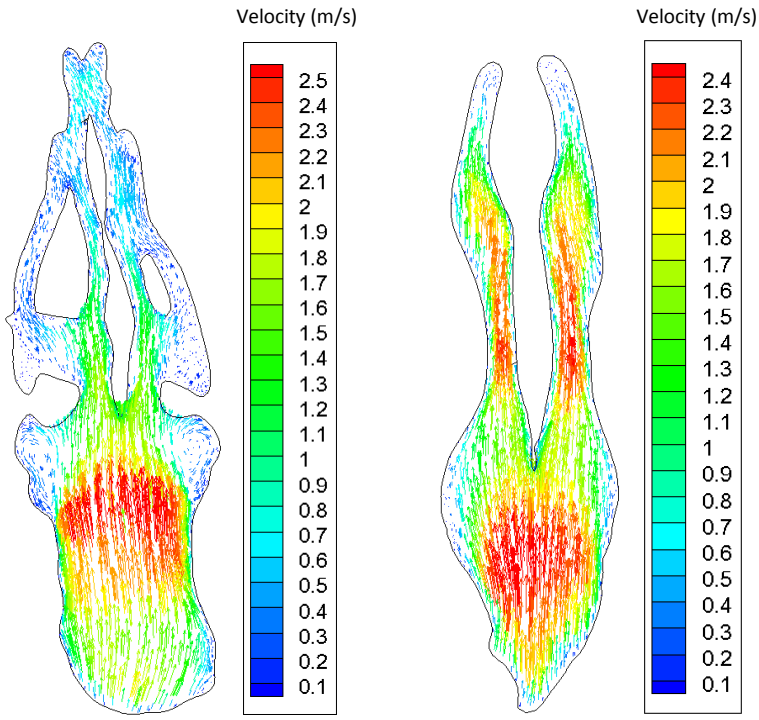


(b)

Figure 4.7: Contours of static pressure at the nasopharynx during a) inspiration and b) expiration for Patients A (left) and B (right)

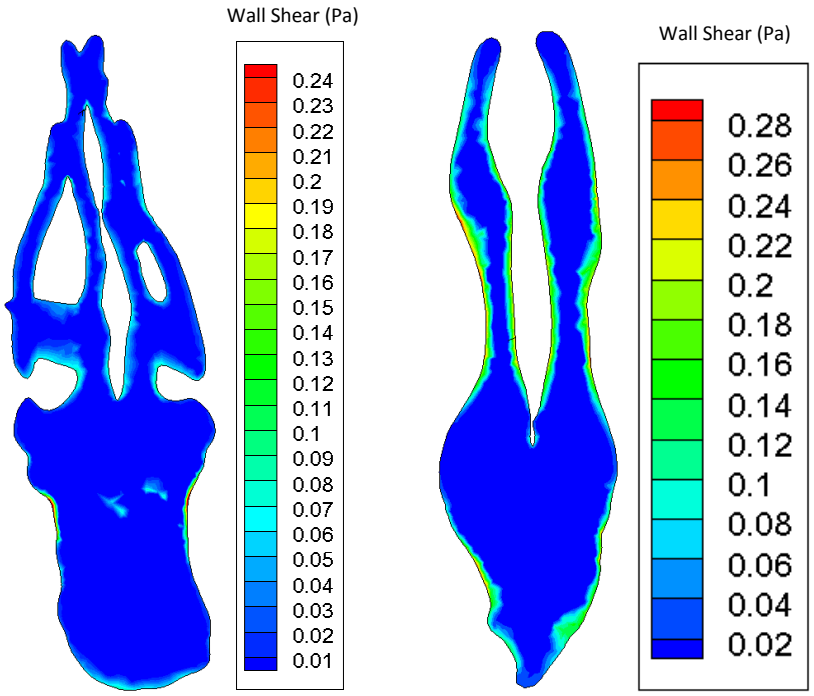


(a)

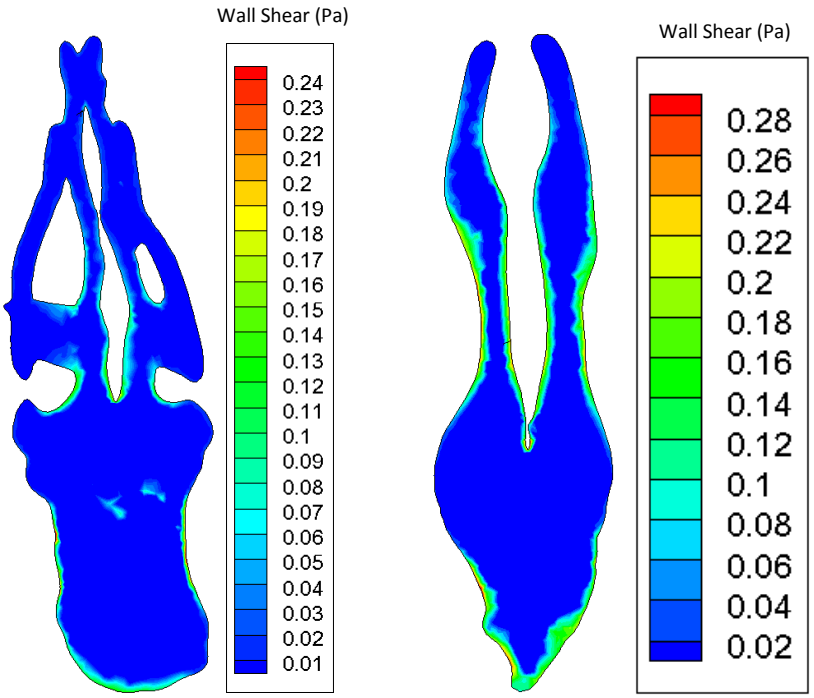


(b)

Figure 4.8: Velocity vectors colored by magnitude at the nasopharynx during a) inspiration and b) expiration for Patients A (left) and B (right)



(a)



(b)

Figure 4.9: Contours of wall shear stress magnitude at nasopharynx for a) inspiration and b) expiration for Patients A (left) and B (right)

4.1.4 Discussion on Validation of Results

As with any computer simulation, results must be verified in some way. In this case empirical data and theoretical calculations are used. Since an outlet pressure boundary condition was used to drive the flow in these models, it is helpful to examine the resulting tidal volumes for each case. Tidal volume was calculated based on the mass flow rate at the nasal inlet and is compared to a range of reference volumes (Table 6). These reference volumes were determined by multiplying the patient's estimated ideal weight(Wilton 1948) by standard values of tidal volume per kg body weight (7-9 ml/kg).

Table 6: Comparison of simulation results with typical reference values

Patient	Estimated Weight (kg)	Actual Weight (kg)	Simulation Tidal Volume (ml)	Reference Tidal Volume (ml)
A	27.92	34.93	244	195.44 – 251.28
B	10.89	unknown	107	76.23 – 98.01
C	14.97	9.07	90	104.79 – 134.73

It is also helpful to compare results to those of previous studies. While there has been little computational fluid dynamics (CFD) performed on airways of children, there have been studies done on adult patients. Since the respiratory airways of Patient A in this research are very similar to the airways of an adult, the results can be compared to an adult study. Wang et al. performed CFD on the upper airways of a 30 year old male and present many different parameters to describe their findings. Among these parameters is velocity distribution, which is shown at many different cross sections along the respiratory tract (Figure 4.10).

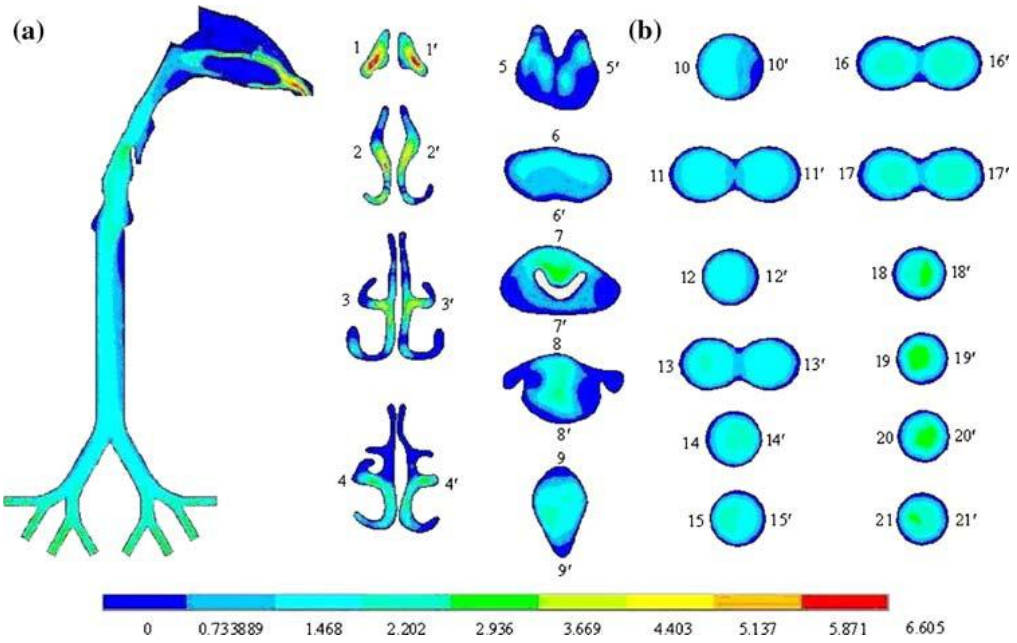


Figure 4.10: Results from a previous study by Wang et al. (Wang, Liu et al. 2009) showing velocity distribution at different cross sections along their airway model

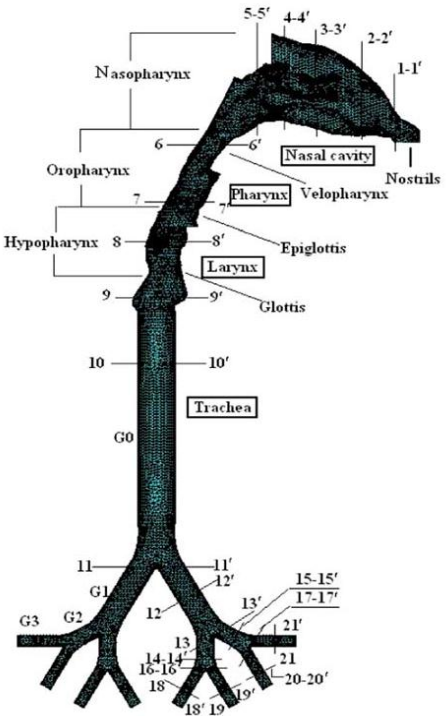


Figure 4.11: Whole airway schematic from Wang et al. illustrating location of cutting planes

Velocity magnitudes in the pharynx and larynx are similar for the present research and the previous study by Wang et al. The locations of the sampling slices shown in Figure 40 are shown in Figure 4.11. The average velocity through these regions is around 1.5 m/s for the previous study and about 2.5 m/s for this research. Both studies also show local flow acceleration due to narrowing of the pharynx. Overall, velocities are higher for Patient A, but this is expected since higher pressure gradients are needed in a child model than an adult model (Xi et al). This is because the airways of a child

are narrower than those of adult, so higher velocities will be needed to accomplish a similar mass flow rate.

In addition to velocity similarities, the overall pressure drop for this simulation is similar to the predicted pressure drop by Wang et al as well as Weinhold et al (Weinhold and Mlynski 2004). The pressure drop for Patient A is slightly higher, in accordance with previous findings that larger pressure gradients are needed for child models than adult models (Figure 4.12).

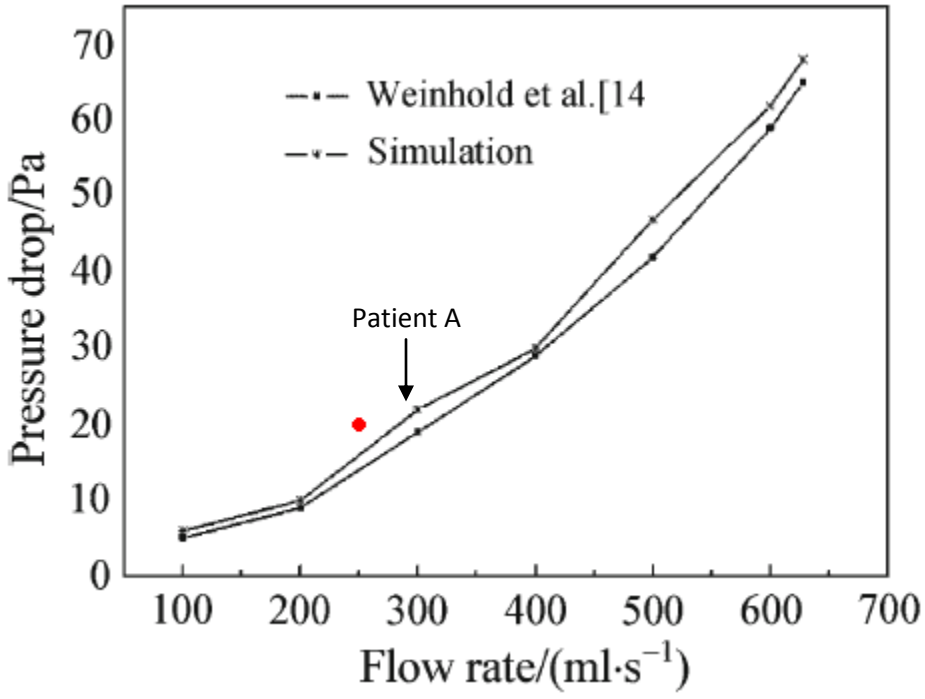


Figure 4.12: Graph taken from Wang et al. showing the relationship between pressure drop and flow rate for an adult model

4.1.5 Age Effects

Comparing the results of Patients A and B shows the differences that age can have on airflow dynamics in the respiratory tract. Two of the major differences between these models are the geometric complexity and the local dimensions. Patient B has a much simpler nasal cavity, which causes a reduction in airway volume in this region. Therefore, average velocity and wall shear stress is higher in the nasal cavity of Patient B than Patient A. Other features of Patient B are undeveloped and irregular, such as the lower trachea and bifurcation. This causes asymmetric flow patterns and wall shear stress. The other main difference between the two models is cross-sectional area. There is a narrowing in the larynx of both patients, but this is especially pronounced in Patient B. Likewise, the cross-sectional area of the trachea is reduced just superior to bifurcation in Patient B but not in Patient A. These local decreases in diameter cause higher velocities and shear stresses in the younger patient.

4.1.6 Intubation Effects

The presence of an intubation tube can cause non-physiologic flow in the lower airways. Results from these simulations show that there is a jet of air leaving the intubation tube during inspiration that leads to increased wall shear stress at the tracheal bifurcation. This jet is caused by the diameter of the tube, which is obviously smaller than the tracheal diameter since it must fit inside this airway. As previously mentioned, a smaller diameter airway means higher velocity for an equivalent mass flow rate.

At the termination of the intubation tube, there is a sharp change in cross-sectional area. During expiration, air is forced to rapidly transition from a large-diameter vessel to a much smaller one. While most of the effects from this change in structure are felt by the intubation tube, there does appear to be a small region of very high wall shear stress on the wall of the trachea. In conjunction with the irritation caused by physical contact between intubation tube and tracheal wall, this shear stress could cause significant damage to the airway.

4.2 Heart Results and Discussion

4.2.1 Dimension vs Scale

The left ventricle is a very complex structure, especially its inner surface. This surface can be described as a fractal that displays level 3 complexity, meaning that its geometry varies with scale and therefore can be termed a multi-fractal. Because of this, it is necessary to examine the surface at many different scales and to determine the fractal dimension at each one.

A fractal dimension value of two means that a surface is very smooth and close to a plane in appearance. However, an object appearing this way could actually have a very rough surface when examined closely, but depending on the observation method this might not be noticed. Conversely, the appearance of a rough surface would be obvious when focusing on small features, but this could lead to a neglect of the overall shape of the object. In Figure 5, fractal dimension is shown for a range of scales. The resolution of the box-counting grid is analogous to scale. In this case, the side length of each grid element ranges from 2% to 25% of the overall bounding box side length. Smaller scales allow the algorithm to capture fine features, similar to observation under a magnifying glass, whereas larger scales enable description of the general shape of the surface.

Figure 4.13 shows that fractal dimension is near 2 at very low scales and then increases rapidly, peaking at around 4% of the largest box size. This means that there are many features present that are around the size of this scale. Dimension then decreases gradually with increasing scale, until reaching a local maximum around 10% scale. At scales above 10%, dimension is around 2 or lower. This is expected since trabeculae

will not be noticed at large scales and the dimension will eventually tend to 1 as scale approaches 100%.

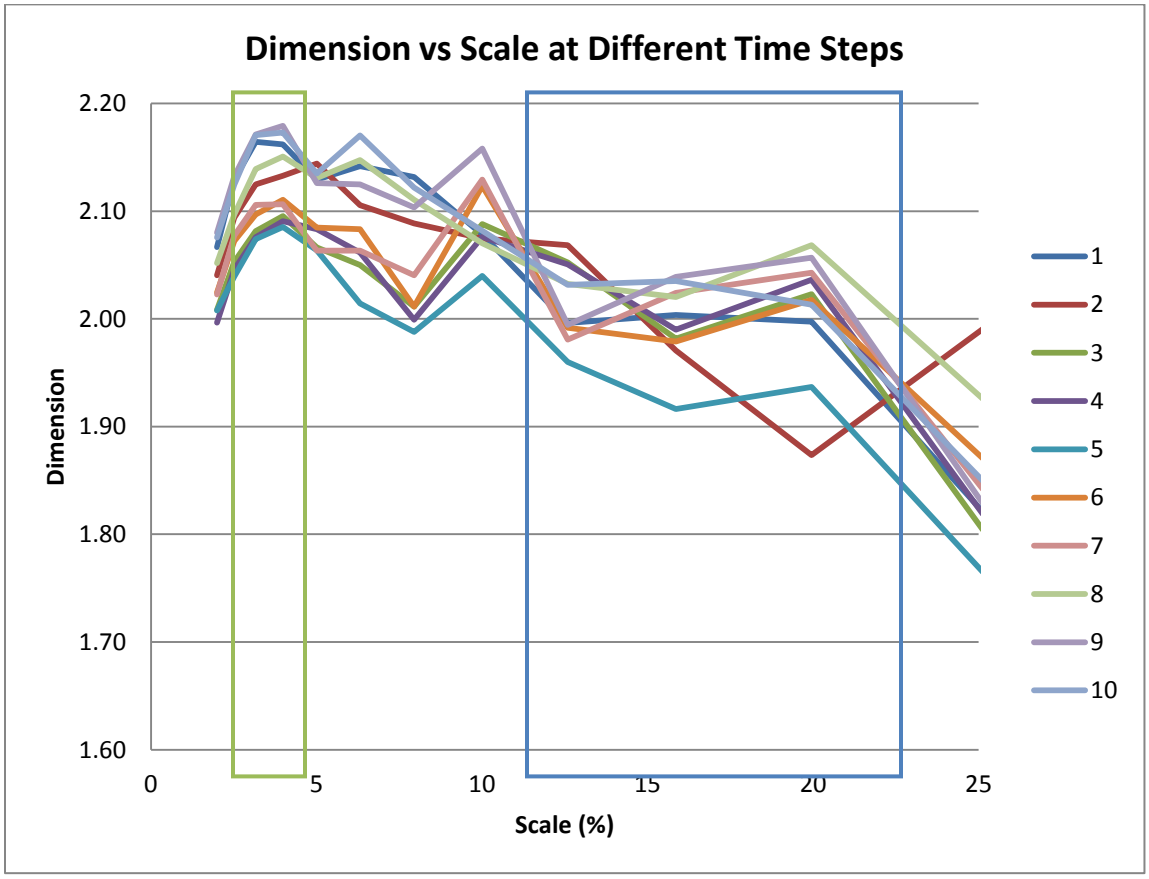


Figure 4.13: Fractal dimension as a function of scale for ten time steps during systole and diastole. Each trend line represents a different time step and bounding boxes are used to show small (green) and large (blue) scales.

There were a total of ten time equally-spaced steps tested over one cardiac cycle, starting with the beginning of contraction and finishing with the end of filling. The first time step corresponds to the start of systole, the fifth time step corresponds to the end of systole (also beginning of diastole), and the tenth time step represents the end of diastole.

4.2.2 Dimension vs Time

To better understand the changing geometry of the ventricle throughout systole and diastole, it is necessary to examine its fractal dimension at different times during these processes (Figure 4.14). This is done by focusing on one scale at a time (or in this case averaging a small range of scales) and comparing the changing dimension values. Two different scales were extracted and used to show fractal dimension as a function of time. Small scale data was gathered from 2 to 4% and large scale data was gathered from 10 to 15%. These scale ranges are highlighted in Figure 4.13, and each trend line was averaged over the highlighted region to yield a single data point in Figure 4.14. Since there are ten trend lines and two highlighted regions, this means that there are a total of twenty data points in Figure 4.14, ten for each region/scale.

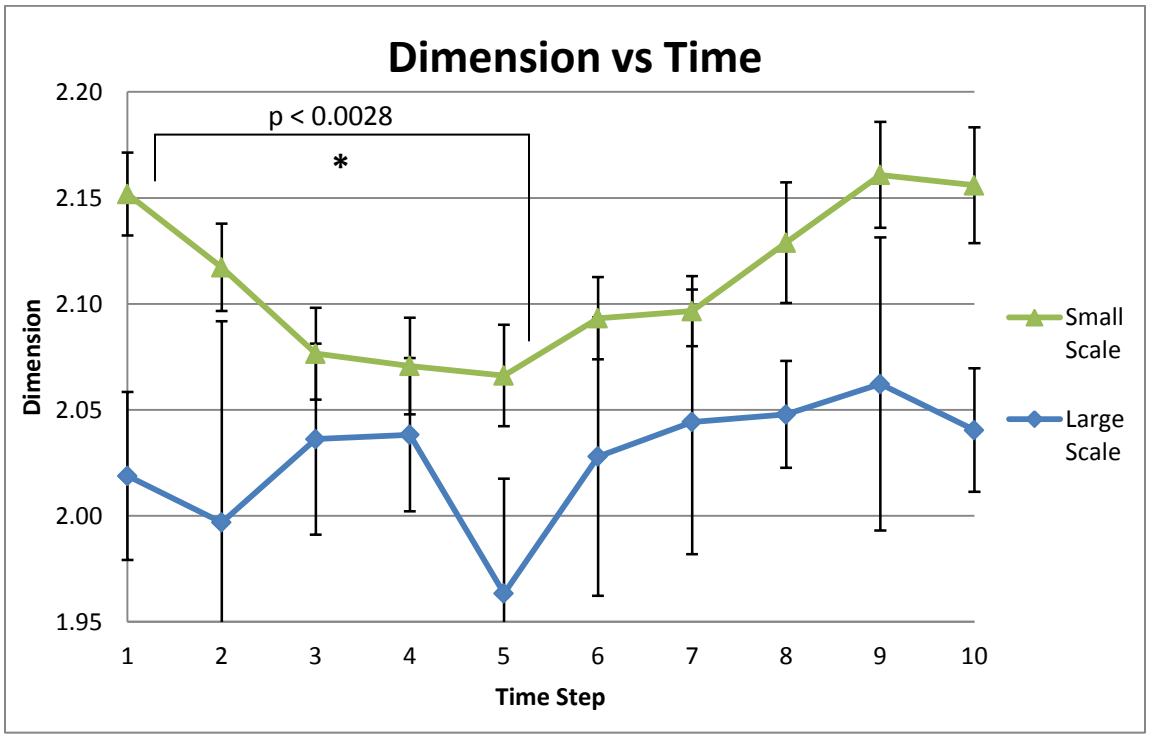


Figure 4.14: Dimension vs time from the beginning of systole (1) to end of diastole for small and large scales

As can be seen in Figure 4.14, the time steps follow a much more noticeable trend at small scales than they do at large scales. Due to somewhat scattered data, it is difficult to make a conclusion about the changing dimension at large scales. Error bars are used to show the statistical significance of both of these trends. It can be seen that there is strong statistical evidence ($p < 0.0028$) that a drop in dimension occurs at small scales, whereas this same argument cannot be strongly supported at large scales.

4.2.3 Correlation between Dimension and Volume

It was found that the fractal behavior of the ventricle's surface relates to the cardiac pumping cycle. At small scales, the fractal dimension is the greatest at the end of diastole and then decreases during systole. A minimum is reached at the end of systole and then dimension increases back to its previous value. The behavior at large scales is not as significant statistically but there does appear to be a minimum at the end of systole.

Using a modified box-counting algorithm, the volume of the ventricle was computed at each time step. This algorithm involved filling the object with boxes and then summing the volume of all of them. As the box size decreased, the volume increased, but this trend leveled-off after boxes became sufficiently small. In order to create a fully-closed object, the base of the ventricle, where the mitral and aortic valves are, was artificially "walled-off" with a plane perpendicular to the long axis of the ventricle. This was done at the same location for all time steps so as to cancel out any possible error that could have been induced.

The results from this algorithm are shown on a plot of changing ventricular volume over time (Figure 4.15). In addition, a similar plot is shown for an average clinical case (Figure 4.16). This plot shows that stroke volume is approximately 60mL as the ventricular volume varies from 120mL just before systole to 60mL afterwards. Results obtained from the box-counting method in this case are also shown and range from a maximum of about 120mL to a minimum of about 85mL, suggesting that the patient used for this study may have had a slightly reduced cardiac output. Regardless, the shape of the curve produced from this study matches the shape of the clinical curve.

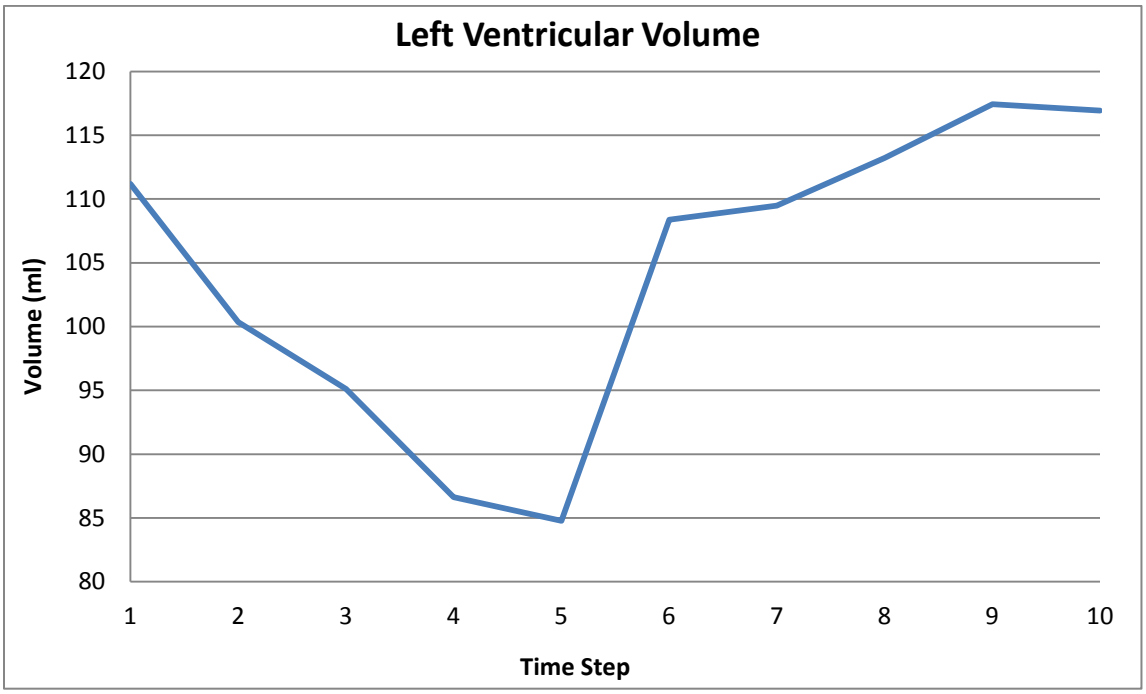


Figure 4.15: LV volume computed by modified box-counting algorithm from beginning of systole (1) to end of diastole (10)

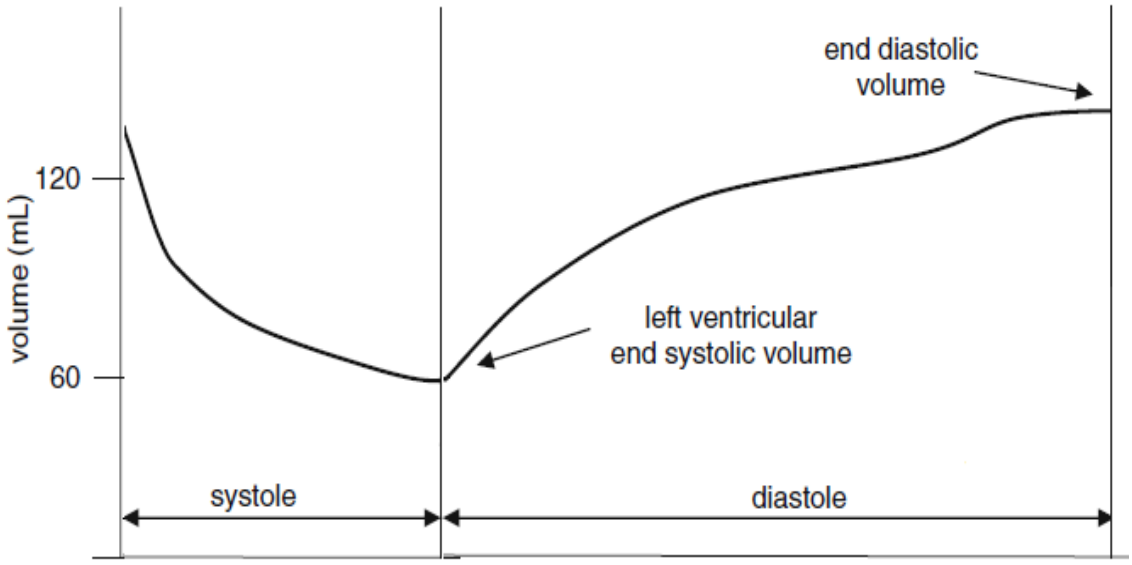


Figure 4.16: Clinical case of LV volume from beginning of systole to end of diastole(laizzo 2009)

In both curves, the ventricular volume decreases at a fairly rapid rate during systole until reaching a minimum and then increases rapidly at the beginning of diastole. This rapid increase slows down and levels off toward the end of diastole, where the cycle then repeats. The shape of the curve(s) can be attributed to the contraction rates of the ventricle and atrium.

As stated previously, there appears to be a relationship between changes in fractal dimension and changes in volume during the pumping cycle. During systole, there is a drop in ventricular volume as well as a drop in fractal dimension. Similarly, both of these aspects increase during diastole. There is a difference between small scale and large scale dimension changes in that small scale changes occur mostly at higher volumes whereas large scale changes occur mostly at lower volumes (Figure 4.17).

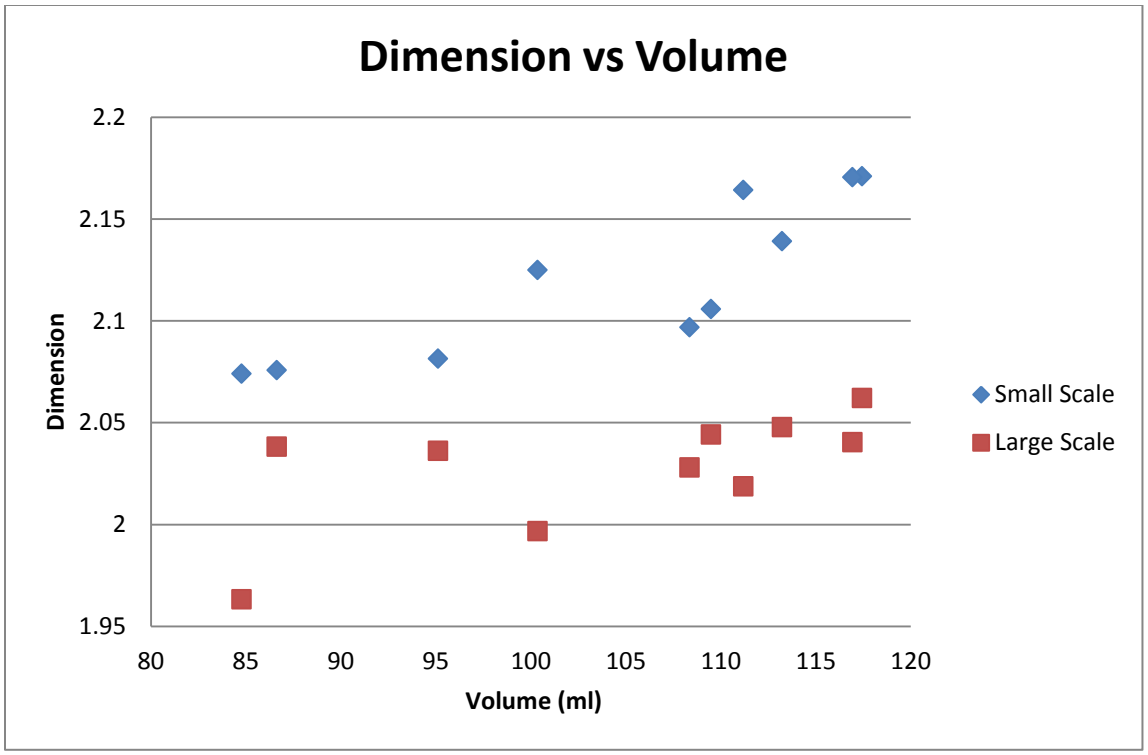


Figure 4.17: There is little variation in small scale dimension but large variation in large scale dimension at low volumes (85-95 ml). Conversely, dimension varies a lot at small scales but not much at large scales for higher volumes (110-120 ml).

In order to more clearly understand the changing geometry of the ventricle, 2D slices were extracted about both its long (Figure 4.18) and short (Figure 4.19) axes for each of the ten time steps tested. This was done by using 2D contour plots taken from the 3D meshed data in Tecplot 360. In both figures the series of images range from time steps 1 to 10, with 1 corresponding to the beginning of systole, 5 to the end of systole, and 10 to the end of diastole. The surface can be seen changing from a very rough texture to a smoother one during systole and then transitioning back during diastole. These images also provide visual evidence of small scale and large scale features. In agreement with box-counting results, most of the small scale features are present mainly during early systole and late diastole whereas large scale features don't smooth out until late systole and then reappear during early diastole.

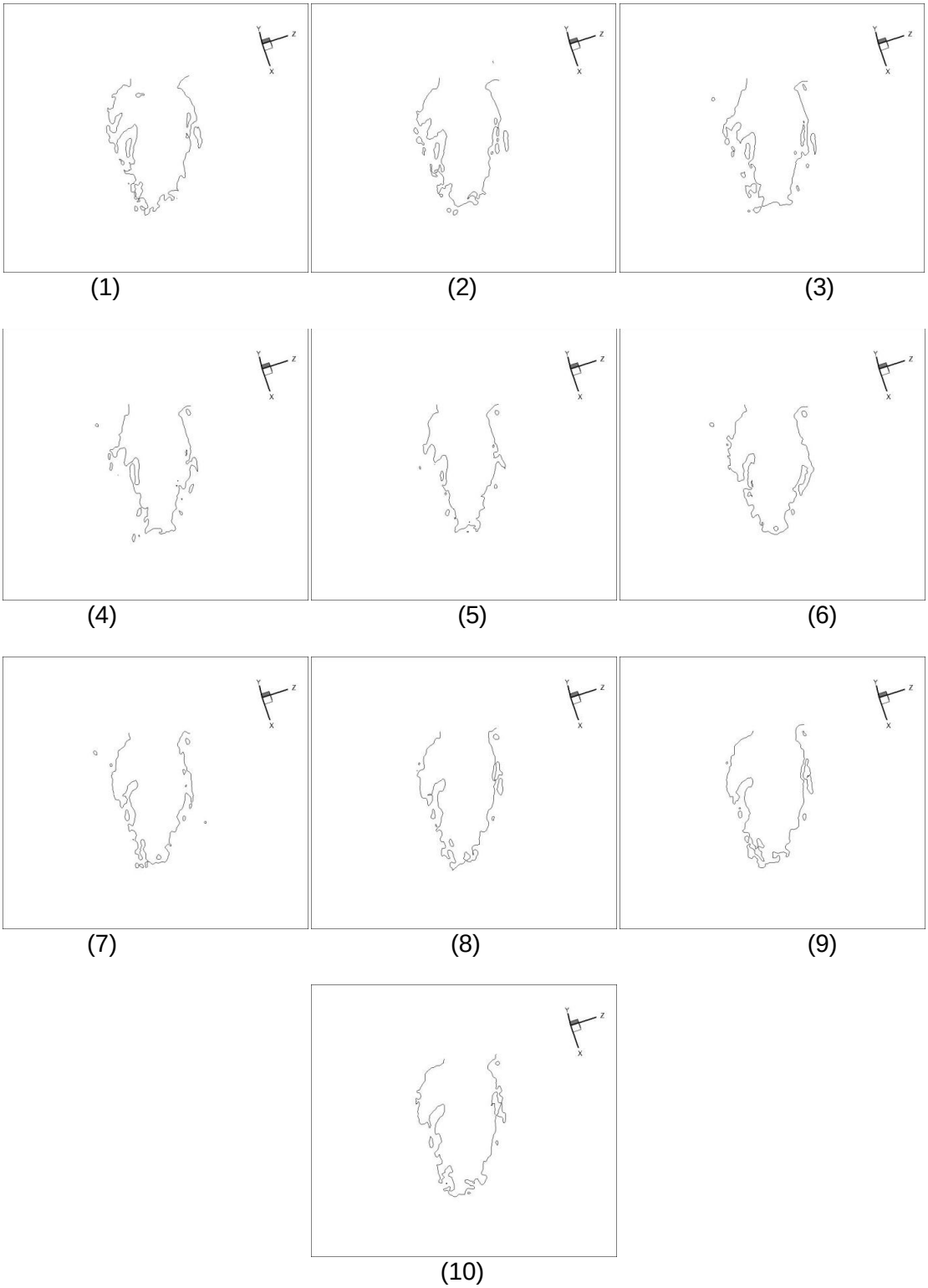


Figure 4.18: 2D long axis (coronal) slices taken at ten time steps during systole and diastole

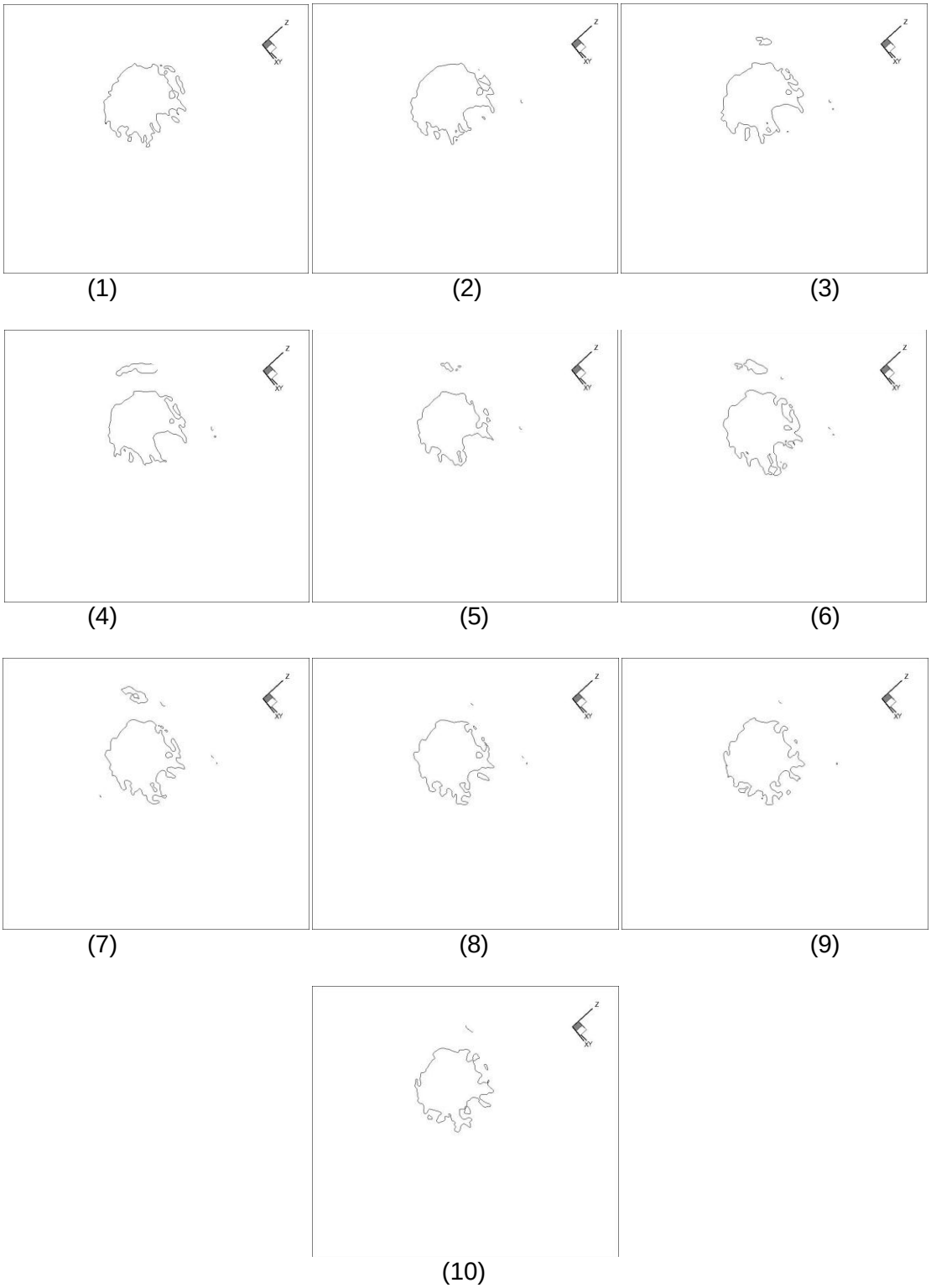


Figure 4.19: 2D short axis (transverse) slices taken at ten time steps during systole and diastole

4.2.4 Implications of Change in Dimension

It has been shown that the left ventricle of a human heart changes its fractal dimension during the cardiac cycle. This change in dimension has also been shown to correlate with the change in volume that occurs during this cycle. It appears that this is largely due the geometry of the ventricular endocardium, which is very rough at the end of diastole and gradually becomes smoother throughout systole. At the end of diastole, trabeculae protruding into the ventricle are quite apparent. However, these features are not nearly as visible when the ventricle is fully contracted. Thus, there is a smoothing-out of the cavities that are initially present due to the numerous trabeculae. Figure 4.20 shows how a smoothing of the interior ventricle wall can cause a change in volume.

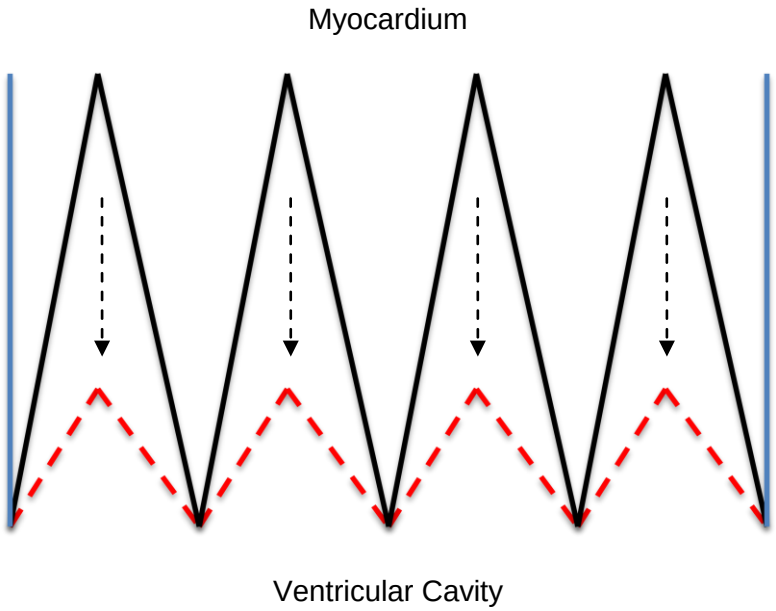


Figure 4.20: As deep recesses in the ventricle walls, formed by trabeculae, flatten-out (black to dashed red) blood is displaced thus reducing ventricular volume

To explain how this smoothing causes a change in dimension, it is helpful to revisit a common mathematically-created fractal: the Koch curve. In this example, the classical shape of the fractal is manipulated slightly but the same means are used to calculate its dimension. The generator (poked-out triangle) is analogous to a cavity in the ventricle wall created by protruding trabeculae. As the generator flattens out, the fractal dimension decreases (Figure 4.21). Dimension is calculated from the number of new line segments created (four) and the division scale of each of these segments ($2 < \lambda < 4$).

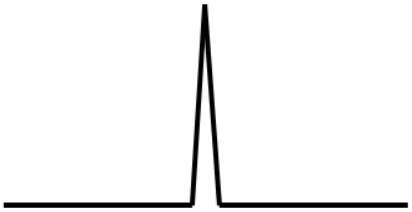
	$D = \frac{\log(4)}{\log(2.01)} = \mathbf{1.986}$
	$D = \frac{\log(4)}{\log(3)} = \mathbf{1.262}$
	$D = \frac{\log(4)}{\log(3.99)} = \mathbf{1.002}$

Figure 4.21: As the Koch curve is modified, its fractal dimension value changes from nearly 2 to just over 1. Limiting cases are shown as a very "spikey" version (top) and a very "smooth" version (bottom).

4.2.5 Biomechanical Considerations

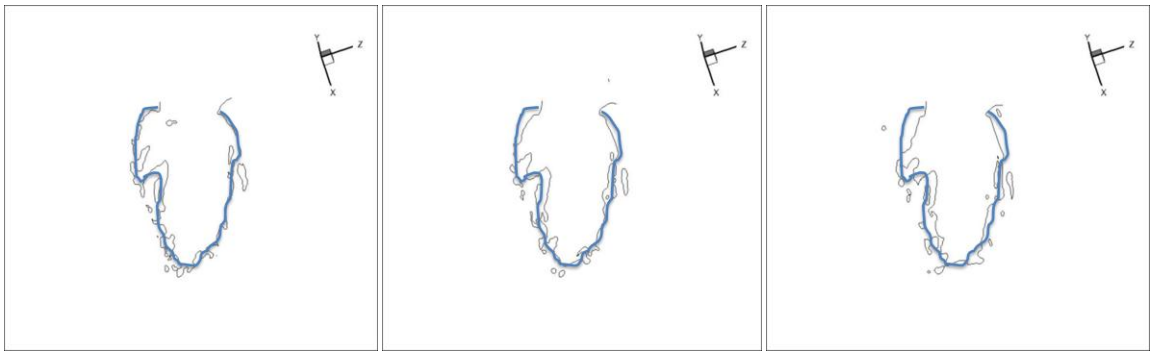
There are basically two methods by which the ventricle can change its volume: by increasing or decreasing the depth of cavities in the walls or by an overall translation of the walls. A visual inspection of the ventricle along with knowledge of volume and dimension can help one understand which of these modes is taking place. In Figure 4.22, a line has been drawn around the perimeter of the long axis slice for the first time step. This line is superimposed on each image to show a comparison of the ventricle's shape at different times. The same thing is done in Figure 4.23. There is little translation of the walls during the first three time steps, but features of the trabeculae start to disappear as the endocardium becomes less rough (refer to Figures 4.18 and 4.19 for better visualization). The fourth and fifth time steps show more noticeable translation along with further smoothing.

At the beginning of diastole (from step 5 to 6) there is a very large increase in volume, which is mainly due to wall translation and reappearance of large features. This rapid expansion may be present to make sure there is adequate room for blood to enter the ventricle so that there is no extra resistance on the atrium. Small features caused by trabeculae gradually reappear and there is little volume change during the rest of diastole (from step 6 to 10).

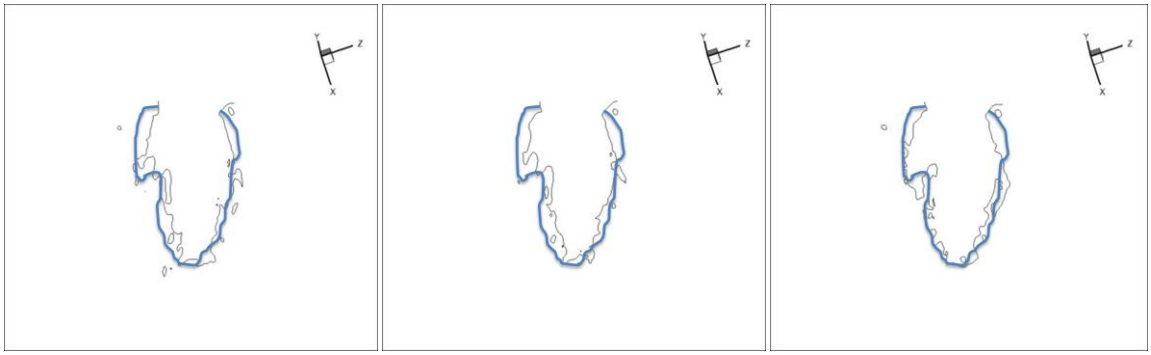
The way the heart changes geometrically may be optimized for pumping efficiency. It is apparent that most blood volume ejected during early systole is forced out of cavities in the ventricular wall. This may be a way to get blood moving at less of an energy expense than an overall translation of the walls. Once flow is started, the walls then move inward to expel the rest of the blood volume contained in the ventricle. This

process is similar to the way a car transmission works, in which mechanical advantage is initially utilized to overcome large forces by sacrificing displacement. Once the vehicle is moving and gains some momentum, the gear ratio can be changed to yield larger displacements.

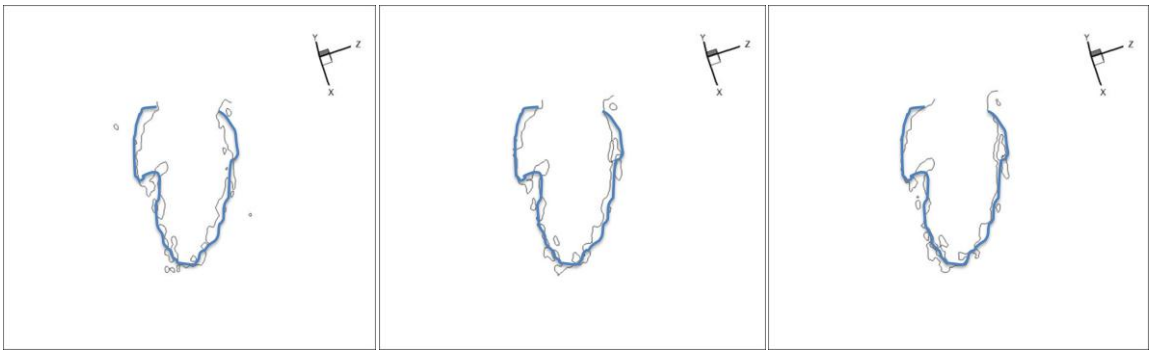
Another possible reason for the presence of trabeculae could be to reduce pressure reflections that are formed due to rapid contraction of the ventricle. If there was a sudden translation of the entire ventricular wall, this would cause a large pressure wave to propagate toward the aortic valve. Such a situation could cause high pressure in the aorta as well as very sudden movements of the aortic valve. On the other hand, if smaller pressures are initiated at many different places along the ventricular wall, they would be dampened significantly before reaching the aortic valve.



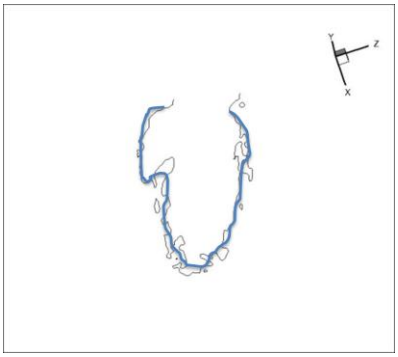
(1) (2) (3)



(4) (5) (6)

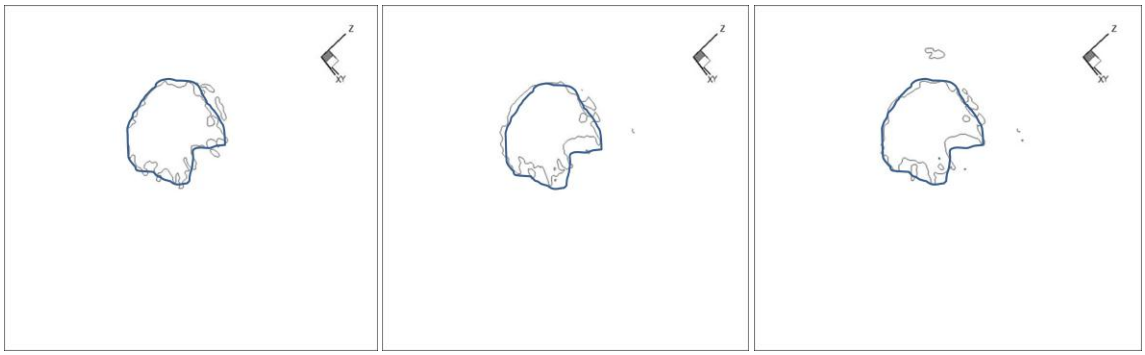


(7) (8) (9)



(10)

Figure 4.22: Long axis slices shown with a boundary around the first slice superimposed over every image for comparison



(1) (2) (3)

(4) (5) (6)

(7) (8) (9)

(10)

Figure 4.23: Long axis slices shown with a boundary around the first slice superimposed over every image for comparison

4.3 Limitations

Airway

- In the CFD simulations that were conducted for this work, an inlet boundary condition was set at the nasal valve and an outlet boundary condition was applied along a plane intersecting the main bronchi. Flow direction was set normal to these boundaries to best approximate actual conditions, however this would not be the case physiologically. In addition, prescribing pressure at the outlet does not exactly mimic the actual pressure gradients that would be seen due to action of the diaphragm.
- The sinusoidal shape of the pressure outlet boundary condition does not exactly match actual breathing patterns, but similar profiles have been used in other studies (Wang, Liu et al. 2009)
- Magnitude of outlet pressure was determined for Patient C using pipe flow calculations and an expected flow rate. However, the tube length and friction factor could only be estimated, leading to some uncertainty in this boundary condition.
- Patients used in this study each had some sort of ailment, which may have impacted airway geometry and therefore flow characteristics
- Airway dimensions were kept constant throughout the simulations. In reality, there would be some deformation due to inspiration and expiration.

Heart

- The box-counting algorithm used to determine fractal dimension is unreliable at large scales. Accuracy of this method relies on a large number of boxes for any given box size, which is not possible when box side lengths are close to the

length of the ventricle. Nevertheless, at small scales the number of boxes was sufficient for accurate calculation of fractal dimension.

4.4 Summary and Future Work

In summary, a methodology was developed to analyze the fluid dynamics of airflow through the respiratory tract of different patients. Computed Tomography (CT) scans were acquired through collaboration, and Mimics software was used to generate accurate 3D models of respiratory airways. These airways were meshes using ANSYS TGrid and imported into ANSYS Fluent for CFD simulations.

Simulation results showed differences in flow due to unique patient airway structure. The relatively narrow geometry of Patient B, a 7-month-old infant, caused overall greater velocities and wall shear stresses than those of Patient A, an 8-year-old child. Additionally, an intubation tube used on Patient C, a 3-year-old child, caused non-physiologic flow near the tube-trachea interface. Results for this portion were validated based on similar studies and basic flow rate calculations.

In the future, models will be adapted so that geometry is extended beyond the current inlets and outlets. This will help eliminate any unrealistic effects caused by truncation of anatomy. Also, boundary conditions that are more patient-specific will be developed to ensure conditions are as close to physiologic as possible. Direct Numerical Simulations could also be run to fully capture the flow characteristics at all locations and scales.

For the geometric analysis of the left ventricle, similar methods were used to extract a 3D model from CT scans. In this case, a single patient was used, but ten different

models of the ventricle were created at equally-spaced time intervals during one cardiac cycle. Surface meshes were created for each of these models and then analyzed using fractal geometry. Specifically, a custom box-counting algorithm was created in MATLAB and validated using mathematically-created fractals of known dimension. The algorithm was then applied to each ventricle model.

Box-counting results yielded changing dimension values over time for a range of scales. It was determined that large-scale dimension changes were statistically insignificant but changes at small scales were indeed present. These dimension values ranged from about 2.16 at the end of diastole to around 2.07 at the end of systole. Such dimension changes could have implications on the biomechanics of heart pumping. Rather than just pumping blood by translating the ventricular walls, the heart appears to squeeze blood out of many small cavities that are created by trabeculae. This could help reduce the energy needed to expel blood as well as reduce pressure wave reflections that would otherwise be much larger.

Future work for this study will include the same fractal analysis on a more patients. These patients should be selected in order to study the effects of different factors, such as age, gender, or race. Additional studies could even be carried out on animals of different species to elucidate the geometric properties of a wide range of ventricles.

References

"AV and Semilunar Valves." Anatomy BY411, 2011, from http://julieteaslickby411.blogspot.com/.

Adhyapak, S. M. and V. R. Parachuri "Architecture of the left ventricle: insights for optimal surgical ventricular restoration." Heart Failure Reviews **15**(1): 73-83.

AHA. (2011). "Cardiovascular Disease Statistics." Retrieved 2011, from http://www.americanheart.org/presenter.jhtml?identifier=4478.

Ansys (2010). Fluent Users Guide.

Burch, G. E., C. T. Ray, et al. (1952). "THE FAHR,GEORGE LECTURE - CERTAIN MECHANICAL PECULIARITIES OF THE HUMAN CARDIAC PUMP IN NORMAL AND DISEASED STATES." Circulation **5**(4): 504-513.

Burkardt, J. (6/26/2010). "DISTMESH A Simple Mesh Generator in MATLAB." 2011, from http://people.sc.fsu.edu/~jburkardt/m_src/distmesh/distmesh.html.

Burton, A. C. (1957). "THE IMPORTANCE OF THE SHAPE AND SIZE OF THE HEART." American Heart Journal **54**(6): 801-810.

Catrakis, H. J. (2000). "Distribution of scales in turbulence." Physical Review E **62**(1): 564-578.

Churchill, S. E., L. L. Shackelford, et al. (2004). "Morphological variation and airflow dynamics in the human nose." American Journal of Human Biology **16**(6): 625-638.

Cotes, J. E., D. J. Chinn, et al. (2009). Lung Function Physiology, Measurement and Application in Medicine. Chichester, John Wiley & Sons, Ltd.

Des Jardins, T. R. (2002). Cardiopulmonary anatomy & physiology essentials for respiratory care. Australia ;, Delmar/Thomson Learning.

Ethier, C. R. and C. A. Simmons (2007). Introductory biomechanics from cells to organisms. Cambridge ;, Cambridge University Press.

Grosberg, A., M. Gharib, et al. (2009). "Effect of Fiber Geometry on Pulsatile Pumping and Energy Expenditure." Bulletin of Mathematical Biology **71**(7): 1580-1598.

Iaizzo, P. A. (2009). Handbook of cardiac anatomy, physiology, and devices. New York, NY, Springer.

Iftekharuddin, K. M., W. Jia, et al. (2000). A fractal analysis approach to identification of tumor in brain MR images. Proceedings of the 22nd Annual International Conference of the Ieee Engineering in Medicine and Biology Society, Vols 1-4. J. D. Enderle. New York, Ieee. **22:** 3064-3066.

Janssen, P. M. L. "Myocardial contraction-relaxation coupling." American Journal of Physiology-Heart and Circulatory Physiology **299**(6): H1741-H1749.

Juliet. "AV and Semilunar Valves." Anatomy BY411, 2011, from http://julieteaslickby411.blogspot.com/.

Klabunde, R. E. (4/6/2007). "Cardiovascular Physiology Concepts." 2011.

knowledgerush. "knowledgerush." 2011, from http://www.knowledgerush.com/kr/encyclopedia/Koch_snowflake/.

Mandelbr.B (1967). "HOW LONG IS COAST OF BRITAIN - STATISTICAL SELF-SIMILARITY AND FRACTIONAL DIMENSION." Science **156**(3775): 636-&.

Mandelbrot, B. (1982). The Fractal Geometry of Nature, W. H. Freeman and Co.

Martino. "Laboratory Dissection: Heart." 2011, from http://academic.amc.edu/martino/grossanatomy/site/Medical/Lab%20Manual/Cardiovascular/Dissections/Heart%20Dissections/Dissected%20Heart%2049.htm.

Materialise. "Materialise." 2011, from www.materialise.com.

mathaware. "Growth Factors." 2011, from http://mathaware.org/mam/00/master/essays/dimension/dimen09.html.

Munson, B. R. (2006). Fundamentals of Fluid Mechanics, John Wiles & Sons, Inc.

Richardson, L. F. (1961). "The problem of contiguity: an appendix of statistics of deadly quarrels." General systems: Yearbook of the Society for the Advancement of General Systems Theory **6**: 139-187.

Rouadi, P., F. M. Baroody, et al. (1999). "A technique to measure the ability of the human nose to warm and humidify air." Journal of Applied Physiology **87**(1): 400-406.

Sauret, V., R. M. Halson, et al. (2002). "Study of the three-dimensional geometry of the central conducting airways in man using computed tomographic (CT) images." Journal of Anatomy **200**(2): 123-134.

Schneider, N. S., T. Shimayoshi, et al. (2006). "Mechanism of the Frank-Starling law - A simulation study with a novel cardiac muscle contraction model that includes titin and troponin I." Journal of Molecular and Cellular Cardiology **41**(3): 522-536.

Sundberg, J., P. Birch, et al. (2007). "Experimental findings on the nasal tract resonator in singing." Journal of Voice **21**(2): 127-137.

ter Keurs, H., T. Shinozaki, et al. (2008). "Sarcomere mechanics in uniform and non-uniform cardiac muscle: A link between pump function and arrhythmias." Progress in Biophysics & Molecular Biology **97**(2-3): 312-331.

Vanderbilt. "Fractals and the Fractal Dimension." 2011, from http://www.vanderbilt.edu/AnS/psychology/cogsci/chaos/workshop/Fractals.html.

Wang, Y., Y. X. Liu, et al. (2009). "Numerical analysis of respiratory flow patterns within human upper airway." Acta Mechanica Sinica **25**(6): 737-746.

Weiford, B. C., V. D. Subbarao, et al. (2004). "Noncompaction of the ventricular myocardium." Circulation **109**(24): 2965-2971.

Weinhold, N. and G. Mlynski (2004). "Numerical simulation of airflow in the human nose." European Archives of Oto-Rhino-Laryngology **261**(8): 452-455.

Wendt, J. F. and J. D. Anderson (2009). Computational fluid dynamics an introduction. Berlin ;, Springer.

White, F. M. (2011). Fluid Mechanics. New York City, McGraw-Hill.

Wikipedia. "How Long Is the Coast of Britain? Statistical Self-Similarity and Fractional Dimension." 2011, from http://en.wikipedia.org/wiki/How_Long_Is_the_Coast_of_Britain%3F_Statistical_Self-Similarity_and_Fractional_Dimension.

Wilton (1948). "A Handbook of the Measurement and Interpretation of Height and Weight in the Growing Child." Monographs of the Society for Research in Child Development **13**(3): 1-68.

Xi, J. X., X. H. Si, et al. "Simulation of airflow and aerosol deposition in the nasal cavity of a 5-year-old child." Journal of Aerosol Science **42**(3): 156-173.

Yokusoglu, M., M. Uzun, et al. (2005). "Elongation index as a new index determining the severity of left ventricular systolic dysfunction and mitral regurgitation in patients with congestive heart failure." Echocardiography-a Journal of Cardiovascular Ultrasound and Allied Techniques **22**(7): 555-560.

Zhang, Z., Z. Q. Zhai, et al. (2007). "Evaluation of various turbulence models in predicting airflow and turbulence in enclosed environments by CFD: Part 2-comparison with experimental data from literature." Hvac&R Research **13**(6): 871-886.

Zubair, F. R. and H. J. Catrakis (2009). "On separated shear layers and the fractal geometry of turbulent scalar interfaces at large Reynolds numbers." Journal of Fluid Mechanics **624**: 389-411.

Appendix

Main section of code “Heart 3D.m”:

```
%Making sure the same "random" numbers aren't generated each time code is
%run
RandStream.setDefaultStream(RandStream('mt19937ar','seed',sum(100*clock)));
uniqueCounter=0;

newData=zeros(408820,3);

for var=1:1;

if(var==1)
count=1;

for i=1:230532;
   newData(i,:)=[data(i,1),data(i,2),data(i,3)+50];
   count=count+1;
end
sizeTwo=count-1;
end

if(var==2)
count=1;
for i=1:199941;
   newData(i,:)=[data2(i,1),data2(i,2),data2(i,3)+50];
   count=count+1;
end
sizeTwo=count-1;
end

if(var==3)
count=1;
for i=1:212657;
   newData(i,:)=[data3(i,1),data3(i,2),data3(i,3)+50];
   count=count+1;
end
sizeTwo=count-1;
end

if(var==4)
count=1;
for i=1:214067;
```

```
    newData(i,:)=[data4(i,1),data4(i,2),data4(i,3)+50];
    count=count+1;
end
sizeTwo=count-1;
end

if(var==5)
count=1;
for i=1:188417;
    newData(i,:)=[data5(i,1),data5(i,2),data5(i,3)+50];
    count=count+1;
end
sizeTwo=count-1;
end

if(var==6)
count=1;
for i=1:385866;
    newData(i,:)=[data6(i,1),data6(i,2),data6(i,3)+50];
    count=count+1;
end
sizeTwo=count-1;
end

if(var==7)
count=1;
for i=1:408820;
    newData(i,:)=[data7(i,1),data7(i,2),data7(i,3)+50];
    count=count+1;
end
sizeTwo=count-1;
end

if(var==8)
count=1;
for i=1:403240;
    newData(i,:)=[data8(i,1),data8(i,2),data8(i,3)+50];
    count=count+1;
end
sizeTwo=count-1;
end

if(var==9)
count=1;
for i=1:213669;
    newData(i,:)=[data9(i,1),data9(i,2),data9(i,3)+50];
    count=count+1;
end
sizeTwo=count-1;
end

if(var==10)
count=1;
for i=1:207740;
    newData(i,:)=[data10(i,1),data10(i,2),data10(i,3)+50];
```

```
    count=count+1;
end
sizeTwo=count-1;
end



heartTwo=zeros(sizeTwo,3);

count=1;
for i=1:sizeTwo;
    if((newData(i,1)>60 && newData(i,1)<80 && newData(i,2)>125 && newData (i,2)<120) ||
newData(i,1)<100)
    else
        heartTwo(count,:)=newData(i,:);
        count=count+1;
    end
end
sizeA=count;

heart=zeros(sizeA,3);
for i=1:sizeA;
    heart(i,:)=heartTwo(i,:);
end



for num=2:2;

    uniqueCounter=uniqueCounter+1;



clear Box
clear X
clear Y
clear Z
clear boxes

BSL=num;

Box=zeros(ceil(200/BSL),ceil(200/BSL),ceil(200/BSL));
boxes=zeros(200000,3);



for i=1:200;
    for j=1:200;
        Box(ceil(100/BSL),ceil(i/BSL),ceil(j/BSL))=1;
        Box(ceil(i/BSL),ceil(j/BSL),ceil(120/BSL))=1;
    end
end



%Find boxes touching heart
for i=1:sizeA;
    if(heart(i,1)>0)
```

```
        %Place box at point
        xBox1=(ceil(heart(i,1)*(1/BSL)));
        yBox1=(ceil(heart(i,2)*(1/BSL)));
        zBox1=(ceil(heart(i,3)*(1/BSL)));
        Box(xBox1,yBox1,zBox1)=1;
    end
end



IP=[ceil(mean(heart(1:sizeA,1))/BSL),ceil(mean(heart(1:sizeA,2))/BSL),ceil(mean(heart(1:sizeA,3
))/BSL)];
box=IP;

Box=checkBoxes3D(box,Box);



countTwo=1;
for i=ceil(150/BSL):ceil(156/BSL);
    for j=1:ceil(200/BSL);
        for k=1:ceil(200/BSL);
            if(Box(i,j,k)==1)
                X(countTwo)=i*BSL;
                Y(countTwo)=j*BSL;
                Z(countTwo)=k*BSL;
                countTwo=countTwo+1;
            end
        end
    end
end



volume=countTwo*BSL^3;
m(uniqueCounter,:)=[BSL,countTwo,volume,angle];

scatter3(X,Y,Z)

end
end

xlswrite('U:\Research\HeartDimensionsRaw.xlsx',m);
```

Subfunction “Checkboxes3D.m”:

```
%Input:  Array of boxes forming a polygon (boxes), Box array, point in
%middle of polygon (box)

%Output:  Adds boxes inside the polygon to the Box array

function Box=checkBoxes3D(box,Box)
```

```
Box(box(1),box(2),box(3))=2;
count=1;

if(Box(box(1)+1,box(2),box(3))==0)
   boxesTwo(count,:)=[box(1)+1,box(2),box(3)];
   Box(box(1)+1,box(2),box(3))=2;
   count=count+1;
end
if(Box(box(1)-1,box(2),box(3))==0)
   boxesTwo(count,:)=[box(1)-1,box(2),box(3)];
   Box(box(1)-1,box(2),box(3))=2;
   count=count+1;
end
if(Box(box(1),box(2)+1,box(3))==0)
   boxesTwo(count,:)=[box(1),box(2)+1,box(3)];
   Box(box(1),box(2)+1,box(3))=2;
   count=count+1;
end
if(Box(box(1),box(2)-1,box(3))==0)
   boxesTwo(count,:)=[box(1),box(2)-1,box(3)];
   Box(box(1),box(2)-1,box(3))=2;
   count=count+1;
end
if(Box(box(1),box(2),box(3)+1)==0)
   boxesTwo(count,:)=[box(1),box(2),box(3)+1];
   Box(box(1),box(2),box(3)+1)=2;
   count=count+1;
end
if(Box(box(1),box(2),box(3)-1)==0)
   boxesTwo(count,:)=[box(1),box(2),box(3)-1];
   Box(box(1),box(2),box(3)-1)=2;
   count=count+1;
end

%Use "number" to access box coordinates from the "boxesTwo" array
number=0;
%Use "done" to terminate the while loop after all boxesTwo have been found
done=0;

while (done==0)

   number=number+1;
   if(boxesTwo(number,1)>99 || boxesTwo(number,2)>99 || boxesTwo(number,3)>99)
      done=1;
   else
      if(Box(boxesTwo(number,1)+1,boxesTwo(number,2),boxesTwo(number,3))==0)
         boxesTwo(count,:)=[boxesTwo(number,1)+1,boxesTwo(number,2),boxesTwo(number,3)];
         Box(boxesTwo(number,1)+1,boxesTwo(number,2),boxesTwo(number,3))=2;
         count=count+1;
      end

      if(Box(boxesTwo(number,1)-1,boxesTwo(number,2),boxesTwo(number,3))==0)
         boxesTwo(count,:)=[boxesTwo(number,1)-1,boxesTwo(number,2),boxesTwo(number,3)];
         Box(boxesTwo(number,1)-1,boxesTwo(number,2),boxesTwo(number,3))=2;
         count=count+1;
```

```
        end

        if(Box(boxesTwo(number,1),boxesTwo(number,2)+1,boxesTwo(number,3))==0)
            boxesTwo(count,:)=[boxesTwo(number,1),boxesTwo(number,2)+1,boxesTwo(number,3)];
            Box(boxesTwo(number,1),boxesTwo(number,2)+1,boxesTwo(number,3))=2;
            count=count+1;
        end

        if(Box(boxesTwo(number,1),boxesTwo(number,2)-1,boxesTwo(number,3))==0)
            boxesTwo(count,:)=[boxesTwo(number,1),boxesTwo(number,2)-1,boxesTwo(number,3)];
            Box(boxesTwo(number,1),boxesTwo(number,2)-1,boxesTwo(number,3))=2;
            count=count+1;
        end

        if(Box(boxesTwo(number,1),boxesTwo(number,2),boxesTwo(number,3)+1)==0)
            boxesTwo(count,:)=[boxesTwo(number,1),boxesTwo(number,2),boxesTwo(number,3)+1];
            Box(boxesTwo(number,1),boxesTwo(number,2),boxesTwo(number,3)+1)=2;
            count=count+1;
        end

        if(Box(boxesTwo(number,1),boxesTwo(number,2),boxesTwo(number,3)-1)==0)
            boxesTwo(count,:)=[boxesTwo(number,1),boxesTwo(number,2),boxesTwo(number,3)-1];
            Box(boxesTwo(number,1),boxesTwo(number,2),boxesTwo(number,3)-1)=2;
            count=count+1;
        end
    end
        %Set done to 1 (true) if all elements of boxesTwo array have been tested
        %for neighboring empty boxesTwo
        boxesSize=size(boxesTwo);
        if(number>=boxesSize(1))
            done=1;
        end

end

end
```

Main code for “Heart2D.m”:

```
%Making sure the same "random" numbers aren't generated each time code is
%run
RandStream.setDefaultStream(RandStream('mt19937ar','seed',sum(100*clock)));
uniqueCounter=0;


heart=zeros(408820,3);
```

```
for var=1:10;

if(var==1)
count=1;
for i=1:230532;
   %Add 50 to z values so entire heart is in first octant
   heart(i,:)=[data(i,1),data(i,2),data(i,3)+50];
   count=count+1;
end
sizeTwo=count-1;
end

if(var==2)
count=1;
for i=1:199941;
   heart(i,:)=[data2(i,1),data2(i,2),data2(i,3)+50];
   count=count+1;
end
sizeTwo=count-1;
end

if(var==3)
count=1;
for i=1:212657;
   heart(i,:)=[data3(i,1),data3(i,2),data3(i,3)+50];
   count=count+1;
end
sizeTwo=count-1;
end

if(var==4)
count=1;
for i=1:214067;
   heart(i,:)=[data4(i,1),data4(i,2),data4(i,3)+50];
   count=count+1;
end
sizeTwo=count-1;
end

if(var==5)
count=1;
for i=1:188417;
   heart(i,:)=[data5(i,1),data5(i,2),data5(i,3)+50];
   count=count+1;
end
sizeTwo=count-1;
end

if(var==6)
count=1;
for i=1:385866;
   heart(i,:)=[data6(i,1),data6(i,2),data6(i,3)+50];
   count=count+1;
```

```
end
sizeTwo=count-1;
end

if(var==7)
count=1;
for i=1:408820;
   heart(i,:)=[data7(i,1),data7(i,2),data7(i,3)+50];
   count=count+1;
end
sizeTwo=count-1;
end

if(var==8)
count=1;
for i=1:403240;
   heart(i,:)=[data8(i,1),data8(i,2),data8(i,3)+50];
   count=count+1;
end
sizeTwo=count-1;
end

if(var==9)
count=1;
for i=1:213669;
   heart(i,:)=[data9(i,1),data9(i,2),data9(i,3)+50];
   count=count+1;
end
sizeTwo=count-1;
end

if(var==10)
count=1;
for i=1:207740;
   heart(i,:)=[data10(i,1),data10(i,2),data10(i,3)+50];
   count=count+1;
end
sizeTwo=count-1;
end

%Rotation stuff


for num=1:10;
   uniqueCounter=uniqueCounter+1;

clear Box
clear Box2
clear X
clear Y
clear Z


BSL=num;
```

```
Box=zeros(ceil(200/BSL),ceil(200/BSL),ceil(200/BSL));
%Use Box2 array for isPoly function
Box2=zeros(ceil(200/BSL),ceil(200/BSL),ceil(200/BSL));

%Find boxes touching heart
for i=1:sizeTwo;
   if(heart(i,3)<120)
      %Place box at point
      xBox1=(ceil(heart(i,1)*(1/BSL)));
      yBox1=(ceil(heart(i,2)*(1/BSL)));
      zBox1=(ceil(heart(i,3)*(1/BSL)));
      Box(xBox1,yBox1,zBox1)=1;
      Box2(xBox1,yBox1,zBox1)=1;
   end
end

%Creating bounding plane to isolate ventricle
for i=ceil(1/BSL):ceil(200/BSL);
   for k=ceil(1/BSL):ceil(200/BSL);
      Box(i,70,k)=1;
      Box2(i,70,k)=1;
   end
end

IP=[ceil(mean(heart(1:sizeTwo,1))/BSL),ceil(mean(heart(1:sizeTwo,2)-
8)/BSL),ceil(mean(heart(1:sizeTwo,3))/BSL)];

%Find boxes inside a slice
for i=ceil(4/BSL):ceil(190/BSL);
   clear boxes
   countWhat=1;
   for j=ceil(10/BSL):ceil(140/BSL);
      for k=ceil(10/BSL):ceil(190/BSL);
         if(Box(i,j,k)==1)
            boxes(countWhat,:)=[i,j,k];
            countWhat=countWhat+1;
         end
      end
   end
   if(countWhat>=8)
      box=[ceil(mean(boxes(:,1))),ceil(mean(boxes(:,2)))-ceil(8/BSL),ceil(mean(boxes(:,3)))];
      [polygon,Box2]=isPoly(box,Box2,BSL);
      if(polygon==true)
         Box=checkBoxes2D(box,Box);
      end
   end
end


countTwo=1;
for i=ceil(100/BSL):ceil(180/BSL);
   for j=ceil(1/BSL):ceil(140/BSL);
```

```
      for k=ceil(10/BSL):ceil(190/BSL);
         if(Box(i,j,k)==2)
            X(countTwo)=i*BSL;
            Y(countTwo)=j*BSL;
            Z(countTwo)=k*BSL;
            countTwo=countTwo+1;
         end
      end
   end
end


volume=countTwo*BSL^3
for i=1:countTwo-1;
   m(i,:)=[X(i),Y(i),Z(i)];
end

n(uniqueCounter,:)=[BSL,countTwo,volume];

%scatter3(X,Y,Z)

end
%end

end

xlswrite('U:\Research\HeartDimensionsRaw.xlsx',n);
```

Subfunction “isPoly”:

```
%Input:  Array of boxes (boxes), Box array (Box), point in
%middle of boxes (box)

%Output:  Returns true if boxes form a closed polygon, false otherwise

function [polygon,Box2]=isPoly(box,Box2,BSL)

polygon=true;
Box2(box(1),box(2),box(3))=2;
count=1;

if(Box2(box(1),box(2),box(3)+1)==0)
   boxesTwo(count,:)=[box(1),box(2),box(3)+1];
   Box2(box(1),box(2),box(3)+1)=2;
   count=count+1;
end
if(Box2(box(1),box(2),box(3)-1)==0)
   boxesTwo(count,:)=[box(1),box(2),box(3)-1];
   Box2(box(1),box(2),box(3)-1)=2;
   count=count+1;
```

```
end
if(Box2(box(1),box(2)+1,box(3))==0)
   boxesTwo(count,:)=[box(1),box(2)+1,box(3)];
   Box2(box(1),box(2)+1,box(3))=2;
   count=count+1;
end
if(Box2(box(1),box(2)-1,box(3))==0)
   boxesTwo(count,:)=[box(1),box(2)-1,box(3)];
   Box2(box(1),box(2)-1,box(3))=2;
   count=count+1;
end

%Use "done" to terminate the while loop after all boxesTwo have been found
done=0;

if(count==1)
   done=1;
   polygon=false;
end

%Use "number" to access box coordinates from the "boxesTwo" array
number=0;

while (done==0)

   number=number+1;

      if(boxesTwo(number,1)==1 || boxesTwo(number,1)==ceil(200/BSL) ||
boxesTwo(number,2)==1 || boxesTwo(number,2)==ceil(200/BSL) || boxesTwo(number,3)==1 ||
boxesTwo(number,3)==ceil(200/BSL))
         polygon=false;
         done=1;

      else

         if(Box2(boxesTwo(number,1),boxesTwo(number,2),boxesTwo(number,3)+1)==0)

boxesTwo(count,:)=[boxesTwo(number,1),boxesTwo(number,2),boxesTwo(number,3)+1];
            Box2(boxesTwo(number,1),boxesTwo(number,2),boxesTwo(number,3)+1)=2;
            count=count+1;
         end

         if(Box2(boxesTwo(number,1),boxesTwo(number,2),boxesTwo(number,3)-1)==0)
            boxesTwo(count,:)=[boxesTwo(number,1),boxesTwo(number,2),boxesTwo(number,3)-
1];
            Box2(boxesTwo(number,1),boxesTwo(number,2),boxesTwo(number,3)-1)=2;
            count=count+1;
         end

         if(Box2(boxesTwo(number,1),boxesTwo(number,2)+1,boxesTwo(number,3))==0)

boxesTwo(count,:)=[boxesTwo(number,1),boxesTwo(number,2)+1,boxesTwo(number,3)];
            Box2(boxesTwo(number,1),boxesTwo(number,2)+1,boxesTwo(number,3))=2;
            count=count+1;
```

```
        end

        if(Box2(boxesTwo(number,1),boxesTwo(number,2)-1,boxesTwo(number,3))==0)
            boxesTwo(count,:)=[boxesTwo(number,1),boxesTwo(number,2)-
1,boxesTwo(number,3)];
            Box2(boxesTwo(number,1),boxesTwo(number,2)-1,boxesTwo(number,3))=2;
            count=count+1;
        end

    end

    %number=number+1;

    %Set done to 1 (true) if all elements of boxesTwo array have been tested
    %for neighboring empty boxesTwo
    boxesSize=size(boxesTwo);
    if (number>=boxesSize(1))
        done=1;
    end

end

end
```

Subfunction “checkBoxes2D.m”

```
%Input:  Array of boxes forming a polygon (boxes), Box array, point in
%middle of polygon (box)

%Output:  Adds boxes inside the polygon to the Box array

function Box=checkBoxes2D(box,Box)

Box(box(1),box(2),box(3))=2;
count=1;

if(Box(box(1),box(2),box(3)+1)==0)
    boxesTwo(count,:)=[box(1),box(2),box(3)+1];
    Box(box(1),box(2),box(3)+1)=2;
    count=count+1;
end
if(Box(box(1),box(2),box(3)-1)==0)
    boxesTwo(count,:)=[box(1),box(2),box(3)-1];
    Box(box(1),box(2),box(3)-1)=2;
    count=count+1;
end
if(Box(box(1),box(2)+1,box(3))==0)
    boxesTwo(count,:)=[box(1),box(2)+1,box(3)];
    Box(box(1),box(2)+1,box(3))=2;
    count=count+1;
end
if(Box(box(1),box(2)-1,box(3))==0)
```

```
   boxesTwo(count,:)=[box(1),box(2)-1,box(3)];
   Box(box(1),box(2)-1,box(3))=2;
   count=count+1;
end


%Use "number" to access box coordinates from the "boxesTwo" array
number=0;
%Use "done" to terminate the while loop after all boxesTwo have been found
done=0;

while (done==0)

   number=number+1;

      if(Box(boxesTwo(number,1),boxesTwo(number,2),boxesTwo(number,3)+1)==0)
         boxesTwo(count,:)=[boxesTwo(number,1),boxesTwo(number,2),boxesTwo(number,3)+1];
         Box(boxesTwo(number,1),boxesTwo(number,2),boxesTwo(number,3)+1)=2;
         count=count+1;
      end

      if(Box(boxesTwo(number,1),boxesTwo(number,2),boxesTwo(number,3)-1)==0)
         boxesTwo(count,:)=[boxesTwo(number,1),boxesTwo(number,2),boxesTwo(number,3)-1];
         Box(boxesTwo(number,1),boxesTwo(number,2),boxesTwo(number,3)-1)=2;
         count=count+1;
      end

      if(Box(boxesTwo(number,1),boxesTwo(number,2)+1,boxesTwo(number,3))==0)
         boxesTwo(count,:)=[boxesTwo(number,1),boxesTwo(number,2)+1,boxesTwo(number,3)];
         Box(boxesTwo(number,1),boxesTwo(number,2)+1,boxesTwo(number,3))=2;
         count=count+1;
      end

      if(Box(boxesTwo(number,1),boxesTwo(number,2)-1,boxesTwo(number,3))==0)
         boxesTwo(count,:)=[boxesTwo(number,1),boxesTwo(number,2)-1,boxesTwo(number,3)];
         Box(boxesTwo(number,1),boxesTwo(number,2)-1,boxesTwo(number,3))=2;
         count=count+1;
      end

      %Set done to 1 (true) if all elements of boxesTwo array have been tested
      %for neighboring empty boxesTwo
      boxesSize=size(boxesTwo);
      if(number>=boxesSize(1))
         done=1;
      end

end

end
```

Creating a Koch Snowflake using “KochSnowflake.m”

```
%Create the Koch Snowflake for any desired number of iterations

%Set the number of iterations
iterations=7;
%Set the initial side length (ISL) for the first iteration
ISL=4;




for i=0:4;
%Set the Square Side Length
SSL=0.0125*2^i;

%Set point 1 far enough away from the origin (in Quadrant I) so no points have
%negative coordinate values.  Set values of other 2 initial points.
point(1,:)=[ISL,ISL];
point(2,:)=[ISL+ISL/2,ISL+ISL/2*sqrt(3)];
point(3,:)=[ISL*2,ISL];

%For loop to execute for each iteration
for i=1:iterations;
   %Set the number of sides based on the iteration using known formula
   sides=3*4^(i-1);
   %Set the side length based on the iteration using known formula
   sideLength=ISL*3^(-(i-1));
   %Use num so that points are "numbered" (i.e. placed into the array) in a clockwise manner
   num=1;

   %Set temporary points to reference later
   for j=1:sides-1;
       temp(j,:)=point(j+1,:);
   end
   count=1;

   %Place three extra points, plus two existing, into point array
   for k=1:sides;

      %If statement to take care of the case of the "closing side"
      if(k==sides)
         change=point(1,:)-point(num,:);
         %Get normal vector from point1 to point2
         magnitude=sqrt(change(1)^2+change(2)^2);
         normal=[change(1)/magnitude,change(2)/magnitude];
         point(num+4,:)=point(1,:);

point(num+1,:)=[point(num,1)+normal(1)*sideLength/3,point(num,2)+normal(2)*sideLength/3];

point(num+3,:)=[point(num,1)+normal(1)*sideLength*2/3,point(num,2)+normal(2)*sideLength*2/3]
;
         %Getting perpendicular vector
         if(normal(1)>0 && normal(2)==0)
            perp=[0,1];
```

```
            elseif(normal(1)<0 && normal(2)==0)
                perp=[0,-1];
            elseif(normal(1)==0 && normal(2)>0)
                perp=[-1,0];
            elseif(normal(1)==0 && normal(2)<0)
                perp=[1,0];
            else
                tempNum=normal(1);
                perp=[-normal(2),tempNum];
            end
            tempPoint=[point(num,1)+normal(1)*sideLength/2,point(num,2)+normal(2)*sideLength/2];
            tempLength=sideLength/6*sqrt(3);
            point(num+2,:)=[tempPoint(1)+tempLength*perp(1),tempPoint(2)+tempLength*perp(2)];
        else
            point(num+1,:)=temp(count,:);
            change=temp(count,:)-point(num,:);
            %Get normal vector from point1 to point2
            magnitude=sqrt(change(1)^2+change(2)^2);
            normal=[change(1)/magnitude,change(2)/magnitude];
            point(num+4,:)=point(num+1,:);

point(num+1,:)=[point(num,1)+normal(1)*sideLength/3,point(num,2)+normal(2)*sideLength/3];

point(num+3,:)=[point(num,1)+normal(1)*sideLength*2/3,point(num,2)+normal(2)*sideLength*2/3]
;
            %Getting perpendicular vector
            if(normal(1)>0 && normal(2)==0)
                perp=[0,1];
            elseif(normal(1)<0 && normal(2)==0)
                perp=[0,-1];
            elseif(normal(1)==0 && normal(2)>0)
                perp=[-1,0];
            elseif(normal(1)==0 && normal(2)<0)
                perp=[1,0];
            else
                tempNum=normal(1);
                perp=[-normal(2),tempNum];
            end
            tempPoint=[point(num,1)+normal(1)*sideLength/2,point(num,2)+normal(2)*sideLength/2];
            tempLength=sideLength/6*sqrt(3);
            point(num+2,:)=[tempPoint(1)+tempLength*perp(1),tempPoint(2)+tempLength*perp(2)];
            num=num+4;
        end
        count=count+1;
    end
end

%Initialize all squares to zero (0=not touching surface, 1=touching surface)
for i=1:max(point(:,1))/SSL+2;
    for j=1:max(point(:,2))/SSL+2;
        Square(i,j)=0;
    end
end

%Finding squares on the boundary of the object
```

```
amount=size(point);

for i=2:amount(1,1);
   %Determine coordinates of each point of the line
   point1=point(i-1,:);
   point2=point(i,:);
   %Find squares that touch the line
   side=[point1;point2];
   Square=FindSquares(side,Square,SSL);
end

counter=1;
for i=1:max(point(:,1))/SSL+2;
   for j=1:max(point(:,2))/SSL+2;
      if(Square(i,j)==1)
         x(counter)=i*SSL;
         y(counter)=j*SSL;
         counter=counter+1;
      end
   end
end

%counter

End
```

Creating the Koch Surface using “Koch Surface.m”

```
%3D Square Koch Surface

%Choose a starting box, defined by side length and corner position

uniqueCounter=0;
for p=4:4;

if(p==1)
   angle=0;
   matrix=[1,0,0;0,1,0;0,0,1];
end

if(p==2)
   %angle=rand(1)*360
   angle=293.3;
   matrix=[1,0,0;0,cos(angle),-sin(angle);0,sin(angle),cos(angle)];
end

if(p==3)
   %angle=rand(1)*360
   angle=326.09;
   matrix=[cos(angle),0,-sin(angle);0,1,0;sin(angle),0,cos(angle)];
end
```

```
if(p==4)
   %angle=rand(1)*360
   angle=45.72;
   matrix=[cos(angle),-sin(angle),0;sin(angle),cos(angle),0;0,0,1];
end



%for q=1:3;
   for s=1:5;
uniqueCounter=uniqueCounter+1;
clear Box
clear squares
clear normal
clear num
clear r
clear rotated
clear squaresTwo

%Initial Box Length
IBL=4;
Vertex=[6,6,6];
center=[Vertex(1)+IBL/2,Vertex(2)+IBL/2,Vertex(3)+IBL/2];
BSL=0.05*2^(s-1);
iterations=3;
normal=[0,0,0;0,0,0];



for i=1:ceil((Vertex(1)+2*IBL)/BSL);
   for j=1:ceil((Vertex(2)+2*IBL)/BSL);
      for k=1:ceil((Vertex(3)+2*IBL)/BSL);
         Box(i,j,k)=0;
      end
   end
end

array(1,:)=Vertex;
array(2,:)=[Vertex(1)+IBL,Vertex(2),Vertex(3)];
array(3,:)=[Vertex(1),Vertex(2)+IBL,Vertex(3)];
array(4,:)=[Vertex(1)+IBL,Vertex(2)+IBL,Vertex(3)];
array(5,:)=[Vertex(1),Vertex(2),Vertex(3)+IBL];
array(6,:)=[Vertex(1)+IBL,Vertex(2),Vertex(3)+IBL];
array(7,:)=[Vertex(1),Vertex(2)+IBL,Vertex(3)+IBL];
array(8,:)=[Vertex(1)+IBL,Vertex(2)+IBL,Vertex(3)+IBL];

squares(1:4,:)=[array(1,:);array(2,:);array(3,:);array(4,:)];
normVec(1,:)=[0,0,-1];
squares(5:8,:)=[array(5,:);array(6,:);array(7,:);array(8,:)];
normVec(2,:)=[0,0,1];
squares(9:12,:)=[array(1,:);array(5,:);array(3,:);array(7,:)];
normVec(3,:)=[-1,0,0];
squares(13:16,:)=[array(2,:);array(6,:);array(4,:);array(8,:)];
normVec(4,:)=[1,0,0];
squares(17:20,:)=[array(1,:);array(2,:);array(5,:);array(6,:)];
normVec(5,:)=[0,-1,0];
```

```
squares(21:24,:)=[array(3,:);array(4,:);array(7,:);array(8,:)];
normVec(6,:)=[0,1,0];
normal(1:6,:)=normVec(1:6,:);

num=25;
for j=1:6;
   n=normVec(j,:);
   [Box,squares,normal,num]=addBox(Box,squares((4*(j-1)+1):(4*(j-
1)+4),:),n,squares,normal,num,BSL);
   num=num+4;
end

if(iterations>=1)

   for j=7:84;
      n=normal(j,:);
      [Box,squares,normal,num]=addBox(Box,squares((4*(j-1)+1):(4*(j-
1)+4),:),n,squares,normal,num,BSL);
      num=num+4;
   end
end

if(iterations>=2)
   for j=85:1098;
      n=normal(j,:);
      [Box,squares,normal,num]=addBox(Box,squares((4*(j-1)+1):(4*(j-
1)+4),:),n,squares,normal,num,BSL);
      num=num+4;
   end
end

if(iterations>=3)
   for j=1099:14280;
      n=normal(j,:);
      [Box,squares,normal,num]=addBox(Box,squares((4*(j-1)+1):(4*(j-
1)+4),:),n,squares,normal,num,BSL);
      num=num+4;
   end
end

if(iterations>=4)
   for j=14281:185646;
      n=normal(j,:);
      [Box,squares,normal,num]=addBoxTwo(Box,squares((4*(j-1)+1):(4*(j-
1)+4),:),n,squares,normal,num,BSL);
      num=num+4;
   end
end

%Option to rotate model
%Create random number between 0 and 360, then rotate whole object about
%x-axis by that amount

%Returns random number between 0 and 360
```

```
%angle=rand(1)*360
%angle=0;
%Define rotation matrix
   %For rotation about x-axis
%matrix=[cos(angle),-sin(angle),0;sin(angle),cos(angle),0;0,0,1];
%matrix=[1,0,0;0,cos(angle),-sin(angle);0,sin(angle),cos(angle)];


for i=1:size(squares);
   r(:,i)=squares(i,:)'-[center(1);center(2);center(3)];
   rotated(i,:)=matrix*r(:,i);
   squaresTwo(i,:)=rotated(i,:)+center;

   %X(i)=rotatedArray(i,1);
   %Y(i)=rotatedArray(i,2);
   %Z(i)=rotatedArray(i,3);
end



for i=1:ceil((Vertex(1)+2*IBL)/BSL);
   for j=1:ceil((Vertex(2)+2*IBL)/BSL);
      for k=1:ceil((Vertex(3)+2*IBL)/BSL);
         Box(i,j,k)=0;
      end
   end
end

if(iterations==0)
   for j=1:6;
      Box=findBoxesSquare(Box,squaresTwo((4*(j-1)+1):(4*(j-1)+4),:),BSL);
   end
end

if(iterations==1)
   for j=7:84;
      Box=findBoxesSquare(Box,squaresTwo((4*(j-1)+1):(4*(j-1)+4),:),BSL);
   end
end

if(iterations==2)
   for j=85:1098;
      Box=findBoxesSquare(Box,squaresTwo((4*(j-1)+1):(4*(j-1)+4),:),BSL);
   end
end

if(iterations==3)
   for j=1099:14280;
      Box=findBoxesSquare(Box,squaresTwo((4*(j-1)+1):(4*(j-1)+4),:),BSL);
   end
end

if(iterations==4)
   for j=14281:185646;
      Box=findBoxesSquare(Box,squaresTwo((4*(j-1)+1):(4*(j-1)+4),:),BSL);
```

```
   end
end


%{
count=1;
for i=1:ceil((Vertex(1)+2*IBL)/BSL);
   for j=1:ceil((Vertex(2)+2*IBL)/BSL);
      for k=1:ceil((Vertex(3)+2*IBL)/BSL);
         if(Box(i,j,k)==1)
            p(count,1)=i*BSL;
            p(count,2)=j*BSL;
            p(count,3)=k*BSL;
            count=count+1;
         end
      end
   end
end


Box=checkBoxes(Box,p,BSL);
%}

count=1;
for i=1:ceil((Vertex(1)+2*IBL)/BSL);
   for j=1:ceil((Vertex(2)+2*IBL)/BSL);
      for k=1:ceil((Vertex(3)+2*IBL)/BSL);
         if(Box(i,j,k)==1)
            X(count)=i*BSL;
            Y(count)=j*BSL;
            Z(count)=k*BSL;
            count=count+1;
         end
      end
   end
end

%count
%scatter3(X,Y,Z)

m=[BSL,count,angle,iterations]
xlswrite('U:\Research\KochSurfaceDimensionsRaw.xlsx',m,uniqueCounter)
   end
end
%end
```