

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)

June 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 | 1 |
| 2 Blood Flow and Oxygen Delivery | 1 |
| 2.1 Blood Flow | 2 |
| 2.2 Oxygen Transport and Delivery | 2 |
| 2.3 Current Trends in Modeling | 2 |
| 3 Simulation Methods | 2 |
| 3.1 Green's Function Method Code | 3 |
| 3.2 <i>DuMu^x</i> | 12 |
| 4 Comparison of the Different Methods Behind the Simulations | 18 |
| 4.1 Comparison of the Results | 18 |
| 4.2 Comparison of the Numerical Methods | 18 |
| 5 Conclusion | 19 |
| 6 Acknowledgements | 19 |
| References | 20 |
| A Appendix | 22 |
| A.1 DuMuX Results Visualized with ParaView | 22 |
| A.2 Green's Function Code Results | 23 |

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. [13] can produce accurate results for new and especially complex networks. A new model using an open source software called *DuMu^x* was implemented to offer a different approach for oxygen transport and diffusion simulations.

1.2 Structure of the Thesis

First, a short inside to the physiological background of blood flow and oxygen transport in the body and especially to the tissue will be given. Linked to this, 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, was evaluated using 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 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.)

2.1 Blood Flow

Blood flow in the body can be approximated by the Navier-Stokes equations.

But rather talk about purely physiological things first.

Put some papers and references in and explain some values and parameters.

Use mathematics paper for blood flow calculations

2.2 Oxygen Transport and Delivery

How does this 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

2.3 Current Trends in Modeling

Do link to modeling and models in this field in general, than link this to computational models in this field and their use. Actual trends and what people hope for in the future (in terms of computational power, computational methods but also from a medical perspective) is also interesting.

Generally Blood Flow in the body can be approximated by the Navier-Stokes equations. 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.

3 Simulation Methods

As mentioned previously, this thesis is focussing on the analysis of simulation methods for oxygen transport and the delivery to tissue surrounding the vessels. In this section I will limit the analysis of simulation methods to the Green's Function Method and accordingly the code developed by Secomb et al. [13], and a new model 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 Code

The Green's Function Method was first used for numerical computations of Oxygen delivery in the tissue by Secomb et al. [13]. It can deliver accurate results with lower computational cost, 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 Green's Function 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

Mathematics Behind Green's Function Method

This part strongly relies on the description by Secomb et al. [13]. In this subsection, the Green's Function Method, its equations and 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. This

$$D\alpha \nabla^2 G = -\delta_3(\mathbf{x} - \mathbf{x}^*) \quad (3.1)$$

3.1 is obtained by putting $G(\mathbf{x}; \mathbf{x}^*)$ as the Green's Function, and saying that it is defined as the potential at \mathbf{x} resulting from a source at \mathbf{x}^* .

$$\mathbf{P}(\mathbf{x}) = \int_{Sources} G(\mathbf{x}; \mathbf{x}^*) q(\mathbf{x}^*) d\mathbf{x}^* \quad (3.2)$$

3.2 gives the potential, with $q(\mathbf{x})$ representing the distribution of source strengths.

$$G = G_1 = \frac{1}{(4\pi D\alpha |(\mathbf{x} - \mathbf{x}^*)|)} \quad (3.3)$$

3.3 gives the solution on an infinite domain, which is the singular function.

$$q_v(s) = \int_0^L F(s - s^*) q_0(s^*) ds^* \quad (3.4)$$

3.15 gives the actual diffusive flux at the blood–tissue boundary.

The distribution of radial flux across the cylindrical surface is obtained by the function $F(s)$, where s is the distance along the vessel.

$$F(s) = \frac{1}{2} \delta_1(s) + \frac{k(K(k) - E(k))}{4\pi r_0} \quad (3.5)$$

3.5 for sources that are uniformly distributed around the circumference.

Where $K(k)$ 3.6 and $E(k)$ 3.7 represent the complete elliptic integrals of the first and second kind respectively :

$$K(k) = \int_0^{\frac{\pi}{2}} (1 - k^2 \sin^2 \Theta)^{-\frac{1}{2}} d\Theta \quad (3.6)$$

$$E(k) = \int_0^{\frac{\pi}{2}} (1 - k^2 \sin^2 \Theta)^{\frac{1}{2}} d\Theta \quad (3.7)$$

The Computational Model

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. Similar to the previous section, this section will strongly rely on [13].

When looking at the code of the Green's Function Method, and especially at the inputs 3.1.2, 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 3.1.2.

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:

$$J = -D \frac{dc}{dx} \quad (3.8)$$

Where J is the diffusive flux, D is the previously mentioned diffusivity coefficient, and dc/dx is the derivative of the concentration with respect to the distance x .

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

$$\frac{\delta\rho}{\delta t} + \nabla(\rho v) = 0 \quad (3.9)$$

These two fundamental equations can be combined to finally get:

$$D\alpha\nabla^2P = M(P) \quad (3.10)$$

3.10 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 3.8. For the consumption, a Michaelis-Menten relationship applied to the problem is used:

$$M(P) = \frac{M_0 P}{(P_0 + P)} \quad (3.11)$$

In 3.11, $M(P)$ 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.

$$f(P_b) = Q(H_D C_0 S(P_b) + \alpha_{eff} P_b) \quad (3.12)$$

In 3.12, 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:

$$S(P_b) = \frac{P_b^n}{(P_b^n + P_{50}^n)} \quad (3.13)$$

In Hill's equation 3.13, P_b ist 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:

$$\frac{df(P_b)}{ds} = -q_v(s) \quad (3.14)$$

3.14 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:

$$q_v(s) = -D\alpha \int_0^{2\pi} \frac{\delta P}{\delta r} r_v d\Theta \quad (3.15)$$

3.15 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]:

$$P_v(s) = P_b(s) - K q_v(s) \quad (3.16)$$

3.16 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 3.1.2 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 3.17.

$$P^* = P + \frac{D_{Mb} C_{Mb} V_m S_{Mb}(P)}{D\alpha} \quad (3.17)$$

In 3.17, 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 General Structure of the Used Code

In this section, an insight to the code provided by Secomb et al. [13] 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 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 relativ 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 pages 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 pa-

rameters from CountourParams.dat Labels nodes with nodevar 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. [13] 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 [14].

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 \downarrow 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)

testconvect.cpp: testing if convect 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 ?

3.1.3 Results on Different Networks

- Present some results on different networks - Plots and Results (maybe 3 I already have + 2 new ones)
- Explanation of plots
- Explaining inputs/outputs and meanings

Krogh Model as an Introduction

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

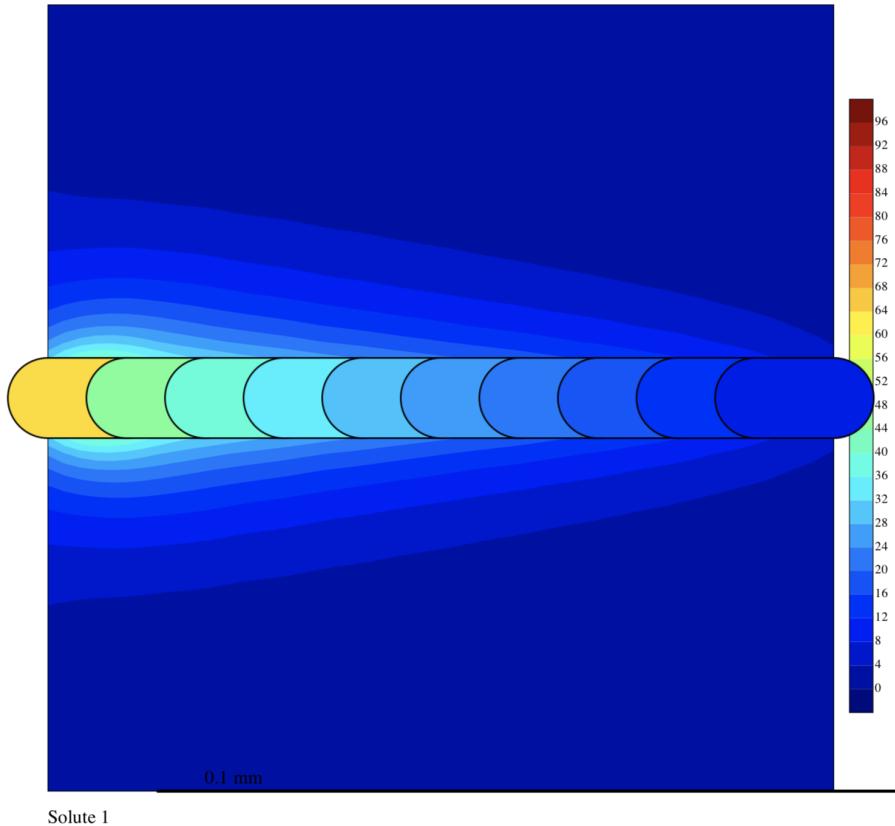


Figure 1: Green's Contour Output for Krogh Network

Cardiac Model

The Cardiac network provided by Secomb (?) was used for further simulations and to validate the produced data.

Figure 2 is the result of a Green's Method blood flow simulation computed on a Cardiac network. The Solute Concentration in the network is visualized.

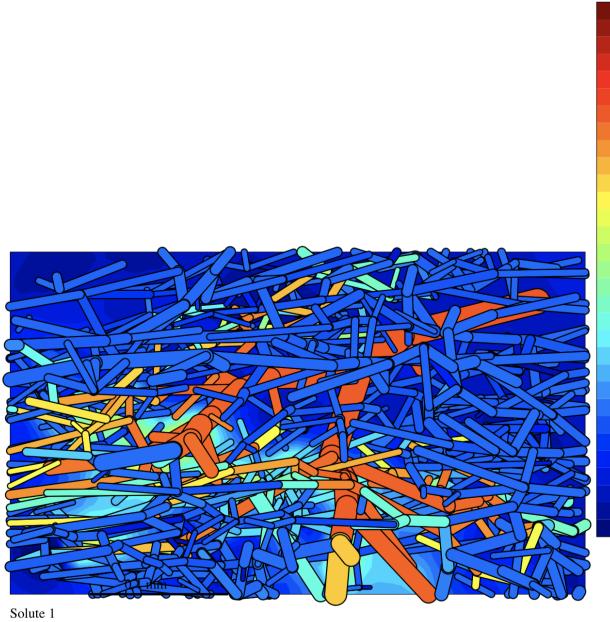


Figure 2: Green's Contour Output for Cardiac Network

Mescent Model

The Mescent network provided by Secomb (?) was used for further simulations and to validate the produced data.

Figure 3 is the result of a Green's Method blood flow simulation computed on a Mescent network. The Solute Concentration of Solute 1 in the network is visualized.

Figure 4 is the result of a Green's Method blood flow simulation computed on a Mescent network. The Solute Concentration of Solute 2 in the network is visualized.

Tumor Model

The Tumor network provided by Secomb (?) was used for further simulations and to validate the produced data. Figure 2 is the result of a Green's Method blood flow simulation computed on a Tumor network. The Solute Concentration in the network is visualized.

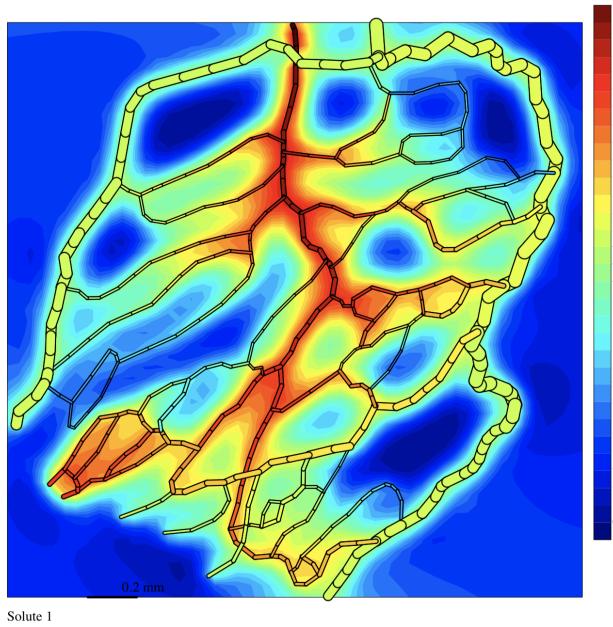


Figure 3: Green’s Contour Output for Mescent Network (Solute 1)

Further Outlook for New Networks

The results provided by the Green’s Method Code were previously discussed for a few networks provided by Secomb (?). In this section the goal is to see how this method can perform in general when it is applied to new networks and to discuss the produced results in detail.

3.1.4 Discussion of Green’s Method and the Produced Results

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

3.2 *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 as described in [3].

3.2.1 General Structure of *DuMu^x*

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.

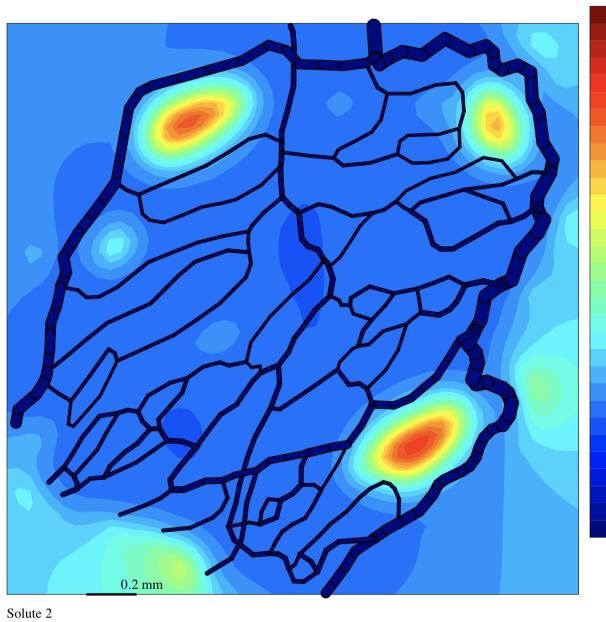


Figure 4: Green's Contour Output for Mescent Network (Solute 2)

Figure 6 is the Structure of *DuMu^x* given from the *DuMu^x* Version 2.12 Documentation [1].

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.

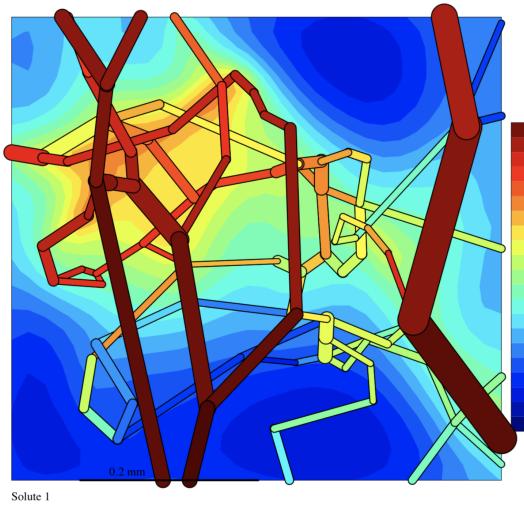


Figure 5: Green's Contour Output for Tumor Network

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:

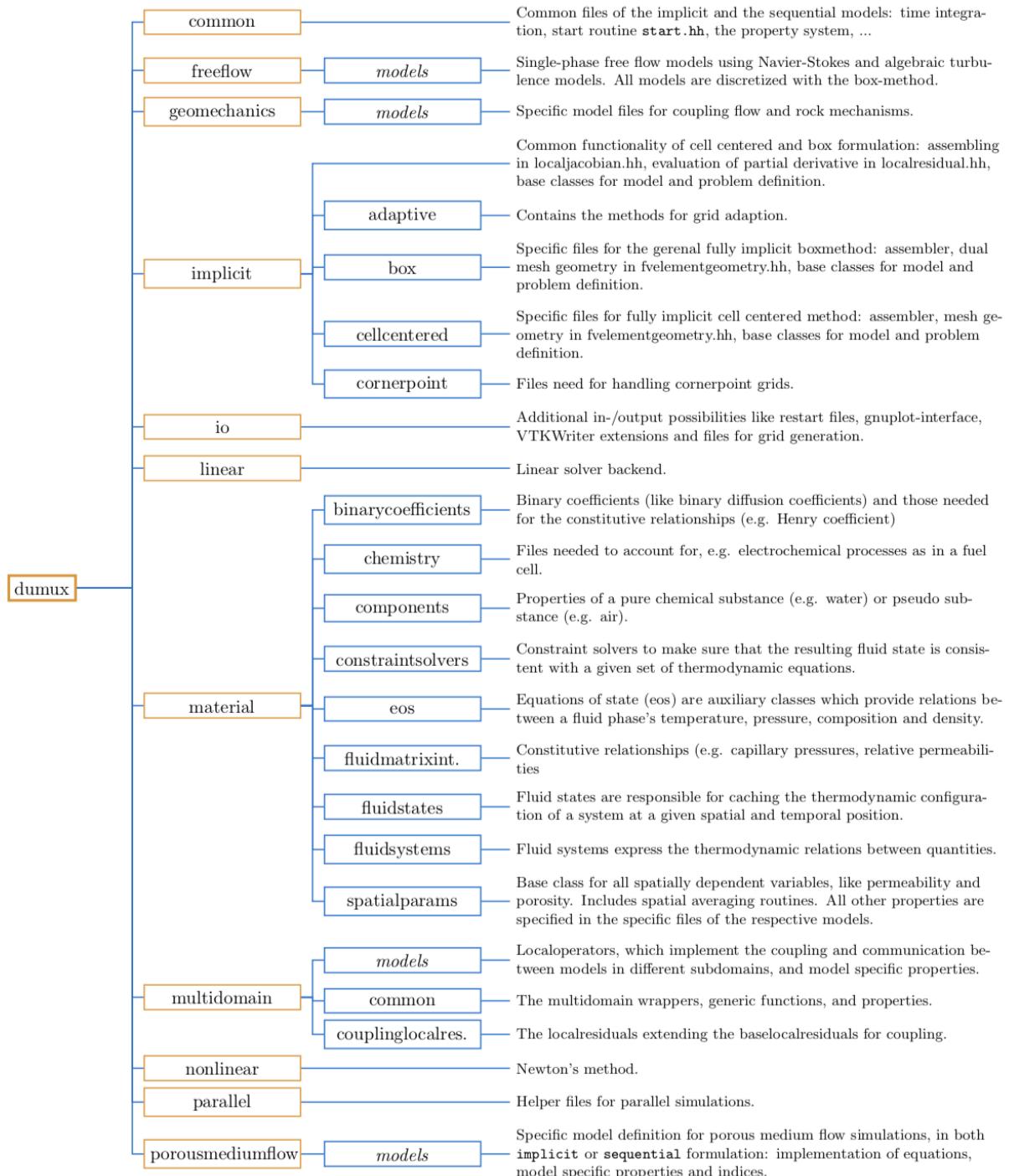
- 1)CT of the kidneys
- 2)Recognition of fat, tissue, arteries and veins by machine learning algorithms
- 3)Three dimensional captions of the organ are build by many virtual two dimensional slices
- 4)Transformation of the recognized artery/vein data into networks that can be used for the simulations.

3.2.4 Results

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

Some Nord Network Simulations

Figure 7 is the result of a *DuMu^x* blood flow simulation computed on a Nord network. The pressure field in the network is visualized.

Figure 6: The *DuMu^x* Structure

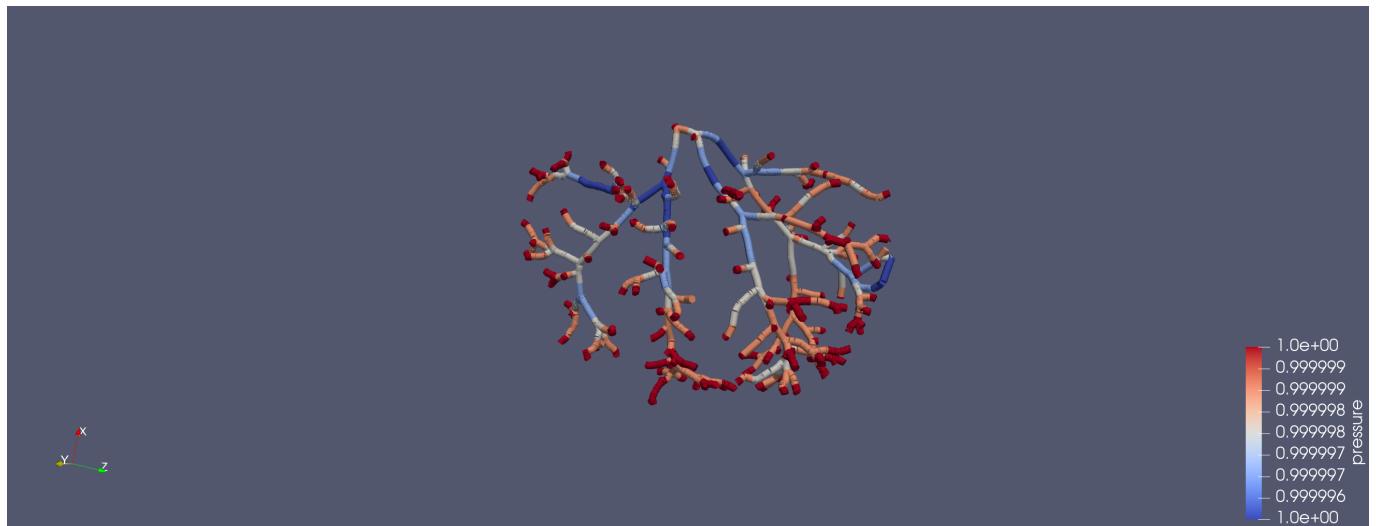


Figure 7: *DuMu^x* Pressure Computations for Nord Network

Figure 8 is the result of a *DuMu^x* blood flow simulation computed on a Nord network. The velocity field in the network is visualized.

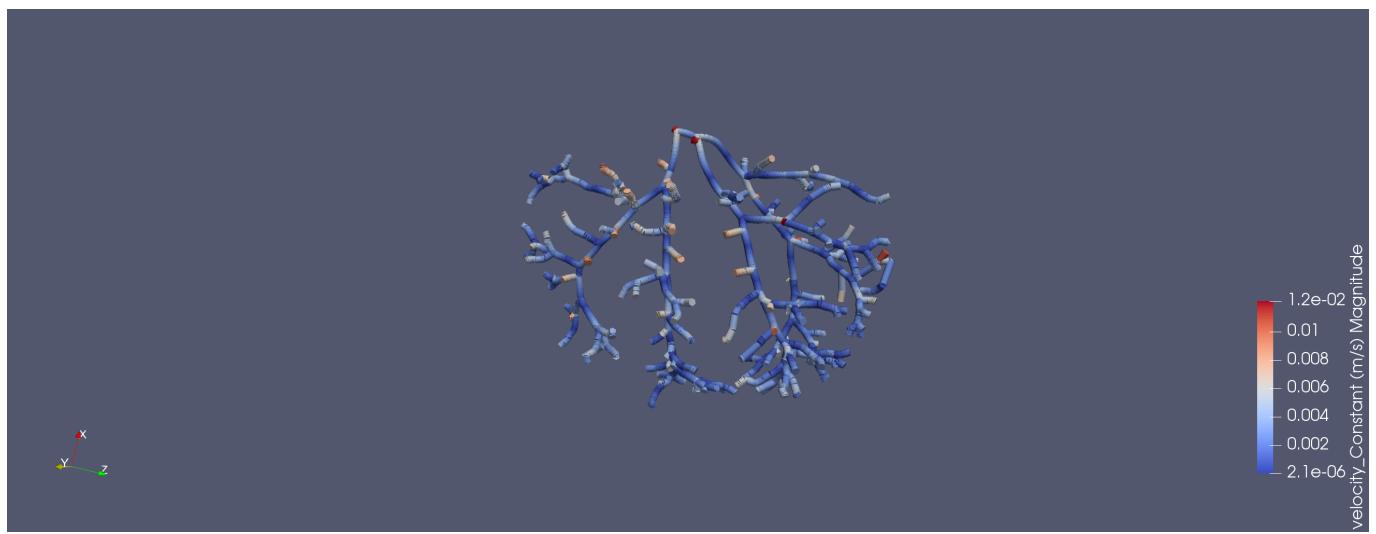


Figure 8: *DuMu^x* Velocity Computations for Nord Network

Some Nephron Network Simulations

Figure 12 is the result of a *DuMu^x* blood flow simulation computed on a Nephron network. The pressure field in the network is visualized.

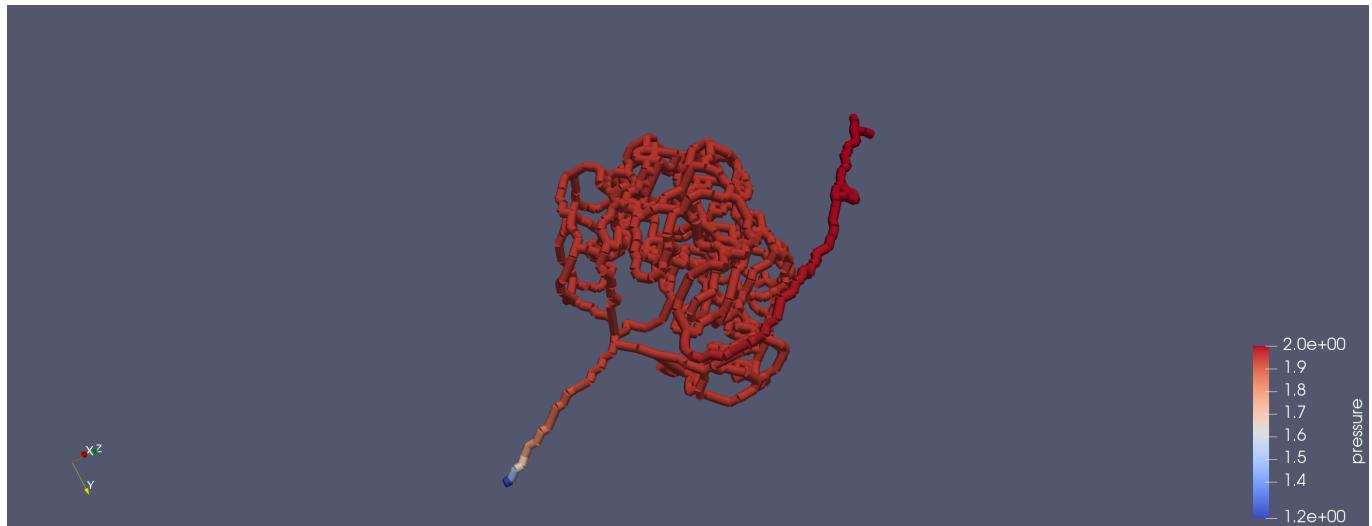


Figure 9: *DuMu^x* Pressure Computations for Nephron Network

Figure 11 is the result of a *DuMu^x* blood flow simulation computed on a Nephron network. The velocity field in the network is visualized.

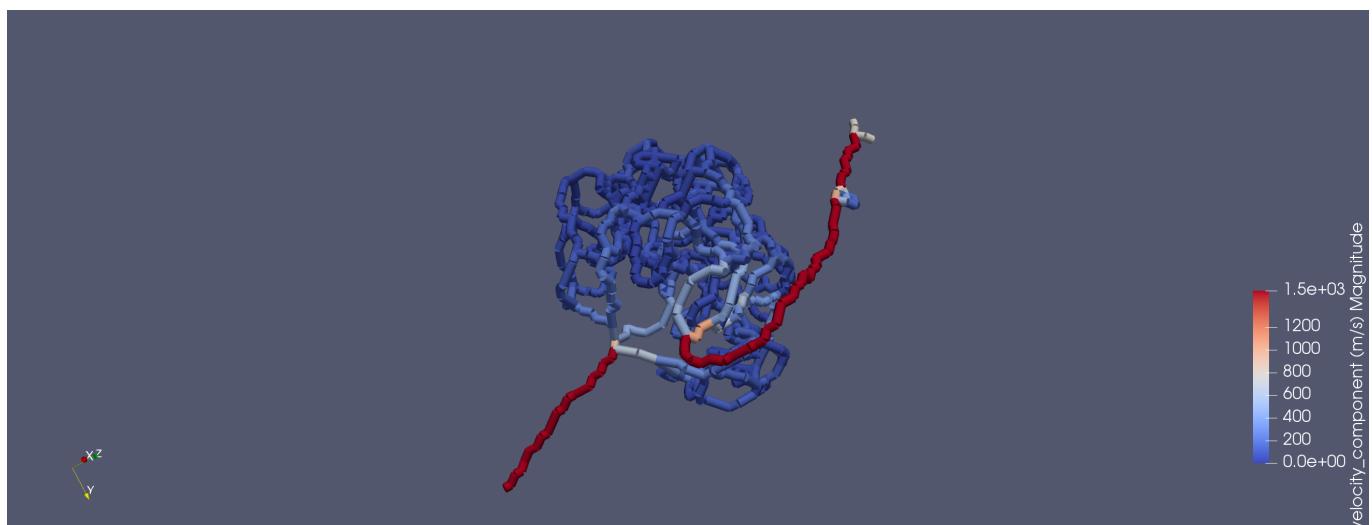


Figure 10: *DuMu^x* Velocity Computations for Nephron Network

4 Comparison of the Different Methods Behind the Simulations

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

4.1 Comparison of the Results

In this section the main differences in the produced results for the same input files/same networks will be discussed.

4.2 Comparison of the Numerical Methods

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.

4.2.1 Explanation of the Methods

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

4.2.2 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.

5 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.

6 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] D. Kuzmin. A guide to numerical methods for transport equations. *University Erlangen-Nuremberg*, 2010.
- [9] 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.
- [10] 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.
- [11] 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.
- [12] 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.

- [13] 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.
- [14] R. Weiss. Parameter-free iterative linear solvers. *Mathematical Research*, 97, 1996.

A Appendix

Additional material such as extra figures.

A.1 DuMuX Results Visualized with ParaView

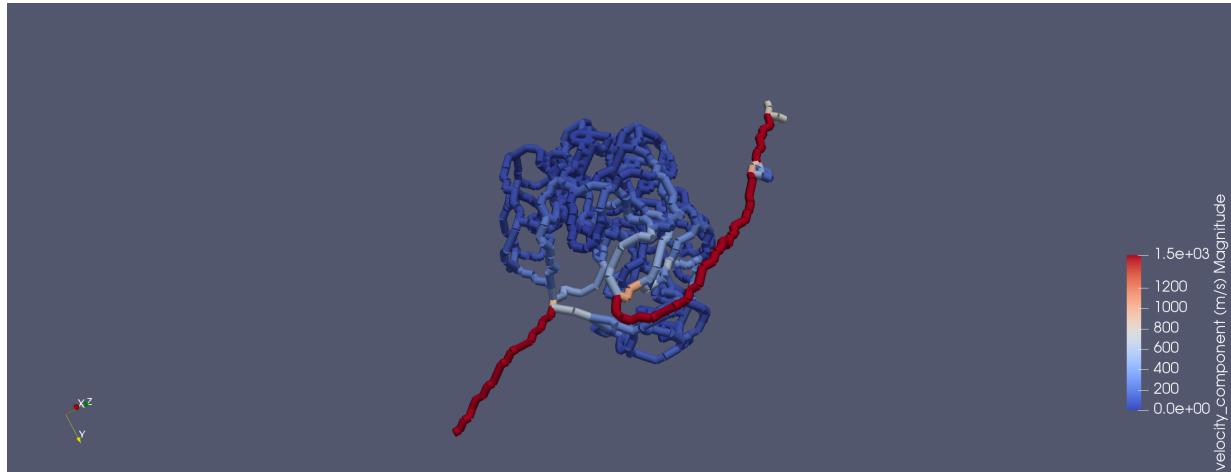


Figure 11: Velocity Field in a Nephron

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

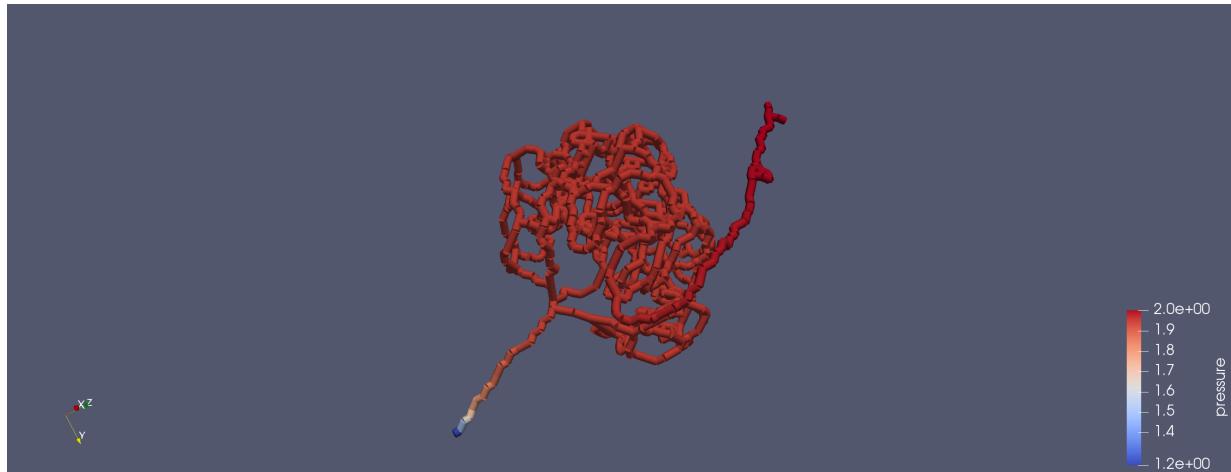


Figure 12: Pressure Field in a Nephron

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

A.2 Green's Function Code Results

Figure 13 is the result of a Green's Method blood flow simulation computed on a Cardiac network. The Solute Concentration in the network is visualized.

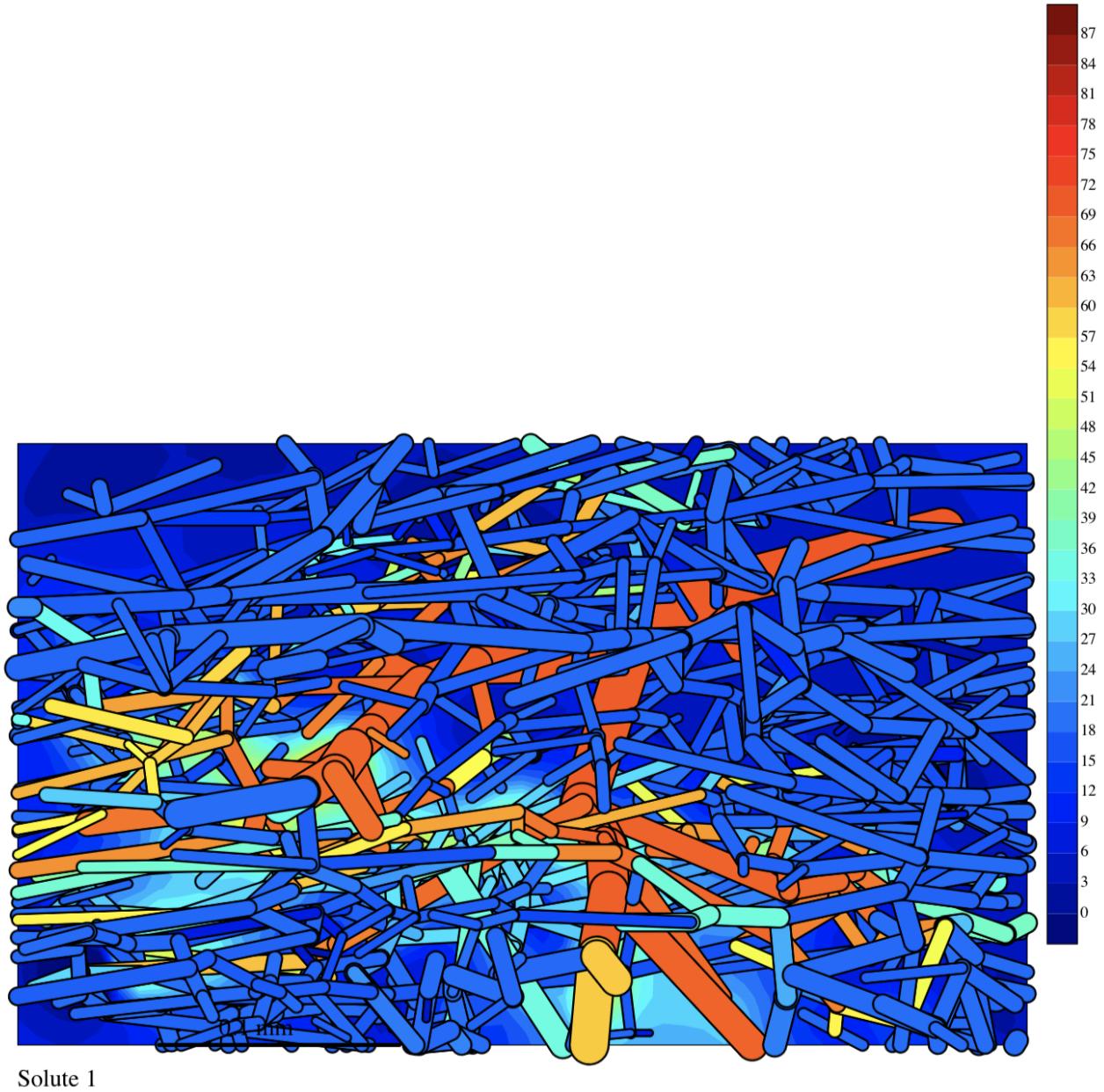


Figure 13: Green's Contour Output for Cardiac Network