Modeling blood vessels and oxygen diffusion into brain tissue

Thesis

Presented in Partial Fulfillment of the Requirements for the Degree Master of Mathematical Sciences in the Graduate School of The Ohio State University

By

Mark Alexander Caldwell, B.S.

Graduate Program in Mathematical Sciences

The Ohio State University

2019

Thesis Committee:

Dr. David Terman, Advisor

Dr. Yulong Xing

Abstract

The delivery of oxygen by blood through blood vessels is a critical process that enables cells and tissue to maintain functionality. This thesis focuses on this process at a small scale in the body. In particular, it examines the flow of blood through capillaries in the brain and how that blood diffuses oxygen into the surrounding tissue. These two models, blood flow and oxygen diffusion, are modeled and the two models are coupled. Two different implementations of this coupled model are used to solve the partial differential equation (PDE). First, an implementation in MATLAB based on the Finite Element Method (FEM) was written for this paper. Additionally, an implementation in C++ is used which is based on Green's Function Methods. This implementation was written by Secomb and can be found at github [3]. These implementations are used to examine oxygen transport and the effects of heterogeneity on it (differences in oxygen content of blood vessels that are close to each other). Heterogeneity tends to occur alongside diseases. We aim to show that this heterogeneity decreases the total oxygen in surrounding tissue. Additionally, we aim to show what the consequences of heterogeneity and the decrease in oxygen in the tissue would be on nearby neurons.

Acknowledgments

I would like to thank my advisor, Dr. David Terman, and all others who helped guide me with my research and thesis. I would also like to thank my family and friends for the support throughout my program.

Vita

North Canton Hoover High School …….2012

The Ohio State University – B.S. Applied Physics …….2015

The Ohio State University – B.S. Computer and Information Sciences …….2015

The Ohio State University – Graduate Teaching Associate …….2015-present

Fields of Study

Major Field: Mathematical Sciences

Table of Contents

List of Tables

List of Figures

Chapter 1. Introduction

Blood is used to transport oxygen throughout the human body and supply it to tissues and organs. As blood travels through vessels, it diffuses this oxygen into the surrounding tissue. Various factors come into play regarding the distance that oxygen can diffuse into tissue; however, due to low solubility of oxygen inside of tissue, this distance is often small [6]. Specific factors that impact diffusion of oxygen examined by this thesis will be noted in Chapter 2. Because of the relatively small diffusion distance, it is important for blood to be delivered reliably and efficiently.

Blood vessels and tissue within the brain are the focus of this thesis. Since the brain is an organ that is vital to human life, it must be supplied with enough oxygen. Without proper amounts of oxygen, problems arise. Cells, tissue, and neurons, in particular, can all fail to reproduce or even die without enough oxygen. This lack of oxygen can occur for many reasons including diseases or conditions such as strokes or heart attacks or even simply old age. In the brain, enough deficiency in oxygen can lead to death. The main purpose of this thesis is to simulate oxygen levels in tissue

surrounding a capillary network based on the blood oxygen content coming into the vessel network. A mathematical model is used to simulate this.

At a young age, vessels near each other tend to exhibit homogeneity. This means that capillaries near each other will have very similar qualities. As a human ages, however, heterogeneity tends to emerge due to plaque buildup among other factors. Differences in blood vessels close to each other tend to become more pronounced. Potentially, certain capillaries could become blocked off entirely. We will show that an increase in heterogeneity leads to a decrease in oxygen diffused into tissue surrounding the vessel network even if the total amount of oxygen in the vessel network is the same. Existing research has been done regarding oxygen diffusion [6][7][8]; however, there is a bit of a gap in the research regarding the effects of the aforementioned heterogeneity. As such, this will also be a focus of the results and conclusions in this paper.

Finally, by examining the oxygen necessary for a neuron to fire or even live, one can determine the amount of oxygen in a given tissue area in or around the vessel network and determine if a neuron will survive in this area or not. If the neuron would survive, tests could then be run under various conditions to see which factors would cause this neuron to stop firing and which factors could cause the neuron to die [9].

A brief outline of the rest of this thesis is as follows. Chapter 2 of this thesis will discuss the mathematical model used for blood flow and diffusion. Within this section, we break down the governing equations as well as variables and the choices made for the implementations. Chapter 3 will review both of the implementations used. Since two different implementations are used, pros and cons of each will be discussed. In Chapter 4,

results and a discussion of them is presented. We will examine the effects of heterogeneity as mentioned above. We also examine the effects of varying different parameters provided in the model. Finally, Chapter 5 will include a brief conclusion to the thesis as well as a discussion about future work and how the results from this thesis can be used in other applications.

Chapter 2. The Mathematical Model for Blood Flow and Blood Diffusion

In order to capture the physiological process of oxygen delivery, one must model the path that oxygen takes from blood into tissue. As such, there are two major models used in this thesis. The first models how oxygen is transported inside of blood vessels and capillaries. This begins with oxygen enriched blood at the start of a capillary and models how the oxygen levels inside of the blood vessel changes. Since oxygen (O_2) is diffusing into the surrounding tissue, vessels will see a drop in the partial pressure of oxygen down the length of the vessel. For reference, the partial pressure of oxygen is simply the measurement of oxygen in arterial blood. Once the values of the O_2 partial pressure (partial pressure of oxygen) have been obtained for each vessel in the network, the second model comes into play. The oxygen diffusion model takes the output of the blood flow model and is used to calculate the O_2 partial pressure in the surrounding tissue [9]. By combining these two models, we obtain the final physiological process of the delivery of oxygen from the beginning of a capillary network and into the surrounding tissue.

2.1 Blood Flow Model

The main purpose of the blood flow model is to analyze the oxygen levels down a capillary. This model can be used multiple times when networks of capillaries are used as opposed to one single capillary. For generality, capillaries modeled are assumed to be cylinders.

2.1.1 Oxygen Transport Rate Equation

The rate of convective oxygen transport along a vessel is given by:

$$\Phi(P_b) = Q\left(\alpha_b P_b + C_D H_D S(P_b)\right)$$

In this equation, Q is the volumetric rate of flow of blood in the capillary. In our model, this can be calculated by using the radius of the blood vessel in question as well as the velocity (in micrometers/sec throughout this thesis) of the blood entering the vessel. In particular:

$$v \times r_c^2 \times \pi$$

Where v is velocity and r is the radius.

α_b is the effective solubility of oxygen within blood plasma. O_2 partial pressure in the blood leads to oxygen being dissolved in the blood. A higher pressure causes more oxygen to be dissolved in the blood. However, oxygen dissolved in blood represents only a relatively small amount of oxygen carried. Most oxygen ends up being bound to a protein known as hemoglobin [5]. This solubility can be represented by the following equation:

$$\alpha_b = (1 - H_D)\alpha_p + H_D \alpha_{rbc}$$

Here, H_D is the same as it is in the transport rate equation. α_p is the solubility of blood in plasma and α_{rbc} is the solubility of blood in red blood cells. As stated previously, dissolved oxygen delivered is generally very small compared to hemoglobin-bound oxygen, but could be more significant in cases of very low hematocrit. Since both α values in this equation are very similar, the dependence of α_b on the hematocrit level is fairly low and this equation is treated as a constant in this thesis [6].

C_D is the concentration of hemoglobin-bound oxygen in a fully saturated red blood cell (how much oxygen is in a fully saturated red blood cell). This helps to determine the amount of oxygen available due to hemoglobin, along with H_D and S(P).

H_D is the hematocrit level. Hematocrit is the ratio of the volume of red blood cells to the total volume of blood. If hematocrit is too low, a person is described as being anemic. Generally, the normal hematocrit levels for a person can very with gender and age. For men, a normal range is considered to be between 38 and 48 percent. For women, the range is shifted down slightly and is considered to be between 35 and 44 percent [4]. For our purposes, the model uses 40% or 0.4 for H_D unless stated otherwise.

P_b is the O_2 partial pressure inside of the blood vessel and is the end result of this model. It is shown this P_b will vary along the length of the vessel. Due to this, we will generally start with an initial P_b and will use that to solve for $P_b(x)$.

Finally, S(P) is the fractional oxyhemoglobin saturation curve. This function is represented by the Hill equation [6]:

$$S(P) = \frac{P^h}{P_{50}^h + P^h}$$

Here, P_{50} represents the O_2 partial pressure at 50% hemoglobin saturation and h represents the hemoglobin saturation Hill coefficient and is constant. P_{50} is generally taken to be around 38 mmHg and h is generally taken to be 2.7 and is dimensionless. A plot of this curve is pictured below:

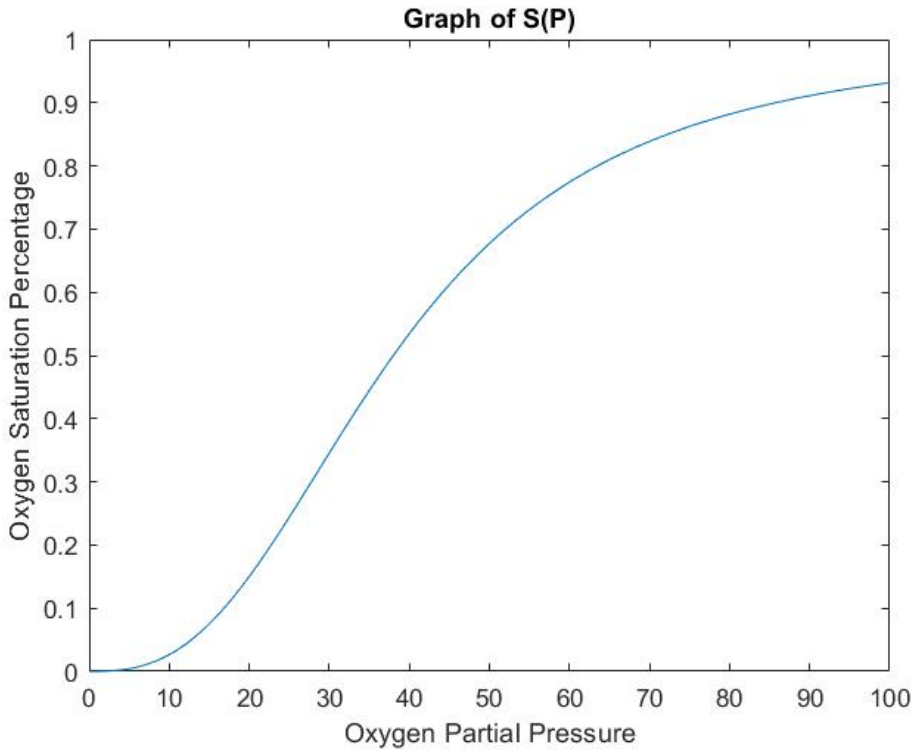

Figure 2.1: Oxyhemoglobin saturation curve

2.1.2 Oxygen Level Differential Equation

Given the oxygen transport rate equation described above, it is possible to formulate a differential equation to find P_b. Due to conservation of oxygen, it is necessary

to balance the flow across the boundary of the capillary with the total flux along the capillary. This gives the equation:

$$\frac{d\Phi(P_b)}{dx} = -q_v(x) \quad \text{where} \quad q_v(x) = -\alpha_T D \int_0^{2\pi} \frac{\partial P_T}{\partial r}(r_c)\, d\theta .$$

Here, D is the oxygen diffusion constant in tissue. α_T is the solubility of oxygen in tissue and P_T is the O_2 partial pressure in the surrounding tissue. Additionally, r_c is the radius of the capillary in question and r is the radial distance from the center line in the capillary. The integral simply integrates around the circumference of the capillary. This equation leads to the following differential equation which can be solved for P_b:

$$\frac{dP_b}{dx} = \frac{2D}{Qr_c\left(\alpha_b + C_D H_D S'(P_b)\right)} \frac{dP_T}{dr}(r_c)$$

Note that the derivative of the oxyhemoglobin saturation is used. This equation is solved by introducing the initial condition of $P_b(0)$, that is the partial pressure of oxygen at the beginning of the capillary. For multiple capillaries, this equation must me solved multiple times. Additionally, if capillaries are connected to each other, branch points must be considered and conservation of blood flow must be maintained. If one capillary branches into two, the flow of blood in the two branch capillaries must equal the flow in the original capillary.

2.2 Oxygen Diffusion Model

After solving for the oxygen levels in each capillary, it is possible to solve for the partial pressure of oxygen in the surrounding tissue by modeling the effects of diffusion of oxygen from capillaries and into the tissue. The O_2 partial pressure in the tissue satisfies the partial differential equation:

$$\Delta P_T = \frac{M_{OX}}{\alpha_T D} \Gamma\left(P_T\right) \equiv \delta \Gamma\left(P_T\right)$$

In this equation, P_T is the O_2 partial pressure in the tissue and is the goal. M_{OX} is the metabolic consumption rate of oxygen when oxygen is not limited. α_T is the solubility of oxygen in the tissue. D is the uniform oxygen diffusivity. Delta is used to combine the previous three variables into one for ease of use. This delta is used to vary the strength of the metabolism in both implementations as all three variables are related to consumption or diffusion of oxygen [6]. Finally:

$$\Gamma\left(P_T\right) = \frac{P_T}{K_T + P_T}$$

Here, K_T is a constant that represents partial pressure of oxygen in the tissue at a half maximum consumption rate.

Although both implementations of the model handle the boundary conditions slightly differently, this chapter includes a brief discussion of the boundary conditions. In general, there are a couple of different ways to represent conditions along the boundary. At the edge of the tissue region, no-flux boundary conditions can be imposed. There are other choices for the edges of the tissue that will be discussed in the chapter on implementation. At the boundary between the tissue and the capillaries, the diffusive

oxygen flux across the boundary must be continuous [6]. Specifically, this relationship can be defined with the equation:

$$P_{av}(x) = P_b(x) - K\, q_v(x)$$

As this equation is a function of x, it can be used to represent the boundary for the entire length of the capillary. Here, P_b is the partial pressure of oxygen within the capillary, as before. P_{AV} represents the O_2 partial pressure of the tissue averaged around the circumference of the capillary. K represents intravascular resistance to radial oxygen transport. This K is assumed to be a constant in this model.

As this partial differential equation (PDE) gives the value of P_b throughout the tissue domain, it can then be used as part of the equation that involves finding oxygen levels in the blood vessel. Since the solution to each of these equations gives a value that can be used in the other, it is possible to go back and forth between the two to see if the O_2 partial pressure in the tissue will settle down to a specific distribution.

Chapter 3. Implementation

Two implementations of the mathematical model were used in this thesis. For each implementation, there will be a brief overview of the numerical algorithm used. Additionally, there will be a discussion of the input to the program as well as the output generated by the program. Following that, there will be a section on the actual implementation, including language specific details. Finally, a discussion of the strengths and weaknesses of each algorithm is presented.

First, an implementation was written in MATLAB. This utilizes MATLAB functions for solving ordinary differential equations [2] and also uses a MATLAB partial differential equation solver that uses the finite element method. Despite having some limitations, this implementation is a relatively good approximation to the solution. The second implementation is written by Secomb and is in C++ [3]. In addition, there is a MATLAB version of this Green's Function implementation which was used to grasp the workings of the code as it was easier to read. Specific benefits and drawbacks to each implementation are discussed in their respective sections.

3.1 Finite Element Method Implementation

As previously stated, the first implementation used was based on the finite element method.

3.1.1 Methodology

There are two main sections for of this program, corresponding to the two sections of the model. The first section is finding the oxygen levels inside of the vessels in the capillary network. For this, a general ordinary differential equation (ODE) solver [2] built into MATLAB is used to solve the differential equation presented in Chapter 2.

$$\frac{dP_b}{dx} = \frac{2D}{Qr_c\left(\alpha_b + C_D H_D S'\left(P_b\right)\right)} \frac{dP_T}{dr}\left(r_c\right)$$

The second main part of this program solves the second half of the model. As stated before, this is a partial differential equation and will require a PDE solver. The finite element method (FEM) is a numerical method for solving boundary value problems for partial differential equations such as the one found in the above model. As a brief aside, some introductory information on the FEM is given.

The finite element method involves taking the domain used and dividing it up into small subdomains, also known as elements. The name is derived from the fact that there are a finite amount of these elements. Approximate solutions are generated in each subdomain and then the entire solution is pieced together in the end. A simple, one dimensional example is given below:

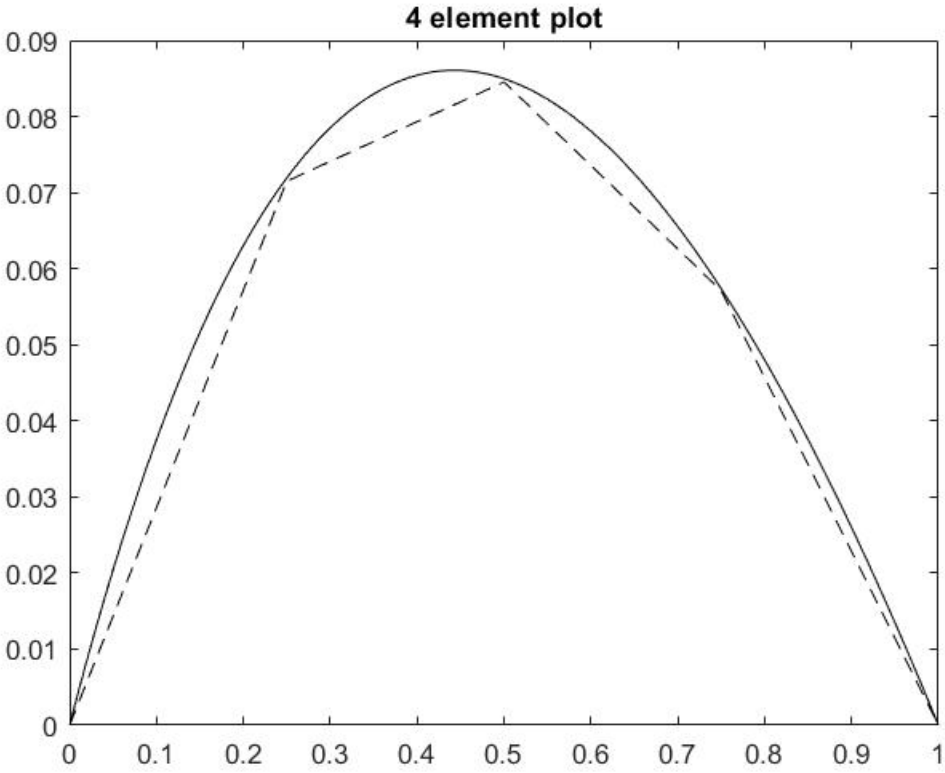

Figure 3.1: Sample FEM plot

This figure is a plot of the solution to the following PDE.

$$-\frac{d}{dx}\left((1+x)\frac{du}{dx}\right) = 1, \quad 0 < x < 1,$$

$$u(0) = u(1) = 0.$$

Note that the solid line is the exact solution to the PDE and the dashed line is the approximation using the FEM with four equal sized, linear elements. The endpoints of each element are equal to the exact solution. With more elements and higher order

elements (cubic, quartic, etc.), the total approximate solution becomes closer to the exact solution.

This method can be expanded to two and three dimensions. With two dimensions, elements used are generally triangles or squares. More dimensions obviously adds more complexity. For three dimensions, many types of shapes combining triangles and squares into 3D objects can be used for elements. Elements are then combined like in the above figure to fill out the mesh. In the case of our model, the tissue domain is the mesh and is broken up into elements by the algorithm. At the edges of the domain boundary conditions must be imposed.

3.1.2 Implementation Details

MATLAB is a tool that implements many mathematical features efficiently and is good for readability and convenience. For these reasons, MATLAB was chosen to implement the FEM version of the model. In this subsection, all implementation details will be discussed. This includes program input, output, and function calls.

To begin, the input has two sections. First is simply input variables. These include most of the variables discussed in Chapter 2. In particular, the input variables that will be changed most are metabolism-based variables (M, D, and delta) as well as blood flow-based variables (blood velocity in each capillary and initial O_2 partial pressure for each capillary).

The other input is the mesh domain which is stored as a .stl file. This file must be generated elsewhere. This is an abbreviation of stereolithography and is used with

computer aided design (CAD) software to describe surface geometry of figures. It is often used in applications such as 3D printing. In our implementation, the .stl file is imported into MATLAB and used as the tissue domain. The software used for the .stl file generation is called SketchUp. SketchUp is a 3D modeling program by Google that allows drawing meshes [1]. Once a 3D model is generated it is exported as a .stl file and can be properly imported into the MATLAB as long as the model is valid. In MATLAB, nested geometries are not allowed. An example of a model made for this implementation is shown in figure 3.2 below.

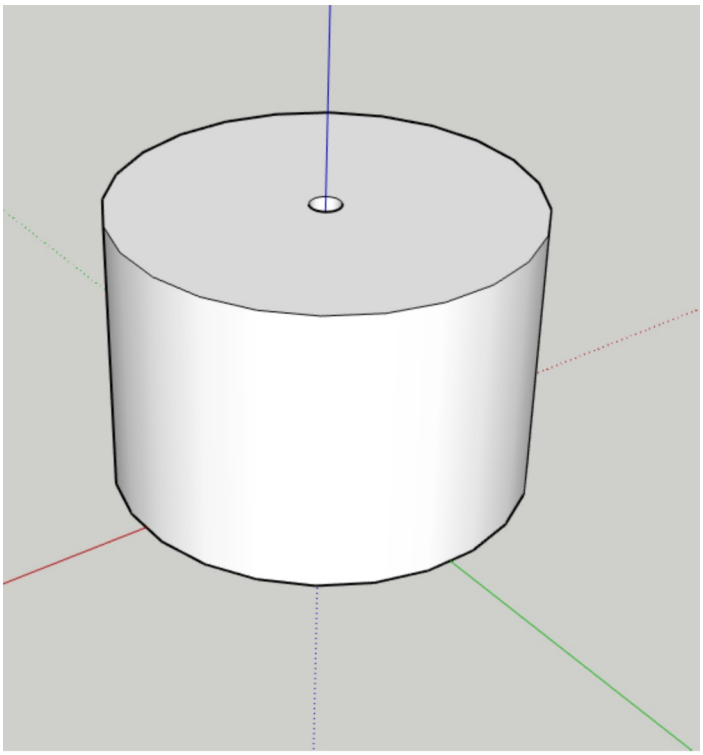

Figure 3.2: Sample SketchUp model

The .stl file pictured in figure 3.2 is of a relatively simple domain with one capillary down the center. Note that the entire domain is a large cylinder that represents the tissue domain. The inside of the capillary is not modeled. Instead the oxygen concentration is imposed as a boundary condition along the inside of the tissue domain where the capillary would be.

The MATLAB code begins with setup of input and variables. Then, the ODE model is implemented. This is done with the use of the ode45 function. There are many functions that can be used to solve ODEs in MATLAB and a table of these can be found in MATLAB documentation [2]. ode45 is useful for most problems and was suited well for this problem. The initial value for the ODE is given as the partial pressure of oxygen at the beginning of the capillary and it is solved down the entire length of the capillary (usually around 100 micrometers [8]). This process is done for each capillary.

The next section of the code solves the second model regarding diffusion of oxygen into tissue. First, the .stl file is imported using the importGeometry function. Then boundary conditions must be applied to the model by using applyBoundaryCondition of each face of the model. Each face can be given either neumann, dirichlet, or mixed boundary conditions. The outside of the model generally has no flux conditions. The faces of the model touching the capillaries are given Neumann conditions related to the oxygen concentration in the blood. Since this varies, a function must be passed into the applyBoundaryCondition function. Careful attention must be paid to make sure that the boundary conditions take into account the oxygen already in the tissue and the oxygen from other capillaries. In order to see how faces are

labeled, the pdegplot shows the numbered labels on the model to make boundary conditions easier to set. The next step of the process is to use the function specifyCoefficients. The MATLAB PDE solver is relatively limited in the forms of PDEs that are solvable. Currently, it solves PDEs of the form [10]:

$$m\frac{\partial^2 u}{\partial t^2} + d\frac{\partial u}{\partial t} - \nabla \cdot (c\nabla u) + au = f$$

In this equation, m, d, c, a, and f are coefficients that must be supplied. This is done through the specifyCoefficients function. For the model presented here, m, d, and a are all set to 0, c is set to 1, and f is set using the value of delta from the original PDE. Finally, the mesh is generated using generateMesh and the PDE is actually solved using solvepde.

After the result is generated, it must be analyzed and output must be generated. The first piece of output in this implementation is the percentage of tissue with sufficient oxygen. The user can supply the value, in mmHg, that is sufficient and the program will output the percentage as well as the average and total partial pressure in the tissue and a 3D plot of which areas of the tissue have sufficient oxygen. Additionally, a separate 3D plot is generated which shows the specific levels of O_2 partial pressure throughout the tissue. Finally, plots are made for each capillary. These plots have the oxygen distribution down the inside of each capillary as calculated from the ODE solver. They also show the oxygen distribution just outside of the capillary by averaging the values of the O_2 partial pressure around the circumference around each capillary at various points. As an

intermediary step, a plot of the values around a section of capillary was also made. This plot can be found below in figure 3.3.

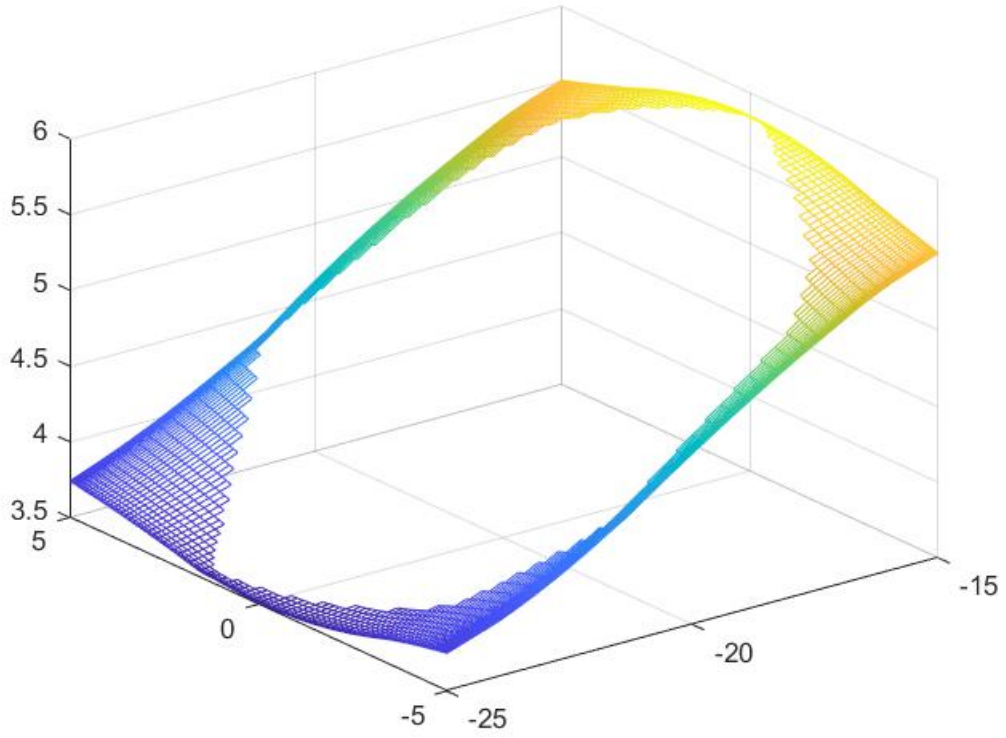

Figure 3.3: Partial pressure surrounding a capillary

Images of other specific output plots described here can be found in the results section.

3.1.3 Limitations

Although this implementation of the model works fairly well, there are a few significant limitations to point out. The first and most important is that the PDE solver is limited to a certain subset of PDEs as noted in the previous subsection. In particular, the model requires that the right side of the PDE be variable with respect to P_T:

$$\Gamma\left(P_T\right)=\frac{P_T}{K_T+P_T}$$

Unfortunately, the right side of the solver requires an inputted 'f' value that can be variable, but cannot depend on P_T. This means that the metabolism of this implementation stays constant. Thus the PDE is slightly simplified, but is still similar to the stated model. Another limitation is on the input to the code. Specifically, for a new network to be generated, an entirely new .stl file must be generated through SketchUp. This can be a difficult process depending on experience with the software and level of complexity of the desired network. As a result many networks used in this implementation are relatively simple with no branching vessels. Additionally, this implementation does not work if all boundary conditions are neumann boundary conditions. Because of this, a few different runs were done with various boundary conditions. The most reasonable seemed to be imposing boundary conditions as the model suggests, but using mixed boundary conditions on the outside of the tissue furthest from the capillary. Here the implementation uses no flux on the boundary as well as setting the O_2 partial pressure to be equal to 0 on the boundary face farthest from the capillaries and assuming that the diffusion would not reach that far. A final limitation is the time taken to run the program

is fairly long. For example, running this code with four capillaries takes roughly 3 minutes. For these reasons, a second implementation of the model was also used to allow for greater accuracy.

3.2 Green's Function Implementation

A second implementation was found using Green's function. This was implemented by Secomb and is used as it does not have nearly as many limitations as the FEM implementation. First, this implementation accurately implements the PDE, allowing for the right side of the equation to be variable in P_T. This alone allows for a more accurate solution by using a variable metabolism. This implementation also alleviates the difficulty with inputting various networks that are more complicated. This is done simply through a network input file and will be discussed in subsection 3.2.2 with the implementation details. The problem with boundary conditions is also negated in this implementation. Secomb states a few different ways of handling conditions on the boundary, including the one used in our FEM implementation, but ends up using a different approach [6]. This will also be discussed later in this chapter. Finally, the program can run quite quickly. Depending on the resolution of results desired, the program can run tests for 10 or more capillaries in a few seconds.

3.2.1 Methodology

The implementation of Green's function also has 2 parts, corresponding to the 2 parts of the model. The blood flow model is implemented by using a series of studies that

developed a number of equations. These equations describe the viscosity of blood in vessels as a function of the diameter of the vessel and the hematocrit. Other equations used describe the partition of hematocrit in diverging bifurcations by using flow rates in each branch as well as vessel diameter and hematocrit. For an in depth proper analysis of this method as well as the equations used, see chapter one of *Microcirculation* [7]. This model results in the prediction of capillary flow rate, O_2 partial pressure within the capillary, and specific hematocrit levels for each capillary.

The second part of the implementation uses Green's function to implement the PDE from the model. The idea behind this implementation is that each blood vessel is represented by a finite number of oxygen sources, determined by the blood flow. The representation of O_2 partial pressure in the tissue is generated by adding up the values of the fields from each oxygen source [3]. This Green's function method was developed to avoid the use of the no flux boundary conditions and allows tissue domains of any shape. Particularly, this method embeds the capillary network and tissue region in an effectively infinite domain. Outside of the tissue region, the domain maintains the same diffusivity as the tissue region, but does not contain sources or sinks for any solutes (for this thesis, oxygen was the only solute used) [6].

According to Secomb, in order to solve our PDE model using the Green's function method, we will reformulate the original PDE into the form:

$$D\alpha\nabla^2 G = -\delta_3(\mathbf{x} - \mathbf{x}^*)$$

Here, G, the Green's function, is defined as the O_2 partial pressure at at a point x that results from a source point x*. Additionally, δ_3 is defined as the delta function in three dimensions. The potential, partial pressure in this model, is given by the equation:

$$P(\mathbf{x}) = \int_{\text{Sources}} G(\mathbf{x}; \mathbf{x}^*) q(\mathbf{x}^*)\, d\mathbf{x}^*$$

Note that the Green's function appears here again. The function q in this equation represents the distribution of source strengths in the domain. When embedding the capillary network and tissue in an infinite domain, the solution for G is simply:

$$G = G_1 = 1/(4\pi D\alpha|\mathbf{x} - \mathbf{x}^*|)$$

This makes sense as the farther away a tissue point is from a source, the larger |x-x*| will be. This in turn makes G smaller, effectively simulating diffusion. When placing the tissue in a finite domain, terms other than G_1 may be required. Oxygen sources are considered to be uniformly distributed around the circumference of the capillary [6].

3.2.2 Implementation Details

This method is implemented by Secomb in C++ [6]. A version with fewer features was also found in MATLAB but was only used in understanding the algorithm. Since the C++ version is more complete, it will be explained in this subsection. Input, output, and a brief explanation of the code will be discussed.

Input to this implementation is very different from input to the first implementation. All input is passed through .dat files that are placed in the folder where

the program is run. Seven input files are used, but only a few will be extensively covered in this thesis as some remain constant throughout testing. Samples of these input files can be found in Appendix A.

The input file modified most often is network.dat. This file essentially provides the structure of the capillary network and tissue domain. The beginning of this file allows for the size of the tissue domain in microns. Generally, sizes of 100x100x100 or 110x110x110 were used for the tests in this thesis. Additionally, the number of tissue points can be provided in a grid. When more tissue points are used, a more accurate result is given. The rest of the file specifies the location and dimensions of each capillary. For a given capillary, the only variables needed are the start and end points, the diameter, a relative flow, and hematocrit. Start and endpoints are defined through nodes. This makes creating new networks very simple. One simply needs to add a line for the new capillary for each capillary and a new node if necessary. The final few lines of this file give information about boundary nodes. Here, it is specified which nodes have blood coming into them, what the partial pressure is in these nodes, and what the incoming flow is.

Another input file of interest is the SoluteParams.dat file. In this file, parameters can be supplied that describe how blood, oxygen, and the tissue interact. Specifically, metabolism constants, solubility of oxygen, and Hill equation constants can be supplied, among others. This file is used to vary these parameters when performing tests. More solutes than oxygen can be provided, but oxygen alone was sufficient for this thesis.

VaryParams.dat is very helpful for testing as it allows for multiple runs. This allows a user to set a number of runs and to change a specific variable during each run.

Although the variables allowed for this are limited, it is still helpful. Allowed variables are noted in the file.

Other input files give intravascular resistance (IntravascRes.dat), formatting output plots (ContourParams.dat), rates of production in the tissue (tissrate.cpp.dat), and dependent variables from greens (postgreens.cpp.dat and PostGreensParams.dat). These files were rarely changed for the work in this thesis. Additionally, postgreens.cpp.dat and PostGreensParams.dat are not used at all and can be considered empty.

Many C++ files are used in combination to implement this method. They can be can be found on Secomb's github page [3]. This implementation approximates the network as a set of uniform segments inside of a cuboidal tissue shape. Each vessel is divided into subsegments that represent the sources of oxygen. The tissue is also divided into subsegments. Each subsegment represents a tissue node point where oxygen is calculated. Tissue node points are determined by outboun.cpp and analyzenet.cpp based on two possible methods which is discussed by Secomb. The tissue is then embedded in an infinite domain. This is implemented through the use of various matrices that correspond to the strengths that sources have on each other. Each element gives the partial pressure of oxygen at the midpoint of a tissue subregion or vessel subsegment resulting from a source at a different tissue or vessel region. In the end, a system of equations using these matrices and source strengths is solved iteratively. Secomb goes in depth into the specifics of this implementation in his paper [6].

The output of this program is also generated in many files and is placed in a folder. As stated, there are quite a few files so only the ones relevant to this thesis will be

described. First is a contour file called CountourXXX.ps where XXX is the run number (for instance, run 20 would be 020). This shows the oxygen levels in the tissue in the domain and overlays the vessel network on top of it.

Another file generated is summary.out. This file gives a summary of all runs generated from this test. It notes the parameters that were changed for each run as well as the average partial pressure of oxygen in the entire tissue domain. This is relatively useful, but is refined later on.

TissueLevelsXXX.out gives the partial pressure of oxygen at each tissue point. For most runs, 1000 tissue points are used. This also gives the mean, standard deviation, min, and max of the oxygen levels. VesselLevelsXXX.out is also similar. This gives the oxygen levels in each vessel. All four other statistics are also used in the vessel levels documents.

A MATLAB code was also written in order to help visualize the results of this implementation. This code finds the average O_2 partial pressure in a specific area in the tissue domain. Particularly, this will calculate the oxygen level in between four capillaries and also in the lower third of the tissue domain in between the four capillaries. If multiple runs are used in a test, this prints the values for each run. This result is interesting if a neuron is placed in the middle of the capillary network. Then it is possible to determine if the neuron is receiving enough oxygen to fire.

Chapter 4. Results

A number of results were made in order to examine the process of oxygen transport through blood vessels and into tissue. Initial runs were first completed in order to visualize the effects of oxygen diffusion and to verify correctness. Additionally, testing was performed on the effects of varying different parameters. These parameters include vessel radius, partial pressure of oxygen into capillaries, metabolism, flow rate, among others. Finally, tests were run to examine the effects of heterogeneity on the surrounding tissue. In particular, the O_2 partial pressure in between four capillaries was examined when varying the amount of heterogeneity in the four capillaries. For instance, if one capillary had a large radius while the remaining three had small radii. A depiction of a network used for such tests is shown below.

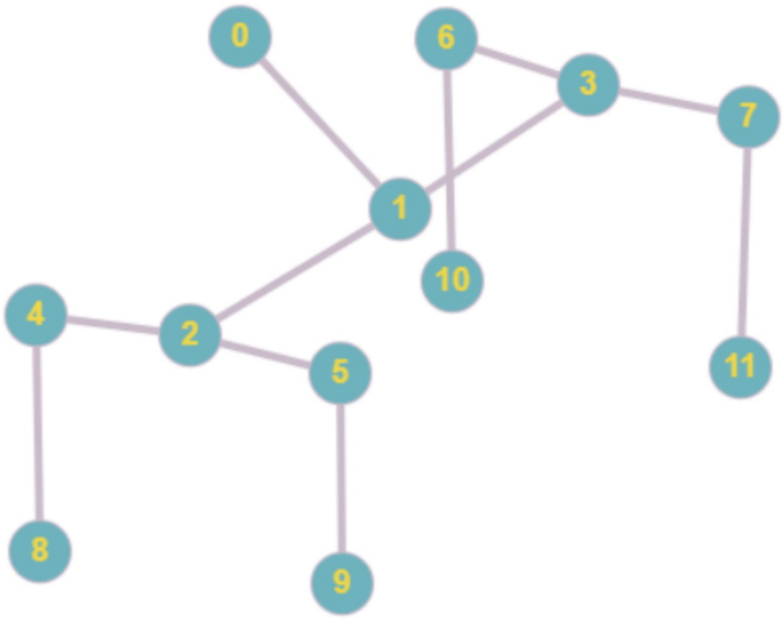

Figure 4.1: Sample capillary network

4.1 F.E.M. Results

Several tests were run with this implementation. Despite the relative inaccuracy of this implementation, it produces nice visuals and can be used to analyze the effects of heterogeneity in the broad sense.

First we examine sample results with two capillaries with a five micron radius. They are 30 microns apart and are embedded in a tissue cylinder with a radius of 60 microns. The tissue region has a height of 80 microns, giving the capillaries that same length. For the two runs presented here, the velocity is varied and the initial partial pressure for both capillaries is 60 mmHg. The first run, depicted in figure 4.2, gives a

velocity of 800 for both capillaries. The second run changes the velocity of the first capillary to 1200 and the second capillary to 400. A 3D plot of each is shown below. Note that the 3D plot shows the O_2 partial pressure in mmHg at any given location in the tissue region by use of the color bar. It is possible to create contour plots of various levels to visualize oxygen levels at locations in the middle of the tissue (as done earlier to find oxygen levels around the circumference of each capillary).

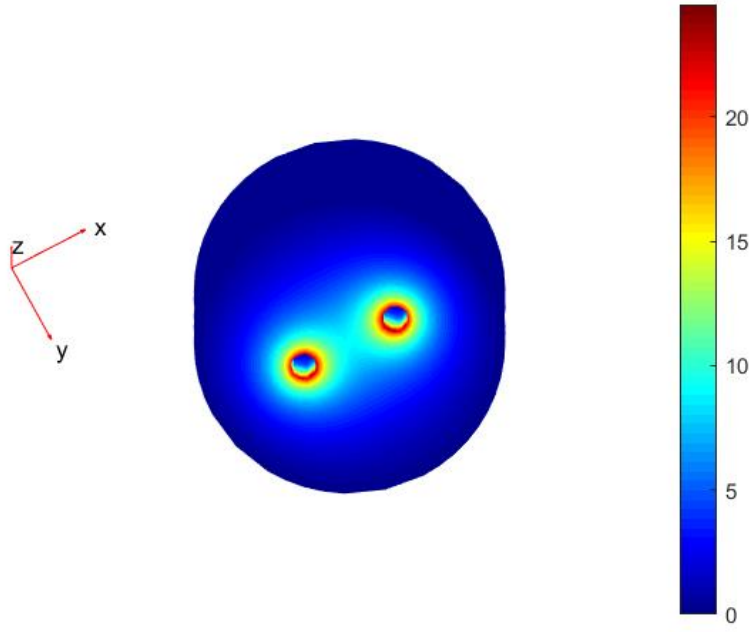

Figure 4.2: 3D homogeneity plot

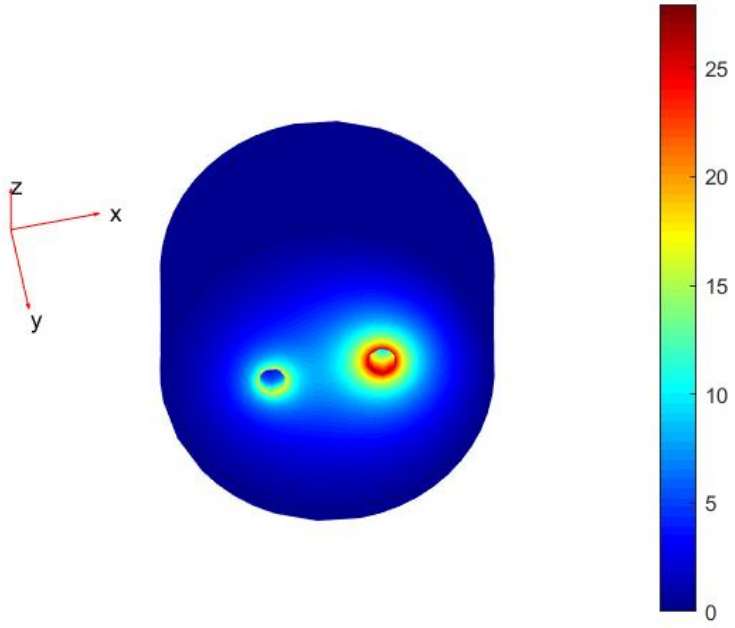

Figure 4.3: 3D heterogeneity plot

Additionally, plots are created to visualize the oxygen levels inside of each capillary and along the circumference of each capillary. For the same tests as above, these plots are shown below, in figures 4.4 and 4.5. As labeled in the legend, the blue line represents oxygen level in the capillary and the orange line represents the oxygen level around the circumference of the capillary.

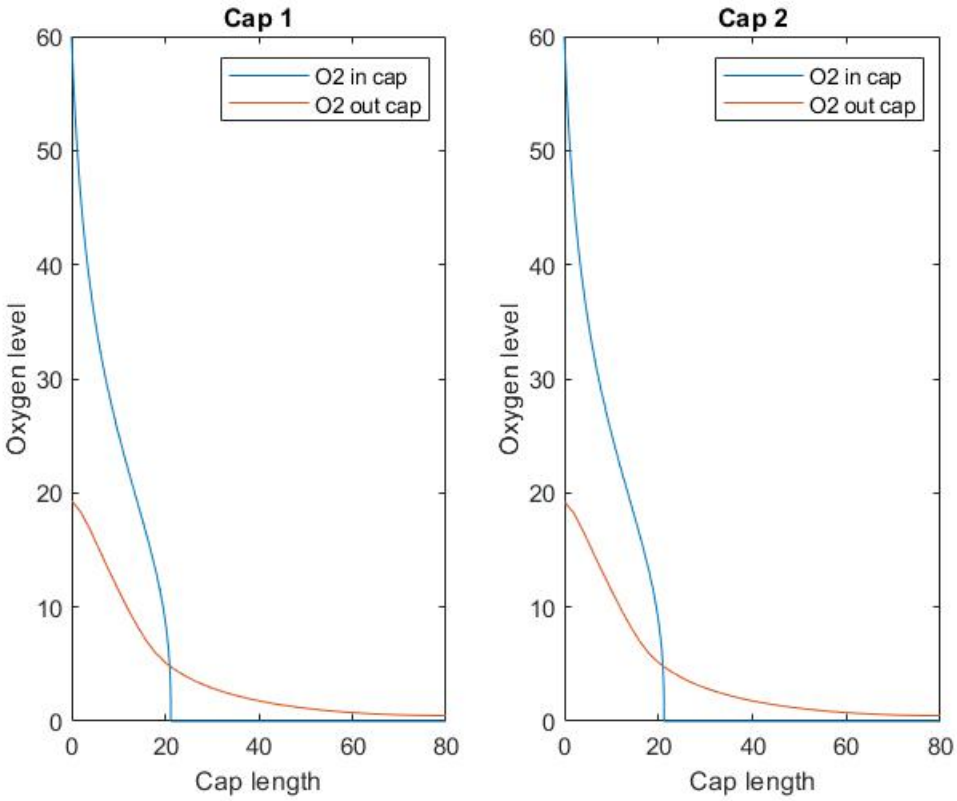

Figure 4.4: Homogeneous capillary O2 levels

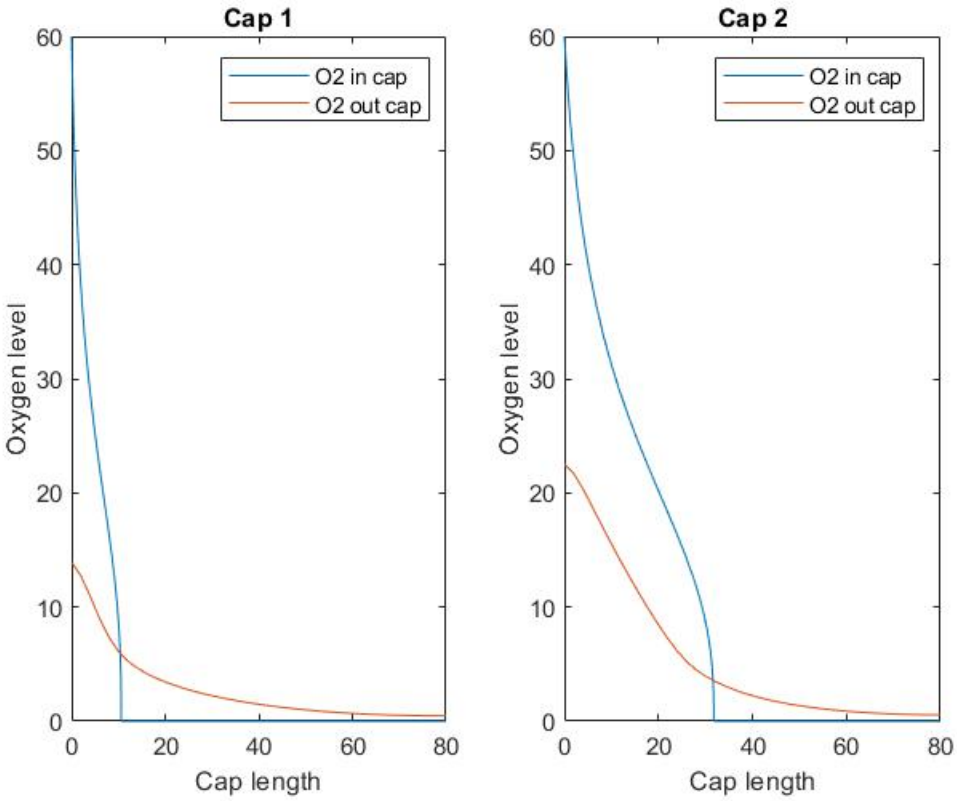

Figure 4.5: Heterogeneous capillary O2 levels

By examining the above plots, it is clear that heterogeneity does have some effect, however, it is hard to tell exactly what those effects are and how significant they are. Because of this, the average O_2 partial pressure in the entire tissue region is calculated. This helps to show the effects of heterogeneity numerically. In the homogeneous test run, the average O_2 partial pressure was 1.6322 and in the heterogeneous test, it was 1.6134 mmHg. Clearly the difference shows a decrease in overall oxygen when heterogeneity is introduced. Further, more in depth studies on heterogeneity based on a four capillary

network follow. One final plot is generated for each run. This plot, depicted in figure 4.6, shows areas where oxygen is above a certain level in red and areas in the tissue where oxygen is below that level in blue. In this case, the level used was 2.5 mmHg, but is a parameter that can be changed.

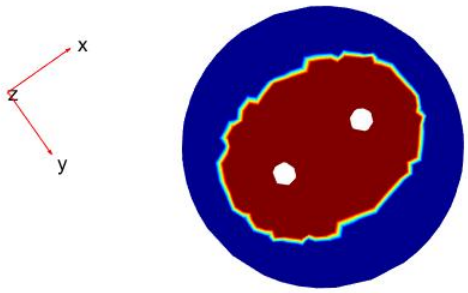

Figure 4.6: Tissue where O2 is above 2.5 mmHg (from above)

More in depth tests were run with a network of four capillaries. A plot of a homogeneous solution to this network is shown below in figure 4.7.

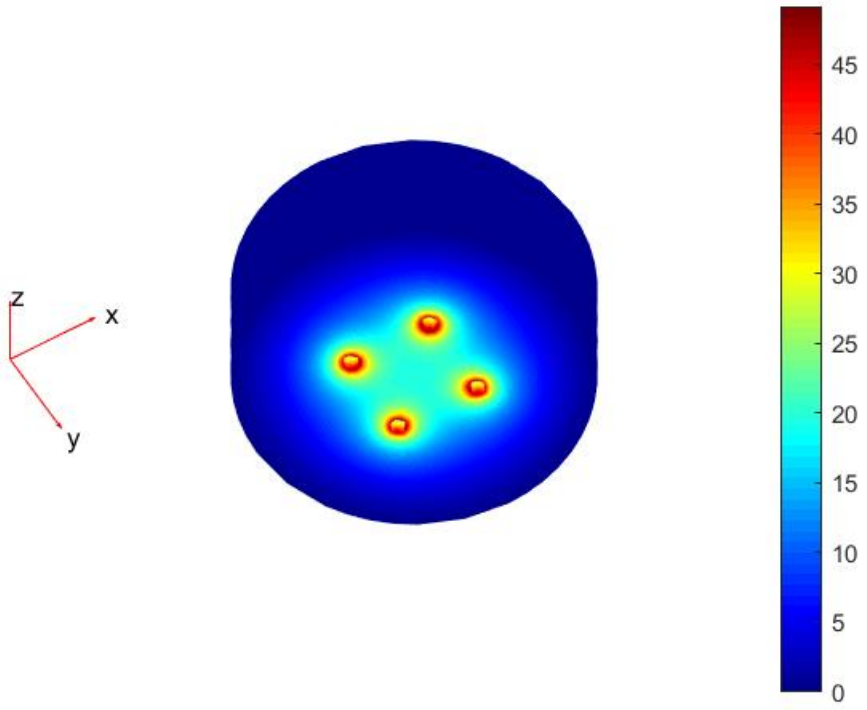

Figure 4.7: Homogeneous capillary network

In order to view the effects of heterogeneity, two tests were run with this network. First, the velocities of the flows coming into the network were changed. Initially, all velocities were 800. To introduce heterogeneity, three capillaries had their velocities reduced by 100 while the fourth capillary had its velocity increased by 300. This is done

until the velocities of the other three capillaries is 100 (making the fourth capillary 2900). The results of these tests are plotted below in figure 4.8.

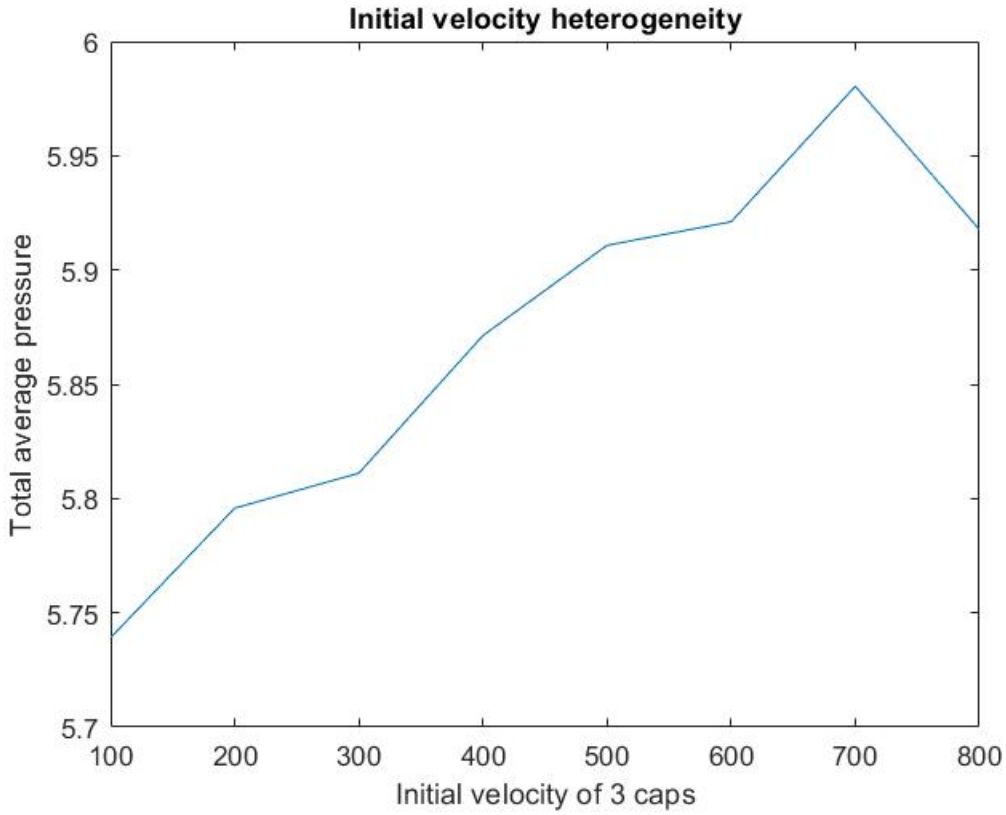

Figure 4.8: Heterogeneity in velocity (FEM)

In this test, we note that the average O_2 partial pressure does not actually vary too much but the model does exhibit some decrease in pressure as the heterogeneity increases. Interestingly, there is a small increase in the average oxygen level when there is a small amount of heterogeneity in this case. The second test performed was based on

the initial partial pressures supplied to each capillary. Initially, all capillaries have a value of 60 mmHg. As with the velocities, three capillaries are reduced (by 10 each time for this test) and one is increased (by 30 each time). A plot is shown below.

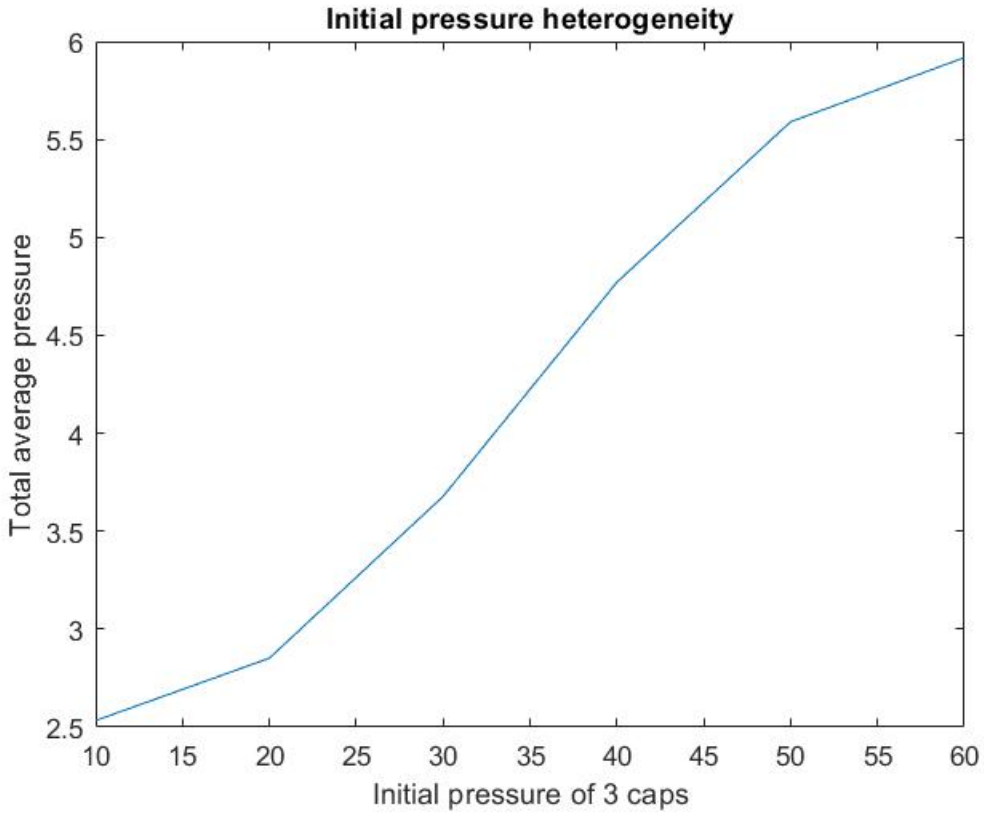

Figure 4.9: Heterogeneity in initial pressure (FEM)

Note that we observe a much greater drop in average pressure due to heterogeneity in the initial pressures. Additionally, we do not observe the slight increase in average pressure when a small amount of heterogeneity is present. Since there are

limitations to this implementation, as mentioned previously, these results are approximations to what we should expect from the Green's function implementation.

4.2 Green's Function Results

The more accurate Green's function implementation was also used to collect results in order to analyze oxygen in tissue using averages more. The code gives users the ability to view contour plots. These plots are not extremely detailed and were generally not used in results calculations, but were helpful when designing networks as the overlaid network is visible in each plot. A sample contour plot generated from a pre-made network is shown below. This is a more complicated network than those used for tests in this thesis, but shows what the Green's function implementation is capable of handling.

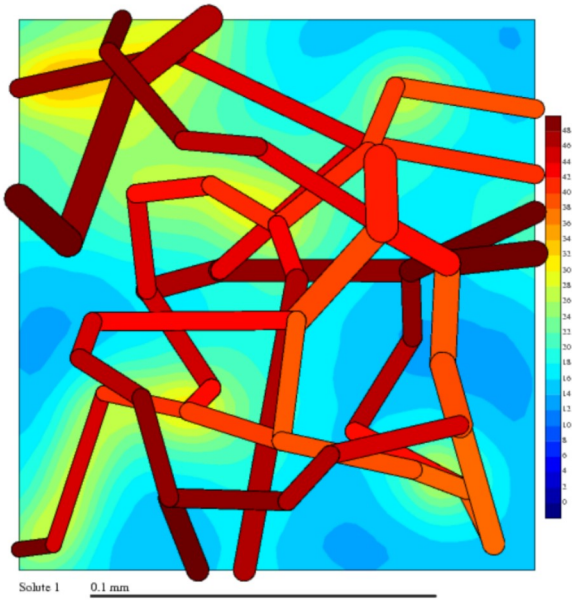

Figure 4.10: Sample capillary network for Green's function method

The Green's function implementation gives output showing the total average O_2 partial pressure in the tissue domain used. It also outputs oxygen levels throughout each capillary as well as oxygen levels in the tissue. This is useful, but for tests used in this thesis, with tests of four capillaries, a slightly different average was used. As the capillary networks used contained four parallel capillaries in a square shape, the average O_2 partial pressure in the tissue in between all four capillaries is calculated. Additionally, the pressure in between all four capillaries and in the lower third of the tissue domain is found. These averages are used when observing the effects of heterogeneity.

First, similar tests to the FEM tests were run. For this network, three variables were changed. First, the input oxygen into each capillary was changed in a similar fashion to the FEM four capillary test. Additionally, tests are done by varying the delta constant, effectively varying the metabolism. The result of the first test is pictured below in figure 4.11. As with before, the initial pressure is 60 mmHg for all capillaries originally. Three capillaries are then decreased by 10 each time while the other is increased by 30. The effects of changing velocity or flow rate is described and tested in a network used later.

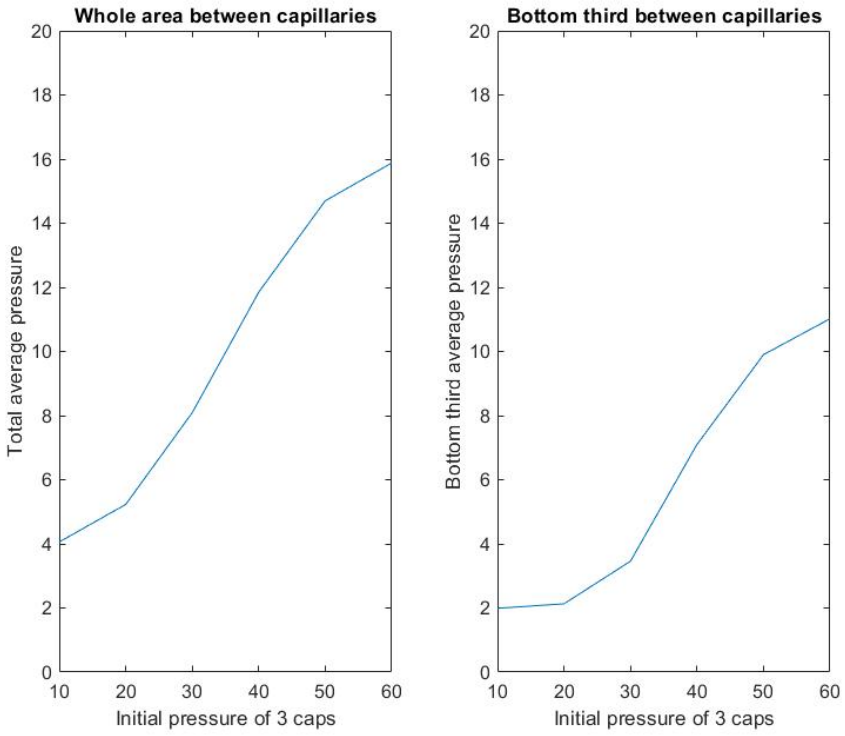

Figure 4.11: Heterogeneity in initial pressure (Green's)

For Green's function tests, two plots will be supplied. Here it is easy to see that the pressure in the tissue decreases in the lower third. Additionally, the curves for the effect of introducing heterogeneity in the initial pressure looks very similar to that of the FEM plots (figure 4.9). A difference here is that the average pressures given by this test are higher in general. This is likely due to the fact that the Green's model is more accurate.

The other test that was run for this network was to vary the metabolism of the surrounding tissue. Specifically, the delta value from the PDE is changed and is varied from 0 to 0.003. A plot is shown below with the initial partial pressure of 50 mmHg.

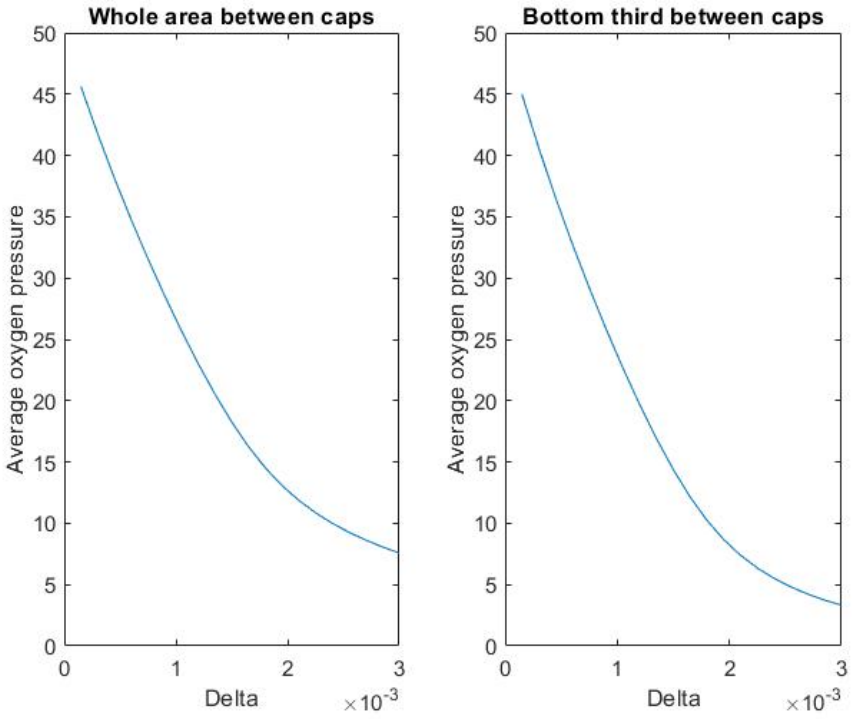

Figure 4.12: Changes in metabolism (Green's)

This results agrees with previous results that the bottom third of the tissue domain has a lower average than the whole domain. Additionally, it shows that a higher value of metabolism leads to a decreased average pressure. A more complicated network is used to

analyze the effects of changing both delta and initial pressure in the homogeneous and heterogeneous case.

Finally, a slightly different network was set up where one main capillary brings in all of the blood flow and then separates into four capillaries which can be used to test heterogeneity. Figure 4.1 is an approximate visualization of this network. A number of tests were done on this configuration. Namely, metabolism was varied and initial pressure was varied. In order to introduce heterogeneity, flow rates were changed. As velocity cannot be changed directly in this implementation, the following equation is used to ensure that the proper velocity is used.

$$q = \pi \cdot r^2 \cdot v \cdot 6 \times 10^{-6}$$

This equation relates the flow, q, from Green's function (in nl/min) with the radius in micrometers and velocity in micrometers/sec. It also means that in order to maintain 500 micrometers/sec, q and r must be changed using this equation. Thus, heterogeneity is introduced by changing both the flow rate and the radius. A heat map of the whole domain in between capillaries is shown below in figure 4.13.

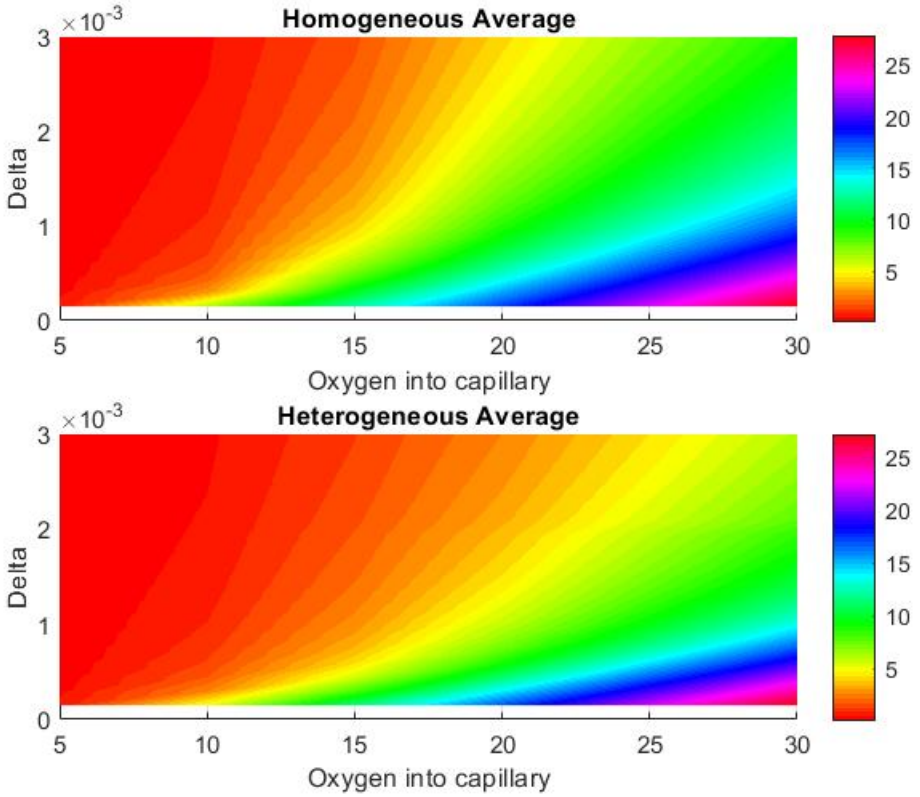

Figure 4.13: Heat map of whole domain

Although the homogeneous and heterogeneous averages are similar, it is visible that the heterogeneous average decreases faster, particularly by observing the yellow band in the plots above. This is as predicted. Also, as expected, a lower delta leads to a higher partial pressure in the tissue since the metabolism is not using as much oxygen in the tissue. Once the oxygen level decreases below about two, it becomes difficult for cells

to survive. These generally correspond to the red areas in these heat maps. A similar heat map of the lower third can be found below.

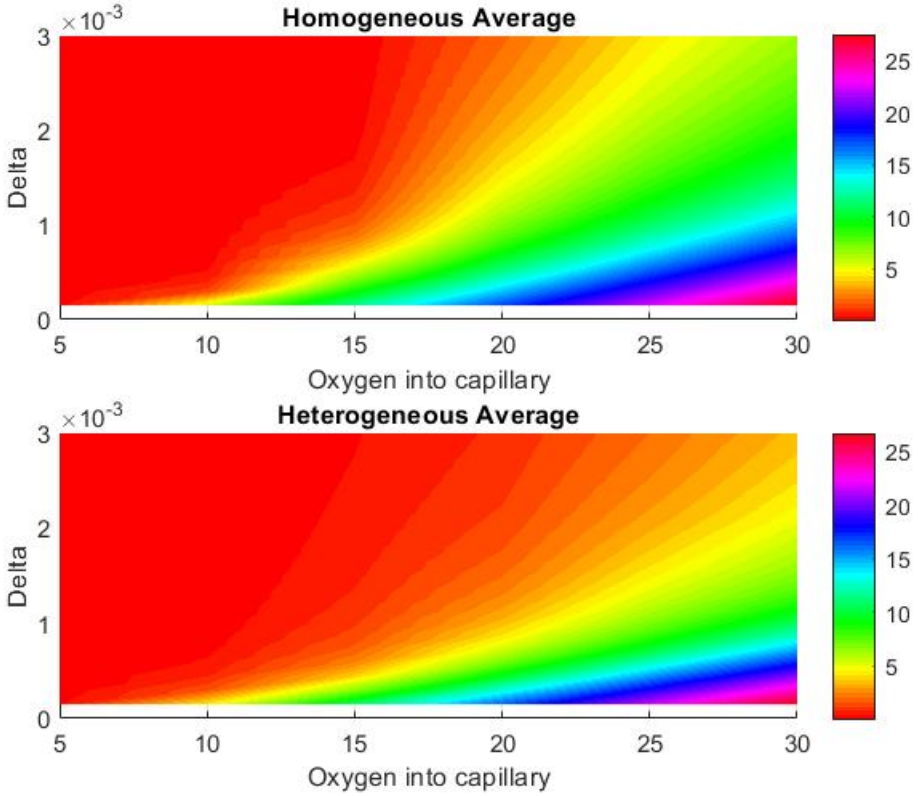

Figure 4.14: Heat map of lower third of tissue

Note that these heat maps exhibit the same traits as those of the whole tissue area, particularly in the yellow band. Additionally, the lower third heat maps tend to have a lower average oxygen level, as expected. Tables of the values shown in these heat maps are also shown below for specificity.

	5	10	15	20	25	30
0.003	0.04048	0.41367	1.6648	4.1912	6.7642	9.1989
0.00285	0.042961	0.43166	1.7366	4.3577	6.9965	9.5018
0.0027	0.045697	0.45136	1.8156	4.5375	7.2493	9.829
0.00255	0.048729	0.47304	1.9027	4.7338	7.5232	10.182
0.0024	0.052108	0.49695	1.9998	4.9483	7.8201	10.563
0.00225	0.055933	0.52363	2.1086	5.1838	8.1456	10.979
0.0021	0.06026	0.55353	2.2316	5.4432	8.5028	11.435
0.00195	0.065187	0.58729	2.3706	5.7322	8.8963	11.937
0.0018	0.07083	0.62578	2.5297	6.0556	9.3362	12.502
0.00165	0.077372	0.67041	2.7155	6.4199	9.8303	13.143
0.0015	0.085051	0.7228	2.9358	6.837	10.398	13.883
0.00135	0.094201	0.78519	3.2006	7.3199	11.06	14.757
0.0012	0.10531	0.86108	3.5281	7.8936	11.85	15.798
0.00105	0.11913	0.95597	3.9512	8.591	12.813	17.039
0.0009	0.13687	1.0788	4.5245	9.4689	13.994	18.484
0.00075	0.16065	1.247	5.3675	10.594	15.413	20.12
0.0006	0.19467	1.4976	6.5091	12.022	17.058	21.911
0.00045	0.24855	1.9278	8.1	13.75	18.885	23.823
0.0003	0.35137	2.9226	10.172	15.714	20.85	25.83
0.00015	0.65555	6.2828	12.497	17.777	22.863	27.86

Table 4.1: Results of whole domain, homogeneous test

	5	10	15	20	25	30
0.003	0.039383	0.3848	1.2898	2.5833	4.2783	6.079
0.00285	0.041735	0.40065	1.3347	2.6745	4.4298	6.2897
0.0027	0.044328	0.41796	1.384	2.7744	4.5955	6.5194
0.00255	0.047198	0.43697	1.438	2.885	4.7772	6.7711
0.0024	0.050394	0.45804	1.4976	3.0073	4.9791	7.0484
0.00225	0.054012	0.48151	1.5637	3.1438	5.2044	7.3556
0.0021	0.058098	0.50786	1.6374	3.2976	5.4574	7.6989
0.00195	0.062763	0.5398	1.7588	3.5819	5.9798	8.4594
0.0018	0.068075	0.57419	1.8559	3.7889	6.32	8.9216
0.00165	0.074242	0.61386	1.9682	4.0307	6.7104	9.457
0.0015	0.081487	0.66039	2.0999	4.318	7.1643	10.089
0.00135	0.090125	0.71588	2.2569	4.6611	7.706	10.855
0.0012	0.10061	0.78328	2.448	5.0799	8.3711	11.821
0.00105	0.11367	0.8677	2.6897	5.6141	9.2213	13.081
0.0009	0.13046	0.97733	3.0085	6.3294	10.366	14.743
0.00075	0.15304	1.127	3.4566	7.3519	11.957	16.801
0.0006	0.18547	1.3491	4.1581	8.8666	14.087	19.156
0.00045	0.23716	1.7242	5.4463	11.096	16.611	21.711
0.0003	0.33658	2.5236	7.9282	13.93	19.334	24.4
0.00015	0.63339	5.0695	11.387	16.913	22.1	27.126

Table 4.2: Results of whole domain, heterogeneous test

	5	10	15	20	25	30
0.003	0.00043133	0.0023221	0.05403	2.0326	4.4237	6.4973
0.00285	0.00048922	0.0027577	0.067205	2.1822	4.6375	6.7891
0.0027	0.00056062	0.0033066	0.083729	2.3463	4.8737	7.1088
0.00255	0.00064954	0.0040079	0.10453	2.5271	5.1345	7.4593
0.0024	0.00076161	0.0049123	0.13075	2.7279	5.4232	7.8442
0.00225	0.00090461	0.0060919	0.16399	2.9524	5.7451	8.2706
0.0021	0.0010906	0.0076531	0.20646	3.2051	6.1051	8.7455
0.00195	0.0013367	0.0097481	0.2671	3.4919	6.5098	9.2778
0.0018	0.0016676	0.012609	0.34935	3.82	6.9702	9.884
0.00165	0.002122	0.016648	0.45911	4.1989	7.4978	10.583
0.0015	0.0027634	0.022443	0.60759	4.6421	8.1142	11.403
0.00135	0.0036928	0.030945	0.81274	5.168	8.8468	12.384
0.0012	0.0050836	0.043808	1.1045	5.806	9.7353	13.566
0.00105	0.0072459	0.064055	1.5324	6.5991	10.833	14.986
0.0009	0.010773	0.097628	2.1887	7.6146	12.194	16.652
0.00075	0.01689	0.15817	3.2646	8.9339	13.841	18.543
0.0006	0.028408	0.27681	4.7086	10.621	15.756	20.618
0.00045	0.052883	0.5524	6.6987	12.664	17.886	22.836
0.0003	0.1165	1.427	9.2401	14.978	20.174	25.163
0.00015	0.37181	5.5007	12.036	17.402	22.518	27.519

Table 4.3: Results of bottom third, homogeneous test

	5	10	15	20	25	30
0.003	0.00052316	0.0083736	0.38833	0.92015	2.0884	3.4936
0.00285	0.00059825	0.0098302	0.40738	0.96862	2.2034	3.663
0.0027	0.00069027	0.011664	0.42832	1.024	2.3326	3.8508
0.00255	0.00080427	0.013981	0.45152	1.0878	2.4787	4.06
0.0024	0.00094702	0.016836	0.47746	1.1612	2.6458	4.2947
0.00225	0.0011281	0.020378	0.5069	1.2468	2.8389	4.5598
0.0021	0.0013615	0.024811	0.54063	1.3481	3.0643	4.8618
0.00195	0.0016683	0.030477	0.58146	1.4797	3.3569	5.275
0.0018	0.0020708	0.037987	0.62806	1.6317	3.6667	5.7018
0.00165	0.0026169	0.048029	0.68425	1.8219	4.0306	6.212
0.0015	0.00338	0.061385	0.75351	2.0648	4.465	6.8372
0.00135	0.0044606	0.079491	0.84172	2.3697	4.9969	7.6265
0.0012	0.0060622	0.10558	0.95825	2.7626	5.6691	8.6733
0.00105	0.0084983	0.14377	1.1185	3.2935	6.5588	10.107
0.0009	0.012397	0.20153	1.3528	4.0507	7.8065	12.071
0.00075	0.019011	0.29626	1.7265	5.1869	9.623	14.527
0.0006	0.031213	0.46086	2.3927	6.9075	12.148	17.318
0.00045	0.056599	0.78617	3.793	9.5005	15.154	20.323
0.0003	0.12125	1.5714	6.689	12.859	18.37	23.47
0.00015	0.37533	4.3275	10.764	16.386	21.616	26.653

Table 4.4: Results of bottom third, heterogeneous test

Chapter 5. Conclusions

Oxygen transport throughout the body is a critical process in order to maintain life. The model proposed by this thesis defines the process of delivering oxygen from the bloodstream into surrounding tissue. This model is then implemented in two ways. While the latter is more accurate, both are valuable and give interesting results.

The results found based on tests done indicate a few key traits of oxygen transport into tissue based on metabolism, diffusion, and heterogeneity. From the results of the Green's implementation, it is clear to see that an increase in the metabolism results in a decrease in oxygen levels in the tissue. As explained previously, this is due to more oxygen being absorbed by the metabolism, leaving less in the tissue. Both implementations exhibit the effect of adding heterogeneity. The FEM implementation shows that heterogeneity in the incoming velocity and the initial O_2 partial pressure causes a general decrease in the total oxygen in the surrounding tissue. By analyzing the tables generated from the Green's function method, this result is confirmed. Other interesting results arise from these tables however. It is important to note that for smaller initial pressures, heterogeneous networks actually give a higher average O_2 partial pressure in the lower third of the tissue domain. This is slightly unexpected as

heterogeneity generally lowers oxygen diffusion. The reason for this is hypothesized to be the fact that a heterogeneous network with the same amount of oxygen entering the network will have at least one capillary that contains more oxygen at the lower third than then homogeneous network. Because of this, it is likely that this capillary will diffuse more oxygen, leading to an overall increase in oxygen levels in the lower third.

The results from this thesis can be used in many applications. Particularly, by noting that heterogeneity generally leads to a decrease in average partial pressure of oxygen, these results can be used to predict the effect of heterogeneity on specific cells in between capillaries. If this network is located within the brain, neurons will likely reside between the capillaries. Therefore, it is important to use the results from these tests in order to determine whether a given neuron would have enough oxygen supplied to produce adenosine triphosphate (ATP) and allow it to fire (ATP being essentially an energy source for cells) [9]. This is important future work that could help to determine the effects of certain blood disorders on neurons and the brain in general.

Bibliography

[1] SketchUp https://www.sketchup.com/ Accessed 2019-04-12

[2] MATLAB OSU Solvers https://www.mathworks.com/help/matlab/math/choose-an-ode-solver.html Accessed 2019-03-23

[3] Secomb GitHub https://github.com/secomb Accessed 2019-07-01

[4] Mayo Clinic Hematocrit Tests https://www.mayoclinic.org/tests-procedures/hematocrit/about/pac-20384728 Accessed 2019-07-05

[5] Mayo Clinic Hemoglobin Tests https://www.mayoclinic.org/tests-procedures/hemoglobin-test/about/pac-20385075

[6] Secomb, T.W., Hsu, R., Park, E.Y.H. et al. *Annals of Biomedical Engineering* (2004) 32: 1519. https://doi.org/10.1114/B:ABME.0000049036.08817.44

[7] Tuma, R. F., Durán, W. N., & Ley, K. *Handbook of physiology a critical, comprehensive presentation of physiological knowledge and concepts*. (2008)

[8] Sefaeian, N, Sellier, M, David, T. *Journal of Theoretical Biology* (2011) 271: 145. https://doi.org/10.1016/j.jtbi.2010.11.038

[9] Attwell, D, Laughlin, S.B. *Journal of Cerebral Blood Flow & Metabolism* (2001) https://doi.org/10.1097/00004647-200110000-00001

[10] Solving Problems Using PDEModel Objects

https://www.mathworks.com/help/pde/ug/solve-problems-using-pdemodel-objects.html

Accessed 2019-04-07

Appendix A. Sample Data Files for Green's Function

16 Solute 1	
D (um)	K (mmHg.um.s/um3O2)
4	2.45
5	2.00
6	1.74
8	1.44
10	1.27
12	1.16
14	1.08
16	1.02
18	0.974
20	0.935
22	0.895
24	0.853
28	0.815
32	0.779
40	0.750
50	0.730

Figure A.1: IntravascRes.dat

```
Brain network
  100.  100.  100.      box dimensions in microns
  10    10    10              number of tissue points in x,y,z directions
  200.                  outer bound distance (100. in test case)
  10.                           max. segment length.  (5. in test case)
  4                             nodsegm, max. allowed number of segments per node
  11                            total number of segments
 name  type   from   to   diam. rel. flow  hem.
    1    5     1      2    8.0    1.51    0.4
       1    5     2      3    8.0    1.51    0.4
       1    5     2      4    8.0    1.51    0.4
       1    5     3      5    8.0    1.51    0.4
       1    5     3      6    8.0    1.51    0.4
       1    5     4      7    8.0    1.51        0.4
       1    5     4      8    8.0    1.51    0.4
       1    5     5      9    8.0    1.51    0.4
       1    5     6      10   8.0    1.51    0.4
       1    5     7      11   8.0    1.51    0.4
       1    5     8      12   8.0    1.51    0.4
 12                    total number of nodes
name  x     y     z
1       0          50    50
2       20     50         50
3       20         50    30
4       20     50         70
5       20         30    30
6       20     70         30
7       20         30    70
8       20     70         70
9       100        30    30
10      100    70         30
11      100        30    70
12      100    70         70
 1      total number of boundary nodes
node bctyp press/flow   HD    PO2    solute 3
 1      2     2.0      0.4      50.             1.
```

Figure A.2: Network.dat

```
        float pcr,m0,po2;

//This version requires that nsp = 1
        if(nsp != 1) printf("*** error: nsp = %i\n",nsp);
//initialize
        for(isp=1; isp<=nsp; isp++){
                mtiss[isp] = 0.;
                mptiss[isp] = 0.;
        }
//isp = 1: oxygen
        m0 = tissparam[1][1];
        pcr = tissparam[2][1];
        po2 = FMAX(p[1],0.);            //get rid of negative values
        mtiss[1] = -m0*po2/(po2 + pcr);
        if(po2 > 0.) mptiss[1] = -m0*pcr/SQR(po2 + pcr);
```

Figure A.3: Tissrate.cpp.dat

```
VaryParams.dat - version for brain simulations
Allowable parameters to vary: q0fac, diff[isp], tissparam[i][isp] - give actual values
solutefac[isp], intravascfac[isp] - give multiplicative factors
3       number of parameters to be varied: list below
q0fac - param1
solutefac[1] - param2
tissparam[1][1] - param3
20 number of runs
run  param1 param2 param3
1     1.0   1.0   0.0000001
2     1.0   1.0   0.000001
3     1.0   1.0   0.00001
4     1.0   1.0   0.000025
5     1.0   1.0   0.00005
6     1.0   1.0   0.0001
7     1.0   1.0   0.00015
8     1.0   1.0   0.0002
9     1.0   1.0   0.00025
10    1.0   1.0   0.0005
11    1.0   1.0   0.00075
12    1.0   1.0   0.001
13    1.0   1.0   0.00125
14    1.0   1.0   0.0015
15    1.0   1.0   0.00175
16    1.0   1.0   0.002
17    1.0   1.0   0.00225
18    1.0   1.0   0.0025
19    1.0   1.0   0.00275
20    1.0   1.0   0.003
```

Figure A.4: VaryParams.dat

```
Solute parameters for brain
1 2 3           rungreens (0 or *1), g0method (*1 or 2 for TPZ modelling), linmethod (1, 2 or *3)
2  2 100        nmaxvessel,nmaxtissue,nmax - iteration limits
1.00E-04        errfac:  overall convergence tolerance (suggest 1e-5 to 1e-4)
50                      lowflowcrit, criterion for low flow segment (suggest 50-200)
38                      P50 in mmHg
3                       n in Hill equation
0.04        Oxygen binding capacity of red cells in cm^3/cm^3/mmHg
3.10E-05        Effective solubility of oxygen in blood in cm^3/cm^3/mmHg
1.                      Factor to vary flows in network (normally use 1.0) - modified in GreensV4
1   3           number of reacting species, number of tissue parameters per species
Solute 1 - oxygen
  1     1   1           permsolute, 0 or 1, diffsolute, 0 or 1; oxygen, 0 or 1, 1 if solute is oxygen
  100.          pref[1] - typical maximum value, mmHg
  6.0e-10       diff[1] - tissue D*alpha in cm^3 O2/cm/s/mmHg
  0.002         tissparam[1][1] - Max. oxygen cons. rate in cm3/cm3/s
  3                     tissparam[2][1] - Michaelis constant of consumption in mmHg - changed April 2016 per Golub and Pittman
  0.            tissparam[3][1] - Extra, not used
  0.2           g0[1] - initial estimate of g0
  1.0           g0fac[1] - use 1, decrease if g0 values oscillate
```

Figure A.5: SoluteParams.dat

```
0. 0. 70.      5            origin of plane slice for PO2 output, subdivision of segments for contours
150. 0. 70. 50              corner of plane slice, number of points
0. 160. 70. 50              corner of plane slice, number of points
0. 2. 25                    Solute 1: minimum contour level, increment, number of levels
```

Figure A.6: ContourParams.dat