



Universiteit
Leiden
The Netherlands

Analysing a model for drug transport and binding in the brain

Hackmann, T.

Citation

Hackmann, T. *Analysing a model for drug transport and binding in the brain.*

Version: Not Applicable (or Unknown)

License: [License to inclusion and publication of a Bachelor or Master thesis in the Leiden University Student Repository](#)

Downloaded from: <https://hdl.handle.net/1887/4171191>

Note: To cite this publication please use the final published version (if applicable).

T. Hackmann

**Analysing a model for drug transport and binding in
the brain**

Bachelor thesis

December 8, 2021

**Thesis supervisors: prof. dr. V. Rottschäfer
dr. ir. B. van 't Hof**



**Leiden University
Mathematical Institute**

Abstract

Drug development for targets in the brain is a difficult research topic. To improve quantitative understanding of drug pharmacokinetics (PK) in the brain, a mathematical model for 3D brain units was established by Vendel, Rottschäfer and De Lange [1][2][3]. CapSys is a software program that has translated this mathematical model into a computational model. In this thesis the quality of the simulation in CapSys will be investigated by comparing it to exact solutions of the differential equations it simulates. The focus of this investigation will be on the diffusion equation in different domains within the software. CapSys correctly simulates the diffusion equation in a cubic domain. The simulation converges to the exact solution at the expected rate with decreasing time steps and grid size. Further qualitative analysis does show some issues appearing with rounded shapes in the domain, such as cells.

Contents

1	Introduction	5
2	Physiology and pharmacology of the brain	5
2.1	Physiology of the brain	6
2.2	Pharmacokinetics of the brain	7
2.2.1	Absorption	8
2.2.2	Distribution	8
2.2.3	Metabolism	10
2.2.4	Excretion	10
2.3	Protein binding of drug	10
3	Mathematical model of the brain	11
3.1	Assumptions	11
3.2	The 3D brain unit	12
3.3	Mathematical description of the drug distribution in the ISF	13
3.4	Brain unit network	14
3.5	Domains for this thesis	14
3.5.1	Cubic domain	14
3.5.2	Spherical domain	15
3.5.3	Cylindrical domain	15
4	Solutions to the diffusion equation	16
4.1	Diffusion equation in a cubic domain	16
4.2	Diffusion equation in a spherical domain	18
4.3	Diffusion equation in a cylindrical domain	22
5	Theoretical results	24
5.1	Local Truncation Error	24
5.2	Rounding Error	25
5.3	Total Error	25
5.3.1	Cubic domain	26
5.3.2	Spherical domain	27
5.3.3	Cylindrical domain	28
6	CapSys results	29
6.1	Diffusion within the cubic domain	29
6.1.1	Qualitative results	29
6.1.2	Quantitative results	30
6.2	Diffusion within the spherical domain	32
6.2.1	Qualitative results	33
6.2.2	Quantitative results	35
6.3	Diffusion within the cylindrical domain	37
6.3.1	Qualitative results	39
6.3.2	Quantitative results	39
7	Discussion	41

Bibliography	43
A Solution to spherical Bessel functions	44
B Solution to cylindrical Bessel functions with no inflow	44

1 Introduction

Two of the most important fields of study in pharmaceutical sciences are the fields of pharmacokinetics and pharmacodynamics. Pharmacokinetics describes what the body does to a drug, including the uptake, transportation and metabolism of the drug. Pharmacodynamics describes what the drug does to the body, including the desired effect at the target site, but also undesirable effects. Together, pharmacokinetics (PK) and pharmacodynamics (PD) can model the concentration of a drug at different areas in the body over time and describe the effect on the body at that dose. This methodology is called pharmacokinetic/pharmacodynamic (PK/PD) modelling. An often expressed adage in the field of toxicology is "*The dose makes the poison*", a quote often credited to Paracelsus [4]. PK/PD modelling allows us to optimize the dosage, such that a drug does not become a poison.

The modelling of pharmacological phenomena has led to the creation of the field of mathematical pharmacology. By combining mathematical models with pharmacological research data, we can get more insight into why something does or doesn't work. This increased understanding can then be experimentally tested in a pharmacological setting.

A model is a simplified description or depiction of a more complex underlying topic. Well known pharmacological models include the well known images of signal transduction cascades in cells. PK/PD modelling is an example of a mathematical model, where reality is simplified and translated into mathematical language. The advantage of describing concepts in a mathematical model, is that it can lead to a better understanding of these concepts. Mathematical models show us the relation between different variables very clearly in equations. By allowing us to easily change parameters, models can show us the outcome of changing a certain parameter at any point in time.

Once a mathematical model has been established and shown to have good predictive capabilities by comparing it to experimental results, the model can become a useful tool in itself. It can predict the uptake and transport speed to the target organ of a certain drug, if the required properties are known. The model can predict the concentration of a drug over time in places where it cannot be measured, such as the brain. A good mathematical model can even lead to *in silico* - in computer simulation - drug research, where computers run mathematical models to aid in drug discovery. This type of research has the potential to reduce the need for *in vivo* - in living organisms - research on animals and humans.

In this thesis, I will look at CapSys, a software program containing a mathematical model for drug transport in the brain. CapSys will be analyzed based on its implementation of diffusion of drug through the brain interstitial fluid. To get there, first we need to understand the physiology and pharmacology that the mathematical model is based on. Then, the mathematical model itself will be laid out. After that we will solve the diffusion equation in the mathematical model for several types of domains. This will give exact solutions to those equations that we can compare the simulations from CapSys to. Fourth will be the theoretical background behind the expected difference between the exact solution and our simulation. Then we will look at the actual results from the simulations in CapSys. Finally, we will discuss the implications of the findings.

2 Physiology and pharmacology of the brain

In order to give a good mathematical description of drug distribution mechanisms within the brain, an understanding of both the physiology and pharmacology of the brain is required.

Physiology describes how living organisms function. It covers the whole range of function, from the function of a single protein in a cell to function of multiple organs working in conjunction in the human body [5]. Central to the study of physiology is the concept of homeostasis. Homeostasis is a

steady range of conditions that a living system attempts to maintain. When a function or condition fluctuates too far from the homeostatic range, a healthy body has mechanisms that can restore that equilibrium.

Pharmacology is the study of the effects of drugs on the function, or physiology, of living organisms. This broad term often captures the fields of pharmacokinetics and pharmacodynamics mentioned previously. Pharmacokinetics describes the uptake, transportation and metabolism of drugs in the body. Many facets of pharmacokinetics can vary between organs, because of physiological differences between them. An important example of a physiological element in the brain that has a large impact on pharmacokinetics is the blood-brain barrier (BBB). This barrier means that brain capillaries have no pores in them, resulting in many drugs that can't penetrate it [6]. Physiology and pharmacokinetics are important to understand in tandem, as illustrated by the following example:

The BBB, an element of the physiology of the brain, normally makes the brain inaccessible to the antibiotic penicillin. This might make it seem like this would rule out penicillin as a treatment for meningitis. However, meningitis causes inflammation, which disrupts the homeostasis of the brain. This disruption in homeostasis causes the physiology of the BBB to change. After these physiological changes, penicillin is able to penetrate the BBB, making it an effective treatment.

2.1 Physiology of the brain

The brain is a vital and vulnerable organ. As such it is protected by mechanisms from forces outside the body and from toxins inside the body. While these layers of protection are all incredibly important for our health, they are obstacles for drug targeting.

Protection from the outside mostly has to cover concussive head trauma. The first layer of protection is the skull, making it incredibly hard to administer drugs directly to the brain. Beneath the skull, there are three layers of membranes called meninges. From the skull inwards we find the *dura mater* covering the inside of the skull, the *arachnoid mater* in the middle and then the *pia mater* around the brain tissue. Between the *dura mater* and *arachnoid mater* the veins are located, draining the blood back to the heart. Below that, the subarachnoid space is found between the *arachnoid mater* and *pia mater*. This space contains a layer of cerebrospinal fluid (CSF) that helps absorb shocks. Because the brain tissue is soft and delicate, this is important to reduce trauma.

Located centrally in the brain is the choroid plexus, an epithelial structure that produces CSF. From the choroid plexus the CSF flows down past the brain stem, where it flows into the subarachnoid space. It drains at the top of the cranium into veins. The choroid plexus produces enough CSF to allow for the CSF to fully renew three times a day [5]. Flow of CSF from the choroid plexus through the brain to the outflow in the cranium can be seen in figure 1.

Protection of the brain from toxins is mainly achieved through the BBB. The BBB is a collection of cells that line the capillary blood vessels in the brain. It consists of both physical barriers, such as tight junctions, as well as physiological structures, such as active transport systems [5]. These junctions almost eliminate passive paracellular (through spaces between cells) diffusion through pores in the capillaries in favour of active transport or passive transport through cells. Passive transcellular (through the cells) transport is possible for lipid-soluble molecules. For water-soluble molecules that are important in the brain, such as glucose, there are active transport solutions. This means that the interstitial fluid (ISF) in the brain is quite different from the blood plasma. The interstitial systems, including the ISF make up an estimated 15-20% of total brain volume [8]. The differences between brain capillaries with the BBB and regular capillaries are illustrated in figure 2.

Besides the BBB there are also barriers between the blood and choroid plexus. In addition to those barriers, the choroid plexus also acts as a sort of barrier in the production of the CSF, not letting unwanted molecules through. This is important, since the CSF and ISF are connected and exist in a

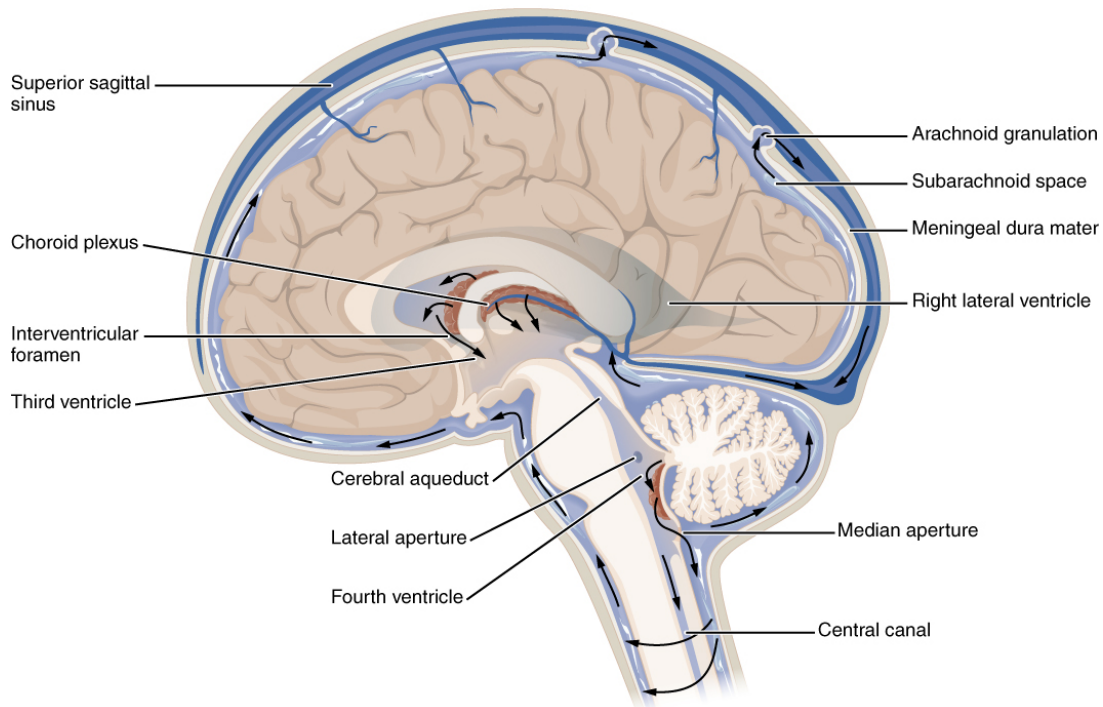


Figure 1: Cross section of the brain showing the production and flow of CSF and all three meninges. The CSF is produced in the choroid plexus in the center before flowing through the ventricles to the brain stem, where it flows into the subarachnoid space between the dura mater and arachnoid mater [7].

diffusion equilibrium.

The brain ISF is supplied by the blood flow. Part of the drainage of the ISF comes directly from drainage by the veins. Another major drainage source is the ventricular CSF. It has been estimated that up to 20% of CSF could be drained ISF [8]. The third drainage path is through cervical lymph nodes (in the neck) [9]. Combined, these methods of drainage cause advection (bulk flow) in the ISF towards the regions of drainage.

Finally, there are the neurons that make up the majority of the brain. These act as physical barriers for molecules in the ISF that are moving through diffusion or advection. Chemically, the neurons are also nearly always active. This is most apparent at the synapses, where a signal is sent from one neuron to the next by excretion of neurotransmitters into the ISF in the synaptic cleft. In these clefts there are also many receptor proteins, making them essential targets for many drugs.

2.2 Pharmacokinetics of the brain

Drugs are delivered to the brain by the blood flow. To understand the drug delivery in the brain from the capillaries onward, as the subject model of this thesis attempts, we need to understand all the steps in the pharmacokinetics of the brain. Most models describing the pharmacokinetics of a drug use the ADME model[10]:

1. Absorption
2. Distribution
3. Metabolism
4. Excretion

We will describe each of these processes separately in detail as is relevant to the pharmacokinetics of the brain.

2.2.1 Absorption

Absorption describes all the steps leading to the drug entering the blood circulation in the body. The first important step here is the drug administration. There are several methods of administering a drug:

- Oral,
- Rectal,
- Intravenous,
- and more.

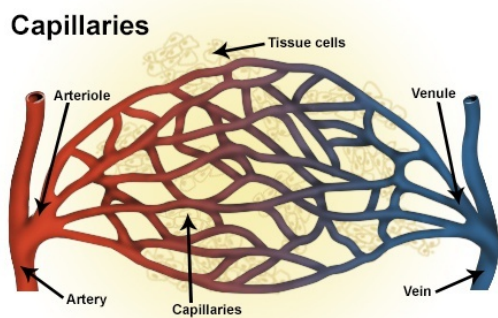
Each of these administration methods can have entire chapters dedicated to their intricacies, so the focus here will mostly be on oral - or gastrointestinal - administration. This is the most common method and features many obstacles that are shared by other administration methods in some way. Generally, the goal of studying absorption is to increase the bioavailability of the drug. The bioavailability is the fraction of administered drug that reaches the systemic circulation. Systemic circulation is the section of blood circulation that supplies all organs with blood, the circulation that leaves the heart through the aorta.

Oral administration of drugs leads to absorption mostly in the small intestine. Uptake in the stomach rarely happens due to the acidic environment and the small intestine is simply the largest of the organs in the gastrointestinal tract in terms of surface area. There are several mechanisms that impact the bioavailability of drugs. First there is metabolism in the gastrointestinal lumen by enzymes. Second, there is the uptake by the gut wall, which prefers lipophilic (soluble in fat) drugs. If a drug does not go through the gut wall it is excreted before reaching circulation, reducing bioavailability. While the gut wall is somewhat selective in favour of lipophilic drugs, it is not nearly as selective as the BBB. In the gut wall, there is another set of enzymes that can metabolise the drug, lowering bioavailability further. After passing through the gut wall, the drug is in the bloodstream, but not yet in the systemic circulation. Blood coming from the gastrointestinal tract does not directly flow into the heart, but flows into the portal vein towards the liver. The liver is where most drugs are eventually metabolized before excretion. For drugs that are administered gastrointestinally, they have a chance to be metabolized by the liver before reaching the circulation. This is called the first-pass effect and has a large impact on bioavailability. A recent example is the Covid-19 drug remdesivir, which cannot be administered orally, because the first pass effect would eliminate nearly all of the drug [11].

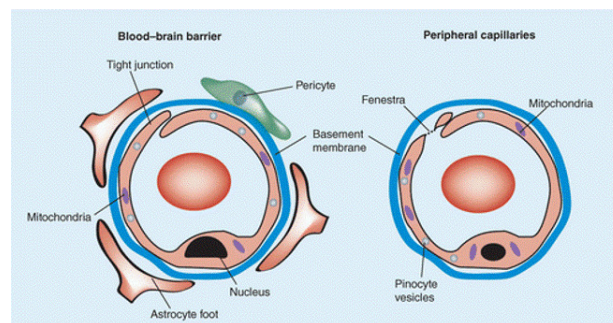
2.2.2 Distribution

After absorption, the drug has entered the circulation. After reaching the heart it is then spread throughout the body by the arteries. Within the arteries, the drug moves with the advection of blood and, in much smaller part, by diffusion within the blood. Since blood flows much faster than the rate of diffusion, this second effect can largely be ignored.

After entering the brain through the brain artery, the artery splits into arterioles, which in turn split into capillaries. These capillary blood vessels transport the drug throughout the brain. The blood vessels are separated from the interstitial fluid (ISF) of the brain by the blood-brain barrier (BBB).



(a) A two dimensional representation of a capillary net structure. The blood flows in from the arteries into the arterioles and into the capillary net from there. As the capillary net passes through the tissue, oxygen is being used by the cells and CO_2 is moved into the capillaries. The CO_2 rich blood then flows out of the capillary net through the venule into the veins. Image from NIH [12].



(b) Zoom-in on a cross section of a capillary, with a capillary surrounded by the blood-brain barrier on the left and a regular capillary on the right. The most important distinctions are the difference between the fenestra (pore) in the peripheral capillary compared to the tight junctions that are present in the BBB. The astrocyte feet in the BBB also give more structure to the BBB. Image from an article in Ther Deliv [13].

Figure 2: Stylised image of a capillary net structure and an image showing the difference between general capillaries and those with the BBB surrounding them.

In figure 2a we can see a 2D representation of how a capillary net looks. In the gaps between blood vessels we can find the ISF with the cells embedded there. From these capillaries, drug can move from the capillaries by active and passive transport.

Passive transport is mostly achieved by diffusion. The rate of diffusion is affected by the environment in which the diffusion takes place and the properties of the drug molecule. On the environment side the important factors are the permeability of the membrane, the surface area for transport and the difference in concentration on either side of the membrane. Specific to the brain is a very low permeability, since there are no pores in the capillaries due to the BBB. This means that there is almost no paracellular (in between the cells) transportation, even for smaller molecules [14],[15]. Most transportation through the barrier is through channels, pumps and transcellular receptor mediated transportation (RMT) [14]. Smaller lipophilic molecules are passively transported through the cell membranes by passive diffusion.

On the drug side, there are three important properties that govern transport. First we have the size of the molecule, smaller molecules are easier to transport. Second there is the lipophilicity of the molecule, how well it dissolves in fat. Finally the charge, or ionization, of the molecule is important. For the BBB specifically, the lipophilicity is the most important factor among these three. Most transport has to go through cells and this mode heavily favour lipophilic compounds [10]. The other factors do still matter, since even very lipophilic drugs are hardly able to penetrate the BBB if they become too large.

After transportation through the BBB, the drug has reached the brain ISF. In the ISF the main method of transportation is diffusion from the capillaries to the brain cells. There is also some advection in the ISF towards the ventricular CSF as mentioned previously. Both the diffusion and advection rates are affected by the presence of physical cells, that block a lot of the movement. In many cases these cells are also the target of the drugs.

2.2.3 Metabolism

Most of the metabolism of drugs takes place in the liver. After reaching the circulation, a portion of the drug goes through the liver with each pass through the body. This results in the drug concentration in the circulation decreasing over time. With this decreased concentration comes decreased diffusion to the target site and reduced effect. Different compounds have different rates of metabolism. In designing medication, taking the rate of metabolism in mind in creating the drug and the administration schedule is of great importance.

2.2.4 Excretion

Drug excretion, or elimination, can take place before or after the compound is metabolized. The most common path for drug excretion by far is renal excretion, excretion through the urinary tract. Besides that path, almost any part of the body that excretes liquids can cause excretion of active or metabolized drugs. Other methods of excretion include through tears, sweat, milk or faeces. Taking excretion into consideration is important in understanding the concentration of drug in the circulation over time.

2.3 Protein binding of drug

Proteins are large molecules in organisms, that perform a variety of functions. They are involved in metabolic processes, DNA replication, transportation of molecules and response to stimuli among other functions. Many of these functions require proteins to bind other molecules. Since proteins are everywhere in the body, this binding needs to be taken into account in most pharmacokinetic models. Receptors, a category of proteins, are the target for many drugs.

There are two important categories in pharmacokinetics for protein binding: specific and non-specific protein binding. Specific protein binding is binding to the intended target protein. Non-specific protein binding is all other binding to proteins that aren't its intended target.

Non-specific binding can occur almost anywhere in the body. After absorption, the first place to take this in consideration are the blood vessels. This non-specific binding is generally reversible and dissociation is generally rapid [10]. This means that bound and unbound drug are almost always in equilibrium in the plasma. Protein-bound drug becomes too large a complex to diffuse through cell membranes. This means that in our situation of diffusion through the BBB, protein-bound drug is essentially removed from the circulation [10]. As mentioned in section 2.2.2, large molecules such as these proteins cannot penetrate the BBB. Therefore there is very little plasma protein binding in the brain ISF [16].

In the brain there are many receptors on the neurons. For drugs targeting these receptors, this of course causes the desired specific binding. However there is also some non-specific receptor binding taking place. As with the plasma protein binding, specific and non-specific receptor binding is generally reversible and over time reaches an equilibrium. With the reduced diffusion into the brain ISF due to the BBB, non-specific binding can slow the rate of specific binding in the brain even more. This makes it relevant to take into account in a model.

3 Mathematical model of the brain

Drug delivery in the brain is a difficult subject to study. Brain physiology is an important part of drug distribution within the brain, but this cannot easily be tested in a laboratory setting. In order to enhance our understanding of the processes described in chapter 2, a mathematical model of the brain was constructed. The model will not cover the metabolism (section 2.2.3) and excretion (section 2.2.4). This model was developed by Vendel, Rottschäfer and De Lange [1],[2],[3]. The base part that was developed is the 3D brain unit. This unit is a small section of brain tissue surrounded by capillary blood vessels. This 3D brain unit is made to contain all physiologically relevant processes that take place in the brain.

First, the assumptions made in the creation of the model will be laid out. Then the 3D brain unit will be described. This will be followed by a mathematical description of the 3D brain unit. After this it will be described how we can simulate larger sections of the brain with a 3D brain unit network. Finally, the different domains that are studied in this thesis will be described.

3.1 Assumptions

As mentioned, any model relies on assumptions and it is important to state those explicitly. We will start with the assumptions about the capillaries:

- a) Blood flows at a constant speed.
- b) Blood can only enter the system through the arteriole and only leave through the venule (there are no other capillaries that it can enter or exit through)
- c) Drug concentration in the blood that flows into the brain unit can be described as a function of time
- d) Diffusion of drug through the blood plasma is a negligible form of transportation in the capillaries and will be left out
- e) Drugs are not bound to proteins in the blood plasma and are all available

Assumption c) is there to exclude absorption, metabolism and excretion from the model. Those factors can be taken into consideration when constructing such a function of time. Assumption e) contradicts the points mentioned in section 2.3. This is covered by constructing the concentration curve of the blood entering the arteriole with the volume of distribution of the drug in mind, since plasma protein binding is part of that volume.

For the ISF, we have a separate set of assumptions:

- a) Drug moves through the ISF by diffusion and advection
- b) The advection is uni-directional in the x-direction
- c) Drug-receptor binding happens on the membrane of cells, drugs don't have to cross a membrane to bind to receptors
- d) The total number of specific and non-specific binding sites remains constant
- e) Drug-receptor binding is reversible

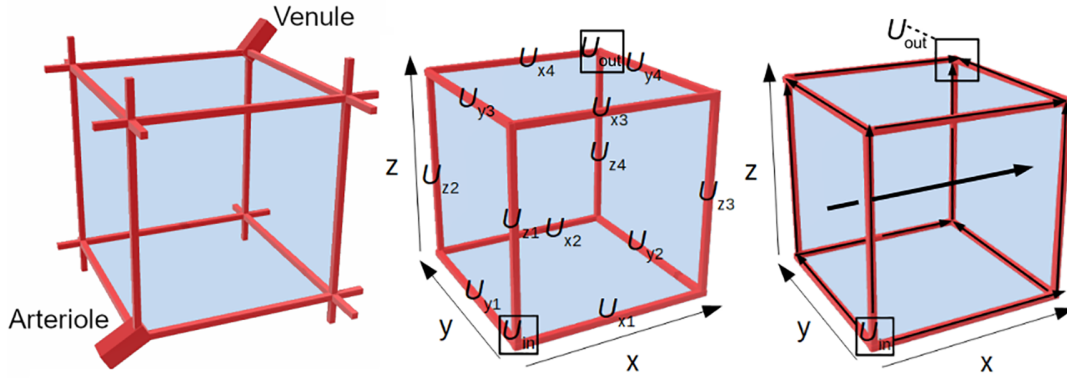


Figure 3: Left: The structure represented by the 3D brain unit. An arteriole carries blood plasma (containing drug) into a brain capillary bed, that is connected to a venule that drains the blood plasma. The brain capillaries (red) surround the brain ISF (blue). Middle: the 3D brain unit and its sub-domains. The unit consists of a brain-ISF-domain (blue) and a blood-plasma-domain (red). The blood-plasma-domain is divided into several subdomains: U_{in} is the domain where the dose of absorbed drug enters the 3D brain unit, U_{x1-x4} , U_{y1-y4} and U_{z1-z4} are the domains representing the x-directed, y-directed and z-directed capillaries, respectively. Right: Directions of transport in the model. The drug enters the brain capillaries in U_{in} . From there, it is transported through the brain capillaries by the brain capillary blood flow in the direction indicated by the small arrows. Drug in the brain capillary blood plasma exchanges with the brain ISF by crossing the BBB. Drug within the brain ISF is, next to diffusion, transported along with brain ISF advection (indicated by the bold arrow). Image from the article by Vendel et al [2].

3.2 The 3D brain unit

In figure 3 (left) there is an overview of the structure of the 3D brain unit. In the lower left is the arteriole that supplies the blood to the 3D unit (U_{in}) and on the far upper corner is the venule that drains the blood from the system (U_{out}). These two blood vessels are connected by a network of capillaries that allow the blood to flow through our unit. To simplify the physical reality for mathematical modelling, the brain unit is represented by a cube. Instead of the complex net of capillaries that make up actual brains, as illustrated in figure 2a, the capillaries are located at the edges of the cube, represented by red rods in figure 3. At the vertices, three capillaries go into the vertex, with some flowing in and some flowing out of the vertex. At the vertices where the arteriole or venule is connected, only capillaries that flow out of and in to the vertex exist respectively. The arteriole and venule exist on opposite sides in all three spatial directions. This means that the blood must travel through a capillary in the x , y and z directions. Because there are four edges in each direction of the cube, there are four capillaries in each of the x , y and z directions, named $U_{x,1} - U_{x,4}$, $U_{y,1} - U_{y,4}$ and $U_{z,1} - U_{z,4}$ as shown in the middle of figure 3.

Between the capillaries the brain ISF and the neurons are found. The drug enters the ISF from the capillaries through passive and active transport, based on the concentrations of drug on either side of the barrier. In the ISF the drug is transported through the ISF by diffusion and advection. The advection is unidirectional in the x direction in the model, as shown on the right of figure 3. Cells are located within the ISF and the drug cannot flow through them by advection or diffusion. The model developed by Vendel, Rottschäfer and De Lange does not physically place cells in the ISF, but considers them through adding so-called tortuosity to the model. Tortuosity is a value that describes how much longer the path of a drug molecule is when compared to a straight path. This mimics the blocking effect that going around cells has on the distribution of the drug through the brain, as discussed in section 2.2.2.

3.3 Mathematical description of the drug distribution in the ISF

The concentration of the drug in the ISF changes over time and location based on the diffusion, advection and protein binding. Describing the dependency of the concentration on protein binding also requires equations for protein binding based on concentration. The partial differential equations that describes the changes in concentration over time are:

$$\frac{\partial C_{ISF}}{\partial t} = \frac{D}{\lambda^2} \Delta C_{ISF} - v_{ISF} \frac{\partial C_{ISF}}{\partial x},$$

$$-k_{1,on}(B_1^{max} - B_1)C_{ISF} + k_{1,off}B_1 - k_{2,on}(B_2^{max} - B_2)C_{ISF} + k_{2,off}B_2, \quad (3.1)$$

$$\frac{\partial B_1}{\partial t} = k_{1,on}(B_1^{max} - B_1)C_{ISF} - k_{1,off}B_1 \quad (3.2)$$

$$\frac{\partial B_2}{\partial t} = k_{2,on}(B_2^{max} - B_2)C_{ISF} - k_{2,off}B_2, \quad (3.3)$$

where:

$C_{ISF}(t, x, y, z)$: gives the free drug concentration at time t and point x, y, z in the ISF,

$B_{1/2}(t, x, y, z)$: the concentration of drug bound to specific (1) and non-specific (2) binding sites, as a function of time and location,

$\Delta C_{ISF} : (\frac{\partial^2 C_{ISF}}{\partial x^2} + \frac{\partial^2 C_{ISF}}{\partial y^2} + \frac{\partial^2 C_{ISF}}{\partial z^2})$, the Laplacian of the concentration,

D : diffusion coefficient,

λ : tortuosity, the effect of going around the cells,

v_{ISF} : advection flow speed,

$k_{1/2,on/off}$: associaton (on) and dissociation (off) rates,

for specific (1) and non-specific (2) binding sites,

$B_{1/2}^{max}$: total concentration of specific and non-specific binding sites.

The different parts of this system of equations will now be examined in more detail, starting with equation (3.1). This equation describes the change in concentration of free drug over time at a certain location within the ISF. On the right/hand side (r.h.s.) of (3.1) the first term, $\frac{D}{\lambda^2} \Delta C_{ISF}$, describes the diffusion of free drug over time. The constant term $D^* = \frac{D}{\lambda^2}$ is the effective diffusion constant and will from now on be abbreviated as such. The Laplacian ΔC_{ISF} describes how much free drug diffuses towards or away from a point, based on the unmixed second partial derivatives.

The second part of equation (3.1) is $-v_{ISF} \frac{\partial C_{ISF}}{\partial x}$. This describes the advection of the drug in the ISF. It depends on the flow speed (v_{ISF}) in the ISF and the derivative in the x -direction.

Next there is the second line of (3.1) that is linked with the two expressions (3.2), (3.3) that represent the changes in concentration due to protein binding as discussed in section 2.3. One is for specific binding (B_1), while the other is for non-specific binding (B_2). Mathematically for the purpose of modeling concentration in the ISF, the two binding types function the same. In many experimental

situations, the specific binding is often the desired outcome. The drug concentration in the ISF decreases with binding, which is governed by the association rate k_{on} . Dissociation, with rate k_{off} , from the proteins increases the concentration in the ISF. Binding is increased with higher concentrations in the ISF (C_{ISF} -term). There are finitely many binding sites, giving a maximum concentration of bounded drug of $B_{1/2}^{max}$ before all binding sites are occupied. As more binding sites are occupied the rate of association reduces with the $(B_{1/2}^{max} - B_{1/2})$ term. We can see that protein binding stops when there is no drug in the ISF ($C_{ISF} = 0$) or if all receptors are saturated ($B_{1/2} = B_{1/2}^{max}$). The rate of dissociation increases with the amount of drug bound to binding sites.

3.4 Brain unit network

The brain unit described above is limited in its representation of drug distribution within the brain due to its limited size and lack of interconnected capillaries [3]. By creating a network of the 3D brain units above, larger sections of brain with interconnected capillaries can be represented. These networks also allow for different brain units to be given different properties, such as a higher specific binding site concentration. This can mimic the fact that some targets can be harder to reach than others, as they can be further from the blood flowing in to the tissue through the arteriole.

The brain unit network consists of many connected brain units. Each brain unit, as described in section 3.2, is a cube with capillaries on the edges. Two connected brain units share one of the faces, where the ISF becomes one large area and the capillaries on the edges of the face are joined together. There are no barriers between the ISF of the different brain units and drug can flow freely between them by diffusion or advection.

3.5 Domains for this thesis

For this thesis, several domains were created in the CapSys software based on the model by Vendel, Rottschäfer and De Lange [2]. The domains that were used will be discussed in more details here.

3.5.1 Cubic domain

First a cubic domain is used to assess the implementation of the diffusion equation. The reason this domain is used, is that the discretization of the physical space works optimally with a cubic region. This way all grid areas are present in full and not cut-off, as is the case in other domain types, such as a sphere. To ensure there is no other interference of elements with the workings of the diffusion equation, the cells and capillaries are removed from the domain. With the removal of the cells, there is also no protein binding in this domain. This means that the domain consists of only ISF with some starting distribution for the drug concentration. The distance between the capillaries, if they would be included, is $50 \mu m$. The cube in between the capillaries has its lower bound placed on the zero-coordinates. This gives us the ranges for x , y and z from 0 to $50 \mu m$. While the capillaries are not included, the area where they would be located is included within the domain as ISF. This means that in all directions, the size of the domain is increased by two times the radius of the capillaries. With the radius of the capillaries set at $2.5 \mu m$, the total domain becomes:

$$D_{cube} = \{(x, y, z) : -5 \mu m \leq x, y, z \leq 55 \mu m\}. \quad (3.4)$$

3.5.2 Spherical domain

For the second domain, the goal was to simulate the diffusion equation in CapSys around a cell. A shape like a cell does not neatly fit in a discretized grid, leading to some grid spaces having parts inside and outside the cell. This led to the choice of a spherical domain with a single cell in the center. As with the cubic domain (3.5.1), there are no capillaries in the spherical domain and no regularly generated cells either. The radius of the sphere is half the distance between the capillaries, giving a radius of $25 \mu m$. The radius of the central cell is set to $4.5 \mu m$. This gives the following domain in polar coordinates:

$$D_{sphere} = \{(r, \theta, \varphi) : 4.5 \mu m \leq r \leq 25 \mu m, 0 \leq \theta \leq 2\pi, 0 \leq \varphi \leq \pi\}. \quad (3.5)$$

3.5.3 Cylindrical domain

Finally, the behaviour of the diffusion equation around a capillary is also of interest. The choice domain for investigating this, is the cylindrical domain with a single capillary running through the center of the cylinder. The cylindrical domain does not contain any cells. The outer radius of the cylinder is the same as the radius of the sphere from section 3.5.2 at $25 \mu m$. The length of the cylinder is set at the length of the sides from the cubic domain in section 3.5.1 from $-5 \mu m$ to $55 \mu m$ in the z -direction. The cylinder has a capillary with radius $2.5 \mu m$ running through the center. Together these aspects give the following domain in cylindrical coordinates:

$$D_{cylinder} = \{(r, \theta, z) : 2.5 \mu m \leq r \leq 25 \mu m, 0 \leq \theta \leq 2\pi, -5 \mu m \leq z \leq 55 \mu m\}. \quad (3.6)$$

4 Solutions to the diffusion equation

CapSys will be tested with the diffusion equation. Diffusion can be taxing for software and can often be difficult to model on the discretized space. It is more likely that we find issues with this equation than the protein binding or advection. For this reason, the assumptions $B_1 = B_2 = B_1^{max} = B_2^{max} = 0$ and $v_{ISF} = 0$ are made. This eliminates equations (3.2) and (3.3) from the system of equations (3.1)-(3.3) and gives:

$$\frac{\partial C_{ISF}}{\partial t} = D^* \Delta C_{ISF} = D^* \left(\frac{\partial^2 C_{ISF}}{\partial x^2} + \frac{\partial^2 C_{ISF}}{\partial y^2} + \frac{\partial^2 C_{ISF}}{\partial z^2} \right), \quad (4.1)$$

where $D^* = \frac{D}{\lambda^2}$.

4.1 Diffusion equation in a cubic domain

The cubic domain is as described in section 3.5.1. For the boundaries x_0 and x_1 it is assumed there is no transportation across them. This results in the boundary conditions:

$$\frac{\partial C_{ISF}}{\partial x}(t, x_0, y, z) = \frac{\partial C_{ISF}}{\partial x}(t, x_1, y, z) = 0 \text{ for } y_0 \leq y \leq y_1, z_0 \leq z \leq z_1, t > 0, \quad (4.2)$$

$$\frac{\partial C_{ISF}}{\partial y}(t, x, y_0, z) = \frac{\partial C_{ISF}}{\partial y}(t, x, y_1, z) = 0 \text{ for } x_0 \leq x \leq x_1, z_0 \leq z \leq z_1, t > 0, \quad (4.3)$$

$$\frac{\partial C_{ISF}}{\partial z}(t, x, y, z_0) = \frac{\partial C_{ISF}}{\partial z}(t, x, y, z_1) = 0 \text{ for } x_0 \leq x \leq x_1, y_0 \leq y \leq y_1, t > 0. \quad (4.4)$$

To increase clarity in the software simulation, we will look for a solution that is constant in the y and z directions. This means that the partial derivatives in the y - and z -direction are both equal to 0 on the entire domain. The initial condition at $t = 0$ will also be independent of y and z . This will be:

$$C_{ISF}(0, x, y, z) = C_{initial}(x) \text{ for } x_0 \leq x \leq x_1, y_0 \leq y \leq y_1, z_0 \leq z \leq z_1, \quad (4.5)$$

where $C_{initial}(x)$ will be determined later.

There are no changes in the coordinate system necessary, so we can use equation (4.1) as is. This equation is linear, which means that the method of separation of variables can be used to write the function of concentration as follows:

$$C_{ISF}(t, x, y, z) = H(t)G(x)K(y)L(z). \quad (4.6)$$

Substituting this into the partial derivatives in equation (4.1) turns those partial derivatives into the following derivatives:

$$\begin{aligned} \frac{\partial C_{ISF}}{\partial t} &= G(x)K(y)L(z) \frac{dH}{dt}, \\ \frac{\partial^2 C_{ISF}}{\partial x^2} &= H(t)K(y)L(z) \frac{d^2 G}{dx^2}, \\ \frac{\partial^2 C_{ISF}}{\partial y^2} &= H(t)G(x)L(z) \frac{d^2 K}{dy^2}, \\ \frac{\partial^2 C_{ISF}}{\partial z^2} &= H(t)G(x)K(y) \frac{d^2 L}{dz^2}. \end{aligned}$$

Now we can substitute these derivatives into equation (4.1) and divide both sides by $H(t)G(x)K(y)L(z)$:

$$\frac{1}{H(t)} \frac{dH}{dt} = D^* \left(\frac{1}{G(x)} \frac{d^2G}{dx^2} + \frac{1}{K(y)} \frac{d^2K}{dy^2} + \frac{1}{L(z)} \frac{d^2L}{dz^2} \right). \quad (4.7)$$

When we apply the condition that the solution is constant in the y and z directions we find that the second derivatives in those directions must also be zero. This simplifies expression (4.7) to:

$$\frac{1}{H(t)} \frac{dH}{dt} = \frac{D^*}{G(x)} \frac{d^2G}{dx^2}. \quad (4.8)$$

This results in an equality between expressions of different variables on the different sides. This means that both sides must be equal to a constant. Let both the right-hand side (r.h.s.) and left-hand side (l.h.s.) equal a constant $-\kappa$. From the l.h.s. we then obtain:

$$\frac{1}{H(t)} \frac{dH}{dt} = -\kappa,$$

which implies that,

$$H(t) = c_1 e^{-\kappa t}. \quad (4.9)$$

Here we find that $H(t)$ is bounded if $\kappa \geq 0$.

The r.h.s. can be solved in a similar way. There we have:

$$\frac{D^*}{G(x)} \frac{d^2G}{dx^2} = -\kappa, \quad (4.10)$$

$$D^* \frac{d^2G}{dx^2} + \kappa G(x) = 0. \quad (4.11)$$

This is a second order differential equation, which we can solve by finding the roots to the characteristic polynomial of (4.11):

$$D^* r^2 + \kappa = 0. \quad (4.12)$$

The solutions to (4.12) are $r = \pm \frac{\sqrt{-\kappa}}{\sqrt{D^*}}$, with $\kappa > 0$. This means both solutions of (4.12) are complex. From this we obtain the solution for G :

$$G(x) = c_2 \cos\left(\sqrt{\frac{\kappa}{D^*}}x\right) + c_3 \sin\left(\sqrt{\frac{\kappa}{D^*}}x\right). \quad (4.13)$$

Substituting expressions (4.9) and (4.13) into (4.6) gives a solution:

$$C_{ISF}(t, x, y, z) = c_1 e^{-\kappa t} (c_2 \cos\left(\sqrt{\frac{\kappa}{D^*}}x\right) + c_3 \sin\left(\sqrt{\frac{\kappa}{D^*}}x\right)) \quad (4.14)$$

To apply the boundary conditions from (4.2) to $G(x)$ (4.13), we determine the derivative:

$$G'(x) = -c_2 \sqrt{\frac{\kappa_x}{D^*}} \sin\left(\sqrt{\frac{\kappa_x}{D^*}}x\right) + c_3 \sqrt{\frac{\kappa_x}{D^*}} \cos\left(\sqrt{\frac{\kappa_x}{D^*}}x\right). \quad (4.15)$$

The boundary conditions in (4.2) give $G'(x_0) = G'(x_1) = 0$. And since there is no x such that $\sin(\sqrt{\frac{\kappa_x}{D^*}}x) = \cos(\sqrt{\frac{\kappa_x}{D^*}}x) = 0$, at least one of c_2 or c_3 must be 0. It will assumed that $c_3 = 0$, which means that:

$$\sin(\sqrt{\frac{\kappa_x}{D^*}}x_0) = 0 \implies \sqrt{\frac{\kappa_x}{D^*}}x_0 = 0 + k_0\pi, \quad (4.16)$$

$$\sin(\sqrt{\frac{\kappa_x}{D^*}}x_1) = 0 \implies \sqrt{\frac{\kappa_x}{D^*}}x_1 = 0 + k_1\pi, \quad (4.17)$$

with $k_0, k_1 \in \mathbb{Z}$.

We set $k_0 = 0$ and $k_1 = k \in \mathbb{Z}$. To ensure $\sqrt{\frac{\kappa_x}{D^*}}x_0 = 0$ where $x_0 \neq 0$, we translate the function by x_0 to get:

$$G'(x) = c_2 \sqrt{\frac{\kappa_x}{D^*}} \sin(\sqrt{\frac{\kappa_x}{D^*}}(x - x_0)), \quad (4.18)$$

which now equals 0 at x_0 . We are left with the second boundary condition which we can solve:

$$\sqrt{\frac{\kappa_x}{D^*}}(x_1 - x_0) = k\pi, \quad (4.19)$$

$$\sqrt{\frac{\kappa_x}{D^*}} = \frac{k\pi}{x_1 - x_0}. \quad (4.20)$$

Applying these findings at the boundary to (4.13) gives the solution:

$$G(x) = c_2 \cos\left(\frac{x - x_0}{x_1 - x_0} k\pi\right). \quad (4.21)$$

The product (4.21) and (4.9) gives the function C_{ISF} by reversing the separation of variables:

$$C_{ISF}(t, x) = H(t)G(x) = c_1 c_2 e^{-D^* \left(\frac{k\pi}{x_1 - x_0}\right)^2 t} \cos\left(\frac{x - x_0}{x_1 - x_0} k\pi\right). \quad (4.22)$$

The constants that are still in this equation can now be chosen. For the best resolution in the simulation we choose $k = 1$, since that gives us a half-period of the cosine function. This also means that the maximum of the function is at x_0 . We see that $C_{ISF}(0, x_0) = c_1 c_2$, is the maximum value. We set $c_1 c_2 = 1$.

The value for the concentration needs to be non-negative, since negative concentrations don't exist. The minimum concentration can be found at $t = 0$ and $x = x_1$, where $C_{ISF}(0, x_1) = -1$. This means that to ensure that our initial solution is non-negative, 1 needs to be added everywhere. This gives the initial function for the simulation and the exact solution:

$$C_{ISF}(0, x, y, z) = \cos\left(\frac{x - x_0}{x_1 - x_0} \pi\right) + 1, \quad (4.23)$$

$$C_{ISF}(t, x, y, z) = e^{-D^* \left(\frac{\pi}{x_1 - x_0}\right)^2 t} \cos\left(\frac{x - x_0}{x_1 - x_0} \pi\right) + 1. \quad (4.24)$$

4.2 Diffusion equation in a spherical domain

The spherical domain that will be used here is a generalized version of the domain laid out in section 3.5.2. To analyze the spherical domain, the Cartesian coordinates (x, y, z) are transformed to

spherical coordinates (r, θ, φ) . In spherical coordinates the only boundary of the domain is based on the radius of the sphere. In the domain we want to study, the outer boundary is the edge of the domain at r_1 and the inner boundary is the central cells, with radius r_0 . There is no flow across these boundaries This gives the following boundary conditions:

$$\frac{\partial C_{ISF}}{\partial r}(t, r_0, \theta, \varphi) = \frac{\partial C_{ISF}}{\partial r}(t, r_1, \theta, \varphi) = 0, \quad t \geq 0, \quad 0 \leq \theta \leq 2\pi, \quad 0 \leq \varphi \leq \pi. \quad (4.25)$$

$$(4.26)$$

For the spherical domain it is useful to look for a solution that is symmetric around the center, as this will increase clarity in the simulation. This adds the symmetry conditions:

$$\frac{\partial C_{ISF}}{\partial \theta}(t, r, \theta, \varphi) = 0, \quad (4.27)$$

$$\frac{\partial C_{ISF}}{\partial \varphi}(t, r, \theta, \varphi) = 0. \quad (4.28)$$

Finally we have an initial condition at $t = 0$. Because of the symmetry conditions that we want to apply to the solution, the initial condition will also have to be symmetric. Because of this, the initial condition will only depend on the radius r :

$$\frac{\partial C_{ISF}}{\partial t}(0, r, \theta, \varphi) = C_{initial}(r), \quad r_0 \leq r \leq r_1, \quad 0 \leq \theta \leq 2\pi, \quad 0 \leq \varphi \leq \pi. \quad (4.29)$$

With the transformation to spherical coordinates, the Laplacian is also transformed to:

$$\Delta C_{ISF} = \frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial C_{ISF}}{\partial r} \right) + \frac{1}{r^2 \sin(\theta)} \frac{\partial}{\partial \theta} \left(\sin(\theta) \frac{\partial C_{ISF}}{\partial \theta} \right) + \frac{1}{r^2 \sin^2(\theta)} \frac{\partial^2 C_{ISF}}{\partial \varphi^2}. \quad (4.30)$$

Symmetry in the solution means that the conditions from (4.27) and (4.28) can be applied to the Laplacian (4.30), resulting in the Laplacian:

$$\Delta C_{ISF} = \frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial C_{ISF}}{\partial r} \right) = \frac{\partial^2 C_{ISF}}{\partial r^2} + \frac{2}{r} \frac{\partial C_{ISF}}{\partial r}. \quad (4.31)$$

When substituting the Laplacian (4.31) into equation (4.1) we find:

$$\frac{\partial C_{ISF}}{\partial t} = \frac{\partial^2 C_{ISF}}{\partial r^2} + \frac{2}{r} \frac{\partial C_{ISF}}{\partial r}. \quad (4.32)$$

Applying the method of separation of variables to (4.32) and assume

$$C_{ISF} = H(t)R(r), \quad (4.33)$$

to get:

$$\frac{1}{H(t)} \frac{dH}{dt} = D^* \frac{1}{R(r)} \left(\frac{d^2 R}{dr^2} + \frac{2}{r} \frac{dR}{dr} \right). \quad (4.34)$$

The l.h.s. only depends on t and the r.h.s. only depends on r . This implies both sides must be equal to some constant, which we will call $-\eta$. Then the solution on the l.h.s. is the same we saw before:

$$H(t) = c_1 e^{-\eta t}. \quad (4.35)$$

We see that $\eta > 0$ must hold for $H(t)$ to converge. The r.h.s., after multiplying both sides by r^2 , becomes:

$$r^2 \frac{d^2 R}{dr^2} + 2r \frac{dR}{dr} + r^2 \frac{\eta}{D^*} R = 0. \quad (4.36)$$

This equation is a spherical Bessel equation, specifically a zeroth spherical Bessel equation, since there is no constant R -term in the equation. The solutions to spherical Bessel equations can be found in chapter 10 of Abramowitz and Stegun [17]. We rewrite the equation in the standard form by setting $\sqrt{\frac{\eta}{D^*}} = \lambda$. The first kind of this Bessel equation is:

$$R_1(r) = J_0(\lambda r) = \frac{\sin(\lambda r)}{\lambda r}. \quad (4.37)$$

It can be checked that (4.36) holds.

The zeroth order Bessel equation of the second kind is:

$$R_2(r) = Y_0(\lambda r) = -J_{-1}(\lambda r) = -\frac{\cos(\lambda r)}{\lambda r}. \quad (4.38)$$

Once again (4.36) holds.

Any linear combination of R_1 and R_2 is again a solution to our differential equation. This gives the general solution of $\alpha R_1 + \beta R_2$. When $a = \frac{\alpha}{\lambda}$ and $b = \frac{-\beta}{\lambda}$ the solution for R becomes:

$$R(r) = \frac{a \cos(\lambda r) + b \sin(\lambda r)}{r} \quad (4.39)$$

Combining (4.39) and (4.35) we find the general solution:

$$C_{ISF}(t, r, \theta, \varphi) = c_1 e^{-\eta t} \frac{a \cos(\lambda r) + b \sin(\lambda r)}{r} \quad (4.40)$$

To find the values for a , b and λ we need to apply the boundary conditions from (4.25) to (4.40). Before they can be applied the derivative:

$$\frac{\partial C_{ISF}}{\partial r}(t, r, \theta, \varphi) = \frac{d}{dr} H(t) R(r) = H(t) R'(r) = c_1 e^{-\eta t} \frac{\lambda}{r} (a \cos(\lambda r) - b \sin(\lambda r)) - \frac{1}{r^2} (a \sin(\lambda r) + b \cos(\lambda r)), \quad (4.41)$$

is required. Since our boundary condition sets the derivative to 0, and $e^{\eta t} \neq 0$ for any (t, r) , we only need to solve for $R'(r_{0/1}) = 0$.

However this expression cannot be used to solve for all three of a , b and λ , since there are only two conditions. We can set $a = 1$. Now only b and λ have to be found from two conditions, which can be done. The expression can be simplified as follows:

$$R'(r) = \frac{\lambda}{r_{0/1}} (\cos(\lambda r_{0/1}) - b \sin(\lambda r_{0/1})) - \frac{1}{r_{0/1}^2} (\sin(\lambda r_{0/1}) + b \cos(\lambda r_{0/1})) = 0, \quad (4.42)$$

$$\left(\lambda - \frac{b}{r_{0/1}} \right) \cos(\lambda r_{0/1}) = \left(\frac{1}{r_{0/1}} + \lambda b \right) \sin(\lambda r_{0/1}), \quad (4.43)$$

$$\frac{(\lambda r_{0/1} - b)}{(1 + \lambda b r_{0/1})} = \tan(\lambda r_{0/1}). \quad (4.44)$$

Applying the boundary condition at r_0 to equation (4.44) allows for a solution to b in terms of λ :

$$\frac{(\lambda r_0 - b)}{(1 + \lambda b r_0)} = \tan(\lambda r_0), \quad (4.45)$$

$$b(r_0 \lambda \tan(\lambda r_0) - 1) = \tan(\lambda r_0) - \lambda r_0, \quad (4.46)$$

$$b = \frac{\tan(\lambda r_0) - \lambda r_0}{\lambda r_0 \tan(\lambda r_0) - 1}. \quad (4.47)$$

The outer boundary condition at r_1 can be applied to equation (4.44). By substituting the expression for b found in (4.47), we get the following expression:

$$\frac{(\lambda r_1 - b)}{(1 + \lambda b r_1)} - \tan(\lambda r_1) = 0, \quad (4.48)$$

$$\frac{\left(\lambda r_1 - \frac{\tan(\lambda r_0) - \lambda r_0}{\lambda r_0 \tan(\lambda r_0) - 1} \right)}{\left(1 + \lambda r_1 \frac{\tan(\lambda r_0) - \lambda r_0}{\lambda r_0 \tan(\lambda r_0) - 1} \right)} - \tan(\lambda r_1) = 0, \quad (4.49)$$

$$\frac{\lambda^2 r_1 r_0 \tan(\lambda r_0) + \lambda(r_0 - r_1) - \tan(\lambda r_0)}{-\lambda^2 r_1 r_0 + \lambda \tan(\lambda r_0)(r_1 + r_0)} - \tan(\lambda r_1) = 0. \quad (4.50)$$

In expression (4.50) we have an equation of just one variable (λ) with two known parameters in r_0 and r_1 that we can solve for numerically.

Using equations (4.50) and (4.47) and the boundary conditions at r_1 and r_0 , a numerical solution for λ and b can be found. The matlab code used for this numerical solution can be found in appendix A. The solutions are $\lambda = 286192.3646$ and $b = 0.62738$ using the radii $r_0 = 4.5 \cdot 10^{-6}$ and $r_1 = 2.5 \cdot 10^{-5}$.

From these values for λ and b the solutions can be determined. This gives the initial function:

$$C_{ISF}(0, r, \theta, \varphi) = \frac{\cos(286192.3646r) + 0.62738 \sin(286192.3646r)}{r}. \quad (4.51)$$

Once again the concentration is not something that can be negative. To make sure the concentration is non-negative everywhere, the minimum of (4.51) is added everywhere. We add the minimum specifically so that the minimum of our actual initial function will be zero. This gives the following initial function:

$$C_{ISF,1}(0, r, \theta, \varphi) = \frac{\cos(2.86 \cdot 10^5 r) + 0.63 \sin(2.86 \cdot 10^5 r)}{r} - \min_{[r_0, r_1]} \left\{ \frac{\cos(2.86 \cdot 10^5 r) + 0.63 \sin(2.86 \cdot 10^5 r)}{r} \right\} \quad (4.52)$$

For the exact solution $\eta = D^* \lambda^2 = 5 \cdot 10^{-12} \cdot 286192.3646^2 = 0.4095$ can also be calculated (since we set $\lambda = \sqrt{\frac{\eta}{D^*}}$. The exact solution is:

$$C_{ISF,1}(t, r, \theta, \varphi) = e^{-0.4095t} \frac{\cos(2.86 \cdot 10^5 r) + 0.63 \sin(2.86 \cdot 10^5 r)}{r} - \min_{[r_0, r_1]} \left\{ \frac{\cos(2.86 \cdot 10^5 r) + 0.63 \sin(2.86 \cdot 10^5 r)}{r} \right\} \quad (4.53)$$

The exact value of this function at $t = 0$ and $r = r_0$ is in the order of 10^5 . This is a huge value and can lead to computational problems. To account for this, we scale the function (4.53), dividing it by the maximum value of the initial function (4.52). This gives us the final function that we can use for simulation:

$$C_{ISF}(t, r, \theta, \varphi) = \frac{C_{ISF, 1}(t, r, \theta, \varphi)}{\max_{[r_0, r_1]} C_{ISF, 1}(0, r, \theta, \varphi)} \quad (4.54)$$

4.3 Diffusion equation in a cylindrical domain

The cylindrical domain that will be used here is laid out in section 3.5.2. To analyze the cylindrical domain, the Cartesian coordinates (x, y, z) are transformed to cylindrical coordinates (r, θ, z) . In this domain there are the radial boundaries at the capillary with radius r_0 and the domain edge at r_1 and the edge boundaries at z_0 and z_1 . Since there is assumed to be no flow through the boundaries of the domain, we find the following boundary conditions:

$$\frac{\partial C_{ISF}}{\partial r}(t, r_0, \theta, z) = \frac{\partial C_{ISF}}{\partial r}(t, r_1, \theta, z) = 0, \quad t \geq 0, \quad 0 \leq \theta \leq 2\pi, \quad z_0 \leq z \leq z_1, \quad (4.55)$$

$$\frac{\partial C_{ISF}}{\partial r}(t, r, \theta, z_0) = \frac{\partial C_{ISF}}{\partial r}(t, r, \theta, z_1) = 0, \quad t \geq 0, \quad r_0 \leq r \leq r_1, \quad 0 \leq \theta \leq 2\pi. \quad (4.56)$$

$$(4.57)$$

For the cylindrical domain it is useful to look for a solution that is radially symmetric around the capillary and constant in the z -direction, as this will increase clarity in the simulation. This adds the conditions:

$$\frac{\partial C_{ISF}}{\partial \theta}(t, r, \theta, z) = 0, \quad (4.58)$$

$$\frac{\partial C_{ISF}}{\partial z}(t, r, \theta, z) = 0. \quad (4.59)$$

$$(4.60)$$

Because of the same reasoning as for the spherical domain, we want the initial condition to be:

$$\frac{\partial C_{ISF}}{\partial t}(0, r, \theta, z) = C_{initial}(r), \quad 0 \leq \theta \leq 2\pi, \quad z_0 \leq z \leq z_1. \quad (4.61)$$

Because of the transformation into cylindrical coordinates, the Laplacian also changes with the coordinates to:

$$\Delta C_{ISF} = \frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial C_{ISF}}{\partial r} \right) + \frac{1}{r^2} \frac{\partial^2 C_{ISF}}{\partial \theta^2} + \frac{\partial^2 C_{ISF}}{\partial z^2}. \quad (4.62)$$

Applying the conditions from (4.58) and (4.59) to the Laplacian (4.62) gives the simplified Laplacian:

$$\Delta C_{ISF} = \frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial C_{ISF}}{\partial r} \right). \quad (4.63)$$

By assuming we can separate C_{ISF} as follows,

$$C_{ISF}(t, r, \theta, x) = cH(t)R(r), \quad (4.64)$$

where $c = T(\theta)L(z)$ is constant, we can separate (4.63) as follows:

$$\frac{1}{H(t)} \frac{dH}{dt} = D^* \frac{1}{R(r)} \left(\frac{d^2R}{dr^2} + \frac{1}{r} \frac{dR}{dr} \right). \quad (4.65)$$

Since the l.h.s. of (4.65) only depends on t and the r.h.s. only on r , both sides of the expression equal a common constant $-\mu$. Similar to the previous solutions, we find:

$$H(t) = c_1 e^{-\mu t}. \quad (4.66)$$

We see that $\mu > 0$ must hold for $H(t)$ to converge. Setting the r.h.s. of (4.65) equal to $-\mu$ and multiplying by r^2 , we find:

$$r^2 \frac{d^2R}{dr^2} + r \frac{dR}{dr} + r^2 \frac{\mu}{D^*} R = 0. \quad (4.67)$$

This is the Bessel equation mentioned in expression (9.1.52) of Abramowitz and Stegun [17]. For $\frac{\mu}{D^*} = \nu^2$ the solution to equation (4.67) is:

$$R(r) = aJ_0(\nu r) + bY_0(\nu r), \quad (4.68)$$

where J_0 is the zeroth Bessel function of the first kind and Y_0 is the zeroth Bessel function of the second kind.

Now we apply the boundary conditions from (4.55). Similar to the spherical equations, $a = 1$ is set so that the two boundary conditions can be used to solve for b and ν . The derivative of our solution is also given by Abramowitz and Stegun in equation (9.1.28) [17]:

$$R'(r) = -\nu J_1(\nu r) - \nu b Y_1(\nu r). \quad (4.69)$$

The outer boundary condition is applied to find an expression for b :

$$b = -\frac{J_1(\nu r_1)}{Y_1(\nu r_1)}. \quad (4.70)$$

This value for b can be substituted into (4.69) at the inner boundary to get an expression with only ν as parameter:

$$-\nu J_1(\nu r_0) + \nu \frac{J_1(\nu r_1)}{Y_1(\nu r_1)} Y_1(\nu r_0) = 0. \quad (4.71)$$

Using the radii $r_0 = 2.5 \cdot 10^{-6}$ and $r_1 = 2.5 \cdot 10^{-5}$, we can numerically solve this to find the parameters $\nu = 38.578$ and $b = 7.305 \cdot 10^{-7}$ (matlab code in appendix B). Additionally, $\mu = D^* \nu^2 = 7.441 \cdot 10^{-9}$ can be calculated from all the constants (since we set $\nu^2 = \frac{\mu}{D^*}$). This gives the initial function:

$$C_{ISF}(0, r) = J_0(38.578r) + 7.305 \cdot 10^{-7} Y_0(38.578r). \quad (4.72)$$

Again, to ensure there are no negative concentrations in the domain, the minimal value is subtracted. We could have chosen any constant larger than the minimum, but choosing the minimum sets the smallest concentration to exactly zero. This gives the following initial function and exact solution:

$$\begin{aligned} C_{ISF}(0, r) &= J_0(38.578r) + 7.305 \cdot 10^{-7} Y_0(38.578r) \\ &\quad - \min_{[r_0, r_1]} \{J_0(38.578r) + 7.305 \cdot 10^{-7} Y_0(38.578r)\}, \end{aligned} \quad (4.73)$$

$$\begin{aligned} C_{ISF}(t, r) &= e^{-7.441 \cdot 10^{-9} t} (J_0(38.578r) + 7.305 \cdot 10^{-7} Y_0(38.578r)) \\ &\quad - \min_{[r_0, r_1]} \{J_0(38.578r) + 7.305 \cdot 10^{-7} Y_0(38.578r)\}. \end{aligned} \quad (4.74)$$

5 Theoretical results

Numerical methods for solving differential equations, such as the diffusion equation, always have an error. There are differences between the numeric approximation of the derivative of a function and the actual derivative. The error that results from approximation is called the truncation error. Another error is caused by the physical restrictions of the memory of a machine. Computers cannot store an infinite number of decimals, so they always have to round real numbers to a decimal place. This is called the rounding error. Together, these two errors sum to the total error of a numerical system. We will look at the truncation error, rounding error and total errors of all used functions.

5.1 Local Truncation Error

The truncation error is the difference between the actual derivative of a function and the estimated derivative as determined by our numerical system. CapSys applies the method of backwards Euler, which approximates the derivative with the following expression:

$$Q_b(\Delta t) = \frac{f(t) - f(t - \Delta t)}{\Delta t}, \quad (5.1)$$

where Q_b is backward difference and Δt is the time step. The truncation error is defined as:

$$R_b(\Delta t) = f'(t) - Q_b(\Delta t) = f'(t) - \frac{f(t) - f(t - \Delta t)}{\Delta t}, \quad (5.2)$$

where R_b is the truncation error for the backward difference. To estimate the truncation error, a Taylor expansion of $f(t - \Delta t)$ around t is used:

$$f(t - \Delta t) = f(t) - \Delta t f'(t) + \frac{\Delta t^2}{2} f''(\xi), \quad (5.3)$$

with $\xi \in (t - \Delta t, t)$. Substituting this into the truncation error in equation (5.2) gives:

$$R_b(\Delta t) = f'(t) - \frac{f(t) - f(t) + \Delta t f'(t) - \frac{\Delta t^2}{2} f''(\xi)}{\Delta t} = \frac{\Delta t}{2} f''(\xi) = \mathcal{O}(\Delta t). \quad (5.4)$$

Since the differential equation contains the first derivative with respect to time, this gives the truncation error with respect to time as well. It is found that the order of convergence in time is linear. In other words, halving the time step, should double the accuracy.

For the truncation error with respect to the distance, the error of estimation for the second derivative is required. Using a central difference method to calculate the second derivative gives the expression:

$$Q_c^{[2]}(\Delta x) = \frac{f(x - \Delta x) - 2f(x) + f(x + \Delta x)}{\Delta x^2}, \quad (5.5)$$

where $Q_c^{[2]}$ is the central difference formula for the second derivative. This leads to the truncation error for the second derivative, which is:

$$R_c^{[2]}(\Delta x) = f''(x) - Q_c^{[2]}(\Delta x) = f''(x) - \frac{f(x - \Delta x) - 2f(x) + f(x + \Delta x)}{\Delta x^2}. \quad (5.6)$$

To calculate the error, Taylor expansions of both $f(x - \Delta x)$ and $f(x + \Delta x)$ of the third order are required. These are:

$$f(x - \Delta x) = f(x) - \Delta x f'(x) + \frac{\Delta x^2}{2} f''(x) - \frac{\Delta x^3}{6} f'''(x) + \frac{\Delta x^4}{24} f^{(4)}(\xi), \quad (5.7)$$

$$f(x + \Delta x) = f(x) + \Delta x f'(x) + \frac{\Delta x^2}{2} f''(x) + \frac{\Delta x^3}{6} f'''(x) + \frac{\Delta x^4}{24} f^{(4)}(\xi), \quad (5.8)$$

where $\xi \in [x - \Delta x, x + \Delta x]$. Substituting both (5.7) and (5.8) into (5.6) shows all terms up to and including the third derivative cancel each other. What is left is a term of $\mathcal{O}(\Delta x^2)$, which is then the order of the error in the spatial dimensions.

This means that the total truncation error of the numerical system is of order $\mathcal{O}(\Delta t + \Delta x^2 + \Delta y^2 + \Delta z^2)$. Since the domain is discretized into the same resolution in all three spatial dimensions, $\Delta x = \Delta y = \Delta z$, those terms can be added together. This leads to the total truncation error of order $\mathcal{O}(\Delta t + \Delta x^2)$.

5.2 Rounding Error

The rounding error is determined by the absolute machine precision. For a 32 bit machine, the rounding error is $2^{-23} \approx 1.19 \cdot 10^{-7}$ and for a 64 bit machine this is $2^{-52} \approx 2.22 \cdot 10^{-16}$. These errors are small, but the impact of these rounding errors increase as we make the time steps smaller. This places a mechanical limit on the lower bound for the possible time step before the rounding error makes the numerical method unreliable. For the method of backwards Euler that CapSys uses, the rounding error can be determined:

$$S_t = |Q_c - \hat{Q}_c|, \quad (5.9)$$

where Q_c is the exact expression from (5.1) and \hat{Q}_c is the rounded machine estimation of that expression. We can substitute expression (5.1) to get:

$$\begin{aligned} S_t &= \left| \frac{f(x) - f(x - \Delta t) - \hat{f}(x) + \hat{f}(x - \Delta t)}{\Delta t} \right| \\ &\leq \frac{|f(x) - \hat{f}(x)|}{\Delta t} + \frac{|f(x - \Delta t) - \hat{f}(x - \Delta t)|}{\Delta t} \\ &\leq \frac{2\varepsilon}{\Delta t}, \end{aligned}$$

where ε is the machine accuracy and $\hat{f}(x)$ is the rounded machine estimation of $f(x)$. From here a 64 bit machine will be considered, giving $\varepsilon = 2.22 \cdot 10^{-16}$.

5.3 Total Error

The total error E is the sum of both the rounding error and truncation error:

$$E = S_t + S_x + R_t + R_x = \frac{2\varepsilon}{\Delta t} + \frac{2\varepsilon}{\Delta x} + \frac{M_t}{2} \Delta t + \frac{M_x}{24} \Delta x^2, \quad (5.10)$$

where M_t is the upper bound for $f''(t)$ in the domain and M_x for $f^{(4)}(x)$. S_t and S_x are the machine rounding errors for the functions of t and x respectively, as discussed in section (5.2). R_t and R_x are as calculated in section (5.1).

5.3.1 Cubic domain

To determine the error in the time step for the cubic domain, an upper bound for the second derivative of the exact function (4.23) is required. Specifically, an upper bound for the absolute value is needed, since a large negative derivative also gives a large truncation error. The second derivative of the exact function with respect to time is given by:

$$|\partial_t^2 C_{cube}(t, x, y, z)| = |D^{*2} \left(\frac{\pi}{x_1 - x_0} \right)^4 e^{-D^* \left(\frac{\pi}{x_1 - x_0} \right)^2 t} \cos\left(\frac{x - x_0}{x_1 - x_0} \pi \right)|. \quad (5.11)$$

The second derivative has its maximum at $x = x_0$ or $x = x_1$, since at those values the absolute value of the cosine is 1. The only part dependent on the time is the e^{-kt} expression. The exponential function decreases with time, meaning that the maximum is found at $t = 0$. At $t = 0$ the exponential function is equal to 1. For the cubic domain we have $D^* = 2.5 \cdot 10^{-10}$ and the values for x_0 and x_1 as mentioned in section (3.5.1). Substituting these gives:

$$|\partial_t^2 C_{cube}(t, x, y, z)| \leq |D^{*2} \left(\frac{\pi}{x_1 - x_0} \right)^4| \approx 0.47, \quad (5.12)$$

as the value for M_t . Substituting (5.12) as M_t into (5.10) gives:

$$E_{cube}(\Delta t) = \frac{4.44 \cdot 10^{-16}}{\Delta t} + 0.24\Delta t, \quad (5.13)$$

where E_{cube} is the total error of the numerical system for the cubic domain.

By finding the root of the derivative of this function, the optimal value for the time step can be found to be $3.3 \cdot 10^{-11}$ seconds. This is such a small time step that it will almost never be approached. Only for time steps smaller than the optimal value, does the effect of the machine error start to outweigh the truncation error. Practically, this means that for most calculations the machine error can be left out. At the standard chosen time step of 0.1 seconds, the upper bound for the error is $2.4 \cdot 10^{-2}$.

For the error in resolution (M_x) the fourth derivative with respect to direction is required. The fourth derivative with respect to x is:

$$|\partial_x^4 C_{cube}(t, x, y, z)| = \left| \left(\frac{\pi}{x_1 - x_0} \right)^4 e^{-D^* \left(\frac{\pi}{x_1 - x_0} \right)^2 t} \cos\left(\frac{x - x_0}{x_1 - x_0} \pi \right) \right|. \quad (5.14)$$

Both the exponential function, as well as the cosine are bounded above by 1 for the variables in the correct domain. Substituting 1 for the exponential function and cosine in (5.14) gives the upper bound for the fourth derivative:

$$|\partial_x^4 C_{cube}(t, x, y, z)| \leq \left| \left(\frac{\pi}{x_1 - x_0} \right)^4 \right| \leq 7.52 \cdot 10^{18}. \quad (5.15)$$

Since the value for M_x is larger than M_t , the effect of the machine error is even smaller for resolution than it was for time step. This means that the machine error is again negligible. We can substitute this value for M_x into the total error (5.10). This gives an overall upper bound for the total error depending on both the time step as well as distance step:

$$E(\Delta t, \Delta x) = 0.24\Delta t + 3.13 \cdot 10^{17} \Delta x^2. \quad (5.16)$$

The upper bound on the error is very high for the spatial term, this means that we need (Δx) to be very small to reduce this error term. Since the maximum domain size is only $60\mu\text{m}$, this value will always be very small. It is still really important to take small spatial steps in order to increase the accuracy. More importantly than the upper bound is the fact that the error term is $\mathcal{O}(\Delta t) + \mathcal{O}(\Delta x^2)$. This means that the error should decrease linearly with a reduction in Δt and quadratically with Δx .

5.3.2 Spherical domain

To determine the error in the time step for the spherical domain, an upper bound for the second derivative of the exact function (4.54) is required. Specifically, an upper bound for the absolute value is needed, since a large negative derivative also give a large truncation error. The second derivative of the exact function with respect to time is given by:

$$|\partial_t^2 C_{ISF}(t, r)| = \left| \frac{0.1677 \cdot 10^{-0.4095t} \frac{\cos(2.86 \cdot 10^5 r) + 0.63 \sin(2.86 \cdot 10^5 r)}{r}}{\max_{[r_0, r_1]} C_{ISF, 1}(0, r, \theta, \varphi)} \right|, \quad (5.17)$$

$$= |0.1677| \left| \frac{\frac{\cos(2.86 \cdot 10^5 r) + 0.63 \sin(2.86 \cdot 10^5 r)}{r}}{\max_{[r_0, r_1]} C_{ISF, 1}(0, r, \theta, \varphi)} \right|. \quad (5.18)$$

The second derivative reaches a maximum in time at $t = 0$, since then the exponential function is equal to 1. Clearly, the trigonometric functions divided by the maximum of those same functions (maximum of (4.52)) can not exceed 1. This means that the upper bound for the second derivative in (5.17) is 0.1677. Substituting this into (5.10) gives:

$$E_{sphere} = \frac{4.44 \cdot 10^{-16}}{\Delta t} + 0.1677 \Delta t. \quad (5.19)$$

This similar to the error of the cubic domain in (5.13) we can also find the optimal value for the time step Δt . For the sphere this optimal time step is $5.145 \cdot 10^{-8}$ seconds. For the same reasons as suggested in section 5.3.1, the machine error is negligible. The error at the regularly used time step of 0.1 seconds is 0.01677.

Determining the exact order of the error for the resolution in terms of Δx , Δy and Δz is difficult for equations in spherical coordinates. As shown in section 5.1, the error in the spatial dimension is of order $\mathcal{O}(\Delta x^2 + \Delta y^2 + \Delta z^2)$. This still holds for equation (4.54).

The difference with the order of the error for the cubic domain, is that the boundary of the spherical domain becomes easier to approximate as the resolution increases. A simple example of the problems that can arise with the spherical domain and a low resolution can be viewed in figure 4. The red areas are part of the grid outside of the domain between the outer boundary and inner cell. In this extreme example of a resolution of only 4, there is not a single grid slot that is fully within the simulated domain. Because of this aspect, the order of convergence may, in practice, be different from $\mathcal{O}(\Delta x^2)$.

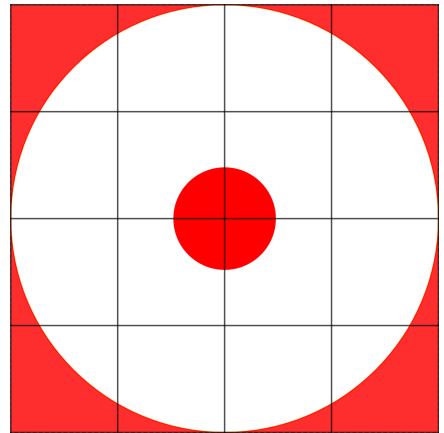


Figure 4: Two dimensional representation of what happens to the spherical domain on a 4 by 4 by 4 grid. The red areas show what falls outside the simulation domain. It is clearly visible in this representation that not a single grid slot is fully within the simulation domain.

5.3.3 Cylindrical domain

For the cylindrical domain there is not much to add over the previous two domains. The cylindrical domain can be viewed as a combination of the cubic and spherical domains. Seeing as both of the previous two domains came out to a theoretical error term of order $\mathcal{O}(\Delta t + \Delta x^2)$, the expectation is that this should also hold for the cylindrical domain. The caveat mentioned in section 5.3.2 about the cut-off around circular elements also counts here, so the convergence may in practice be different from the expected order.

6 CapSys results

CapSys has been modified for this thesis to allow for an exact solution to be entered into the program, besides the regular simulation that it is meant for. The results that we get from the software are the simulation results and the results comparing the simulation with the exact solution. Within those two types, there are different cross sections of the domain and overall timeseries that cover the maximum, minimum and average values, as well as the added root mean square (RMS).

There are two types of results that are important from the CapSys simulations, the qualitative and quantitative results. Qualitative results mostly look at the images from the simulation and the comparison of the simulation with the exact solution. If there are large differences, especially large peak differences, then those differences warrant a deeper look. These large differences can point to issues within the software. Quantitative results will be looked at to examine the overall effectiveness of the software. Within the quantitative results section the RMS will be an important metric.

One of the differences between the different domains is a change in the diffusion coefficient D^* . This is done to get a good amount of convergence within the time range of 5 seconds. If the concentrations converge too quickly, the diffusion coefficient will be lowered and if it converges too slowly it will be increased. Unless stated otherwise, the time step used in the simulation is 0.1 seconds.

6.1 Diffusion within the cubic domain

Diffusion within a cubic domain is one of the easier tasks for the program. CapSys was also programmed based on cubic domains and all sections fall entirely within the cubic domain. No matter the resolution or time step size, there is never any difference in total drug volume in the ISF between the simulated and exact solutions. Moreover, there is no change in total drug volume with changes in resolution. This allows for a good examination of the convergence as the resolution of the grid increases. In appendix A, the Jupyter notebook containing the diffusion within a cubic domain with resolution 16 in each dimension can be found. These runs are with the cosine initial function in the x-direction alone. The only changes between different runs are changes in the resolution of the grid. D^* has been set to $2.5 \cdot 10^{-10}$ for the cubic domain, since that gives a good amount of diffusion over the time frame of 5 seconds.

6.1.1 Qualitative results

First, it is important to look for errors in the simulation or large differences between the simulation of the initial solution and exact solution from section 4.1. Visually, in figure 5 it can be seen that the distribution at time $t = 0$ does appear to follow the expected distribution of a cosine. As the time increases the concentration also approaches the average concentration of 1.

Another important check is the total drug concentration in the domain. The size of the domain is $(60\mu m)^3 = 2.16 \cdot 10^{-13} m^3$. The average concentration is $1 \frac{mmol}{m^3}$, giving a total concentration of:

$$1 \cdot 10^{-3} \frac{mol}{m^3} 2.16E^{-13} m^3 = 2.16 \cdot 10^{-16} mol. \quad (6.1)$$

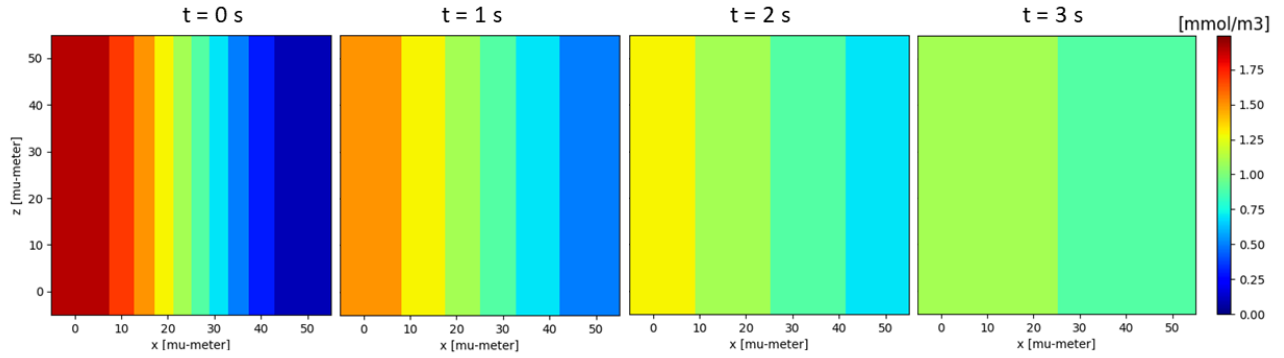


Figure 5: Visualization of the simulation in CapSys. On the left at $t = 0$ s the initial condition can be seen. This is as expected, with a high value of $2 (= \cos(0) + 1)$ on the left and a low value of $0 (= \cos(\pi) + 1)$ on the right. As time progresses, the values everywhere converge towards 1. At $t = 3$ s the convergence is very close to being finished.

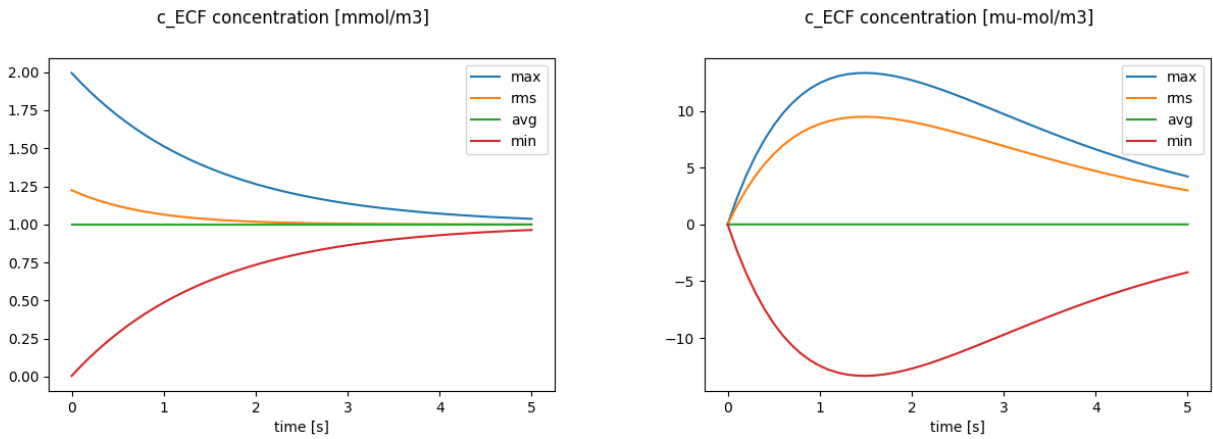
This means that the total drug concentration should be 0.2160 fmol (femto mol, 10^{-15} mol), which is also the value that CapSys returns for the total concentration in both the exact solution and simulated solution.

Next the timeseries (figure 6a) of the simulation is examined to see if there are any grid slots where something goes wrong with the values. Often a grid slot where the simulation doesn't work correctly becomes the maximum or minimum value and stays constant instead of converging. For the cubic grid of resolution 16, the concentrations seem to converge correctly.

Similarly, by looking at the timeseries of the difference in concentration (figure 6b) between the exact solution and simulated solutions, mistakes in either the exact solution or simulation can also be seen. For the timeseries of the comparison at a resolution of 16 there is no visual indication of any mistakes. The maximum value of the RMS is $9.48 \frac{\mu\text{mol}}{\text{m}^3}$, much smaller than the average concentration of $1000 \frac{\mu\text{mol}}{\text{m}^3}$. The average difference is also 0, which is expected, since the total concentration always remained the same in both the simulation and exact solution.

6.1.2 Quantitative results

The maximum difference between the exact solution as measured by the RMS is at 1.5 seconds. As the resolution of the grid increases, the maximum value of the RMS decreases. There appears to be an asymptote larger than 0 that the maximum RMS converges to. It does not look like the RMS will decrease to 0 at 1.5 seconds as the resolution continues to increase.



(a) The timeseries of the concentration shows very even convergence as expected. The maximum and minimum values converge towards the average evenly at the same exponential rate.

(b) Timeseries of the comparison between the simulation and exact solution. The absolute value of the maximum difference is less than $15 \frac{\mu\text{mol}}{\text{m}^3}$. These values are much smaller than the actual concentrations, measured in $\frac{\text{mmol}}{\text{m}^3}$. The maximum error is found around $t = 1.5$ seconds and converges as time increases.

Figure 6: Timeseries of the concentrations in the simulation (6a) and of the comparison between the simulation and the exact solution (6b). Both timeseries are with time steps of 0.1 seconds and resolution of 16.

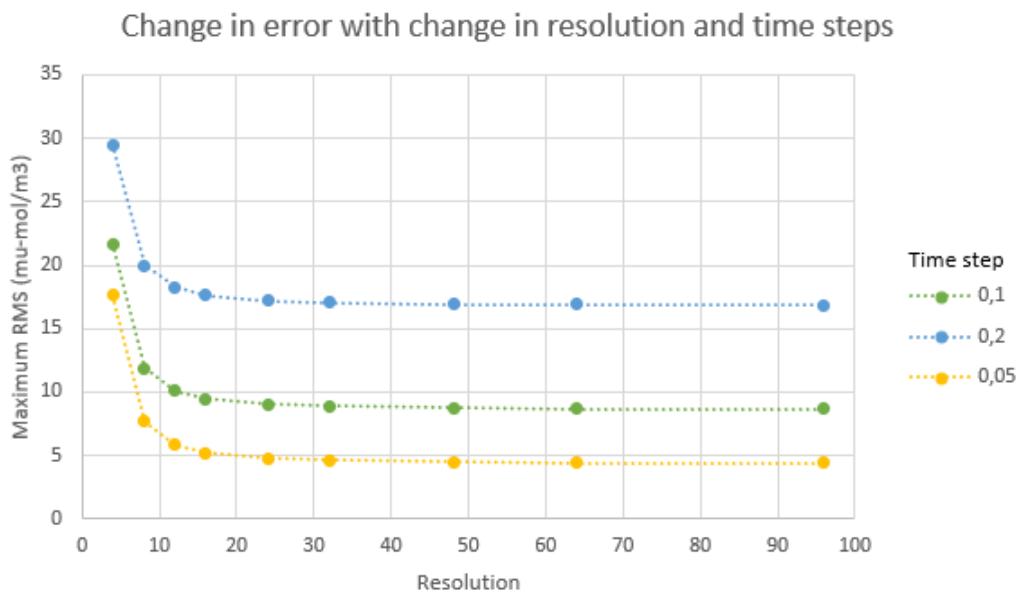


Figure 7: Graph plotting the maximum value for the root mean square (RMS) against the resolution for different time steps. For each time step there is a clear reduction in maximum RMS as the resolution increases, but there seems to be an asymptote for each. This asymptote appears to be explained by the error caused by the time step, since the error at 96 resolution seems to half every time the time step is halved.

Besides the convergence at the regular time step of 0.1 seconds, the same data has also been recovered for time steps of 0.2 seconds and 0.05 seconds. These were chosen since they are double and half the standard 0.1 seconds. As explained in section 5.3.1, the expected order of convergence with

Δt	Maximum RMS	Factor
0.2	16.88	–
0.1	8.70	1.94
0.05	4.42	1.97
0.025	2.24	1.97

Table 1: The factor of difference in the maximum value of the RMS for different time steps. The factors of 1.94, 1.97 and 1.97 are very close to the expected value of 2. Therefore it can be assumed that the numerical system converges linearly with time step.

Δx	Adjusted RMS	Factor
4	13.214	–
8	3.286	4.02
16	0.816	4.03
32	0.199	4.10

Table 2: The factor of difference in the adjusted value of the maximum RMS for different resolutions at a time step of 0.1s. The factors of 4.02, 4.03 and 4.10 are very close to the expected value of 4. Therefore it can be assumed that the numerical system converges quadratically with resolution.

respect to the time step is linear. This means that, at a high resolution, it is expected that doubling the time step, should double the error. As we can see in figure 7, this is what happens. At a resolution of 96, the RMS nearly doubles from 8.70 to 16.88 $\frac{\mu\text{mol}}{\text{m}^3}$ as the time step doubles from 0.1 to 0.2 seconds. Similarly, when halving the time step from 0.1 seconds to 0.05, the RMS nearly halves from 8.70 to 4.42 $\frac{\mu\text{mol}}{\text{m}^3}$. The exact differences can be found in table 1. It can be concluded that the error is indeed of order Δt .

At the time step of 0.05 seconds there is a higher accuracy in time, so this time step will be used to check the rate of convergence with resolution. Between the resolution of 64 and 96, the maximum RMS decreases from 4.44 to 4.42 $\frac{\mu\text{mol}}{\text{m}^3}$. Looking at that convergence, it seems like the maximum RMS would decrease to somewhere around 4.4 $\frac{\mu\text{mol}}{\text{m}^3}$ as the resolution increases further. This is the value that will be used to adjust the maximum RMS to account for the error caused by the time step. By subtracting 4.4 from the RMS at the resolutions of 8, 16 and 32, we can see the change in error with increasing resolution. We can compare this to our expected order of convergence, which is quadratic convergence, as discussed in section 5.3.1. The adjusted values and the factor compared to a resolution of 16 can be found in table 2. It can be concluded that the error is indeed of order Δx^2 .

6.2 Diffusion within the spherical domain

For the spherical domain, the standard time step is 0.1 seconds and standard resolution is 16 for all the first-look qualitative results. The chosen value for the diffusion coefficient is $D^* = 1 \cdot 10^{-11}$.

For the spherical domain, it was not possible to recreate all the simulations run for the cubic domain. There were no simulations possible for spatial resolutions higher than 48 (all resolution multiples of 4 from 48 to 196 were attempted), since they all returned RunTime errors. Resolutions of 52, 56 and 60 gave a 'no convergence' error, meaning the numerical method didn't convergence to any answer within the acceptable amount of steps. For the resolution 64 the error was 'factor is exactly singular'. At resolutions higher than 64 the returned error was 'too many polygons'.

Besides these error messages, there were two other results that were unobtainable. The simulations at resolutions of 40 and 48 were not possible with the time step of 0.2 seconds. There was no error message with these simulations, but after running for 12 hours, not a single step was calculated. These resolutions were functional with time steps of 0.1 and 0.05 seconds.

6.2.1 Qualitative results

Once again the first thing to look at is the qualitative visual results of our baseline simulation with time steps of 0.1 seconds and a resolution of 16. The visuals of the simulation can be found in figure 8. The concentration inside the cell has been set to zero. Specifically, the concentration is 0 for all points in the domain that are less than $4.5\mu\text{m}$ from the center. However, it does show up as higher in the visuals in figure 8. A possible explanation for this is that for a grid with resolution 16 with a length of $50\mu\text{m}$, the edge length of a single grid slot is $3.125\mu\text{m}$. The Pythagorean theorem in three dimensions then gives us that the furthest corner of each of the 8 cubes around the center are at $5.4\mu\text{m}$ from the center, which is larger than the cell radius of $4.5\mu\text{m}$. This means that part of these central grid volumes is inside the cell, while another part is outside the cell. This might explain why the cell shows up as if it contains some drug. The simulation also does seem to work well with the concentration that shows up inside the cell in the initial condition.

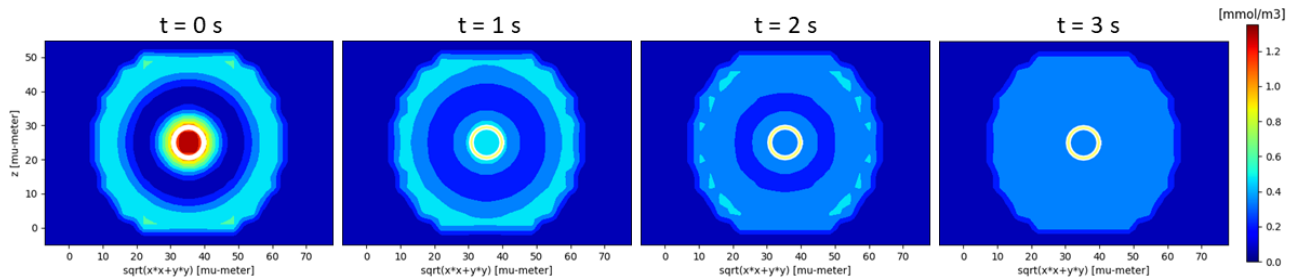
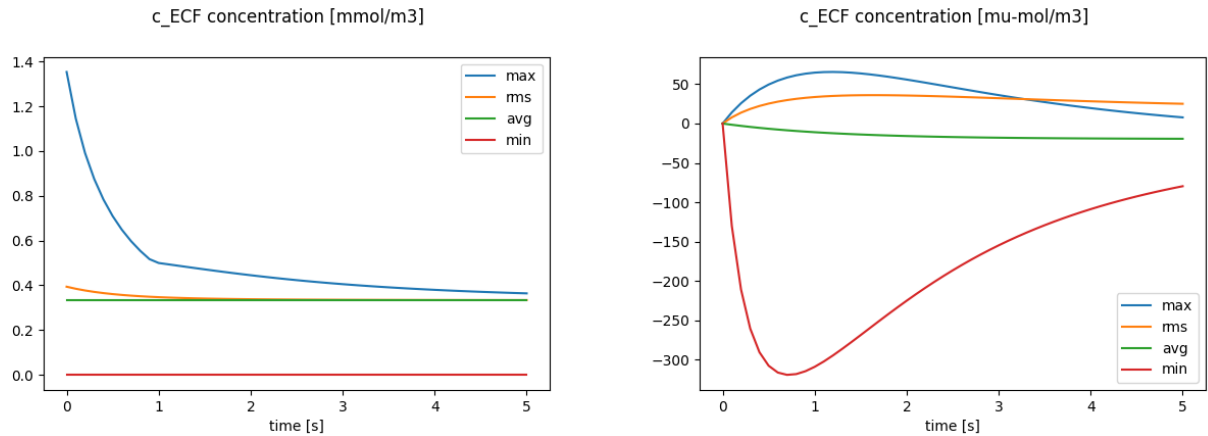


Figure 8: Visualization of the simulation in CapSys. On the left at $t = 0\text{ s}$ the initial condition can be seen. This is as expected from the initial function, with an exception for the area inside the cell, where the concentration was set to 0, but shows up at high values. The simulation also seems to run in that area, where diffusion should not work within the cell. As time progresses, the values everywhere converge towards the average. At $t = 3\text{ s}$ the convergence is very close to being finished.

Looking at the timeseries for the concentration (figure 9a) and the comparison between the simulation and the exact solution (figure 9b) there are a few noticeable things. The convergence of the maximum value decreases towards the average concentration at a high rate and there is a sudden change in the rate of convergence at around 1 seconds. This sudden change makes it seem like another location with slower convergence takes over as the maximum value. Looking at figure 8, it seems like the maximum starts at the center near the cell, but moves to the edge of the sphere after 1 second, where the convergence seems to be slower. Secondly, the minimum difference between the exact solution and simulation is very large at $\left| -300 \frac{\mu\text{mol}}{\text{m}^3} \right|$. A negative difference means that the exact solution has a smaller value than the simulation. By looking at the visual of the comparison (figure 10a) at the time of 1 seconds, it can be seen that this is most likely caused by the cell in the center. Thirdly, the difference in average concentration between the simulation and exact solution slowly becomes negative. This is likely caused by a difference between the total amount of drug that is present in the simulated and exact solution. At the end point of 5 seconds, there is a total amount of 21.48amol (atto mol, 10^{-18}mol) in the simulated solution and 22.72amol in the exact solution. The cause for this is a small increase in total concentration in the exact solution with each time step. As the resolution increases this difference in total amount at 5 seconds becomes smaller.

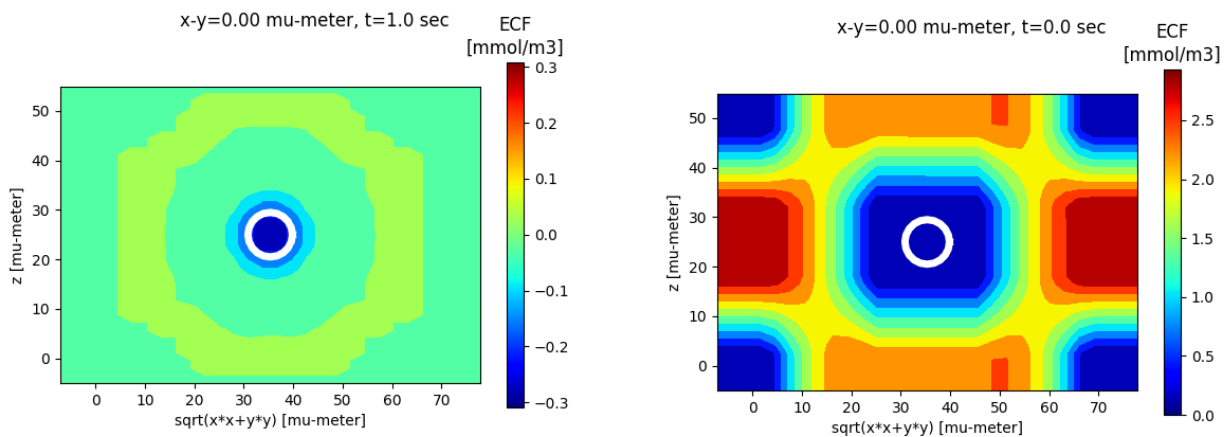
Besides these findings in the initial setup, there are other issues that were found at different resolutions. The case of resolution 4 was already explained in section 5.3.2 with an expectation that



(a) The timeseries of the concentration of the Spherical domain is not exactly as expected. The minimum concentration remaining at 0 is expected, since the area outside the sphere and inside the cell are always at 0. The maximum concentration starts at an exponential rate of convergence, but seems to slow down after 1 second, appearing to switch to a different rate of convergence.

(b) Timeseries of the comparison between the simulation and exact solution. Looking at the scale on the left of the graph, it seems like the error is in the order of $1E^{-5}m$, about an order larger than in the cubic domain. The minimum difference is also really large at 0.8 seconds, at around $300 \frac{\mu\text{mol}}{m^3}$ difference between exact and simulation. This almost as large as the average concentration. While the maximum and minimum values seem to converge with time, the RMS converges very slowly.

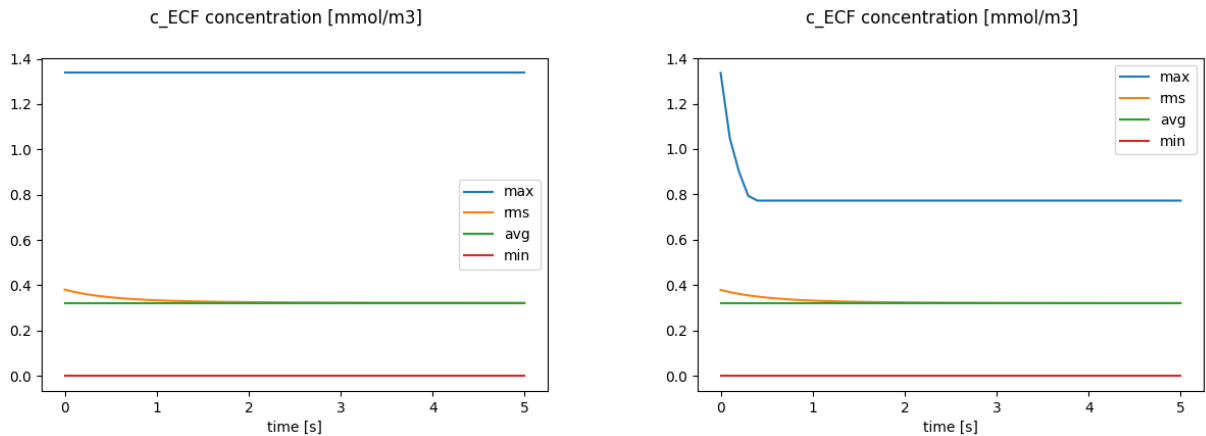
Figure 9: Timeseries of the concentrations in the simulation (9a) and of the comparison between the simulation and the exact solution (9b). Both timeseries are with time steps of 0.1 seconds and resolution of 16.



(a) This image visually shows the results from the timeseries in figure 9b at $t = 1$ second. The large negative minimum value is clearly in the cell in the center. It seems like the large concentration there diffuses much faster in the simulation than in the exact solution. On the contrary, the maximum value is found near the edge of the sphere, where the exact solution converges faster than the simulation.

(b) Visual of the initial condition of a simulation with resolution 4. It looks nothing like the intended initial condition as seen in figure 8. The expected issues considered in figure 4 are evident here.

Figure 10: Visualizations of large differences between the expectation and results. Figure 10a shows the comparison between simulation and exact solution at resolution 16 and time step 0.1. Figure 10b shows the initial condition of the simulation with a resolution of 4.



(a) From this time series it is clear that there is a maximum concentration value at some point in the domain that isn't affected by the simulation, which should push all values towards the average. The maximum value never changes.

(b) This time series shows there is a large value in a grid area that isn't affected by simulation, which should push all values towards the average. The global maximum does decrease, but this area with a concentration of around 0.8 never changes.

Figure 11: Two time series that show error that prop up in certain grid slots. These cannot always be found in the visual cross-sections, but do appear in these. On the left is the time series of the simulated concentrations with resolution 40 and on the right, the same type of time series of resolution 48.

it wouldn't behave correctly. Indeed it does not, as figure 10b shows. It does not even remotely resemble the initial function.

Besides this expected issue, there are also issues that show at higher resolutions. In the timeseries of the concentration over time at resolutions of 40 and 48 we see more of these. In figure 11a it is apparent that the maximum value does not converge at all. The maximum value is found either within the central cell or at its boundary. This gives the idea that the maximum value is inside the central cell and the simulation does not affect this location. If this is the case, the value at this location should have been set to zero, as all values within the central cell should be. Slightly differently, for a resolution of 48, the maximum value does decrease, but there is a floor to the maximum value at $0.8 \frac{\text{mmol}}{\text{m}^3}$, as seen in figure 11b. This also looks like there is a grid point in the domain that does not behave as expected.

6.2.2 Quantitative results

For the quantitative results of the spherical domain we will leave out the domain with resolution 4, because of the reasons stated in section 6.2.1. For the graph in figure 12, resolution 8 is also left out, since the maximum RMS is so high that the differences between time steps will no longer be visible. As mentioned in section 6.2, there is no data for resolutions over 48 and for the time step of 0.2 seconds at resolutions 40 and 48.

Δt	Maximum RMS	Factor
0.2	31.944	–
0.1	31.66	1.01
0.05	32.282	0.98

Table 3: The factor of difference in the maximum value of the RMS for different time steps at a resolution of 32. The factors of 1.01 and 0.98 are nowhere near the expected value of 2 if it would converge linearly. It cannot be concluded that the numerical system converges linearly with time in a domain with spherical elements.

Δx	Maximum RMS	Factor
8	124.0	–
16	36.0	3.44
32	31.7	1.14

Table 4: The factor of difference in the adjusted value of the maximum RMS for different resolutions. The factors of 3.44 and 1.14 are not close to the expected value of 4. Therefore it cannot be concluded that the numerical system converges quadratically with resolution in a domain with spherical elements.

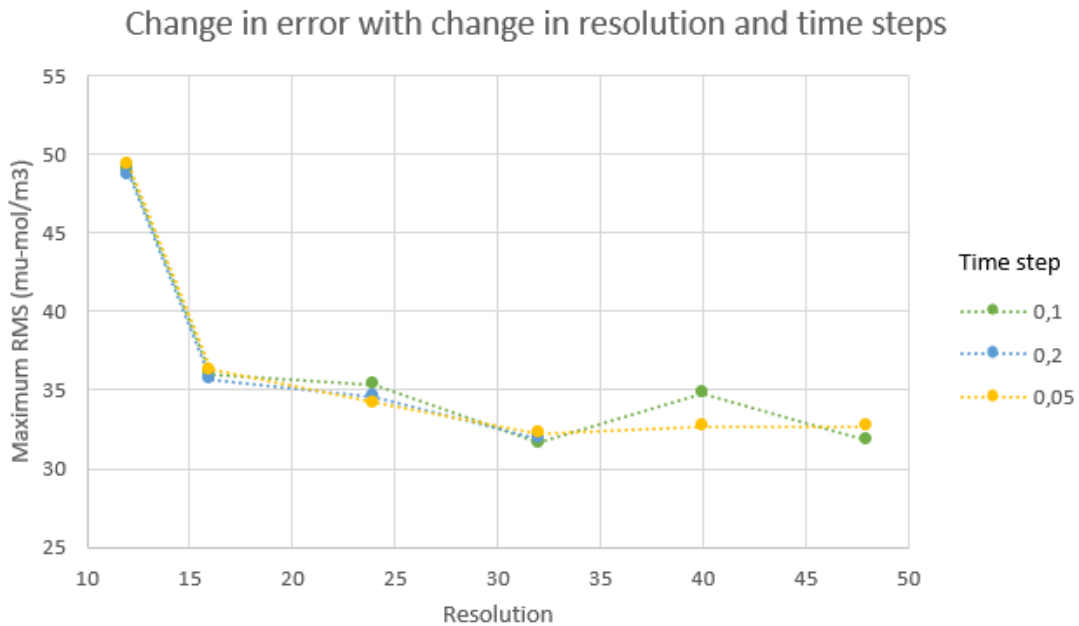


Figure 12: Graph plotting the maximum value for the root mean square (RMS) against the resolution for different time steps. The error seems to decrease with increasing resolution, but only up to a point. The lowest errors are found at a resolution of 32. This may be caused by the erroneous maximum values shown in figure 11a and 11b. There does not seem to be any convergence with decreasing time steps.

In figure 12 the convergence with resolution is visible, but it is not as uniform as with the cubic domain. Possible causes for this are the two edges of the spherical domain. At both the inner and outer edge, the cubic grid areas cut through the edges of the domain. For all three chosen time steps the lowest maximum RMS was with a resolution of 32. Specifically, the RMS is $31.66 \frac{\mu\text{mol}}{\text{m}^3}$ at 32 resolution, $34.83 \frac{\mu\text{mol}}{\text{m}^3}$ at 40 resolution and $31.83 \frac{\mu\text{mol}}{\text{m}^3}$ at 48 resolution. This does seem to follow from the problems that were observed in the maximum values in figures 11a and 11b. There the observed maximum value is higher at a resolution of 40 compared to the maximum value at resolution 48, leading to a larger error.

Looking at the graph in figure 12 and the tables 3 and 4, it seems like there is baseline error factor of around $30 \frac{\mu\text{mol}}{\text{m}^3}$ in the spherical domain that do not seem to reduce with increased resolution in space or time. This large error term that seems present everywhere also restricts the ability to adjust

the RMS, like in table 2, since it is not clear what portion of the error is caused by the time step. With the change in resolution from 8 to 16, the factor seems to approach the expected number of 4, if the convergence is of order $\mathcal{O}(\Delta x^2)$ as explained in section 5.3.2. The next doubling in resolution does not show that kind of convergence however, with a factor of only 1.14.

6.3 Diffusion within the cylindrical domain

The configuration of the simulation has been set up at the standard values of 16 resolution and 0.1 seconds time steps. Based on this, the diffusion coefficient has been set to $D^* = 1 \cdot 10^{-10}$. At this configuration setup the initial qualitative results were gathered. These results were images of the simulation and the comparison between the simulation and exact results, as well as the timeseries of both the simulation and comparison.

Similar to the issues discussed in section 6.2 with recreating the same results from the cubic domain, there were also some issues with the results here. Once again RunTime errors appeared at higher resolution counts, but they only came in the form of 'too many polygon' errors. While the errors for the spherical domain started from resolutions of 52, here they didn't start until resolution 68.

Further, there were no issues with certain resolutions not working for specific time steps, like those discussed in the last paragraph of section 6.2. All results up to the resolution of 64 were obtainable here.

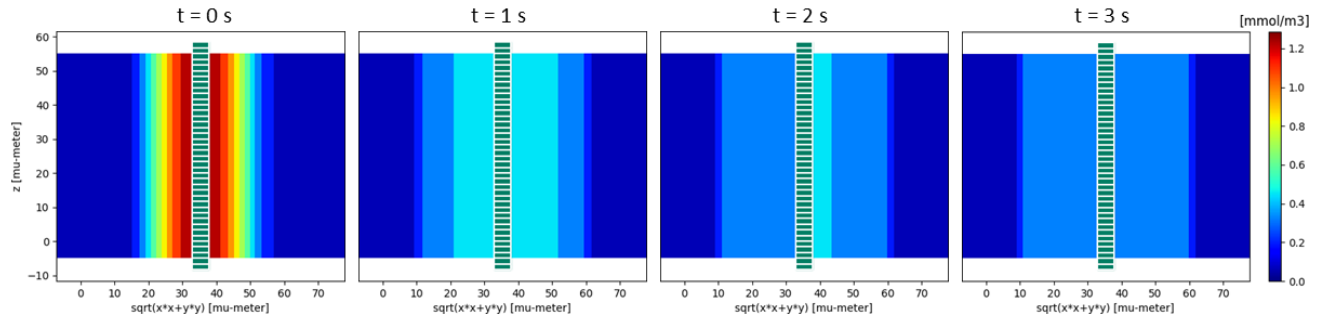
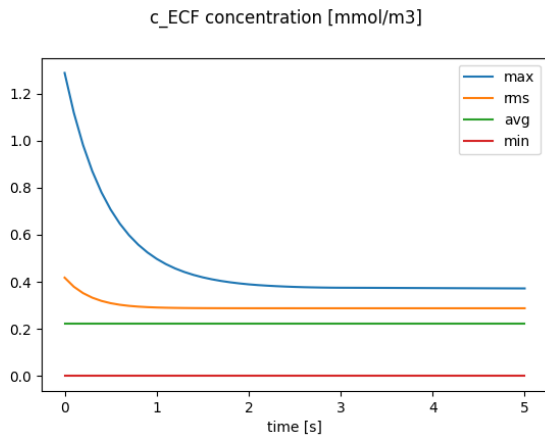
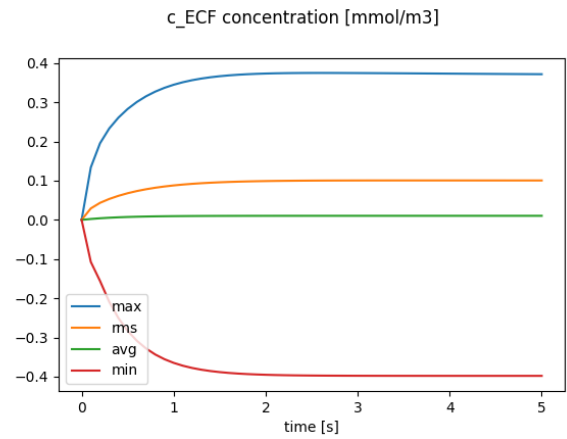


Figure 13: Visualization of the simulation of the cylindrical domain in CapSys. Each image is a cross section of the domain with the cylinder visible in the center. On the y-axis of each image is the z-direction of the domain and in the x-axis is an intersection of the cylindrical element of the domain. On the left at $t = 0$ s the initial condition can be seen. This is as expected from the initial function. As the simulation progresses there seems to be a skew in the concentrations to the right. The values everywhere converge towards the average. At $t = 3$ s the convergence is very close to being finished.

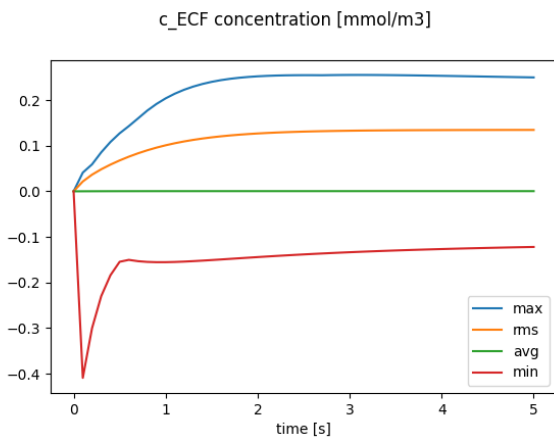


(a) The time series of the concentration of the cylindrical domain is not exactly as expected. The minimum concentration remaining at 0 is expected, since the area outside the cylinder is always at 0. The maximum concentration as well as RMS don't converge to the average concentration as time progresses. Both reach a steady level at some value higher than the average.

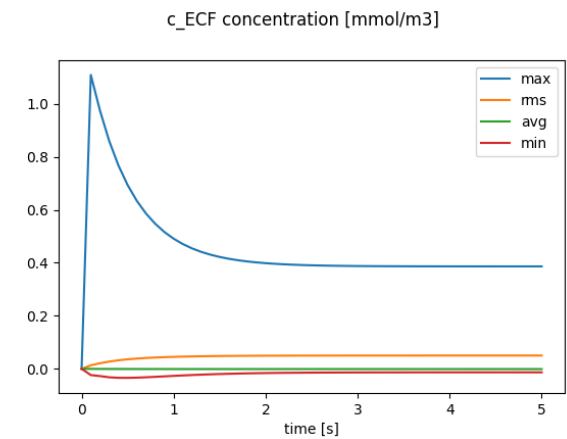


(b) Time series of the comparison between the simulation and exact solution. Looking at the scale on the left of the graph, it seems like the error is in the order of $1E^{-4}m$, about an order larger than in the spherical domain and two orders larger than the cubic domain. The minimum and maximum values are both around equal in magnitude, remaining steady as time progresses instead of converging. The RMS also converges to a value different from the average.

Figure 14: Timeseries of the concentrations in the simulation (14a) and of the comparison between the simulation and the exact solution (14b). Both timeseries are with time steps of 0.1 seconds and resolution of 16.



(a) The time series of the comparison between the simulation and exact solution at resolution of 8 and time steps of 0.1 seconds. This shows the minimum difference appears to go to $-\infty$ as time goes to 0, but it is set to the exact same as the initial function at $t = 0$.



(b) The time series of the comparison between the simulation and exact solution at resolution of 64 and time steps of 0.1 seconds. This shows the maximum difference appears to go to ∞ as time goes to 0, but it is set to the exact same as the initial function at $t = 0$.

Figure 15: Time series at different resolutions that show strange behaviour around $t = 0$, where the behaviour of the minimum (figure 15a) and maximum (15b) appear asymptotic.

6.3.1 Qualitative results

The first results will be the qualitative visual results from the simulation at 16 resolution and 0.1 seconds time steps. In figure 13 it is clearly visible at the images at 1 and 2 seconds into the simulation that the simulation is not symmetric. Even in the images at 3 seconds, it appears that the area where the concentration is not zero extends slightly further. A possible explanation for this may be that the grid is not fully symmetric around the center of the domain used for the simulation. The differences seem significant enough to cause problems at the boundary for the comparison with the exact solution.

Next are the timeseries for both the simulation and the comparison between the simulation and the exact solution. In figure 14a it the maximum value converges to a constant value as expected. This value is not the average, as we expect and saw with the spherical and cubic simulations. There appears no clear reason as to why this is, so this seems like another issue with this simulation. On the right in figure 14b there are also some striking differences with the previous timeseries of this type found in figures 6b and 9b. Here the maximum and minimum differences between the exact solution and simulation do not converge. This also causes the RMS to increase with time to its maximum value at 5 seconds. Finding the maximum RMS at 5 seconds holds for almost all resolutions.

Of course these large differences between the simulation and exact solution should also visibly show up in some of the images. They were found in figure 16. Both the maximum and minimum differences show up in this cross-section in the z-direction. This shows that these differences show up at the edges of the domain. There are issues with almost all sides of the cylindrical part of the domain, but the differences are largest for high values of x and y. This is also where the concentration seemed to skew towards in figure 13.

There are some other resolutions where interesting situations arise. At a resolution of 8, it clearly looks like the minimum difference between the exact solution and simulation would go to negative infinity if the time went to 0. This can be seen in figure 15a. Similarly, at resolutions over 32, the maximum difference seems like it would go to infinity if time went down to 0. The timeseries to illustrate this in figure 15b is at 64 resolution.

6.3.2 Quantitative results

On the quantitative side there is once again convergence with increasing resolution, but similarly to the spherical domain, this is not uniform. Once again this may be linked to how well the grid slots align with the simulation domain. Specifically, the simulation with resolution 32 stands out because it is less accurate than the one with resolution 24.

For all resolutions the error terms of the time steps have smaller differences than 1%, with many not even surpassing 0.1% difference. In many cases, this is even the error term increasing, rather than decreasing, with reduced time steps. This shows there is no notable convergence of the error term with reduction in time steps. Clearly, there seems no linear convergence with time steps, as would be expected.

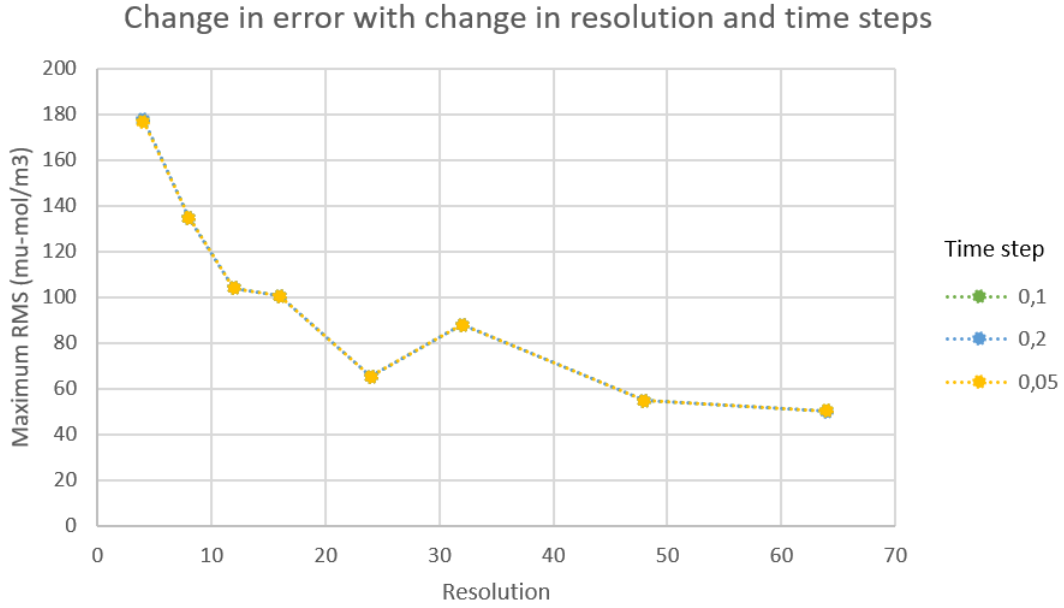


Figure 17: Graph plotting the maximum value for the root mean square (RMS) against the resolution for different time steps. The error seems to decrease with increasing resolution, but not at the same rate as in the cubic domain. Reductions in error with increasing resolution are also not uniform, as shown by the error at resolution 32 being larger than the one at resolution 24. The differences between the errors at different time steps are also negligible.

Δt	Maximum RMS	Factor
0.2	50.00	–
0.1	50.16	0.99
0.05	50.17	1.00

Table 5: The factor of difference in the maximum value of the RMS for different time steps at a resolution of 64. The factors of 0.99 and 1.00 are nowhere near the expected value of 2 if it would converge linearly. It cannot be concluded that the numerical system converges linearly with time in the cylindrical domain.

Δx	Maximum RMS	Factor
8	134.7	–
16	100.7	1.34
32	87.9	1.15

Table 6: The factor of difference in the adjusted value of the maximum RMS for different resolutions. The factors of 1.34 and 1.15 are not close to the expected value of 4. Therefore it cannot be concluded that the numerical system converges quadratically with resolution in cylindrical domain.

This lack of convergence with the expected order of $\mathcal{O}(\Delta t + \Delta x^2)$ is clearly visible in the comparison tables as well (5 and 6). Similar to the spherical case, there is no clear share of the error that is caused by the time step. This means that the maximum RMS cannot easily be adjusted with that value to more accurately compare the convergence in space. In both types of convergence it is clear that it is not well behaved at the expected order.

7 Discussion

In this article we have looked at the implementation of the diffusion equation by the software program CapSys, that intends to simulate drug distribution through the brain. Exact solutions to the equations have been found and used to compare with the software implementation. After that the expectations of the software performance were established, with expected convergence of $\mathcal{O}(\Delta x^2 + \Delta t)$ from the simulation to the exact solution. We have found that the implementation of the diffusion equation in an empty square domain converges at the expected rate. When the domain is not square and contains a cell or capillary, we do not find convergence at the expected rate.

Overall, the implementation of the diffusion equation in CapSys is successful, as seen by the accuracy of the convergence shown in section 6.1.2. The cubic domain offers no challenges to the software and is a pure assessment of numerical method and its convergence. The results of the rate of convergence match the expectation that was determined in 5.3.1, this tells us that the implementation of the equation is correct.

Issues do start appearing when the domain does not neatly follow the cubic grid spaces as set by the resolution. The addition of a spherical outer boundary and central cell in the spherical domain (as described in section 3.5.2) introduced errors in the software. At several resolutions, we found specific areas where the simulation did not behave correctly. These areas seem to be at the boundary of the domain, cell or capillary. Visually, it appears that these are areas that are outside the simulated domain and the concentration there should have been set to 0. It seems most of these issues appear around the cell and not the outer boundary of the domain. Since the software intends to use cells both as targets and obstacles for diffusion, it is important to improve the interaction between an initial concentration and the cells. Since these issues didn't appear at lower resolutions, such as 16 and 24, but did at resolutions of 40 and 48, tells us that simply increasing the resolution further may not be a sufficient solution.

Since the quantitative results discussed in 6.2.2 show that the convergence does not follow the expected value of $\mathcal{O}(\Delta t + \Delta x^2)$. This means that for any domain with spherical elements, such as cells and capillaries, that rate of convergence cannot be expected. If the use of larger 3D networks is required, then higher resolutions are a necessity for accuracy. With resolutions that are too low, issues such as illustrated in figure 4 may appear, with many of the grid slots cut off by capillaries or cells. For this, the 'too many polygons' error message needs to be resolved.

These issues with the spherical domain appear to carry over in the cylindrical domain with the results shown in section 6.3. Once again there are issues around the edges of circular parts of the domain. There is also no visible convergence with differences in time steps.

The issues found in the cylindrical domain mostly appear around the outer edge of the domain. The area around the central capillary seems to work as intended. This could imply that the issues disappear in a cubic 3D-unit network as discussed in section 3.4. In that case the outer boundaries

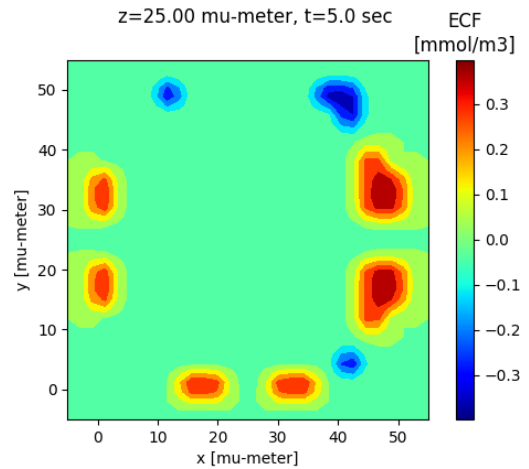


Figure 16: The large maximum and minimal comparison differences seen in figure 14b were visually found in a z-directional cross section. These all seem to be around the edge of the spherical section of the domain. The largest differences are seen at higher values of x and y , which corresponds with the skew seen in 13.

are of the cubic domain type with the embedded capillaries causing no issues.

This article focused entirely on the diffusion equation. To get a more complete picture of the accuracy of CapSys in simulating the drug distribution in the brain further investigation is needed. Especially the addition of inflow of drug through the blood vessels and flux through the blood brain barrier is an important addition that should be made in future studies. Further, the cells were considered empty obstacles here, while in reality there is diffusion through their membranes and there are important drug targets inside cells and on their membranes. It is possible that many issues around the cell membranes found in this article may be less pronounced when cells are permeable to some degree.

Finally, there are several possibilities for improving CapSys. A solution should be found to the issues that have appeared near the central cell in the spherical domain. Since the full brain model contains many more cells, these issues can cause more problems. Another avenue for improvement could be found in the speed with which the program can process certain operations, such as matrix inversion. Using more efficient heuristic solutions to matrix inversion may allow for the program to process domains with higher resolution more easily. These higher resolutions are likely a necessity if you want to model domains with multiple cells.

Bibliography

1. De Lange ECM, Rottschäfer V, Vendel E. The need for mathematical modelling of spatial drug distribution within the brain. *Fluids Barriers CNS*. 2019;16(12).
2. De Lange ECM, Rottschäfer V, Vendel E. A 3D brain unit model to further improve prediction of local drug distribution within the brain. *PLOS ONE*. 2020;15(9):1-24.
3. De Lange ECM, Rottschäfer V, Vendel E. The 3D Brain Unit Network Model to Study Spatial Brain Drug Exposure under Healthy and Pathological Conditions. *Pharm Res*. 2020;37(137).
4. Parcelsus . Dritte defensio. 1538.
5. Widmaier EP, Raff H, Strang KT. *Vander's Human Physiology*. McGraw-Hill 12th international students ed. 2013.
6. Rang HP, Ritter J, Flower R, Henderson G, Loke YK, MacEwan D. *Rang and Dale's Pharmacology*. Elsevier:7th ed. 2013.
7. *OpenStax, Anatomy & Physiology*. OpenStax CNX. 2016.
8. Lei Y, Han H, Yuan F, Javeed A, Zhao Y. The brain interstitial system: Anatomy, modeling, in vivo measurement, and applications. *Prog Neurobiol.* 2017 Oct;157:230-246.
9. Alpern RJ, Moe OW, Caplan M. , eds. *Seldin and Giebisch's The Kidney (Fifth Edition)*. Academic Press:5th ed. 2013.
10. Rowland M, Tozer TN. *Clinical Pharmacokinetics & Pharmacodynamics: Concepts and applications*. Wolters Kluwer:4th ed. 2013.
11. Schooley RT, Carlin AF, Beadle JR. Rethinking Remdesivir: Synthesis of Lipid Prodrugs that Substantially Enhance Anti-Coronavirus Activity. *Preprint. bioRxiv*. 2020.
12. National Cancer Institute NIOH. Illustration of blood vessels including artery, arteriole, capillaries, vein and venule..
13. Geldenhuys W, Mohammad A, Adkins C, Lockman P. The blood–brain barrier is a specialized micro-vascular unit comprised of several cell types..
14. Wong AD, Ye M, Levy AF, Rothstein JD, E BD, Searson PC. The blood-brain barrier: an engineering perspective. *Front. Neuroeng.* 2013;6(7).
15. Reese TS, Karnovsky MJ. Fine structural localization of a blood-brain barrier to exogenous peroxidase. *J Cell Biol.* 1967;34(1):207-217.
16. Kawai , Tadashi . *Bodily Distribution of the Plasma Proteins*:114–130. Berlin, Heidelberg: Springer Berlin Heidelberg 1973.
17. Abramowitz M, Stegun IA. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. New York: Dover ninth dover printing, tenth gpo printing ed. 1964.

A Solution to spherical Bessel functions

```
1 r_0 = 4.5*10^-6;
2 r_1 = 2.5*10^-5;
3
4 syms x;
5 eqn = (x^2*r_1*r_0*tan(x*r_0)+x*(r_0-r_1)-tan(x*r_0))/(-x^2*r_1*
        r_0 + x*tan(x*r_0)*(r_1+r_0)-1)-tan(x*r_1) == 0;
6 fplot([lhs(eqn) rhs(eqn)], [0 300000]);
7 lambda = double(vpasolve(eqn, x, [285000 300000]));
8 disp(['lambda = ', num2str(lambda)])
9
10 b = (tan(lambda*r_0)-lambda*r_0)/(lambda*r_0*tan(lambda*r_0)-1);
11 disp(['b = ', num2str(b)]);
```

B Solution to cylindrical Bessel functions with no inflow

```
1 r_0 = 2.5*10^-6;
2 r_1 = 2.5*10^-5;
3
4 syms x;
5 eqn = besselj(1,x*r_1)/bessely(1,x*r_1) == besselj(1,x*r_0)/
        bessely(1,x*r_0);
6 fplot([lhs(eqn) rhs(eqn)], [1 200000]);
7 nu = double(vpasolve(eqn, x, [1 200000]));
8 disp(['nu = ', num2str(nu)])
9
10 b = -besselj(1,r_1*nu)/bessely(1,r_1*nu);
11 disp(['b = ', num2str(b)]);
12
13 f = -nu*besselj(1,nu*r_0) - nu*b*bessely(1,nu*r_0);
14 g = -nu*besselj(1,nu*r_1) - nu*b*bessely(1,nu*r_1);
15 disp(['f = ', num2str(f)]);
16 disp(['g = ', num2str(g)]);
```