

Bachelor Thesis



The Interface Group

1D Computations of Flow and Oxygen Transport in Micro-Vascular Networks

Berk Nergiz

Advisers

Dr. Kartik Jain

Prof. Dr. Vartan Kurtcuoglu

Interface Group, Institute of Physiology (UZH)

Supervision

Prof. Dr. Bradley J. Nelson

Institute of Robotics and Intelligent Systems

Swiss Federal Institute of Technology Zurich (ETH)

July 2018

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

D MAVT

Contents

Abstract	1	
1	Introduction	1
1.1	Motivation	1
1.2	Structure of the Thesis	2
2	Blood Flow and Oxygen Delivery	3
2.1	Blood Flow	3
2.2	Oxygen Transport and Delivery	4
2.3	Current Trends in Modeling	4
3	Simulation Methods	6
3.1	Green's Function Method	6
3.2	<i>DuMu^x</i>	11
4	Simulation Results and Analysis	16
4.1	Results of the Green's Function Method	16
4.2	Discussion of the Green's Method and the Produced Results	24
4.3	Results of <i>DuMu^x</i>	24
4.4	Comparison of the Results	25
5	Comparison of the Different Methods Behind the Simulations	26
5.1	Comparison of the Numerical Methods	27
5.2	Stability Questions and Computational Cost	27
6	Conclusion	28
7	Acknowledgements	29
References	30	
A	Appendix	32
A.1	Additional Material for the Green's Function Method	32
A.2	Additional Material for DuMuX	35

Abstract

Oxygen is carried by red blood cells in the blood. It is supplied to organs, muscles and generally to the tissue through complex networks. This microvascular network consists of micro-sized capillaries, that can be extracted from ex vivo mice organs. This work is part of the NCCR Kidney project, and aimed to develop and analyze computational models to understand the gradients of Oxygen in the tissue, and have a better understanding of what methods might be the most accurate ones in order to get physiologically correct results.

1 Introduction

1.1 Motivation

This thesis aims to understand and specify how blood flow and oxygen transport has been simulated until today. The goal is to see how far existing methods, especially the Green's Function Method applied to this field by Secomb et al. [15] can produce accurate results for new and especially complex networks. A new model using an open source software called *DuMu^x* [1] was implemented to offer a different approach for oxygen transport and diffusion simulations.

Oxygen from blood is carried on to the tissue of living organs through a complex network of vessels, in particular by capillaries that provide oxygen to the tissue, which is attached to the red blood cells. The complex biochemistry involved in the release of oxygen from hemoglobin and its transport to the surrounding tissue is a complex process that cannot be fully understood by experimental methods. With the advent of mathematical models and advanced computing architectures, the process of Oxygen delivery to the tissue can be simulated to understand various physiological and pathophysiological processes.

This thesis aims to simulate transport of oxygen through capillary networks of various organs using mathematical models. Specifically, the goal is to evaluate the efficacy of the existing numerical methods that simulate flow and transport of O_2 through capillaries. The Green's function method developed by Secomb is evaluated in detail and applied to various microvasculature networks. Further, coupling of 1D network with a 3D surrounding tissue is evaluated in the open source package called *DuMu^x*. A simplified approach for the simulation of O_2 transport was implemented in *DuMu^x* and compared with the results of Green's function method.

1.2 Structure of the Thesis

First, a short insight to the physiological background of blood flow and oxygen transport in the body will be given. The general process of oxygen delivery to the tissue and oxygen consumption by the tissue will be explained. The main subject will be the transport of oxygen in the body, starting in the blood flow and ending in the diffusion into the tissue through a general diffusion process.

The results of oxygen transport simulation methods, especially the Green's Function Method, will later be discussed and evaluated for different networks. In addition to this, a new oxygen delivery model was implemented using an open source software called *DuMu^x*. The focus of this thesis is to understand when and why the Green's Function Method produces physiologically accurate results, how it can perform on new networks and compare these results to alternatives such as the implemented method on *DuMu^x*.

2 Blood Flow and Oxygen Delivery

In this section the physiological aspects of O₂ transport through vessels into the tissue will be discussed. This part is mainly dedicated to the understanding of physiological processes that are involved in the transport of oxygen through capillaries into the tissue, and its consumption therein.

In this section the physiological aspects will be discussed, and the main goal of computations and simulations in this field will be explained. This part is mainly about the physiological points in blood flow and oxygen transport, as well as oxygen consumption in tissue and further outlook in this field. (i.e. EPO and tumor oxygen delivery etc.)

Blood is pumped through the heart into the whole body. This results in pressure gradients, accelerating the fluid through the arteries.

In this thesis, the flow and transport through micro-vessels in the capillary range is the main concern, where generally the time-dependency of the velocity of blood will be neglected. We use a steady-state approximation for the flow rate, and thus assume a constant velocity. This means that there is no variation of velocity depending on time for any given point in the network.

2.1 Blood Flow

Explain this in a bit more detail. Also mention that blood is usually modeled as a homogeneous fluid using Navier Stokes equations despite that it is a mixture of plasma and cells etc. Also mention that why it is extremely inappropriate for us to model it like that.

Blood is basically composed of two phases: liquid and cells. The liquid phase is called plasma, and the cells that are our main concern are the red blood cells. This will later play an important role when considering the oxygen transport in blood. For computational models, blood is generally modeled as an incompressible Newtonian fluid [13]. This basically means that the viscous stress is linearly proportional to the strain rate. This assumption generally holds, even though the particles in the blood, which are the cells, start playing a more important role for the smallest vessels. This leads to the Navier-Stokes equations 2.1, 2.2, which basically describe the flow dynamics of an incompressible Newtonian fluid:

$$\rho \frac{\delta \mathbf{v}}{\delta t} + \rho \mathbf{v} \nabla \mathbf{v} + \nabla p - \eta \nabla (\nabla \mathbf{v} + \nabla \mathbf{v}^T) = \mathbf{f} \quad (2.1)$$

$$\nabla \cdot \mathbf{v} = 0 \quad (2.2)$$

In the Navier-Stokes equations, \mathbf{v} is the velocity vector, p is the pressure, and ρ and η are the blood density and viscosity respectively.

The external forces acting on the fluid are represented by the vector $\mathbf{f} = (f_x f_y f_z)^T$.

Finish this chapter using the Rahman paper...

2.2 Oxygen Transport and Delivery

How does this oxygen related stuff physically and physiologically work ?

Again put some papers and references in and explain some physical diffusion equations, values and parameters.

Why do we care about this etc. ?

Diffusion, Parameters, etc. etc.

Use Vasomotion paper for oxygen delivery [5]

Papers to cite: [10]

Oxygen Transport in the Blood

-Plasma and Red Blood Cells

-Hemoglobin

Blood is basically composed of two phases: liquid and cells, which is the plasma and the red blood cells. The red blood cells build nearly half of the total blood volume, and are the main carriers of oxygen in the blood [12].

Oxygen Diffusion Into the Tissue

As previously described, diffusion will play an important role in oxygen transport in the body, as oxygen is reaching the tissue by diffusing out of the vessels. This physical phenomenon is described by Fick's Law of Diffusion:

$$J = -D \frac{dc}{dx} \quad (2.3)$$

Where J is the diffusive flux, D is the previously mentioned diffusivity coefficient, and $\frac{dc}{dx}$ is the derivative of the concentration with respect to the distance x, which is the concentration gradient [12], [11].

The diffusion process can be helped and enhanced by a substance (molecule?) called Myoglobin. This substance can be found in some tissues in the body, especially muscle tissue (?) and enhances the diffusivity of oxygen through the tissue. The enhancement effect can obviously depends on the Myoglobin concentration in the given tissue, and these effects can be taken into account by (?).

2.3 Current Trends in Modeling

In order to simulate flows, Computational Fluid Dynamics (CFD) is a tool that can be used to produce results which can not be obtained through in vitro or in vivo experiments [13]. For this reason and because the computational capabilities available are growing

everyday, modeling and simulating processes in the body is taking an increasingly important role in biological and medical sciences.

In our case, the focus will lie on oxygen transport simulations and their computational implementation.

As already mentioned in the section 2.1, blood flow in the body can be approximated by the Navier-Stokes equations 2.1, 2.2. The transport of oxygen in the blood can then be linked to this by the transport equations. These physical equations are quite straightforward to implement and solve numerically. The next task remains the biggest problem in actual research: Link this transport to the diffusion process out of the vessels and into the tissue. The biggest problems here seem to be in choosing the right/appropriate boundary conditions to get exact results. Also, many numerical methods fail or more specifically become unstable when the vessel network becomes complicated and heterogeneous. There are already good methods to produce results for the Krogh model, but to compute oxygen fields in very complex networks like in the kidney, these models do not provide the desired results.

Steady-State approximation for simulations: Need this assumption for Michaelis-Menten relation.

3 Simulation Methods

As noted before, this thesis focuses on the evaluation of existing methods that simulate flow and oxygen transport in micro-vasculature networks and applies them to several vasculature morphologies from various organs. In this section I will try to give an overall understanding to the theoretical background to the Green's Function Method and accordingly the code developed by Secomb et al. [15]. Later on I will present a new model that I implemented using *DuMu^x*. Both the methods and the produced results will be discussed in the following sections.

3.1 Green's Function Method

The Green's Function Method was first used for numerical computations of oxygen delivery in the tissue by Secomb et al. [15]. It can deliver accurate results with lower computational cost when compared to implicit methods, due to the fact that the number of unknowns is reduced. For simple networks like the examples specified in the next chapter, the method seems to be stable and produce accurate results. However, using new and more complex networks, the Green's Function Method seems to produce bad results and shows characteristics of instability (at least we think so for now). Obviously as the results get bad, the computational advantages of this method are not of any use (bad formulation?).

The mathematical background behind the Green's Function Method and the numerical method derived from it will be discussed in the upcoming section. The accuracy of produced results as well as the physiological meaning behind these results will be the main subject in this chapter.

3.1.1 The Computational Model

General explanation about Green's Method code and how it works, especially mathematics behind the Green's Function Method (many mathematical formulas) and why it makes sense to use this. Also especially why is this interesting in physiological applications and oxygen delivery/consumption simulations ?

Goals:

- Avoid no flux boundary condition
- Find a systematic method that can be applied to tissue domains of arbitrary shape
- Find a cheap solution in terms of computational cost (when compared to implicit methods)
- Green's function method is computationally less expensive than implicit methods, due to the fact that there is a lower number of unknowns

In this subsection, the used physical quantities, the assumptions and governing equations will briefly be explained. The equations cited here are, in my opinion, the most

important ones to describe the physical background for the treated problem. This part strongly relies on the description by Secomb et al. [15].

When looking at the code of the Green's Function Method, and especially at the inputs A.1.1, one can see that the tissue is modeled as a homogeneous medium. This results in constant oxygen diffusivity and solubility coefficients D and α respectively. These quantities usually have to be specified in the SoluteParams.dat-file. A more detailed explanation to the Inputs can be found in the Inputs section A.1.1.

The governing equations are listed here, with a short explanation for each of them:

As previously described, diffusion will play an important role in oxygen transport in the body, as oxygen is reaching the tissue by diffusing out of the vessels. This physical phenomenon is described by Fick's Law of Diffusion 2.3. As a reminder, I want to mention that Fick's first Law gives the diffusive flux, which we are interested in, as a function of the diffusivity and the concentration gradient in the tissue.

In addition to this, the principle of conservation of mass applies to oxygen, and we have:

$$\frac{\delta \rho}{\delta t} + \nabla(\rho v) = 0 \quad (3.1)$$

These two fundamental equations can be combined to finally get:

$$D\alpha\nabla^2 P = M(P) \quad (3.2)$$

3.2 describes the dependency between the oxygen partial pressure P and the consumption rate $M(P)$. Here I would like to mention the fact that partial pressure is physically nothing else than the concentration used in 2.3. For the consumption, a Michaelis-Menten relationship applied to the problem is used:

$$M(P) = \frac{M_0 P}{(P_0 + P)} \quad (3.3)$$

In 3.3, $M(P)$ is the consumption, M_0 the demand and P_0 the partial pressure at half-maximal consumption.

As previously mentioned, the second important physical phenomenon to describe our transport problem is the convective transport in the blood flow. To model this, we need to know the rate of convective oxygen that is transported through a single vessel segment.

$$f(P_b) = Q(H_D C_0 S(P_b) + \alpha_{eff} P_b) \quad (3.4)$$

In 3.4, the flow rate of blood is Q , and H_D , S , C_0 , P_b and α_{eff} are parameters such as discharge hematocrit, oxyhemoglobin saturation or partial pressure of oxygen in the blood. (Here specify everything yes or no ? Maybe use a symbols table at beginning to make things easier).

The previously mentioned oxyhemoglobin saturation can be computed by Hill's equation:

$$S(P_b) = \frac{P_b^n}{(P_b^n + P_{50}^n)} \quad (3.5)$$

In Hill's equation 3.5, P_b is the previously mentioned partial pressure of oxygen in blood, P_{50} is the partial pressure of oxygen at 50% saturation and n is a constant.

Another important equation that we need to describe the given problem is the relationship describing the rate of diffusive oxygen efflux per unit vessel length:

$$\frac{df(P_b)}{ds} = -q_v(s) \quad (3.6)$$

3.6 is obtained by using the conservation of oxygen along a vessel segment.

The continuity condition for the partial pressure of oxygen on the vessel-tissue interface gives a second equation for the diffusive oxygen flux:

$$q_v(s) = -D\alpha \int_0^{2\pi} \frac{\delta P}{\delta r} r_v d\Theta \quad (3.7)$$

3.13 is an integral form of the diffusive flux $q_v(s)$.

The relation between the partial pressure of oxygen in the tissue and the blood can be approximated with the following equation [6]:

$$P_v(s) = P_b(s) - K q_v(s) \quad (3.8)$$

3.8 is the Hellums relationship.

Here $P_v(s)$ represents the partial pressure of oxygen averaged around the circumference of the vessel, K is the intravascular resistance to radial oxygen transport, which is assumed to be constant but depends on the vessel diameter. The value for K for each vessel can be found in the input file IntravascRes.dat A.1.1 which has to be given to the code.

As discussed in the first part, myoglobin is a substance (molecule?) that supports the diffusion of oxygen in the tissue. The effects of myoglobin-facilitated diffusion can sometimes be neglected, when the myoglobin concentration is very low. For the case where the effects of myoglobin are relevant, one can simply replace the partial pressure P with the partial pressure P^* , given in the following equation 3.9.

$$P^* = P + \frac{D_{Mb} C_{Mb} V_m S_{Mb}(P)}{D\alpha} \quad (3.9)$$

In 3.9, D_{Mb} , C_{Mb} , V_m and S_{Mb} are the diffusion coefficient, the concentration, the molar volume and the oxygen saturation of Myoglobin respectively. D and α are the same constants mentioned previously, which were the diffusivity and the solubility coefficients of oxygen.

3.1.2 Mathematical Details of the Green's Function Method

Similar to the previous section, this section will strongly rely on [15]. In this subsection, the Green's Function Method, its equations and the application to the given problem/field will be explained.

In this section, I will focus on giving a good and understandable insight to the Green's Function Method and its specific application, instead of getting lost in details. More specific low-level informations about the code and its implementation will follow later.

The main idea of the Green's Function Method approach is to model blood vessels as discrete oxygen sources. The oxygen consumption is then modeled by a set of discrete oxygen sinks. As previously described, our goal is to simulate the oxygen transport to finally obtain the oxygen concentration for every node in our tissue and vessel network. In general terms, the oxygen field is computed by a superposition of the fields resulting from each of the sources and sinks.

The strengths of the sources are unknowns in this problem. The sinks are unknowns as well, due to the fact that the oxygen consumption depends on the local partial pressure of oxygen, as one can see in 3.3.

When defining the Green's Function G as the potential at a point \mathbf{x} and putting this potential equal to the local oxygen concentration or equivalently to the partial pressure of oxygen P , we get:

$$D\alpha \nabla^2 G = -\delta_3(\mathbf{x} - \mathbf{x}^*) \quad (3.10)$$

The equation 3.10 is basically the same as 3.2, where $G(\mathbf{x}; \mathbf{x}^*)$ is the partial pressure of oxygen at \mathbf{x} , resulting from a unit point source at \mathbf{x}^* .

The superposition of the fields gives the overall potential $P(\mathbf{x})$ in each point \mathbf{x} , which can be computed by integrating the distribution multiplied the local potential over all the sources:

$$P(\mathbf{x}) = \int_{Sources} G(\mathbf{x}; \mathbf{x}^*) q(\mathbf{x}^*) d\mathbf{x}^* \quad (3.11)$$

3.11 gives the general potential (here the partial pressure of oxygen) in a point \mathbf{x} , with $q(\mathbf{x})$ representing the distribution of source strengths.

The solution computed in an infinite domain is the singular function as defined in the following equation:

$$G = G_1 = \frac{1}{(4\pi D\alpha |(\mathbf{x} - \mathbf{x}^*)|)} \quad (3.12)$$

Coming back to the finite domain that we are looking at, boundary conditions on the domain boundaries have to be taken into account and thus the solution will be different depending on the boundary conditions that we choose. In this part I would like to mention the fact that choosing boundary conditions has a substantial consequence on the results,

and there is not yet one type of boundary condition that can produce overall exact results in terms of hypoxic tissue estimation (this should be mentioned somewhere again, as it is a big deal in this field of research). For the boundary conditions that were used in the research until now (put some references in), the computed hypoxic tissue areas were either way too small or way too big compared to what seems to be physiologically realistic. Again I would like to mention the fact that no good measurements can be made about the specific size and the distribution of hypoxic regions in tissue surrounding micro-sized capillaries.

The general approximation that is used uses a distribution of point sources along the centerline of each vessel segment. The field resulting from these sources can then be computed, but has a small error due to the mentioned approximation (give more detail about this error either here or later in the discussion part about Green's Method).

The diffusive flux at the blood-tissue boundary can generally be expressed by the following equation:

$$q_v(s) = \int_0^L F(s - s^*) q_0(s^*) ds^* \quad (3.13)$$

In 3.13, the distribution of radial flux across the cylindrical surface is obtained by the function $F(s)$, and $q_0(s)$ represents the distribution of source strength per unit length, s being the distance along the vessel.

The main case is a uniform distribution of sources around the circumference, where we get the following formula for $F(s)$:

$$F(s) = \frac{1}{2} \delta_1(s) + \frac{k(K(k) - E(k))}{4\pi r_0} \quad (3.14)$$

In 3.14, $K(k)$ and $E(k)$ are the following functions, also called elliptic integrals:

$$K(k) = \int_0^{\frac{\pi}{2}} (1 - k^2 \sin^2 \Theta)^{-\frac{1}{2}} d\Theta \quad (3.15)$$

$K(k)$ 3.15 and $E(k)$ 3.16 represent the complete elliptic integrals of the first and second kind respectively.

$$E(k) = \int_0^{\frac{\pi}{2}} (1 - k^2 \sin^2 \Theta)^{\frac{1}{2}} d\Theta \quad (3.16)$$

3.2 *DuMu^x*

As my intention was to implement a simple simulation for oxygen transport, I had to decide which solver I want to use. At first sight, the decision had to be made between OpenFOAM, FEniCs and *DuMu^x*.

When starting this project, we knew that the biggest limitations and challenges to face will be the given time, which was quite short to get into such a complex subject and understand the methodology behind it. When researching which solvers might produce the best results, FEniCS seemed like a very good option.

(Paper to cite here:<https://arxiv.org/pdf/1803.04896.pdf>)

However, getting a good understanding of this software to be able to implement a new method seemed unrealistic for the short time of a Bachelor Thesis, which is limited to a few months.

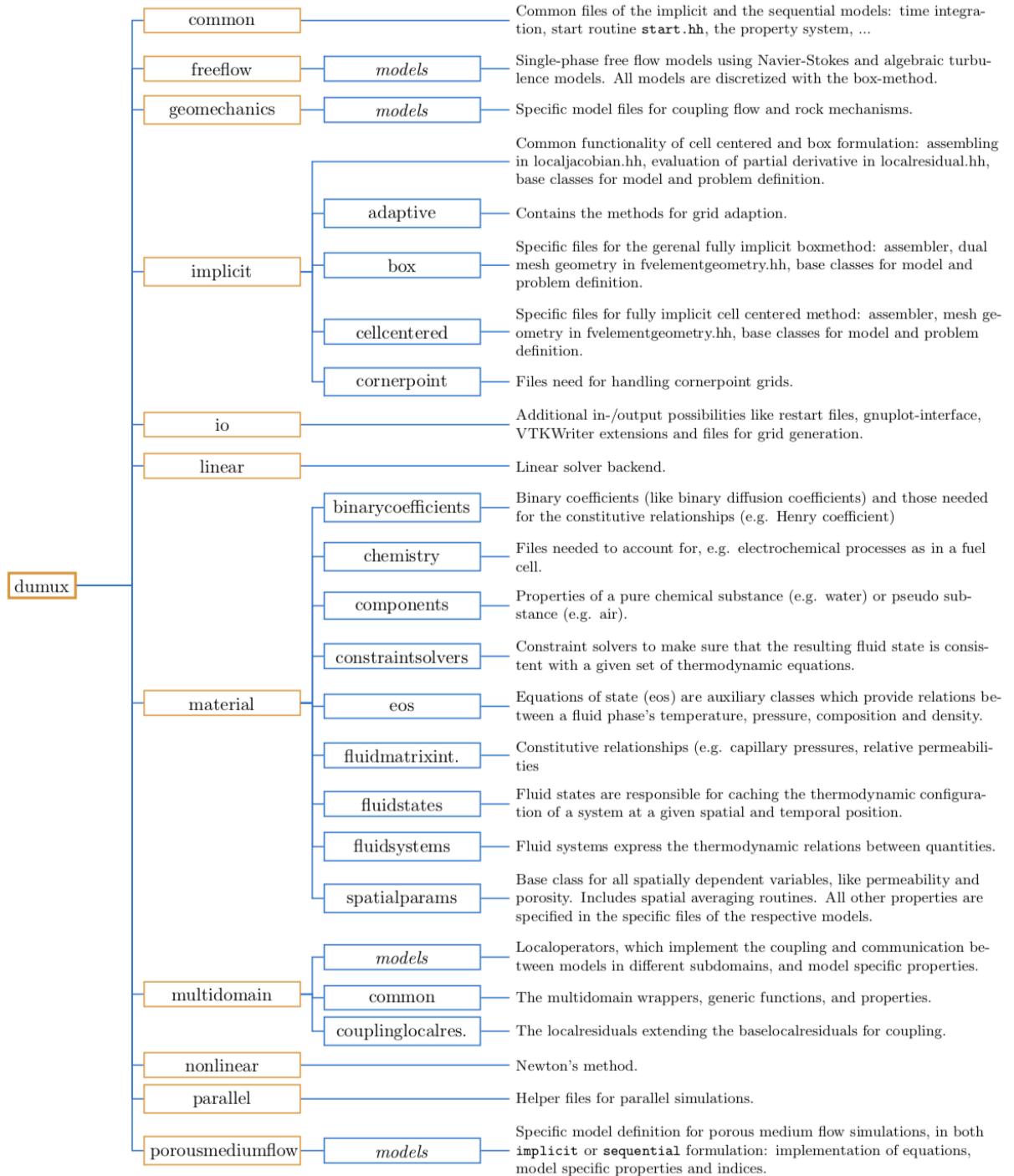
For this reason I decided to use *DuMu^x*, as it already has a few models which compute blood flow simulations. This helped me get a quick insight to how *DuMu^x* works, and how I can use it for my purposes.

3.2.1 General Structure of *DuMu^x*

DuMu^x is an open source software developed by a strong user community everywhere in the world. The software was basically developed by the University of Stuttgart, and provides a C++ based library of numerical methods and implemented test models, to simulate and solve transport and flow processes in porous media [3].

DuMu^x is a multi-scale multi-physics toolbox that aims to describe the physical properties of a specific problem as correct as possible, by focussing on minimizing the computational cost [3].

The structure of *DuMu^x* is specified in this chapter and illustrated with some figures. Figure 1 is the Structure of *DuMu^x* given from the *DuMu^x* Version 2.12 Documentation [1].

Figure 1: The *DuMu^x* Structure

Available Numerical Solvers

DuMu^x offers a large variety of numerical solvers that can be implemented when defining the problem-file. This provides large flexibility in terms of computational cost and stability questions. [1]

One has to take care to choose a solver that is compatible with the given problem...

3.2.2 Implemented Model

While *DuMu^x* has a modular structure, some of the existing test cases can be combined to solve a more sophisticated and specific problem. In our case, this refers to the coupling of the existing tracer model to the 1p-1p model. While the 1p-1p model can simulate the flow of a solute in a network by convection and the diffusion of this solute into tissue, the tracer model can literally trace the path of this solute into the tissue, which can be used to obtain oxygen transport and diffusion into the tissue.

The 1p-1p Model

The 1p-1p model can compute pressure fields and velocities in vessel networks. The velocity outputs can then be used for the tracer model.

1p-1p Model Impressions

Figure 2 is the result of a *DuMu^x* blood flow simulation computed on a Nephron network. The pressure field in the network is visualized.

Figure 3 is the result of a *DuMu^x* blood flow simulation computed on a Nephron network. The velocity field in the network is visualized.

The Tracer Model

The tracer model can trace substances in networks of different geometries, and accordingly compute concentrations and concentration fields of solutes in dynamic systems.

This is the reason I decided to choose the Tracer model to simulate the oxygen transport and diffusion in the given networks.

Generally, the tracer model is computing the distribution of a solute for every time step. It can either take velocities or flows as inputs, and gives the concentration of the traced

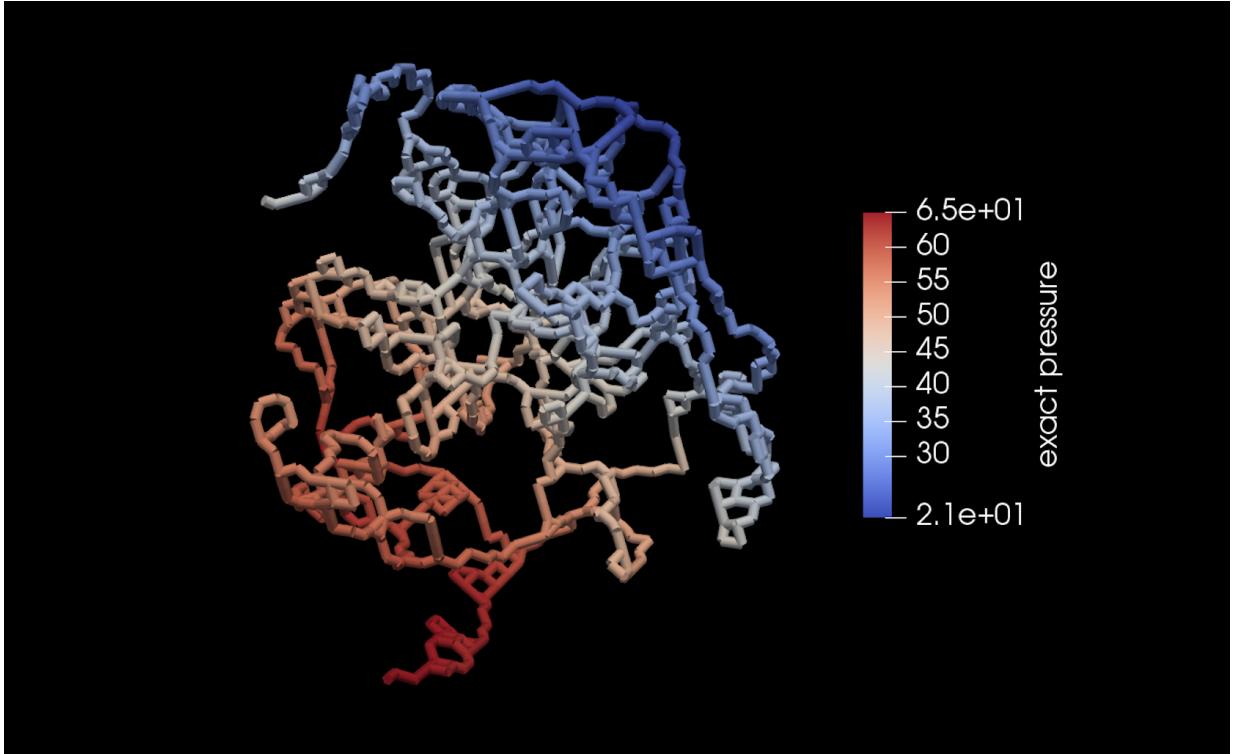


Figure 2: *DuMu^x* Pressure Computations for Nephron Network

solute as an output.

To briefly show what is meant by tracing a solute, a visualization for the first and last time of a tracing simulation is pictured in the following figures.

This simulation is a simple diffusion of Groundwater in a porous rock. The velocity field of the solute is assumed to be constant.

Some Tracer Simulations

Figure 4 is the result of a *DuMu^x* tracer simulation computed on a rectangular geometry. The solute field in the network for the first time step is visualized.

Figure 5 is the solute field for the same simulation for the last time step is visualized.

Figure 6 is the constant velocity field of the solute.

The Coupled Model

To achieve the goal of tracing oxygen in the given vessel-networks, I coupled the previously described models to obtain a 1p-1p-tracer model...

Using the velocity output of my previous 1p-1p model as an input for the tracer model, I wanted to get the concentration of my solute of interest, which is oxygen, for every time step.

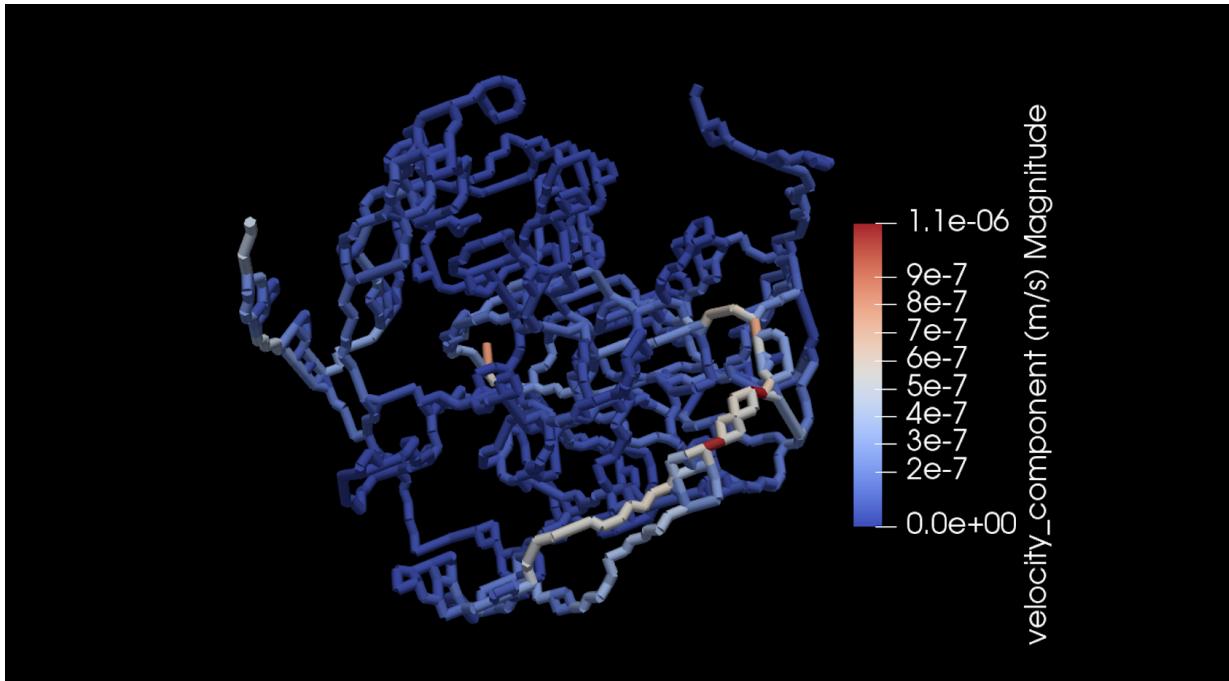


Figure 3: *DuMu^x* Velocity Computations for Nephron Network

3.2.3 Input Data for the Simulations

The networks that were used for the simulations are extracted from real organs coming from mice. This process is done by scientists from different areas of expertise and is usually performed in the following order:

- 1)CT of the kidneys
- 2)Recognition of fat, tissue, arteries and veins by machine learning algorithms
- 3)Three dimensional captions of the organ are build by many virtual two dimensional slices
- 4)Transformation of the recognized artery/vein data into networks that can be used for the simulations.

The previously described tasks can be all be done by Interface Group members, which makes the working environment very special in terms independent research capabilities in this partially unchartered/unexplored field.

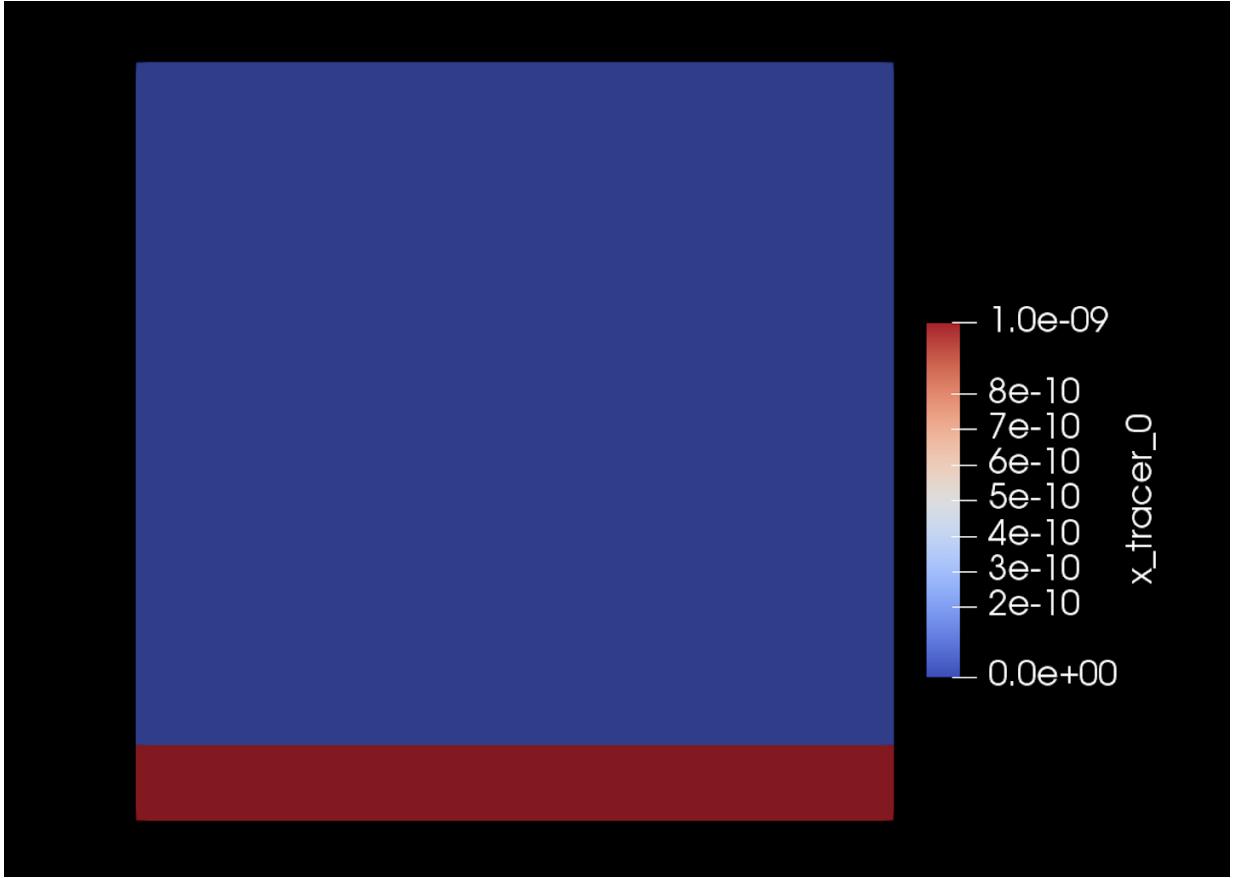


Figure 4: *DuMu^x* Tracer Concentration Simulation for a Rectangular Geometry (Time Step 1)

4 Simulation Results and Analysis

In this section, the produced results using the two introduced methods will be presented and analyzed for a few different networks. Finally the results will be discussed and the methods will be compared.

4.1 Results of the Green's Function Method

The Green's Function Method is providing many outputs which are specified in more detail in the Appendix A.1.1. In this section, some of these outputs will be presented and explained. The physiological meaning will be described and finally the results will be discussed.

Krogh Model as an Introduction

The Krogh model describes the tissue as a cylinder around the vessels and assumes that the capillaries are straight and parallel. The blood flow in all the capillaries is assumed to be unidirectional, and the distribution of capillaries is assumed to be homogeneous [8]. Obviously this model has a few unrealistic assumptions, but as it is a straightforward

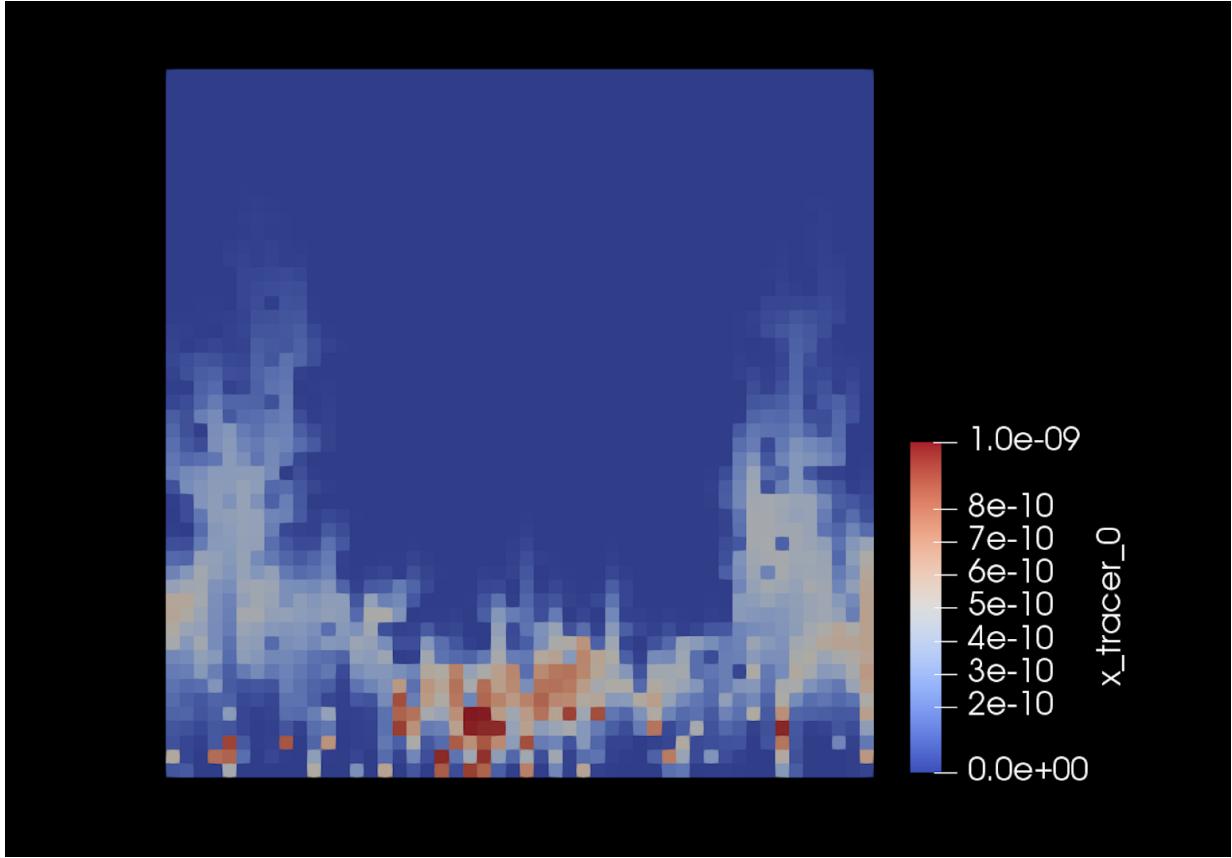


Figure 5: *DuMu^x* Tracer Concentration Simulation for a Rectangular Geometry (Time Step 10)

network, it is the most basic model to start with.

The Krogh model has been a standard simulation network used as an initial point for developments as it has a very easy and simple structure. Figure 7 is the result of a Green's Method blood flow simulation computed on a Krogh network. The Solute Concentration in the network and in the tissue is visualized in the form of a slice (See Output Section A.1.1 for more information). The illustration basically shows a virtual cut into the tissue and gives local oxygen concentrations in mmHg.

One can very clearly see that the results produced are physically very realistic. On 7 the blood flow goes from left to right. The O_2 concentration in mmHg accordingly goes down in flow direction. The concentrations as well as the diffusive distance in the tissue also decrease in flow direction, which can be explained by the fact that a lower concentration gradient goes with a lower Diffusive Flux. This makes sense when looking at Fick's Law of Diffusion 2.3. As one can see from the Input files A.1.1, the Diffusivity constant α is constant for homogeneous tissue. Due to the fact that the diameter of the vessels is constant as well when considering the Krogh Model, the intravascular resistance to radial oxygen transport K for each diameter/vessel is also constant. Differences in diffusive fluxes and Oxygen penetration distances into the tissue therefore mainly depend on the PO_2 gradients themselves.

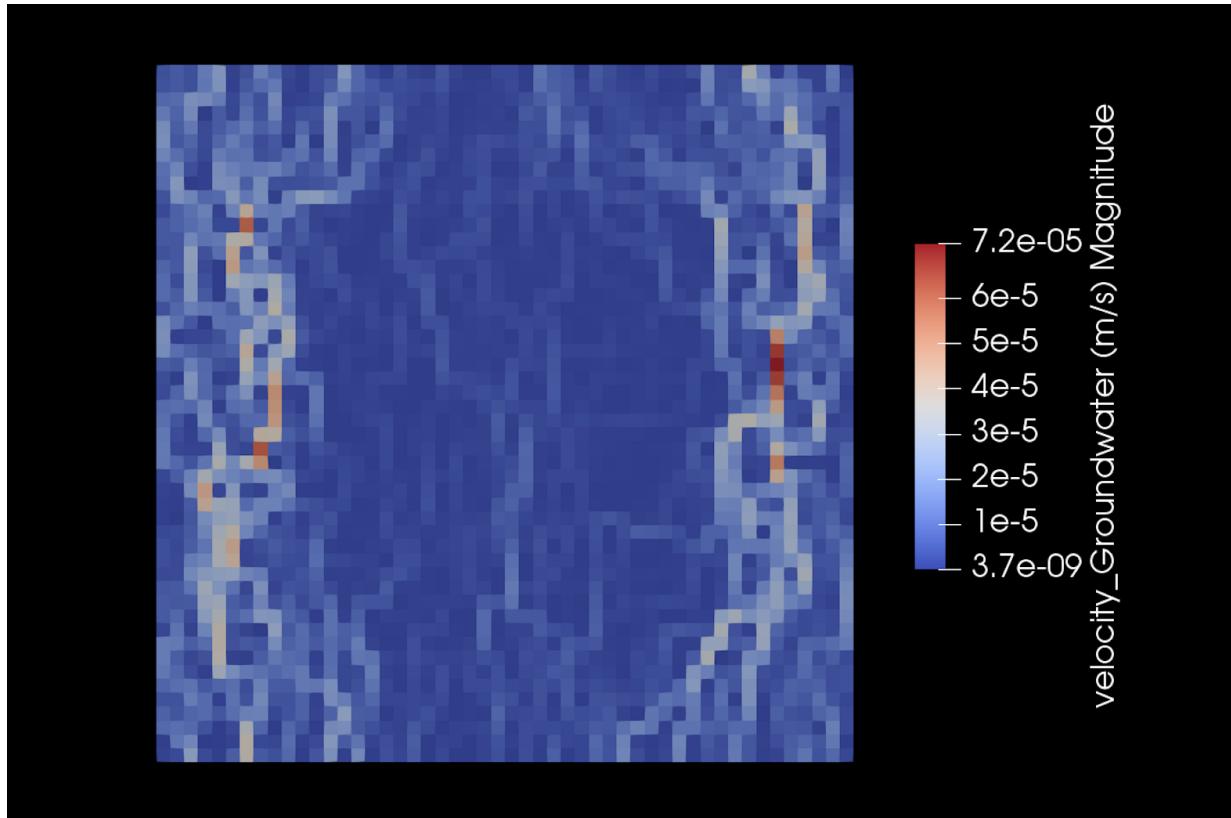


Figure 6: *DuMu^x* Tracer Velocity Simulation for a Rectangular Geometry

Eventually, one can see that the results computed on the Krogh Network show that the Green's Function method performs well and produces physically accurate results on simplified and perfectly homogeneous networks as this one.

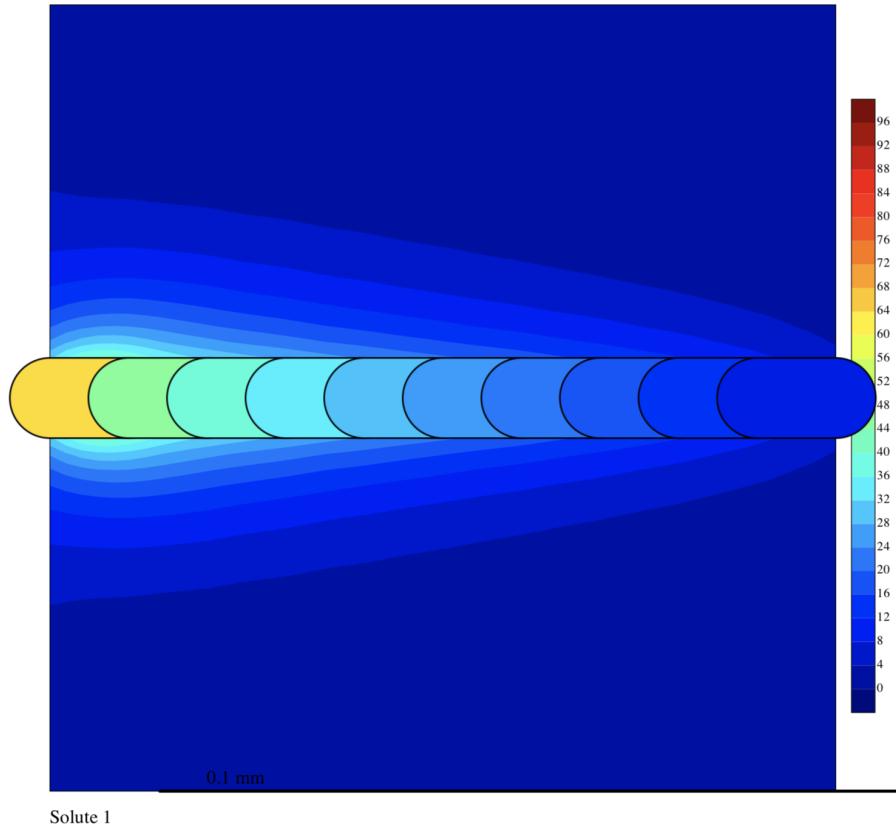


Figure 7: Green's Contour Output for Krogh Network

The Cardiac Model

The Cardiac network provided by Secomb was used for further simulations and to analyze results in more complex networks.

This network is obviously more complicated than the previously presented Krogh model. The result is visualized on Figure 8. Here one can see the result of a Green's Method blood flow simulation computed on a Cardiac network. As for the Krogh model, the Oxygen Concentration in the network and in the tissue is visualized in the form of a slice (See Output Section A.1.1 for more information). Again the illustration shows a virtual cut into the tissue and gives local oxygen concentrations in mmHg.

Obviously looking at the Contour plot doesn't make much sense here, as the high number of vessels makes an analysis of the PO_2 distribution in the tissue impossible. For this reason, other files must be used but which ones ?

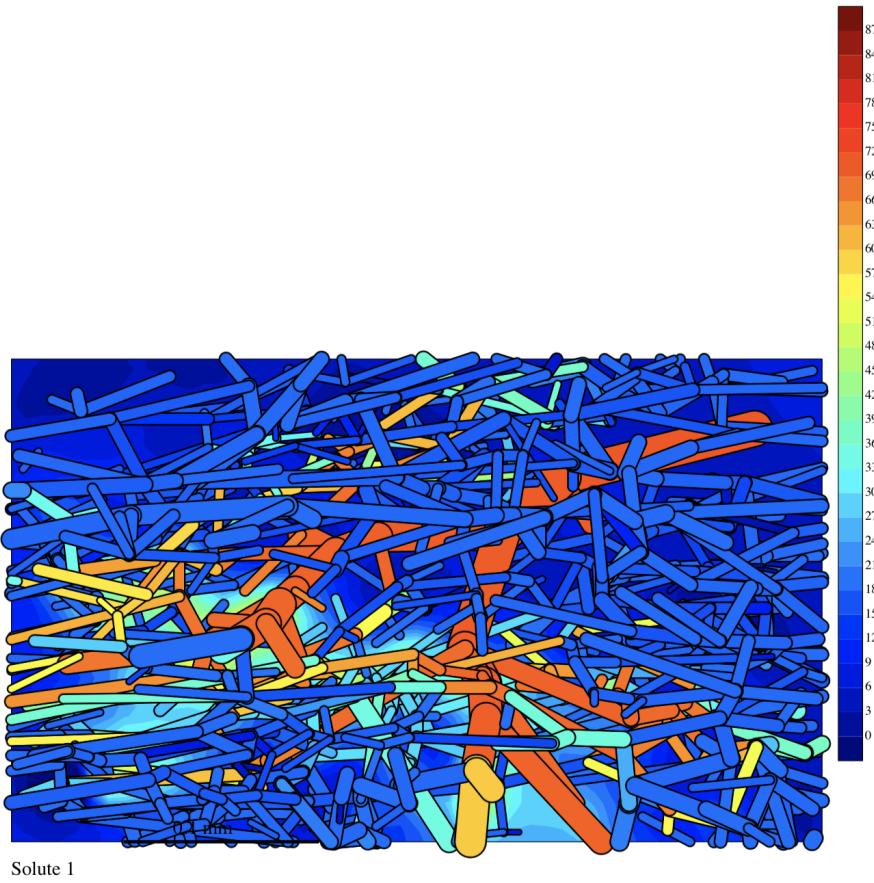


Figure 8: Green's Contour Output for Cardiac Network

The Mesentery Artery Model

The Mesentery Artery network provided by Secomb was used for further simulations and to get more computational results.

This network is obviously more complicated than the previously presented Krogh model but luckily gives a way better insight to the results than the Cardiac Network simulation, due to the much lower vessel density. The result is visualized on Figure 9. Here one can see the result of a Green's Method blood flow simulation computed on a Mesentery Artery network. As for the previous models, the Oxygen Concentration in the network and in the tissue is visualized in the form of a slice (See Output Section A.1.1 for more information). Again the illustration shows a virtual cut into the tissue and gives local oxygen concentrations in mmHg.

Due to the low density of capillaries for this network, this figure can give a good insight to the values computed by the Green's Model. One can clearly see that partial

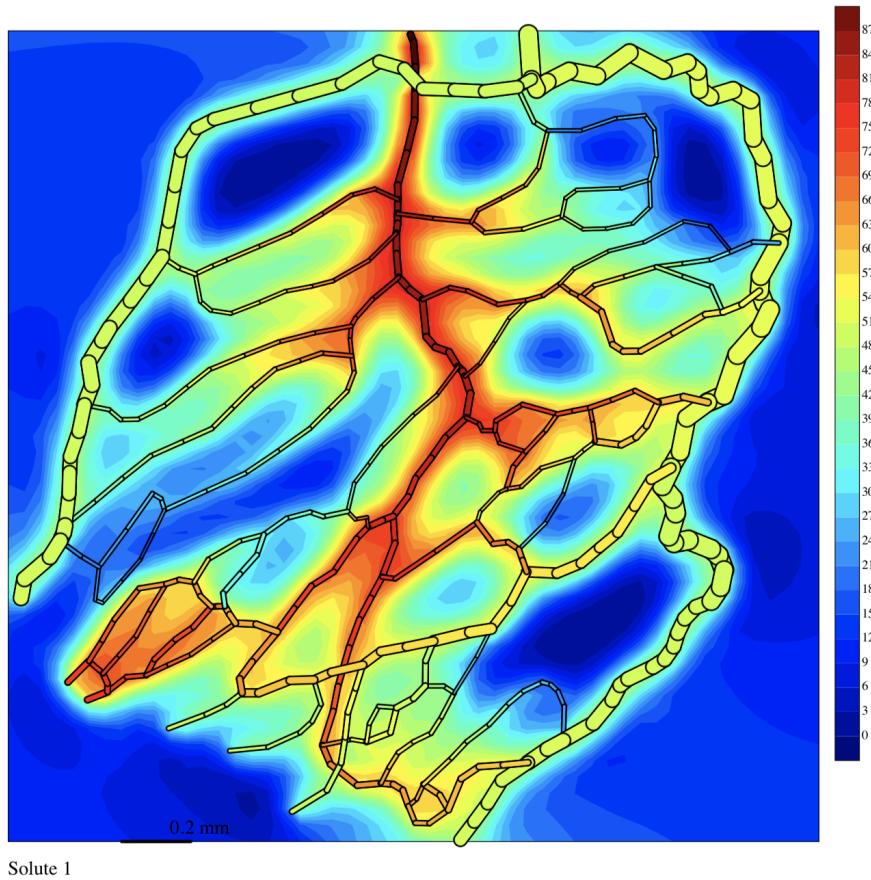


Figure 9: Green's Contour Output for a Mesentery Artery Network (Solute 1)

pressure of O_2 goes down in radial direction with growing distance from the vessels. At hotspots where the capillary density is higher due to branching and ramification, the diffusion comes from two vessels and the superposition of two close source elements obviously leads to peaks in PO_2 values.

A First Tumor Model

The Tumor network provided by Secomb was used for further simulations and to validate the produced data. Figure 10 is the result of a Green's Method blood flow simulation computed on a Tumor network. The Solute Concentration in the network is visualized.

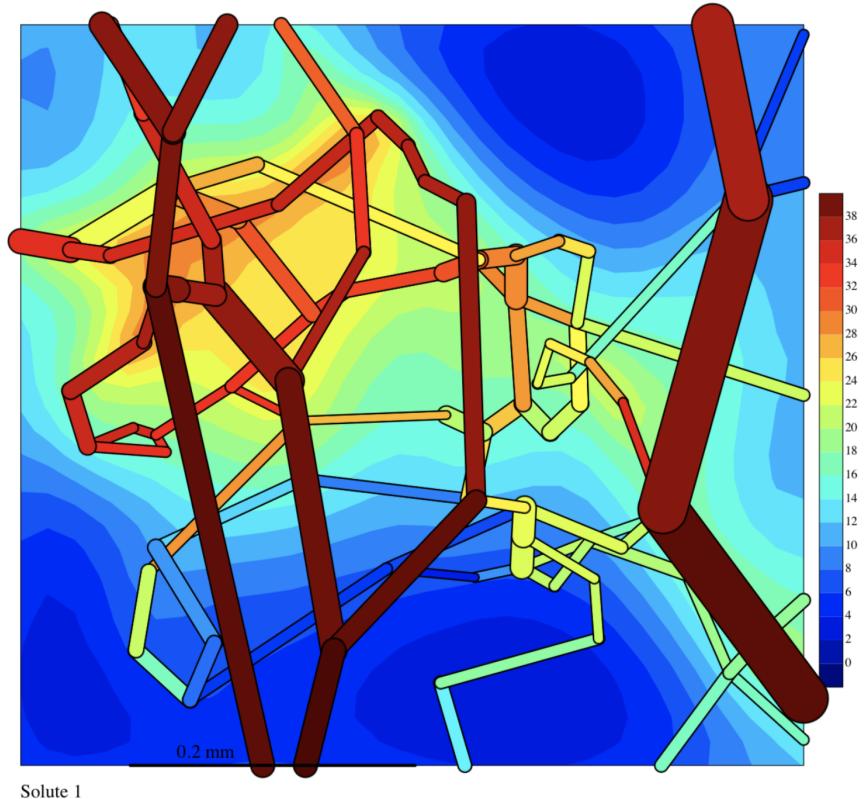


Figure 10: Green's Contour Output for Tumor 1

A Second Tumor Model

The Tumor network provided by Secomb was used for further simulations and to validate the produced data. Figure 11 is the result of a Green's Method blood flow simulation computed on a Tumor network. The Solute Concentration in the network is visualized.

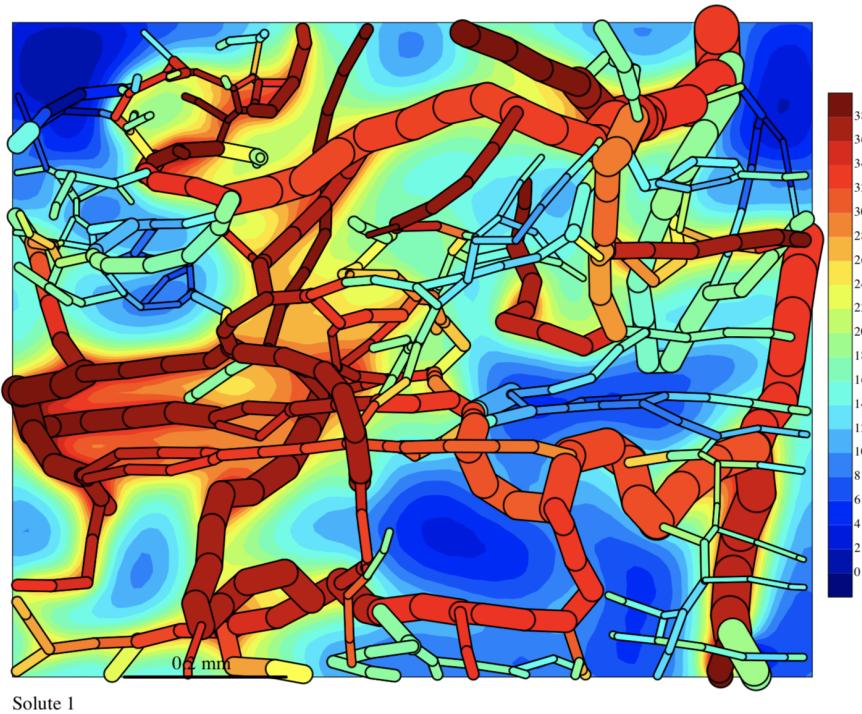


Figure 11: Green's Contour Output for Tumor 2

Further Outlook for New Networks

Put one or two new networks from Kartik in and analyze the results like for the previous networks.

4.2 Discussion of the Green's Method and the Produced Results

Final thoughts about Green's Method for physiological applications and oxygen delivery simulations/computations.

4.3 Results of *DuMu^x*

- Pros/Cons
- Quality of results
- Paraview Visualization of results

Some Glomerulus Network Simulations

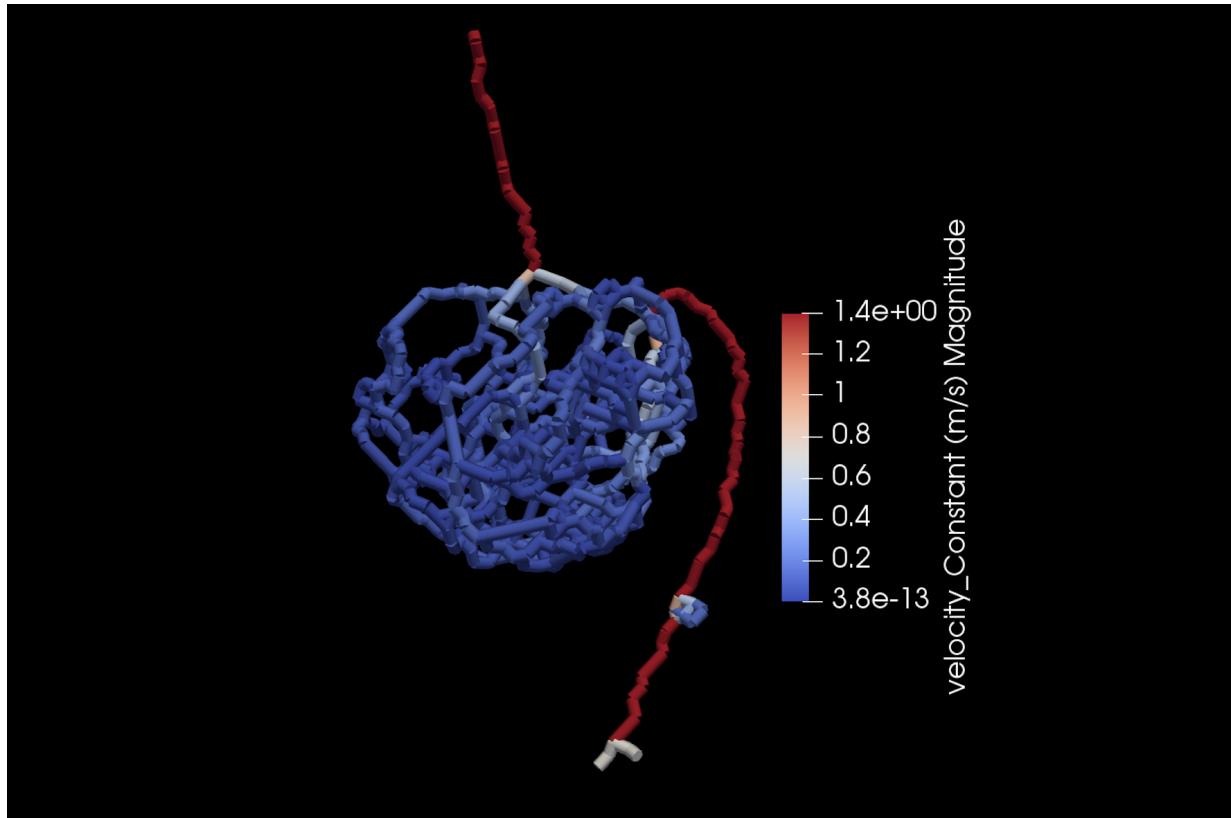


Figure 12: Velocity Field in a Glomerulus

The figure 17 is the result of a DuMuX 1p-1p blood flow simulation computed on a glomerulus network. The velocity field of the flow is visualized.

The figure 18 is the result of a DuMuX 1p-1p blood flow simulation computed on a glomerulus network. The pressure field of the flow is visualized.

4.4 Comparison of the Results

In this section the main differences in the produced results by Green's Function Method and *DuMu^x* for the same input files/same networks will be discussed.

General differences in terms of computational cost and implementation issues can be discussed here as well when explaining the differences in the results.

I might include the whole next section here, as this might be more compact.

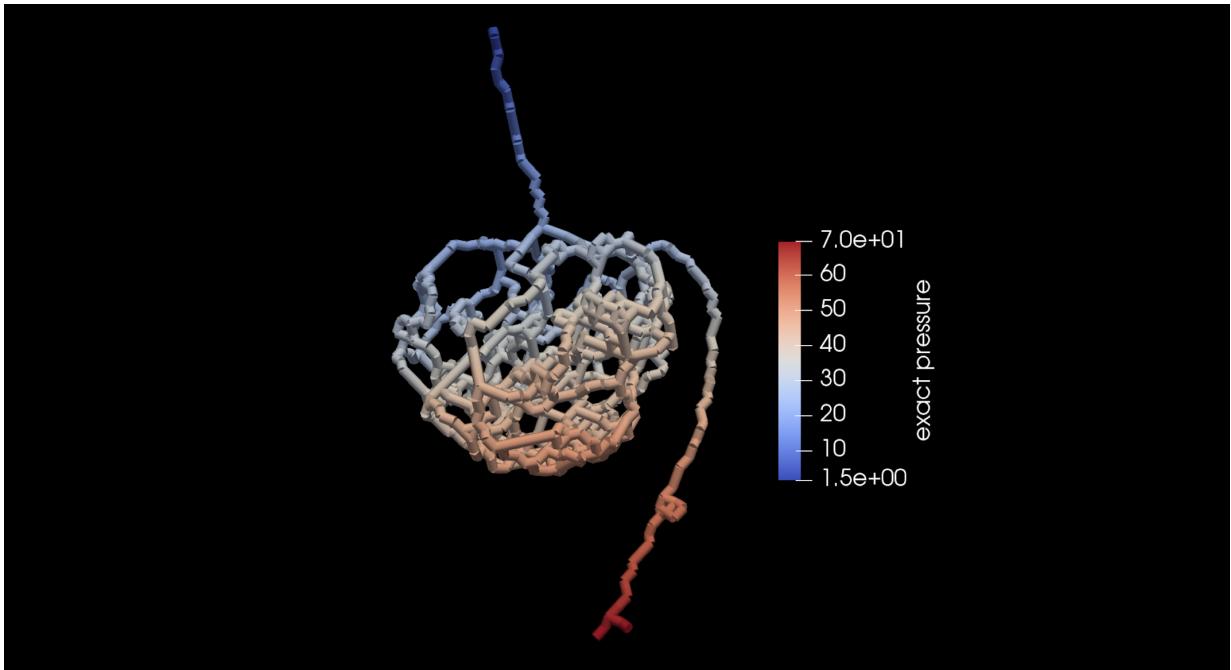


Figure 13: Pressure Field in a Glomerulus

5 Comparison of the Different Methods Behind the Simulations

In this section the main differences between these two simulation methods described previously will be discussed.

Kartik do you think I need this 5th chapter at all ? I think I would rather delete it. I could include the subsections into the Simulation Methods part as well, and discuss some of these in the Conclusion as well ?

In this section the main differences between the solvers and the associated numerical methods will be discussed.

As mentioned previously, one of the huge advantages of *DuMu^x* is the flexibility it provides in terms of solvers, and thus in terms of the numerical methods used for the computations. This means that it is often possible to pick a solver which is stable for the specific problem and the chosen discretization.

This is obviously different for the Green's Method code, as there is no option of choosing a different solver than the one provided in the code. As discussed previously, this code is specifically implemented to solve and compute oxygen (or other Solute) concentrations in the blood and the surrounding tissue, and the implemented method strongly relies on

the stability of the Green's Method and the numerical methods behind the solver.

5.1 Comparison of the Numerical Methods

In this section, the focus will be on comparing the numerical methods behind the Green's Method code and *DuMu^x* and the differences in their implementation.

5.2 Stability Questions and Computational Cost

In this section, the consequences of the previously described differences will be discussed. The focus lies on the differences in terms of stability and computational cost for similar simulations.

6 Conclusion

As seen previously, both the Green's Function Method and the implemented *DuMu^x*-based method can offer very interesting options. While the Green's Function Method is presenting a specifically created C++ based software for oxygen Transport and Diffusion simulations, the results show that the obtained outputs are not always satisfying (eventually will happen for new networks?). The reasons for this have been discussed previously (maybe repeat them shortly here). Compared to this, the *DuMu^x* based model is still in its very initial phase and cannot produce correct results yet (very probably). Still, the fact that it is *DuMu^x* based offers both a large flexibility to the user in terms of further development, as well as in terms of computational flexibility when choosing a time- and space-discretization through choosing discretization modules and different grid creators. The solvers can also be chosen by the user, which doesn't guarantee an overall stable numerical solver, but can provide great flexibility when adapting the model to specific networks.

Main conclusions and final remarks.

7 Acknowledgements

Thank you all for the support blabla...

References

- [1] Dumux homepage. URL <http://www.dumux.org/documents/dumux-handbook-2.12.pdf>.
- [2] D. A. Beard and J. B. Bassingthwaite. Modeling advection and diffusion of oxygen in complex vascular networks. *Annals of biomedical engineering*, 29(4):298–310, 2001.
- [3] B. Flemisch, J. Fritz, R. Helmig, J. Niessner, and B. Wohlmuth. Dumux: a multi-scale multi-physics toolbox for flow and transport processes in porous media. In *ECCOMAS Thematic Conference on Multiscale Computational Methods for Solids and Fluids*, pages 82–87, 2007.
- [4] J. Fletcher. On facilitated oxygen diffusion in muscle tissues. *Biophysical journal*, 29(3):437–458, 1980.
- [5] D. Goldman and A. S. Popel. A computational study of the effect of vasomotion on oxygen transport from capillary networks. *Journal of Theoretical Biology*, 209(2):189–199, 2001.
- [6] J. Hellums. The resistance to oxygen transport in the capillaries relative to that in the surrounding tissue. *Microvascular research*, 13(1):131–136, 1977.
- [7] L. Hoofd. Updating the krogh model-assumptions and extensions. In *SEMINAR SERIES-SOCIETY FOR EXPERIMENTAL BIOLOGY*, volume 51, pages 197–197. Cambridge University Press, 1992.
- [8] F. Kreuzer. Oxygen supply to tissues: the krogh model and its assumptions. *Experientia*, 38(12):1415–1426, 1982.
- [9] D. Kuzmin. A guide to numerical methods for transport equations. *University Erlangen-Nuremberg*, 2010.
- [10] R. Law and H. Bukwirwa. The physiology of oxygen delivery. *Update in Anaesthesia*, 10:6, 1999.
- [11] C.-J. Lee, B. S. Gardiner, J. P. Ngo, S. Kar, R. G. Evans, and D. W. Smith. Accounting for oxygen in the renal cortex: a computational study of factors that predispose the cortex to hypoxia. *American Journal of Physiology-Renal Physiology*, 313(2):F218–F236, 2017.
- [12] R. N. Pittman. Regulation of tissue oxygenation. In *Colloquium series on integrated systems physiology: from molecule to function*, volume 3, pages 1–100. Morgan & Claypool Life Sciences, 2011.

- [13] M. S. Rahman and M. A. Haque. Mathematical modeling of blood flow. In *Informatics, Electronics & Vision (ICIEV), 2012 International Conference on*, pages 672–676. IEEE, 2012.
- [14] T. Secomb, R. Hsu, M. Dewhirst, B. Klitzman, and J. Gross. Analysis of oxygen transport to tumor tissue by microvascular networks. *International Journal of Radiation Oncology* Biology* Physics*, 25(3):481–489, 1993.
- [15] T. W. Secomb, R. Hsu, E. Y. H. Park, and M. W. Dewhirst. Green’s function methods for analysis of oxygen delivery to tissue by microvascular networks. *Annals of Biomedical Engineering*, 32(11):1519–1529, Nov 2004.
- [16] R. Weiss. Parameter-free iterative linear solvers. *Mathematical Research*, 97, 1996.

A Appendix

Additional material such as extra figures.

A.1 Additional Material for the Green's Function Method

A.1.1 General Structure of the Used Code

In this section, an insight to the code provided by Secomb et al. [15] will be given. Especially some explanations about what data this code uses and what data it computes will be given. The main goal is to understand what the results are, how correct their accuracy is. For this, a general explanation of the structure and organization of the code and the methods behind the computations will be given.

Inputs

What are the inputs for this code ? Where do they come from ?

The Green's Function Method takes .dat-files as inputs, which basically contain the network files.

The input data in form of .dat-files is basically build out of four files:

- SoluteParams.dat: This file contains all the information about the solutes we are looking at. This information can be things like tissue solubility, Michaelis constant of consumption, etc.
- Network.dat: This file contains all the information about the network we are looking at. This information is basically in form of an array. This array defines for each segment the type, the start point, the end point, the diameter (in microns), the relative flow (?) and the hematocrit (?).
- IntravascRes.dat: This file contains all the information about a diameter (which one?) and the intravascular resistance to radial oxygen transport K for each diameter/vessel.
- ContourParams.dat: This file contains all the information necessary to create the contour-plots. This means that we create a plot, where the oxygen concentration can be visualized on a 2-dimensional slice. (not sure how plot is created from this data?)

The part of the code where the read-in of the network data and the sources (not sure about this) through the input files is being done is the input.cpp code.

The analyzenet.cpp-part of the code analyzes the input-files...

Outputs

What are the outputs for this code ? How are they produced ? Where do they come from ?

The Green's Function Method gives .dat-files and images (contour.ps-files) as outputs, which basically contain the results that will be discussed later in the next chapters.

The contour.ps-files are produced by the contour.cpp-part of the code, which basically generates and writes the data for the contour-plots. This data is a plot of the vessel from bottom to top according to the z-coordinate. One could see this plot as a 2D picture of a 3D network when looking from the bottom. A new page is generated for each solute, and the computed solute concentration for each area is shown in colors.

`picturenetwork.cpp`: `picturenetwork.cpp` - project network on $z = 0$ plane. Uses parameters from `CountourParams.dat`. Labels nodes with `nodvar` and segments with `segvar` (must be float). Generates a postscript file.

The general outputs of the code can be classified into two categories:

- The first category is in the form of PostScript-files that can be visualized similar to pictures.
- The second category is in the form Text-files, containing the computed data.

Organization of the Code

The code provided by Secomb et al. [15] consists of a few .cpp-files, so that tasks like input read-ins, calculations and output-file generations are separated. Each file has a specific role, which will be explained in detail in this section. As one can think, the most important part of this code is the implementation of the Green's Function Method and its application to the given data.

The `bicgstab.cpp`-part of the code is the implementation of a solver based on the Parameter-free iterative linear solver by R. Weiss [16].

The `blood.cpp`-part of the code is quite interesting, as it basically does an important part of the calculation to obtain the oxygen concentration and distribution in the vessels and the tissue. In this part of the code

The `contr-lines.cpp`-part of the code generates the contour lines for these plots, whereas the `contr-shade.cpp`-part of the code generates the colors and the color bar on the side of the generated picture/plot.

The convect.cpp-part of the code does the convective (flow) part of the calculation.

The eval.cpp-part of the code does the evaluation of the solute field depending on the source strengths provided.

There are some numerical methods and mathematical calculations behind the simulation, as for example the Gauss-Jordan elimination or the Lower-Upper decomposition, which is implemented in the gaussj.cpp-part and the ludcmp.cpp-part of the code.

The main implementation of the Green's Function Method is in the greens.cpp-part of the code, where Green's Function approach for multiple reacting species is implemented. (Talk more about this, heart of the code)

Histograms of solute levels are evaluated in the histogram.cpp-part of the code.

Initial tissue source strengths (given uniform solute field) are computed in the initgreens.cpp-part of the code.

The code main.cpp calls the greens.cpp-code.

The code nrutil.cpp/nrutil.hh is declaring variables and is allocating them.

Outboun.cpp: method = 1: finds the smallest convex region inside the cuboid which excludes tissue node points that have a distance to the nearest vessel greater than a value specified by the user (lb). Any point outside a region between two planes containing the required points is excluded. This is repeated for multiple plane orientations, determined by am.

method = 2: finds all tissue points within a distance lb of the vessels, but does not make a convex region. Fills in 'holes' in interior, whether 2D or 3D.

Output: nnt, total tissue node points inside the region. nbou \neq 1 if inside region, value gives tissue point index

putrank.cpp: generate nodes in order of flow direction

readsources.cpp: read source strengths from file (from TissueSources.out and Vessel-Sources.out)

setuparrays0.cpp: set up arrays with fixed dimensions (for Green's)

setuparrays1.cpp: set up arrays with dimensions of nnod and nseg (for Green's)

setuparrays2.cpp: set up arrays with dimensions of nnv and nnt (for Green's)

testconvekt.cpp: testing if convekt is giving correct results for alpha (?) matrix. Comparing matrix values with values obtained by numerical differentiation.

tissrate.cpp: computing tissue uptake rates of solutes as a function of solute levels (concentration)

Everyone of these code-parts is using the nrutil.hh code, why ?

A.1.2 Some More Green's Results

Figure 14 is the result of a Green's Method blood flow simulation computed on a Cardiac network. The Solute Concentration in the network is visualized.

A.2 Additional Material for DuMuX

A.2.1 Some Nord Network Simulation Results

Figure 15 is the result of a $DuMu^x$ blood flow simulation computed on a Nord network. The pressure field in the network is visualized.

Figure 16 is the result of a $DuMu^x$ blood flow simulation computed on a Nord network. The velocity field in the network is visualized.

A.2.2 Some Glomerulus Network Simulation Results

The figure 17 is the result of a DuMuX 1p-1p blood flow simulation computed on a Glomerulus network. The velocity field of the flow is visualized.

The figure 18 is the result of a DuMuX 1p-1p blood flow simulation computed on a Glomerulus network. The pressure field of the flow is visualized.

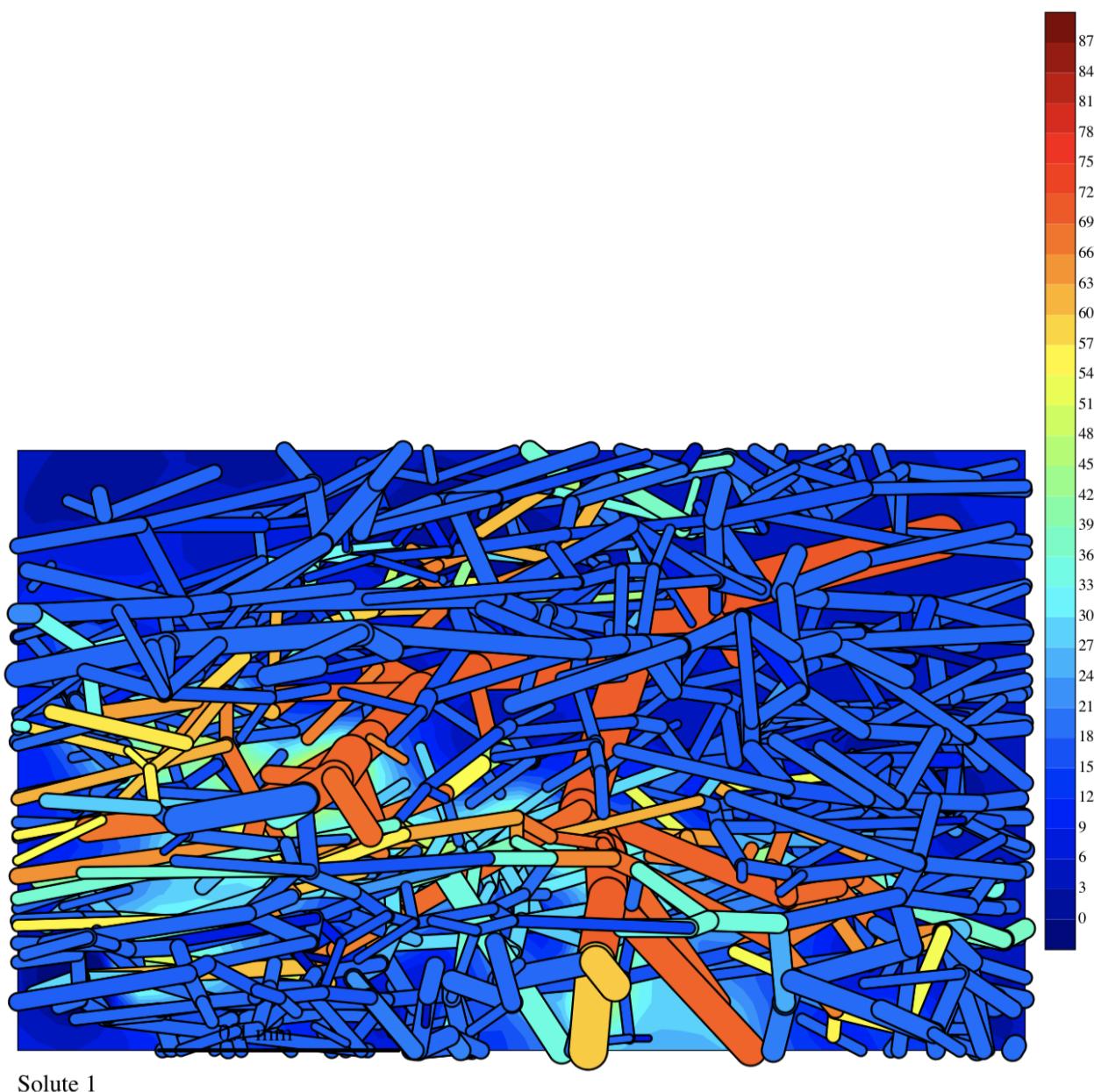


Figure 14: Green's Contour Output for Cardiac Network

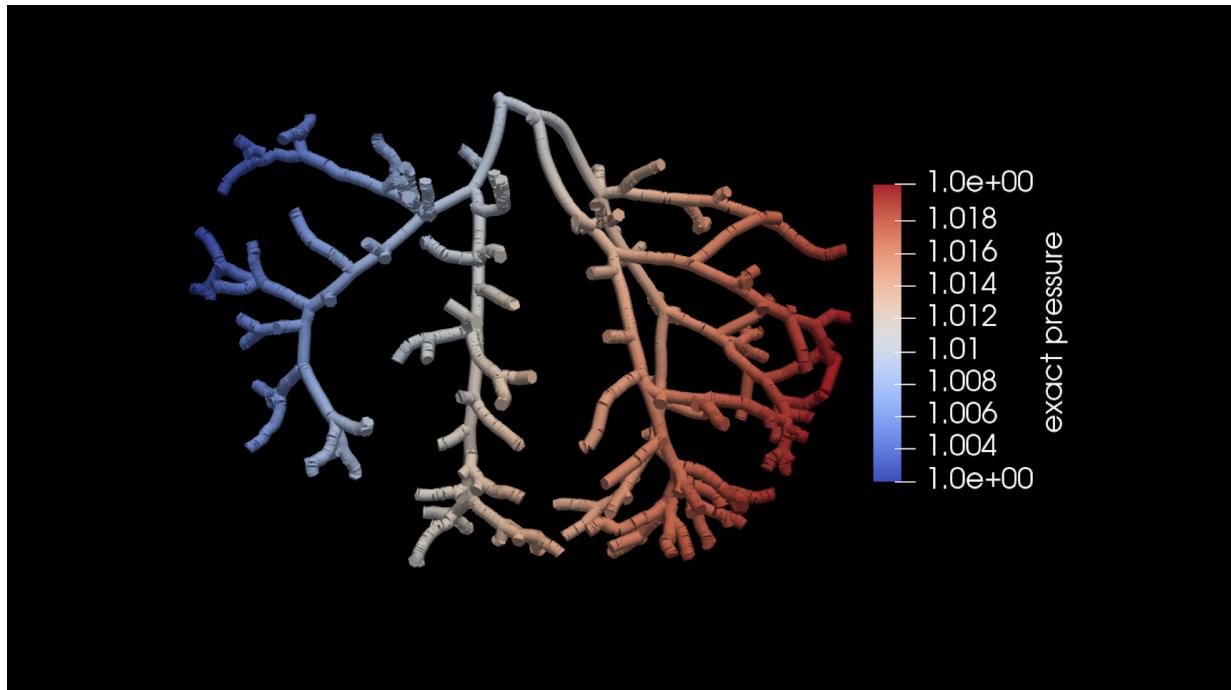


Figure 15: *DuMu^x* Pressure Computations for Nord Network

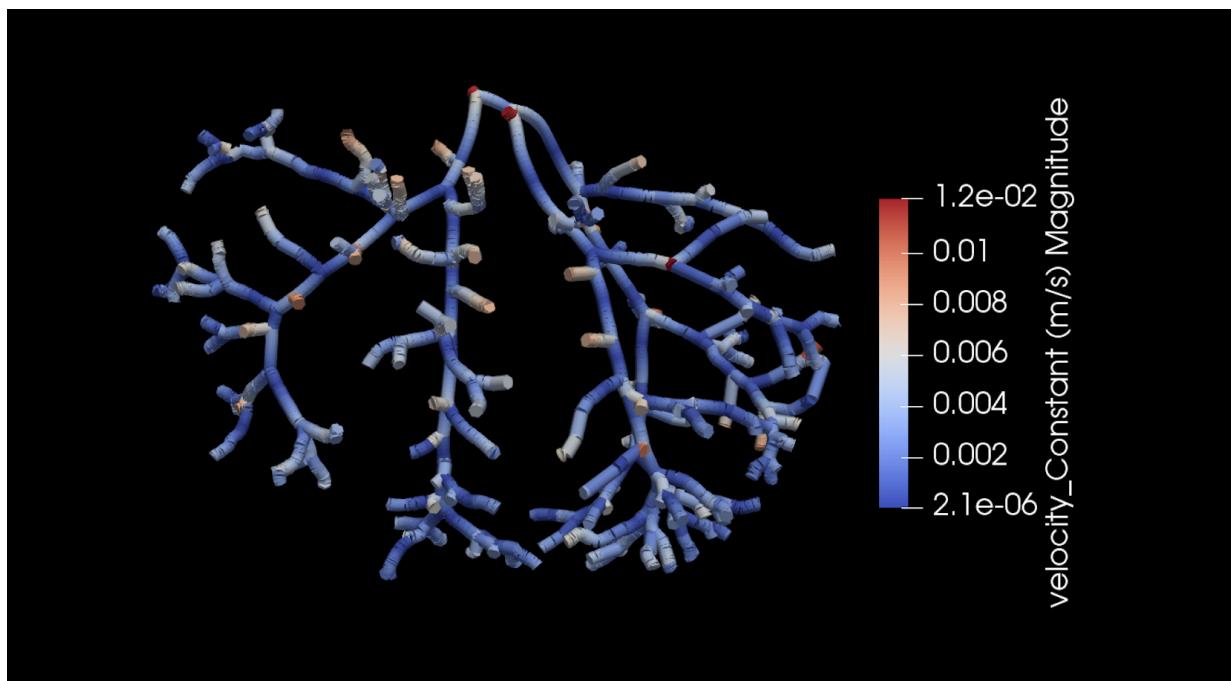


Figure 16: *DuMu^x* Velocity Computations for Nord Network

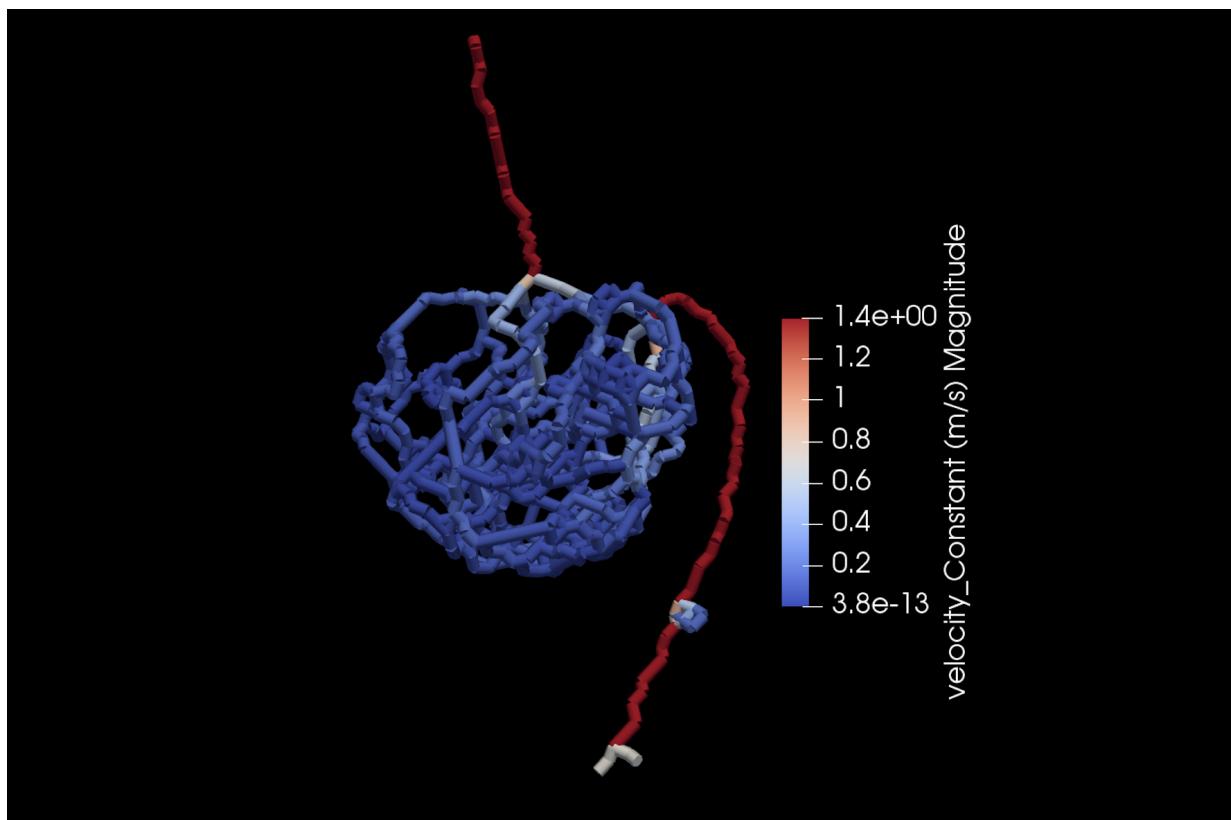


Figure 17: Velocity Field in a Glomerulus

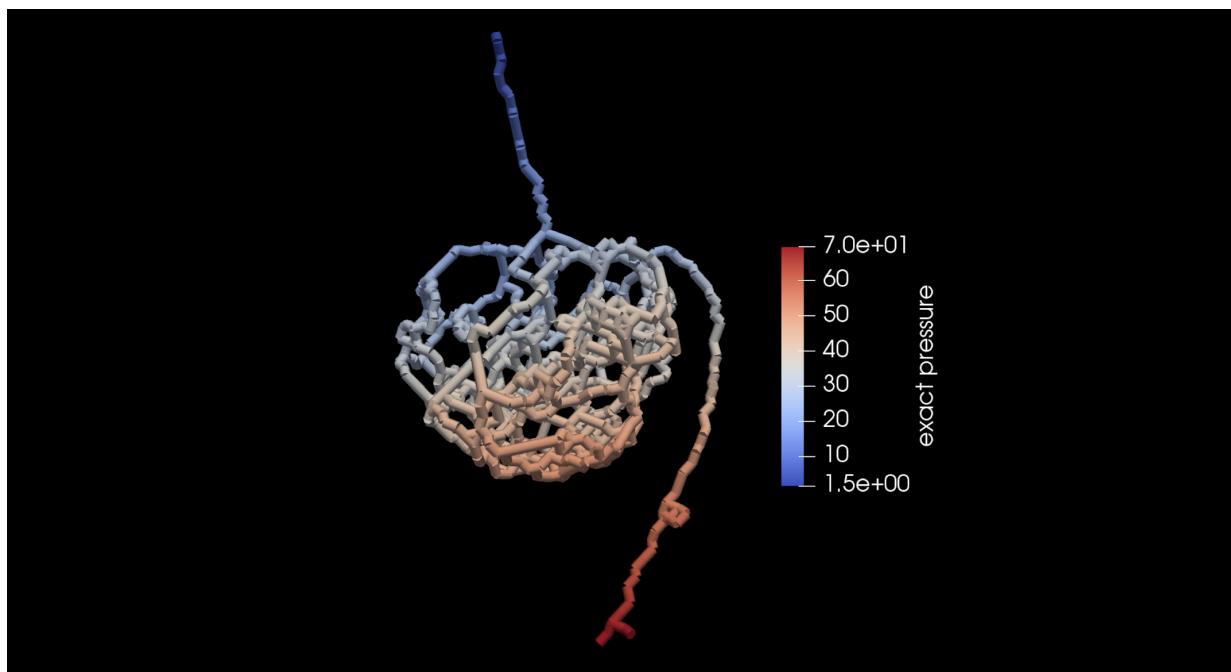


Figure 18: Pressure Field in a Glomerulus