UNIVERSITY OF BERGEN

Renal Function Estimation using Magnetic Resonance Images

Author:
Eyram Schwinger

Supervisor:
Antonella Z. Munthe-Kaas

Thesis submitted in partial fulfilment of the requirements for the degree of Master of Science in Mathematics

Department of Mathematics
Faculty of Mathematics and Natural Sciences
University of Bergen
Bergen, Norway
2009

Acknowledgements

No work is complete without the support, criticisms and encouragement from different groups of people. This is the page where I get to say thank you to all those without whom this project would only have been a dream. First of all, I want to thank the Almighty God from whom all good and perfect gifts come. He has been my strength through the difficult times times.

I also want to thank my supervisor, Antonella Zanna Munthe-Kaas. You have been there through the good and the bad times. I remember when I finished my first semester exams and things did not go as I planned. You provided a smile to radiate a dark period and a hand to reach out to and that I will never forget. To the Bergen BildebehandlingsGruppe (BBG) who were there for my seminar sessions, asked insightful questions and provided feedback, your input was very much appreciated. To Helwig Hauser, your answers to my questions regarding the visualization chapter is greatly appreciated. To Are Losnegård, thanks for taking time off your busy schedule to answer my questions concerning MRI and the kindey. Your replies were always prompt.

To my mother Grace Sakoe and my brother-in-law Pastor Domenyo Hoedzoadey I want to say thanks for your prayers and encouragement. Your ability to spur me on in times of discouragement has been appreciated. To my girlfriend Gladys Quaye, time and space could still not set us apart. You cried and laughed with me, lent a listening ear at all times and trusted in my abilities even when I had no reason to trust in them. You are the brightest part in my world.

To my friend Justine Namakula, thank you for taking time to read through this project and providing insights and spotting those grammatical errors. It has been fun knowing you these two years and friends like you are kept for life.

To everyone else who has contributed in one way or another towards the successful completion of this project, I say thank you and may the good Lord continue to shine His grace upon you.

Contents

Abstract

This study is aimed at testing the use of Magnetic Resonance (MR) images and mathematical models for renal parameter estimation.

The study was based on four models; the Patlak model, Cortical Compartment model, Separable Compartment/Sourbron model and Deconvolution method.

The project included the mathematical derivation of the model. The models were then applied to whole Kidney MRI images in order to get and compare parameters. Part of the project also included the visualization of the parameters produced by the models on a voxel-by-voxel basis.

The project showed that of the four models the Deconvolution method produces the highest parameter values followed by the Patlak model. The Cortical Compartment and Sourbron models produce almost similar results. The voxel-by-voxel visualization also showed that only the renal cortex produces high flow results.

Chapter 1

Background

Human beings generally have two kidneys but can live with only one, but no human being can live without a kidney. The kidneys are organs that regulate the volume of body fluid, acidity and mineral composition. The kidneys perform these functions through the processes of glomerular ultrafiltration, tubular reabsorption and tubular excretion. This is very important because the body needs only a minimum amount of most minerals. Too much of these minerals are harmful to the body. The kidney therefore helps regulate these substances. Certain diseases like diabetes however cause the kidneys to fail to perform their function. People faced with such problems might have to undergo dialysis and others undergo kidney transplants. It is therefore important to know the rate of kidney function especially for people with such diseases. One important measure of renal function is the Glomerular filtration rate. Glomerular filtration rate (GFR) is the rate of flow of filtered fluid through the kidney. It is so far the best test of kidney function.

1.1 The Glomerular Filtration Process

As blood flows through the aorta into kidneys, it flows into the renal arteries which has a smaller diameter and then into the arterioles whose diameter is also smaller than the arteries. The reduction in the vessel diameter causes an increase in the pressure of the blood. This causes the blood to exert pressure on the walls of the vessels. This is called hydrostatic pressure. The walls of the capillaries within the glomeruli are semi-permeable and allow small particles to be filtered through. Mineral salts and acids are small enough and are filtered into the bowman's capsule. Blood plasma and proteins on the other hand are bigger and therefore stay inside the arterioles.

Another type of pressure that acts on the blood is oncotic pressure. This happens because blood plasma displaces water from the blood which causes the water concentration in the blood to be low. On the other hand, water concentration in the tissues ouside the blood vessels is high. The process of osmosis forces water from the extracellular tissues into the blood vessels.

In the kidney, glomerular filtration is possible because the hydrostatic pressure in the glomerulus is higher than oncotic pressure. The Bowman's capsule also exerts its own hydrostatic and oncotic pressure on the glomerulus. The net filtration within the glomerulus is the GFR and is expressed mathematically by Starling's law as

$$GFR = L_p \cdot S \times (\Delta \text{ hydrostatic pressure } - \Delta \text{ oncotic pressure})$$

$$GFR = L_p \cdot S \times [(H_g - H_b) - s(O_g - O_b)],$$

where L_p is the unit permeability in the capillary wall and S is the surface area available for filtration. H_g, H_b are the hydrostatic pressure in the glomerular capillaries and the Bowman's capsule respectively. O_g, O_b the oncotic pressure in the glomerulus and the Bowman's capsule respectively and s is the reflection coefficient of proteins across the capillary walls. If the capilliary walls are permeable, $s = 0$ whereas $s = 1$ if they are impermeable. Since blood plasma and proteins do not filter into the Bowman's capsule, blood plasma and proteins are absent in the Bowman's capsule hence $O_b = 0$ and $s = 1$ which reduces the equation to

$$GFR = L_p \cdot S \times [(H_g - H_b) - O_g].$$

1.2 Traditional Method Of Calculating GFR

The primary method for calculating GFR in hospitals is the Creatinine Clearance test. Creatinine is a chemical produced naturally in the human body as a result of the breakdown of creatinine phosphate in the muscles. The amount of creatinine produced by the human body is constant. To test for GFR, a urine sample is collected over a twenty-four hour period from the patient. The urine is then tested for creatinine concentration and compared to the concentration of creatinine in a sample of blood from the patient at the end of the twenty-four hour period. From these measurements, creatinine clearance is given as

$$C = \frac{U_c \times V}{P_c},$$

where U_c is the concentration of creatinine in the urine, V is the urine flow rate and P_c is the concentration of creatinine in blood plasma. This method, however is not totally accurate since some amount (very little) of creatinine is reabsorbed by the body. Inulin on the other hand, is a polysaccharide used by plants to store energy. Inulin is totally excreted by the human body and therefore produces a more accurate measurement of GFR. Calculating GFR from inulin is a more complicated process used only for reseach purposes.

1.3 Statement Of The Problem

The traditional method of estimating GFR produces values for the whole human body and does not differentiate between the left or the right kidney. It can therefore predict kidney mulfunction or failure but cannot tell which of the kidneys is failing.

The method also produces one parameter; the GFR and does not provide any other parameters of renal function. This creates the need for the use of other methods in order to get detailed information about renal function.

1.4 Aim Of This study

The aim of this study is the estimation of GFR from registered tracer enhanced 3-dimensional Magnetic Resonance Images (MRI).

Signal intensity / time series will be obtained from these images and perfusion and GFR will be calculated using various tracer kinetic models. This study is based on the use of four models. These are: Patlak model, Cortical compartment model, Sourbron/Separable compartment model and Deconvolution method. The thesis begins with the derivation of the mathematical models from the basic assumptions used in the modelling process. The data are then prepared from the available 4-dimensional MRI images. The models are then tested on the data to obtain single kidney perfusion information. Part of the thesis will also be a visualization of voxelwise measurement of GFR on the prepared kidney data.

Chapter 2

Mathematical Models

2.1 Introduction

The success of renal parameter estimation with MRI images relies on well built models that simplify, yet accurately represent the processes that take place within the kidney. Renal paramenter estimation with MRI images is one of the topics at the core of medical research today. A lot of progress is being made in that direction from the simple Patlak plot technique to the much more complex Cortical Compartment Model, but nothing concrete has been implemented in healthcare facilities yet.

The MRI images used in these models provide two paramenters for the modelling process. The first is the region of interest curve denoted $C_{roi}(t)$ which represents the average attenuation per time t over the region from where the renal parameters are to be calculated. In the case of a voxel-by-voxel analysis, the attenuation curve for each voxel is its own region of interest curve. In whole kidney analysis however, this is either the average attenuation over the whole kidney parenchyma or just the renal cortex. The second parameter provided by the MRI images is the arterial input function also called the aortic attenuation function. This function represents the amount of tracer supplied to the kidney by the aorta at a given time t. This is represented by $C_a(t)$.

The curves obtained from MRI images are signal intensity/time curves, which for most research purposes are transformed into concentration/time curves which represent the concentration of the tracer within the system. One method of making this conversion is used by Aumann et al. [4] as

$$C(t) = \frac{-k}{TE} \cdot \ln(\frac{S(t)}{S_0}) \ .$$

Where TE is the echo time which can be retrieved from the diacom file. S_0 denotes the baseline signal. The average baseline signal can be used in this case to account for noise during image acquisition. $S(t)$ denotes the signal level at time t and k is a proportionality constant which relates the change in relaxation rate and the concentration $C(t)$. To find k, information is needed on the characteristics of the contrast agent, pulse sequence and tissue. The raw signal intensity/time curves were used in this project because no information on the contrast agent were available for the images used.

In this chapter four methods for calculating renal parameters from MRI were analyzed. They are the Patlak Plot technique [14, 15], the Cortical Compartment Model [3], the Separable Compartment Model [26] and the Deconvolution Method [16].

The first three of the methods listed are two compartment models. What makes the difference between the parameters returned by these models are the processes that are assumed to take place within the compartments. The general assumptions made in these models however is that the contrast agent is unidirectional, which is an accurate depiction of blood flow within the human body in general. As a result of the combination of valves, arteries and veins, blood and contrast agent carried by the blood move in one direction. Two principles are used frequently in this chapter. These are stated below as theorems:

Central Volume Principle. *The normalized first moment of the efflux concentration curve or mean transit time MTT, is the ratio of the tissues's volume to its flow.*

Conservation of Mass. *Mass cannot be created or destroyed. This can also be stated as the amount of mass within a closed system remains constant regardless of what processes take place within the system.*

Since the input used in this project is the concentration vrs time curves of a contrast agent, mass as used here is calculated as

$$mass = concentration \times volume. \tag{2.1}$$

2.2 The Patlak Model

Patlak model for the assessment of renal function is an adaptation of the original Patlak plot technique originally used for the evaluation of nuclear medicine imaging after the injection of a radioactive tracer. The model uses the whole kidney parenchyma for renal parameter estimation and assumes that:

1. There is a linear relation between the renal parameters.
2. The vascular and nephron space are the two compartments used in this model.
3. The contrast agent is completely mixed within these two compartments.
4. The outflow of the contrast agent from the nephron space is negligible, hence assumed to be zero.

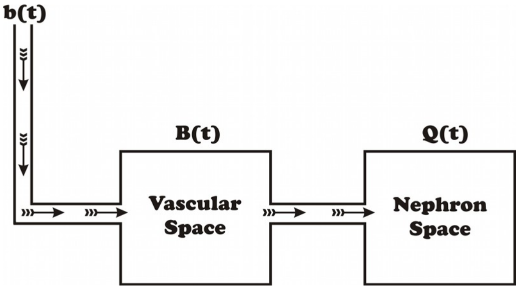

Figure 2.1: An Illustration of the Patlak Model

2.2.1 Description Of The Patlak Model

Let $K(t)$ be the total amount of tracer in the kidney region of interest at time t and $B(t)$ and $Q(t)$ be the amount in the vascular and the nephron space respectively. $b(t)$ is the mass of contrast agent supplied by the arteries at time t. It is assumed that $b(t)$ will be carried instantaneously into the vascular space. The mass of tracer within the vascular space is therefore assumed to be proportional to $b(t)$

$$B(t) = c_1 \cdot b(t) \; . \tag{2.2}$$

Where c_1 is a constant representing the proportion of the contrast agent that flows into the vascular space. From assumption three, the contrast agent that flows into the nephron space is not dissipated. This means that the system is a closed system with a single source and no sinks. The amount of tracer in the nephron space will therefore be increasing and the rate of increase is proportional to $b(t)$ and is given by

$$\frac{dQ(t)}{dt} = c_2 \cdot b(t) \; . \tag{2.3}$$

Solving this equation, we arrive at the equation representing total amount of contrast agent in $Q(t)$ in the nephron space which is given by

$$Q(t) = c_2 \cdot \int_0^t b(y)dy + M \ . \tag{2.4}$$

At time $t = 0$, there is no contrast agent in the nephron space so the above equation becomes

$$Q(t) = c_2 \cdot \int_0^t b(y)dy \ . \tag{2.5}$$

From conservation of mass, the total mass of contrast agent in the whole system is equal to the sum of the masses in the vascular and the nephron space.

$$K(t) = B(t) + Q(t) \ . \tag{2.6}$$

Substituting equations 2.2 and 2.5 into 2.6 results in the fundamental equation of the Patlak model

$$K(t) = c_1 \cdot b(t) + c_2 \cdot \int_0^t b(y)dy \ . \tag{2.7}$$

Dividing equation 2.7 by $b(t)$ results in the equation

$$\frac{K(t)}{b(t)} = c_1 + c_2 \cdot \frac{\int_0^t b(y)dy}{b(t)} \ , \tag{2.8}$$

which is of the form

$$Y = c_1 + c_2 \cdot X \ ,$$

which is the equation of a straight line and this is the central idea behind the Patlak model. The constant c_2 is the slope of the line and represents the GFR and c_1 the y-intercept represents the size of the vascular space.

2.2.2 Calculating Input Parameters Using the Patlak Model

To find the whole kidney $K(t)$, each 3D volume of the kidney was converted to a vector. The result of this process is a set of time series data, one for each signal intensity / time curve of a voxel. The time series were then stripped of all data that were not part of the kidney region of interest. The method used to achieve this was to delete a time series row from the data array if the sum of the row was equal to zero. This worked because those sections of the MRI which were not part of the kidney had been set to zero by the segmentation process. After this process, the number of rows left in the array represents the actual number of voxels that make up the kidney. The array was then averaged by columns to create a row data which represents the average attenuation of the region of interest. The minimum of the average attenuation curve was subtracted from the average attenuation to create a baseline shift which sets the minimum of the attenuation curve before contrast arrival to zero. The result was multiplied by the kidney volume of the kidney which was calculated as the volume of a single volume as obtained from the diacom file multipled by the total number of voxels which represents the kidney region of interest. The final result of these processes is $K(t)$, which is the net attenuation of the kidney.

$b(t)$ was calculated using the same procedure as that used for $K(t)$. The input in this case was the arterial region of interest segmented from the MRI images and the volume calculated from the number of voxels used to calculate the arterial input function $C_a(t)$.

Alternatively, $C_a(t)$ can be used for the value of $b(t)$ while $C_{roi}(t)$ is used for the value of $K(t)$. In this case, since the values are not multiplied by the volume of the regions, the input being used is equal to the mass per unit volume. The c_2 calculated has to be multiplied by the volume of the region of interest to get the whole volume GFR.

2.2.3 Solving The Patlak Model

Since the Patlak model is linear in nature, the key to solving it is discretizing the integration of $b(t)$ and solving the resulting system of equations. To discretize the integral, the trapezoidal rule for unequal intervals was used since the image aquisition time for the data was not with equal intervals

$$\int_0^t b(y)dy = \sum_{i=0}^{t} \frac{b_i + b_{i+1}}{2}(t_{(i+1)} - t_i) \ .$$

Alternatively, the function $b(t)$ can be interpolated using either linear or cubic spline. It can then be integrated by using the trapezoidal rule for equal intervals.

Solving this system of equations is equivalent to solving an $n \times 2$ matrix. According to [15], if the values of $K(t)$ and $b(t)$ and its intergral are known for two time periods t_1 and t_2, then the system can be set up as a simultaneous equation

$$K(t_1) = c_1 \cdot b(t_1) + c_2 \cdot \int_0^{t_1} b(y)dy \ , \tag{2.9}$$

$$K(t_2) = c_1 \cdot b(t_2) + c_2 \cdot \int_0^{t_2} b(y)dy \ . \tag{2.10}$$

Solving for c_1 and c_2 in the above results in the equations

$$c_1 = \frac{K(t_2) - c_2 \cdot \int_0^{t_2} b(y)dy}{b(t_2)} \ , \tag{2.11}$$

$$c_2 = \frac{\frac{K(t_1) \cdot b(t_2)}{b(t_1)} - K(t_2)}{\frac{b(t_2)}{b(t_1)} \int_0^{t_1} b(y)dy - \int_0^{t_2} b(y)dy} \ . \tag{2.12}$$

The problem with this method however is that the dataset in use has unequal intervals, meaning that the resulting values of c_1 and c_2 will be dependent on the times chosen for the calculation. In addition to the above, although the Patlak model assumes a linear relation between the data, the dataset used are nonlinear in nature and simultaneous equations are much more suitable for linear data. This means that simultaneous equations would not work in this situation. Simultaneous equations are therefore not used in this project. The Patlak model results in an $n \times 2$ system of equations

$$Bx = k \tag{2.13}$$

where the vector x contains the unknown parameters c_1 and c_2. The first column of the matrix B contains the vector $b(t)$ and the second column contains the values of the cumulative integral $\int_0^{t_n} b(y)dy$ for each time t_n. Using the normal equations, equation 2.13 becomes

$$B^T Bx = B^T k, \tag{2.14}$$

which is now a 2×2 system of equations which can easily be solved as

$$x = V_B \Sigma_B U_B^{'} l, \tag{2.15}$$

where $l = B^T k$, the assumption of negligible outflow of contrast agent leads to the overestimation of GFR which in this case is c_2. This is usually compensated for by hermatocrit correction. The hermatocrit is the amount of the blood volume that is occupied by the red blood cells. Hermatocrit level can be estimated from the unenhanced $b(t)$ as [15]

$$hct = 0.0083 \cdot b_{un} + 0.0244.$$

From this value the GFR can be calculated from c_2 as

$$GFR = (1 - hct) \cdot c_2. \tag{2.16}$$

2.3 The Cortical Compartment Model

The Cortical Compartment model uses only the renal cortex for the region of interest in its renal paramater estimation function. The model makes the following assumptions;

1. The two compartments that are responsible for glomerular filtration are the glomeruli and the renal tubules.

2. Both inflow and outflow of contrast agent are assumed in this model. Compared to assumption 4 of the Patlak model, the Cortical compartment model does not neglect the outflow of contrast agent.

3. A slight delay in the arrival time of the contrast agent in the cortex from when it is seen in the aortic region.

4. A dispersion of the contrast agent within the glomeruli.

In the cortical compartment model, the contrast agent is carried by the arterial input function $C_a(t)$, delayed and dispersed into the glomeruli.

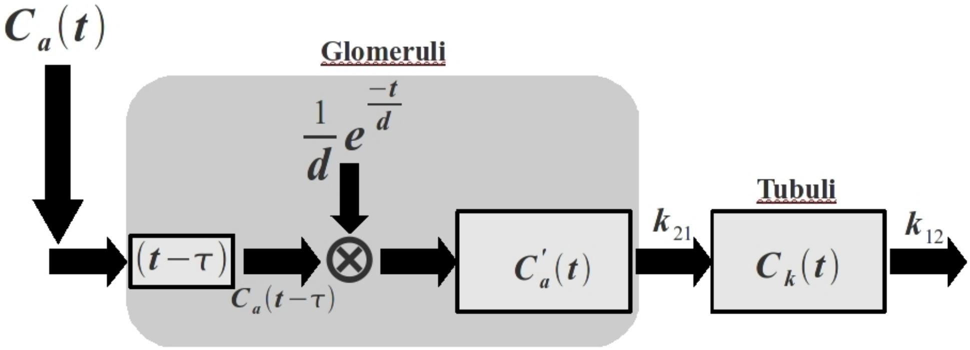

Figure 2.2: A pictorial depiction of the cortical compartment model. The contrast agent is carried by k_{21} from the glomerulus $C_a^{'}$ into the renal tubules C_k it is then carried out of the renal tubules by k_{12}, k_{21} is the GFR.

This model takes into the consideration the fact that glomerular filtration takes place in the renal cortex and not in the kidney as a whole. The contrast agent as seen in the glomeruli $C_a^{'}(t)$ is assumed to be equal to the arterial input function $C_a(t)$ shifted in time and dispersed according to the function [25]:

$$C_a^{'}(t) = C_a(t-\tau) \otimes (\frac{1}{d})e^{-\frac{t}{d}} \ , \tag{2.17}$$

where τ is a time delay and d is a dispersion constant. $\otimes$ denotes the convolution operation.
The flow k_{21} carries the contrast agent into the renal tubules from the glomeruli and the flow k_{12} carries the contrast agent from the renal tubules out of the system. The change in the amount of contrast agent in the tubules at time t is therefore proportional and is given by the equation:

$$\frac{dC_k(t)}{dt} = k_{21}C_a^{'}(t) - k_{12}C_k(t) \tag{2.18}$$

Solving this model using the technique of integrating factor, equation 2.18 becomes

$$\begin{aligned}\frac{dC_k(t)}{dt} + k_{12}C_k(t) &= k_{21}C_a^{'}(t)\\ \mu(t)\frac{dC_k(t)}{dt} + \mu(t)k_{12}C_k(t) &= \mu(t)k_{21}C_a^{'}(t)\end{aligned}$$

Where $\mu(t)$ is the integrating factor that has to be found. From this, it is seen that

$$\begin{aligned}\frac{d\mu(t)}{dt} &= k_{12}\mu(t)\\ \Rightarrow \ln\mu(t) &= k_{12}t + m\\ \mu(t) &= Me^{k_{12}t}\end{aligned}$$

Substituting this back into the differential equation gives:

$$\begin{aligned}e^{k_{12}t}\frac{dC_k(t)}{dt} + k_{12}e^{k_{12}t}C_k(t) &= k_{21}e^{k_{12}t}C_a^{'}(t)\\ \frac{d}{dt}[Me^{k_{12}t}C_k(t)] &= k_{21}e^{k_{12}t}C_a^{'}(t)\\ e^{k_{12}t}C_k(t) &= \int_0^t k_{21}e^{k_{12}y}C_a^{'}(y)dy\\ C_k(t) &= \int_0^t k_{21}e^{k_{12}y}e^{-k_{12}t}C_a^{'}ydy.\end{aligned}$$

This leads to the equation

$$C_k(t) = k_{21}\int_0^t e^{-k_{12}(t-y)}C_a^{'}(y)dy\ . \tag{2.19}$$

Equation 2.19 can also be written as

$$C_k(t) = k_{21}e^{-k_{12}t} \otimes C_a^{'}(t)\ .$$

The total mass of the contrast agent inside the system is the sum of the mass within the glomerular compartment and that in the tubular compartment.

$$C_{roi}(t) = C_a^{'}(t) + C_k(t) \tag{2.20}$$

Since the value $C_a^{'}(t)$ is the mass per unit volume of the glomeruli compartment which is also the plasma compartment, then, according to the principle of buoyancy which states that an object displaces a volume of liquid equal to its own volume, the contrast agent within the plasma compartment displaces a volume of plasma equal to its own volume. Therefore the contrast agent takes up a fraction f_a of the plasma space and this has to be accounted for in the mass balance equation. 2.20 becomes

$$C_{roi}(t) = f_a C_a^{'}(t) + C_k(t), \tag{2.21}$$

Where f_a is the fractional plasma volume - the fraction of plasma space taken up by the contrast agent. Substituting equation 2.19 into equation 2.21 gives the equation

$$C_{roi}(t) = f_a C_a^{'}(t) + k_{21} \int_0^t e^{-k_{12}(t-y)} C_a^{'}(y) dy. \tag{2.22}$$

Equations 2.17 and 2.22 make up the cortical compartment model with five parameters f_a, τ, k_{12}, k_{21}, d.

2.3.1 Implementation Of The Cortical Compartment Model

Since the functions provided by MRI images are discrete and finite, the integrals in the model are discretized. In addition, since the convolution operation is being used, the input data $C_a(t)$ and $C_{roi}(t)$ are interpolated. The time shift operation also requires that the data from the arterial input function $C_a(t)$ be right shifted, which makes its sample times different from the sample times of the original function and the region of interest function. The new function $C_a(t-\tau)$ must therefore undergo both interpolation and extrapolation to resynchronize with the original time samples. However, since it is known that before the start time of the original function, there is no contrast enhancement, the extrapolatin function used is a constant function with value equal to the minimum or the average of the unattenuated part of the arterial input function $C_a(t)$. The integral is also replaced with the cumulative discrete trapezoidal integral. Let

$$f(y) = e^{k_{12}y} C_a^{'}(y),$$

Then

$$T(t) = \int_0^t f(y)dy = \frac{\Delta y}{2}(f(t_0) + 2f(t_1) + 2f(t_2) + \cdots + 2f(t_{n-1}) + f(t_n)),$$

where n is the number of time samples. Then the equation of the cortical compartment model gives

$$C_{roi}^{est} = f_a C_a^{'}(t) + k_{21}e^{-k_{12}t} \cdot T(t). \tag{2.23}$$

The parameters of the two equations that make up the cortical compartment model can then be found using nonlinear least squares.

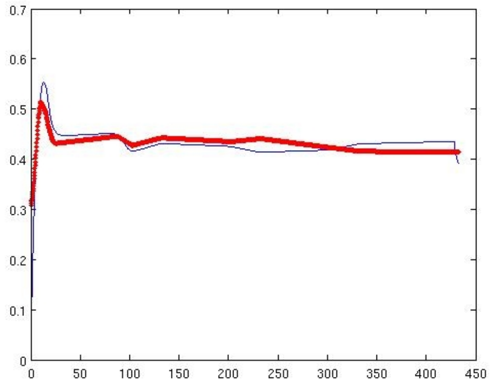

Figure 2.3: The function C_{ROI} (line) and the refitted version (thick line)

2.4 The Separable Compartment / Sourbron Model

The Separable Compartment Model also divides the kidney into 2 compartments, the plasma and tubular compartment. This model assumes the following:

1. Plasma flow denoted by F_P carries the contrast agent from the arteries $C_a(t)$ into the kidneys.

2. Time delay in humans is negligible, hence set to zero.

In the kidneys, the contrast agent first enters into the vascular system where it distributes over the plasma volume denoted V_P. Through ultrafiltration, a fraction of the contrast agent is transported by the tubular flow where is distributes over the tubular volume V_T [26]. From there, the contrast agent leaves the kidney transported by vascular and tubular outflow. Figure 2.4 shows the separable compartment model.

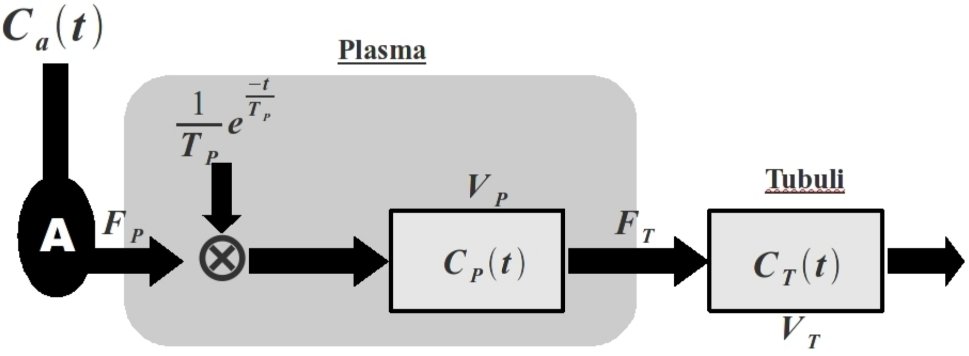

Figure 2.4: The Separable Compartment Model

2.4.1 Description Of The Separable Compartment Model

Let the mean transit time in the plasma and tubular compartments be denoted T_P and T_T respectively. The model assumes that the contrast agent in in the arteries $C_A(t)$ is dispersed within the plasma region by an exponential function [26]

$$C_P(t) = T_P^{-1} e^{-\frac{t}{T_P}} \otimes C_A(t). \tag{2.24}$$

Since some reabsorption of the contrast agent takes place within the tubuli, not all the contrast agent within the tubuli flows out of the tubuli. A fraction f of the contrast agent is reabsorbed, then $(1-f)$ of the contrast agent will then flow out. By the concentration of mass, the change of tracer mass in the tubular compartment is the difference between the concentration that flows into the tubular compartment from the plasma compartment and the concentration that flows out

$$\frac{dV_T C_T}{dt} = F_T C_P - (1-f) F_T C_T \ . \tag{2.25}$$

Although the volume of most organs in the human body is affected by deformation due to respiration and other motions, if we assume the change in time dt is so small that the volume of the tubular compartment is fixed, then equation 2.25 can be rewritten as

$$V_T \frac{dC_T}{dt} = F_T C_P - (1-f) F_T C_T \ . \tag{2.26}$$

We solve this equation using integrating factor

$$\begin{aligned}
&V_T\frac{dC_T}{dt} = F_TC_P - (1-f)F_TC_T \\
&\frac{dC_T}{dt} + \frac{(1-f)F_T}{V_T}C_T = \frac{F_T}{V_T}C_P \\
&\mu(t)\frac{dC_T}{dt} + \mu(t)\frac{(1-f)F_T}{V_T}C_T = \mu(t)\frac{F_T}{V_T}C_P \ .
\end{aligned}$$

Where $\mu(t)$ is the integrating factor. From this we get

$$\frac{d(\mu(t)C_T)}{dt} = \mu(t)\frac{dC_T}{dt} + C_T\frac{d\mu(t)}{dt} \ .$$

From which we get

$$\frac{d\mu(t)}{dt} = \frac{(1-f)F_T}{V_T}\mu(t) \ .$$

Which we solve to arrive at

$$\mu(t) = ce^{\frac{(1-f)F_T}{V_T}t} \ .$$

Which we plug back into our differential equation to get

$$\begin{aligned}
&\frac{d}{dt}(e^{\frac{(1-f)F_T}{V_T}t}C_T(t)) = e^{\frac{(1-f)F_T}{V_T}t}\frac{F_T}{V_T}C_P(t) \\
&e^{\frac{(1-f)F_T}{V_T}t}C_T(t) = \int_0^t e^{\frac{(1-f)F_T}{V_T}y}\frac{F_T}{V_T}C_P(y)dy \\
&C_T(t) = \frac{F_T}{V_T}\int_0^t e^{-\frac{(1-f)F_T}{V_T}(t-y)}C_P(y)dy
\end{aligned}$$

Which is the convolution equation

$$C_T(t) = \frac{F_T}{V_T}e^{-\frac{t}{T_T}} \otimes C_P(y) \ , \qquad (2.27)$$

where

$$T_T = \frac{V_T}{F_T(1-f)} \ . \tag{2.28}$$

The total amount of tracer in the system is the sum of the tracer in the plasma and tubular compartment

$$C = V_P C_P + V_T C_T \tag{2.29}$$

substituting equation 2.27 into 2.29 results in the the final equation

$$C(t) = V_P C_P(t) + F_T e^{-\frac{t}{T_T}} \otimes C_P(t) \ . \tag{2.30}$$

The separable compartment model is defined by the two equations 2.24 and 2.30 and the four parameters T_P, V_P, F_T, T_T. F_T is the renal GFR. Using the central volume theory and the four parameters calculated from the model, plasma flow F_P can also be calculated as

$$F_P = \frac{V_P}{T_P} \ . \tag{2.31}$$

The extraction fraction, which is the percentage of plasma entering the glomeruli that is filtered into the tubuli can also be calculated as

$$E = \frac{F_T}{F_P} \tag{2.32}$$

Comparatively, this model is very similar to the cortical compartment model. First, it should be noted that the type of filtration that takes place in the glomeruli is selective. Blood plasma is therefore not filtered into the tubuli but only remains in the glomeruli. The glomerular space in the cortical compartment model is therefore equivalent to the plasma space in the separable compartment model. Secondly, the parameters calculated by the cortical compartment model are analogues to some of the parameters in the separable compartment model. These parameters are the dispersion constant d which is labelled T_P in the separable compartment model, the filtration fraction f_a which is equivalent to V_P, k_{21} for F_T and k_{12} for $1/T_T$. The major difference between the two models is that while the cortical compartment model assumes a time delay between the tracer in the arterial input function before it enters the glomerular space, the separable compartment model asserts that this delay in humans is small and therefore negligible [26] so in this model $\tau = 0$.

2.5 The Deconvolution Method

The deconvolution method is, in a sense, different from the other three models. This method does not make any simplifying assumption about the underlying structure of the kidney neither does it divide the kidney into compartments of any sort. All it does is to analyze the evolution of the function that makes up the concentration function. The basic assumption made by this model is that the region of interest concentration curve is a dispersion of the arterial input function. The result is the ability to decompose the concentration time curve into its impulse response so as to remove the dependency from the arterial input function [16]. As a result of the above assumption, the arterial input function $C_a(t)$ can be seen as a unit impulse function which is convolved with an unknown function $h(t)$ to produce the cortical region of interest enhancement curve $C_{roi}(t)$.

$$C_{roi}(t) = C_a(t) \otimes h(t) = \int_0^t C_a(\tau) \cdot h(t-\tau) d\tau \tag{2.33}$$

which can be discretized to obtain the discrete deconvolution function

$$C_{roi}(t) = \sum_{\tau=0}^{t} C_a(\tau) \cdot h(t-\tau) \Delta\tau. \tag{2.34}$$

The convolution equation 2.34 can then be inverted to find the renal impulse response function $h(t)$ as

$$h(t) = C_{roi}(t) \otimes^{-1} C_a(t). \tag{2.35}$$

2.5.1 The Deconvolution Operation

From the discretization of the convolution operation,

$$C_{roi}(t) = C_a(t) \otimes h(t) = \sum_{0}^{t} C_a(\tau) \cdot h(t-\tau) \Delta\tau.$$

Assuming $\Delta\tau = 1$ we have

$$\begin{aligned}
C_{roi}(0) &= C_a(0) \cdot h(0) \\
C_{roi}(1) &= C_a(1) \cdot h(0) + C_a(0) \cdot h(1) \\
C_{roi}(2) &= C_a(2) \cdot h(0) + C_a(1) \cdot h(1) + C_a(0) \cdot h(2) \\
&\vdots \\
C_{roi}(n-1) &= C_a(n-1) \cdot h(0) + C_a(n-2) \cdot h(1) + \ldots + C_a(0) \cdot h(n-1) \,,
\end{aligned}$$

which we can decompose into matrix form as

$$\begin{bmatrix} C_{roi}(0) \\ C_{roi}(1) \\ C_{roi}(2) \\ \vdots \\ C_{roi}(n-2) \\ C_{roi}(n-1) \end{bmatrix} = \begin{bmatrix} C_a(0) & 0 & 0 & \dots & 0 & 0 \\ C_a(1) & C_a(0) & 0 & \dots & 0 & 0 \\ C_a(2) & C_a(1) & C_a(0) & 0 & \dots & 0 \\ \vdots & & & \ddots & & \vdots \\ C_a(n-2) & C_a(n-3) & \dots & C_a(1) & C_a(0) & 0 \\ C_a(n-1) & C_a(n-2) & \dots & C_a(2) & C_a(1) & C_a(0) \end{bmatrix} \begin{bmatrix} h(0) \\ h(1) \\ h(2) \\ \vdots \\ h(n-2) \\ h(n-1) \end{bmatrix}$$

The above equation can be expressed as

$$c_{roi} = C_a h \ ,$$

where c_{roi} is a vector representing the cortical region of interest and C_a is a Toeplitz matrix with its first column representing the arterial input function $C_a(t)$ and its first row having its first element to be the first element of the arterial input function and the the rest of the elements being zero and the matrix itself being diagonal-constant. C_a is therefore lower triangular in nature. h is the renal impulse response function we wish to recover by solving the equation. This problem can be solved using the singular value decomposition of $C_a(t)$, we have

$$c_{roi} = C_a h \Rightarrow c_{roi} = U\Sigma V' h \Rightarrow h = V\Sigma^{-1}U' c_{roi},$$

where U, V are unitary matrices.

2.5.2 Curve Fitting Within The Deconvolution Method

The renal impulse response function $h(t)$ exhibits three sequential peaks that identifies the flow of contrast agent through the glomeruli, the proximal convoluted tubule and the distal convoluted tubules [16]. The function also has a lot of other minor oscillations that could be the result of noise from image acquisition. According to [16], the first two of the major peaks can be fitted by the sum of two gamma variate functions

$$G(t) = a_1(t-a_4)^{a_2} e^{-(t-a_4)/a_3} + b_1(t-b_4)^{b_2} e^{-(t-b_4)/b_3}, \tag{2.36}$$

where t is the time and the a_i and b_i are parameters to be found. The problem with this method however is that the parameters $a_i, \ i = 1,2,3,4$ on the left side of the addition symbol determine one gamma variate function while the $b_i, \ i = 1,2,3,4$ determine a second. For the function defined by the parameters a_i, a_4 determines the time at which the function starts. To guarantee that the function is real, $t_a \geq a_4$ where t_a is the time input for the curve defined by the parameters a_i. In the same way, for the function defined by the parameters b_i, b_4 determines the start of the function. To guarantee a real function therefore $t_b \geq b_4$ where t_b is the time input for the curve defined by the b_i parameters. Since the a_i, b_i are fitted to unique peaks on h, they have different starting points. Since the a_i are the parameters defining the first peak, $a_4 > b_4$. This means that $t_a \neq t_b$. To avoid the possibility of fitting $G(t)$ as a complex function, it is better to break in up into two separate functions

$$G(t) = G_{vas} + G_{prox} \ , \tag{2.37}$$

where G_{vas} is the fit to the glomerular peak and is defined as:

$$G_{vas}(t) = a_1(t - a_4)^{a_2} e^{-(t-a_4)/a_3} \tag{2.38}$$

and G_{prox} is the fit to the proximal tubule peak

$$G_{prox}(t) = b_1(t - b_4)^{b_2} e^{-(t-b_4)/b_3} \ . \tag{2.39}$$

G_{vas} can therefore be fitted over the whole of $h(t)$ and G_{prox} can be fitted to the part of $h(t)$ where the proximal distal tubule peak begins. The parameters of the G curves are such that a_1, b_1 are scale factors, a_2, b_2, a_3, b_3 decide the shape of the curves and as stated earlier, a_4, b_4 determine where the curves begin. The parameters are fitted by a nonlinear least squares procedure.

2.5.3 Calculating Renal Parameters For The Deconvolution Method

Application of the central volume principle implies that renal perfusion RP is the ratio of the fractional plasma volume FPV and the vascular mean transit time MTT_{vas},

$$RP = \frac{FPV}{MTT_{vas}} \ . \tag{2.40}$$

The fractional plasma volume which is the fraction of the plasma volume occupied by the contrast agent can be found by the equation

$$FPV = \kappa \frac{\int C_{roi}(t)dt}{\int c_a(t)dt} \ , \tag{2.41}$$

where κ is a scale factor that that accounts for tissue density [22]. However from the matrix representation of the deconvolution, C_{roi} can be represented as

$$C_{roi} = C_a(t) \otimes h(t) = C_a(t) \cdot h(t) \ . \tag{2.42}$$

Therefore equation 2.41 can be rewritten as

$$FPV = \frac{\int C_a(t) \cdot h(t)dt}{\int C_a(t)dt} \approx \int h(t)dt \; . \tag{2.43}$$

The fractional plasma volume is therefore calculated by taking the integral over the $G_{vas}(t)$.

To calculate the Mean transit time, it it should be noted that the net volume of the system can be defined as

$$Net\ Volume = Flow \cdot t$$

To find the total volume therefore is to find the volume of of particles having transit times t.

$$\begin{aligned} dV &= t \cdot \text{rate of flow} \\ dV &= tFh(t)dt \\ V &= \int tFh(t)dt = F \int th(t)dt = F \cdot \text{Transit time} \; , \end{aligned}$$

which means that the mean transit time of the system is given by:

$$MTT = \int_0^\infty th(t)dt \; . \tag{2.44}$$

The vascular mean transit time is therefore given by:

$$MTT_{vas} = \int_0^\infty tG_{vas}(t)dt. \tag{2.45}$$

To calculate the GFR, the following formula was used [16]

$$GFR = \frac{\text{maximum slope of the proximal tubule peak}}{(C_{vasc})_{max}}. \tag{2.46}$$

Where $(C_{vasc})_{max}$ is the ratio of the maximum peak of the vascular function $G_{vas}(t)$ divided by FPV. From the calculation of the renal perfusion and GFR, we can calculate the filtration fraction as the ratio of the GFR and renal perfusion

$$FF = \frac{GFR}{RP}. \tag{2.47}$$

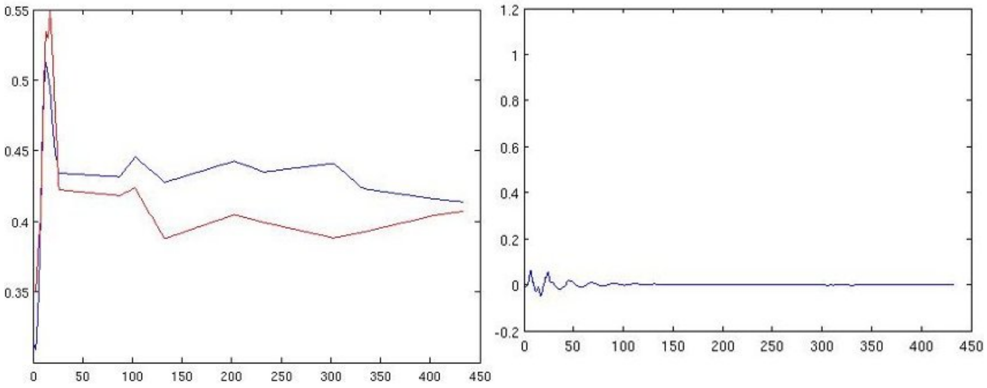

Figure 2.5: Concentration curves $C_{roi}(t)$ blue and $C_a(t)$ red and their impulse response $h(t)$ right

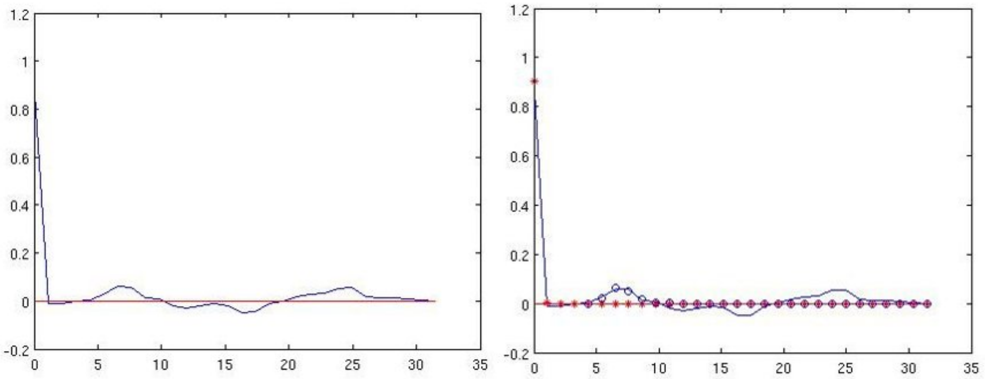

Figure 2.6: A scaled up view of part of the original impulse response curve $h(t)$ and the gamma fitted $G_{vas}(t)$ (red *) and $G_{prox}(t)$ (blue o)

Chapter 3

Data Preparation

3.1 Introduction

Successful renal function estimation using MRI images requires the availability of MRI images for the process. Once these are acquired, the dataset needs to be registered to correct for motion in the organs due to respiration. The next stage is to accurately identify the region of interest (ROI) needed for mathematical assessment. This stage is the segmentation stage. Once the regions of interest have been acquired, the mathematical models can be applied. In this project, the MRI images used were pre-registered images. The data preparation stage of this project therefore involves only the process of data segmentation.

This chapter describes the processes used to segment the kidneys for this project. For some of the images used in the project the first stage in segmentation is to define a rectangular area that contains the kidney. This is necessary because the kidney is located within an area which also has the liver and the stomach. These organs have tissue structures identical to the kidney and might also have similar properties. Defining a local region where segmentation is supposed to take place ensures the accuracy of the segmentation process.

3.2 Manual Segmentation

The first method used to segment the kidney parenchyma was by hand drawing the region of interest(ROI) mask. To achieve this, one of the time sequences was selected as a reference volume. On each slice of the reference volume, a region of interest was drawn using matlab's *roipoly* command.

This results in a matrix of size $m \times n$ for each slice with values 0 and 1 with the value 1 where the pixel is inside the region of interest and 0 otherwise. In locations where the kidney could not be detected or where there was no kidney present, an $m \times n$ matrix of all zeros was created. This is important to ensure that the size of the mask is equal to the size of one time slice of the dataset. Pixelwise multiplication between each time slice of the dataset and the mask is then performed. The result of these operations is the segmented dataset.

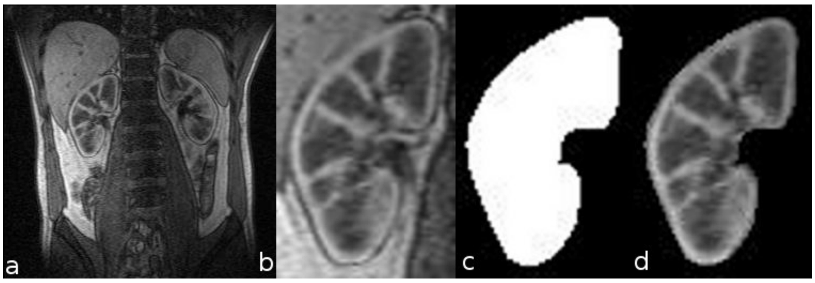

Figure 3.1: Manual Segmentation of the human kidney a) 2D slice of the MRI image b) Rectangular area around the right kidney c) Manual mask around the kidney d) pixelwise product of the kidney and the mask

3.3 Statistical Segmentation

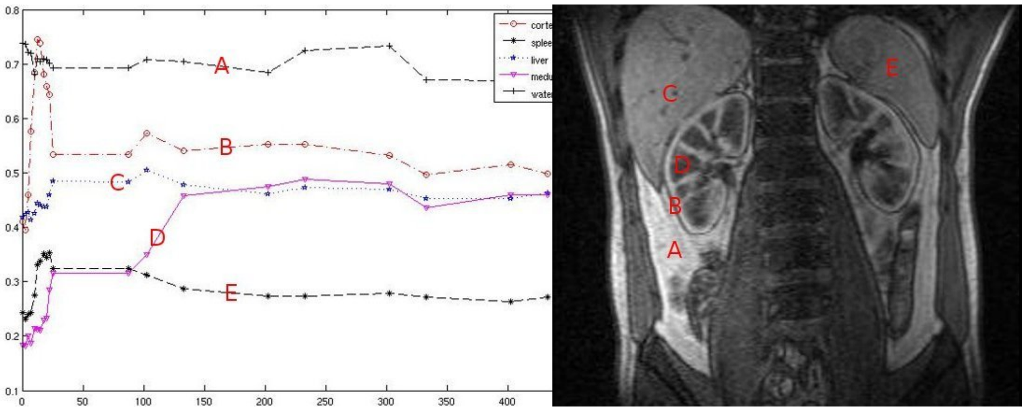

Figure 3.2: Enhancement pattern of different tissue structures in the renal MRI image. Notice how the cortex peaks very early compared to the other tissues and the mean of the maximum value also higher that most of the other tissues. The functions were plotted using six randomly selected pixels from the tissue.

Other segmentation methods tested in this project are based on temporal statistics of each voxel in the dataset. Since the contrast agent is carried by the blood into the system, it is assumed that the part of the body that takes up blood will have the intensity of the voxel changing over time. Some voxels will however change faster than others especially in their tracer uptake phase as shown in figure 3.2. It is therefore assumed that methods of statistical dispersion performed on the voxels in time will show which voxels change rapidly. The cortex for example peaks very early in the sequence. It is therefore assumed to have a high deviation if the sequence used was up to the peak of each voxel and not the whole time sequence. Two methods of dispersion were used, the standard deviation given by

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(X_i - \mu)^2},$$

and the mean absolute deviation

$$mad = \frac{1}{N}\sum_{i=1}^{N}|X_i - \mu| \; .$$

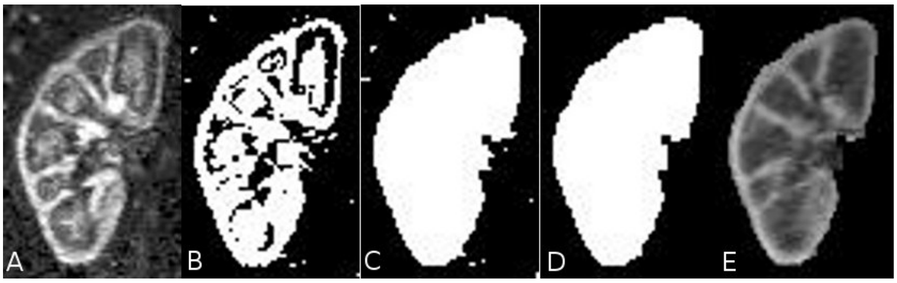

Figure 3.3: Image segmentation using mean absolute deviation a) mean absolute deviation of the dataset from time 0 to peak time. b) thresholding of image a). c) Image b) filled with *imfill* and morphological closing d) Image c) eroded and dilated to remove small pixels not part of kidney e) Pixelwise product of original image and image d)

Segmenting the cortex can be done by increasing the threshold of the deviations. Another method is to use the time to peak information of each voxel in the dataset in combination with the standard or the mean absolute deviation segmentatioin already shown in figure 3.3. Since the mean absolute segmentation method performed already gives the whole kidney parenchyma, and from figure 3.2, some other tissues also peak very early, the time to peak information alone would not produce a good segmentation of the renal cortex. This is further compounded by the presence of noise in the image. However performing a logical *and* operation between the ROI mask of the parenchyma and the time to peak information takes us closer to the goal of cortical segmentation. Morphological opening and closing followed by a morphological dilation and erosion fine tunes the segmentation operation.

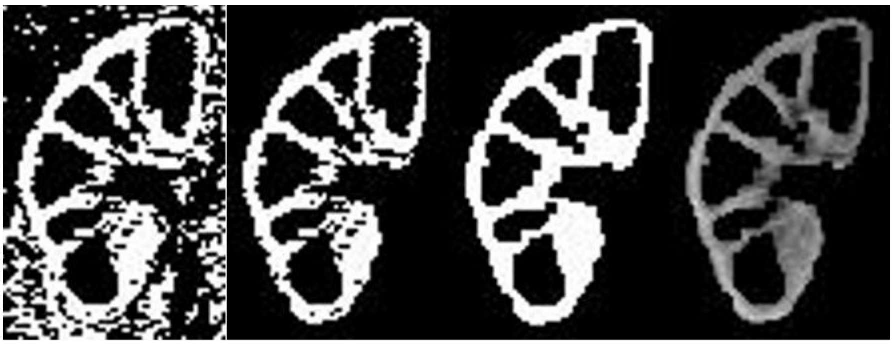

Figure 3.4: Stages of the segmentation of the cortex. a) The time to peak map of the kidney b) image a) combined with figure 3.3 c) using logical *and* c) Image b) after morphological opening and closing and dilation and erosion d) pixelwise multiplication of image c) and figure 3.1 b)

3.4 Kidney Segmentation By k-means Clustering

K-means clustering is an unsupervised cluster analysis method which is used for dimensionality reduction. Given a set of n voxels with t samples per voxel with $t <= n$, then the system is at most $t-$dimensional. K-means provides a way to partition the system into k clusters, $k <= t$. Given the 4 dimensional dataset that makes up our MRI images with n voxels $\{v_1, v_2, \cdots, v_n\}$ per 3 dimensional timestep, each 3 dimensional timestep is reshaped into a vector. The result is a set of t vectors. K-means therefore partitions the voxels into k partitions $S = \{S_1, S_2, \cdots, S_k\}$ by minimizing the objective function

$$f = \sum_{i=1}^{k} \sum_{v_j \in S_i} ||v_j - \mu_i||^2 , \tag{3.1}$$

where μ_i is the mean of S_i. Using k-means clustering for segmentation is usually a trial-and-error routine which will partition the volume into k mutually exclusive partitions. K-means might have to be run a couple of times until the right number of partitions that produces the best kidney partitions is achieved. Image processing routines can then be used to fine-tune the segmentation process.

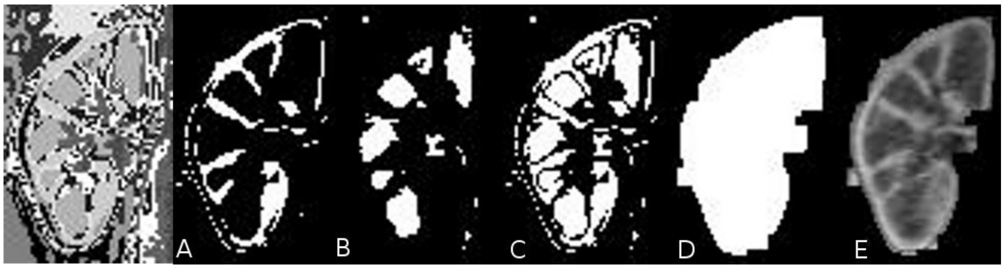

Figure 3.5: K-means segmentation of the kidney. The dataset was partitioned into 11 partitions. The first image shows the partitions within the volume a) The cortical partition b) The medula partition c) sum of image a) and b). d) Morphologically transformed version of c) e) Pixelwise product of d) and the original image

Chapter 4

Numerical Experiments

Having developed the mathematical models associated with renal parameter estimation, the aim of this chapter is to test the models on available MRI images. In a way, this is also a comparison of the models as it will enable us to compare the values returned by the models.

As stated in the chapter 3, the arterial input function was segmented by hand. This raises the issue of how large a region of interest selected for the arterial input function should be.

From figure 4.1, analysing the image from left to right and from top to bottom, the contrast agent enters the kidney from the regions marked A and B. It then flows into the cortex and enhances it. After that, it flows into the other parts of the kidney and enhances them. Since the first sighting of the contrast agent will be in the region marked A, one option is to use this region as the arterial input function for the slice. However, since the whole section marked B makes up the arteries, we might also be tempted to use that section for the arterial input function. The downside to this choice however is the reduction in the function due to averaging effect. This will happen because apart from the region marked A, most of the region B is not enhanced during the initial sightings. It is therefore possible that the selection of B might reduce the output of the models.

To test the models, three MRI images were used to perform four tests. Test 1 was performed on the right kidney of the test subject. Test 2 was performed on the left kidney of the same subject. Test 3 was performed from the same kidney used for test 1 registered by a different person (this was registered by Andrea Anderlik, a PhD fellow at the Biomedicin Department). Tests 1 through 3 were performed using A as the arterial input function. Tests 4 is a repetition of test 1 using the region B.

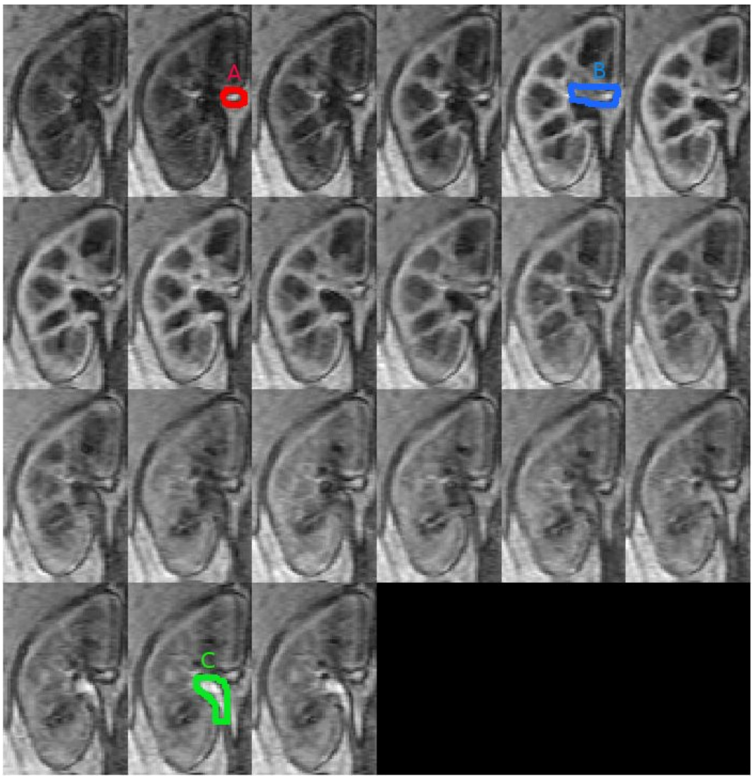

Figure 4.1: A montage in time of a slice of a functional MRI image

The results of the tests were tabulated using the following scheme.

Table 4.1: Parameter comparison of the different models

Model	τ	d	f_a	k_{21}	k_{12}
Patlak	-	-	c_1	c_2	-
Cortical	τ	d	f_a	k_{21}	k_{12}
Sourbron	-	T_P	V_P	F_T	$\frac{1}{T_T}$
Deconvolution	-	-	FPV	GFR	-

Since the cortical compartment model had the most parameters which are five in total, the parameters of the other models were compared to its parameters. Below are the test results;

Table 4.2: Test 1

Model	τ	d	f_a	k_{21}	k_{12}
Patlak	-	-	0.79727546	0.00016595	-
Cortical	3.47380033	0.11456115	0.06615905	0.00379473	0.11951595
Sourbron	-	0.18897981	0.09965853	0.00778876	0.12607557
Deconvolution	-	-	1.09544367	0.02594675	-

Table 4.3: Test 2

Model	τ	d	f_a	k_{21}	k_{12}
Patlak	-	-	0.84903070	0.00009305	-
Cortical	4.59511014	0.24342930	0.15944592	0.01388819	0.26296464
Sourbron	-	0.33050968	0.19419390	0.01824565	0.19152907
Deconvolution	-	-	0.99502602	0.09333561	-

Table 4.4: Test 3

Model	τ	d	f_a	k_{21}	k_{12}
Patlak	-	-	0.74906297	0.00035882	-
Cortical	3.52731078	0.26247502	0.12040167	0.01019533	0.08231768
Sourbron	-	0.27714202	0.11823087	0.01093042	0.07805641
Deconvolution	-	-	0.86909071	0.07237362	-

From tables 4.2 through 4.4, it can be seen that the Patlak model produces very low estimates as compared to the rest of the models. This is the effect of the negligible flow assumption. To see what happens in the estimation, we have to take a look at the typical region of interest curve for the kidney cortex. From figure 4.2, the cortical concentration rises to its peak very early and takes a longer time in the washout phase. Using the whole curve therefore negates the negligible flow assumption. To solve this problem, we take the part of the cortical concentration curve up to the peak time and use that to calculate the Patlak model paramenters.

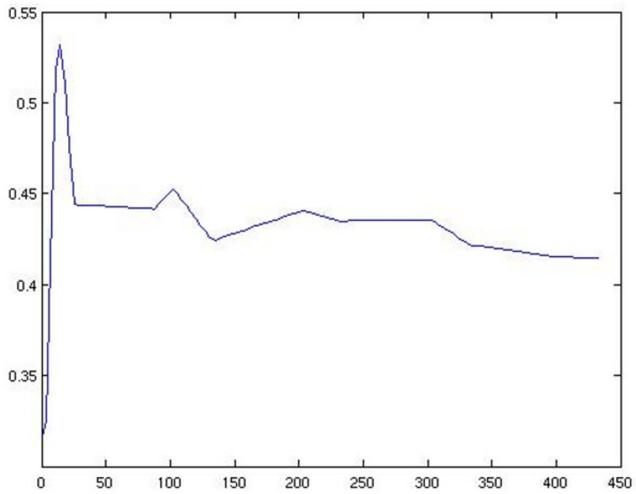

Figure 4.2: The cortical region of interest curve

Table 4.5: Repetition of the calculation of Patlak models using the region of interest curve from initial sighting to peak

Model	τ	d	f_a	k_{21}	k_{12}
Patlak Test 1	-	-	0.53521060	0.02428748	-
Patlak Test 2	-	-	0.59197647	0.02799499	-
Patlak Test 3	-	-	0.44867735	0.01168330	-

We now compare the new Patlak GFR values to the values from the other methods to give us an idea of how the models perform. From figure 4.3, the deconvolution method produces the largest estimates of GFR followed by the Patlak model. The Sourbron model and the Cortical compartment models produce almost parallel values with the Sourbron values being higher than those of the Cortical compartment.

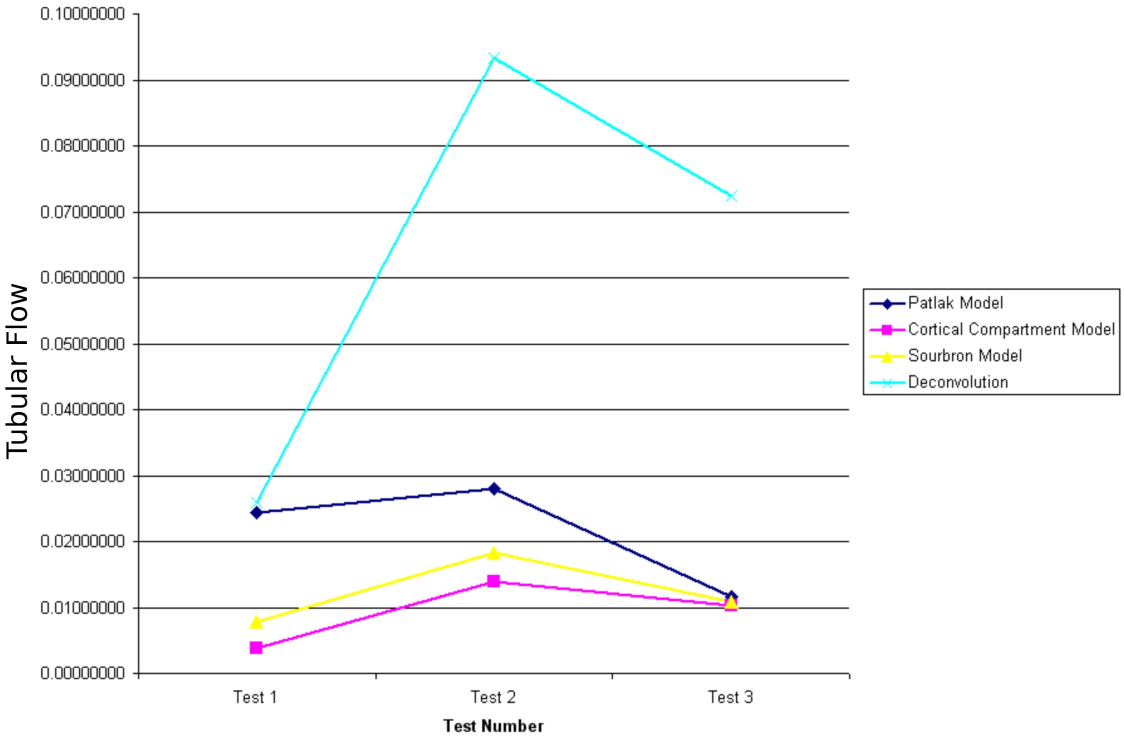

Figure 4.3: A comparison of tubular flow (k_{21}) values from tests 1 to 3

Using a different region of interest for the arterial input function also produces different results for the different models considered.

Table 4.6: Test 4 - Using a Different arterial input function

Model	τ	d	f_a	k_{21}	k_{12}
Patlak	-	-	0.94695893	0.01067041	-
Cortical	0.20226234	0.23029272	0.16283277	0.01187411	0.19224050
Sourbron	-	0.17318121	0.12214594	0.00888276	0.18980771
Deconvolution	-	-	1.35232386	0.00893228	-

Comparing tables 4.2 and 4.6 which are the same kidney image with different arterial input functions, we find that there are differences between the values obtained for the same subject using different segmentations of arterial input function. From this, we can say that two different analysts using the same functional MRI images of the kidney to calculate renal parameters, we are likely to arrive at different results based on segmentation process and the model that is used by the analysts. The results of test 3 also shows that a different registration process might also affect the result of using MRI images for renal parameter estimation. These results mean the results of parameter estimation using the models is not accurately reproduceable. Another example that shows the importance of the selection of a good arterial input function happened accidentally as we were testing an arterial input function. The arterial input function that was corrupted by the value of voxels from other tissues around the arteries. The function that was selected had lower values than the cortical region of interest function.

Table 4.7: Test 4 - Cortical and Sourbron estimation from defective arterial input function

Model	τ	d	f_a	k_{21}	k_{12}
Cortical	-1.35922299	1.39848958	1.26472377	0.00409365	0.02479976
Sourbron	-	0.84283906	0.81964625	0.00228073	0.02339123

Looking at the values of the Sourbron parameters in table 4.7, it is not obvious that there is an error in the estimated parameters but the error is flagged in the Cortical compartment model values as a negative delay time. This is easily seen as an error because a negative time delay implies the contrast agent arrived in the cortex before it arrived in the artery which is cleary false. Figure 4.4 shows the curves used in tables 4.2, 4.6, 4.7. The blue curve represents the arterial input function used in table 4.2. The red curve was used in table 4.6 while the black curve was the erroneous arterial input function used in table 4.7. The curve in magenta with the diamond markers represent the cortical region of interest curve.

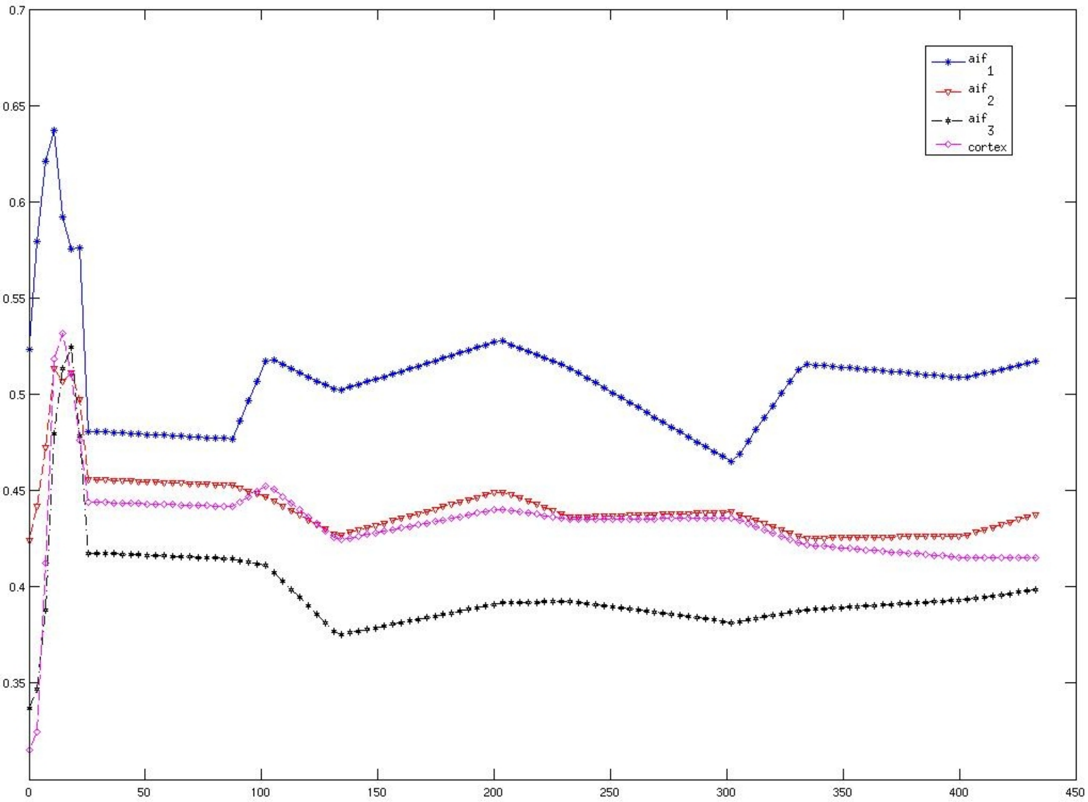

Figure 4.4: Comparison of the different arterial input functions used in tables 4.2, 4.6, 4.7

Chapter 5

Model Visualization

Apart from providing results for single kidney function estimation, functional MRI of the kidney also provides the added advantage of being able to give insight into renal function at the voxel level, using the same models developed for whole kidney function. Since this gives perfusion parameters for each voxel, the number of parameters that are extracted can be rather overwhelming given the high resolution images that today's scanning equipment can produce. The best way to analyze the result of a voxelwise estimation of renal function therefore is to visualize the results using various interactive and informative visualization schemes. This thesis also presents some slice and volume visualization techniques to provide insight into the result of voxelwise parameter estimation.

5.1 Volume Slicing

In visualizing with volume slices, we are interested in seeing what the perfusion data looks like on slices planes perpendicular to the x, y, z axes and on arbitrary slice planes. In this section of the chapter, I will be primarily interested in visualizing with different types of slice based visualization techniques in Matlab.

5.1.1 Single Slices

Since voxel-by-voxel analysis is based on the initial MRI images, the result will equally be a 3-dimensional volume with the size of a single 3-d time-slice of the original MRI image. Given a uniform, rectilinear, or structured grid, a slice is defined as all grid points that have one of the structured integer coordinates $n_1, n_2, ..., n_d$ equal [27]. In effect, taking a slice perpendicular to the z direction will have n_3 integer coordinates equal. This dataset will have dimension one less than the dimension of the original dataset. To implement slicing, we iterate over all the sample points in the slice in order of the structured coordinates n_i, $i \neq s$ where s is the slicing axis and save the sampled points in the slice plane [27]. We can then use a color map to map different intensity levels in the slice plane to colours. In this project, slicing using slice planes results in a 2-dimensional image that allows an insight into one part of the kidney at a time so using slices, we can for example see the perfusion parameters in the centre slice of the kidney.

An example in Matlab is, given a volume of size 70 × 95 × 8, we can take a slice perpendicular to the $z - axis$ as (:,:,p), which will result in an image of the p^{th} slice of the volume. Slicing perpendicular to the x and y axis works a little differently. Just slicing as (:,p,:) will still result in a 3-dimensional image except that the second dimension will have the value 1. To be able to display this image, the data is first reshaped to create a 2-dimensional image. Alternatively, we can use the *permute* command to rearrange the volume's dimension in a manner that matches the axis we want to work with, then we can use (:,:,p).

Apart from axes aligned slices, it is possible to also extract arbitrary slice planes from an image. Matlab defines a slice plane as a surface that takes on colouring based on values of the volume data in the region where the slice is positioned [1]. To generate arbitrary slices in matlab, we use the following procedure [2]:

1. Define a slice surface in the region bounded by the domain of the volume using Matlab's $surf$ and *linspace* commands. In essence, this creates an axis aligned slice.

2. Rotate the surface around an axis to orient it in the direction you want it to go. This is achieved using the *rotate* command.

3. Obtain the XData, YData and ZData for the surface using the *get* command. Each of these results will be a 2-dimensional array.

4. Use the XData, YData, ZData obtained to draw the slice plane within the volume using the *slice* command.

An abritrary slice, unlike an axis aligned one, is not fixed in any one direction which means that the sample locations will not be integral in nature. An interpolation scheme will therefore have to be used to obtain the sample data for the slice. Matlab uses trilinear interpolation by default to achieve this. The value for a pixel location in an arbitrary slice will therefore be based on contributions from other pixel locations within the volume.

Slices extracted from a volume can be used in many ways to convey information to an audience. Using a colormap, already discussed above will help in easy identification of areas of high and low values. In the case of multiple parameters, color information for one variable can be combined with altitude information for another variable to create a 3-dimensional height map with the z values being the altitude..

5.1.2 Multiple Slices

Slice based visualization is usually very easy to implement, but in using it, we tend to lose the 3-dimensional picture as we end up with a flat image devoid of contribution from any other slice. However, given some $n - D$ geometric structure, it is possible to view the structure of the object using stacks of $(n - 1) - D$ substructures. For example, a stack of $2 - D$ objects made up of a point, then a circle of radius r_1, then a bigger circle of radius r_2, then another circle of radius r_1 then another point in that order, the structure can be deduced as a sphere in $3 - D$. Another example is to stack up n circles in $2 - D$ to give a cylinder in $3 - D$ or multiple squares to give a cube. It is also possible to use multiple slices to visualize a slice in relation to other slices strategically placed to get an idea about the consistency of our data across the volume and possible some idea of the volume's $3 - D$ structure. In our particular case, we can use multiple slices, both perpendicular and arbitrary, together with transparency to see if the areas of high and low GFR have particular spatial locations within the kidney. Transparency becomes important in this situation to see through slices while showing other slices behind them.

5.2 3-D Volume Visualization By Raycasting

Volume rendering as opposed to slice based visualization creates an image that is made up of contributions from all voxels in the volume. One of the common volume rendering techniques is raycasting. Raycasting is not available in Matlab so it had to be coded from scratch.

In raycasting, the volume is assumed to be a cube placed in a 3-dimensional coordinate system in front of a viewport and a light source. In this implementation, given a volume of dimension $m \times n \times p$, the volume is placed in a coordinate system with its minimum point being (1,1,1) and its maximum point being (m, n, p). This volume is placed in front of a viewport with its bottom-left located at (1,1,$-q$) and top-right located at (m,n,$-q$) where q is length of the hypothenus of one z-slice of the volume. This is calculated as $ceil(\sqrt{m^2 + n^2})$. This distance allows the viewport not to cut through the volume when it is rotated.

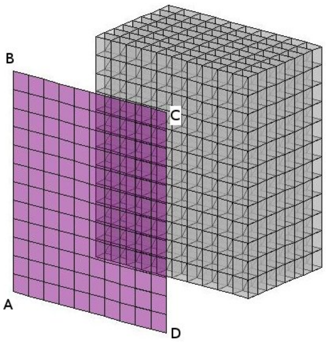

Figure 5.1: Pictoral depiction of viewport (rectangle ABCD) and volume (gray cube)

The basic idea in raycasting is that for every pixel in the viewport ABCD, we shoot a ray perpendicular to the viewport into the volume and determine the intersection of the ray and the volume. For every sample location in the ray, we calculate the contribution of the point in the volume to the pixel value.

5.2.1 Bounding Box Intersection

Given the rectangle ABCD shown in figure 5.1, $\vec{BC}$ and $\vec{BA}$ are vectors in the plane defined by ABCD. The normal $\vec{n_r}$ to the plane is therefore defined as

$$\vec{n_r} = \vec{BA} \times \vec{BC},$$

where × is defined as the vector cross product. For every pixel location ($\vec{p}$) in the viewport, we can define the ray through that location as

$$\vec{r}(t) = \vec{p} + t\vec{n_r},$$

where t is a step parameter, $\vec{p}$ is the origin and $\vec{n_r}$ is the ray direction.

To compute the intersection of the ray with the cube, we can take each face of the cube to be a plane. Then each of the planes is defined by the equation

$$(\vec{a} - \vec{x}) \cdot \vec{n_p} = 0,$$

which is true for all $\vec{x}$. In this equation, $\vec{a}$ represents a point on the plane and $\vec{n_p}$ the normal to the plane. To find if the ray intersects the plane, we plug the ray equation into the plane equation to give

$$(\vec{a} - \vec{p} - t\vec{n_r}) \cdot \vec{n_p} = 0,$$

and solve for t to get

$$t = \frac{(\vec{a} - \vec{p}) \cdot \vec{n_p}}{\vec{n_r} \cdot \vec{n_p}}.$$

If the ray is parallel to the plane, then $\vec{n_r} \cdot \vec{n_p} = 0$ so t will be undefined. To calculate the intersection of the ray and cube, we use the ray-box intersection method. In this method an axis aligned box is defined as two parallel planes which is specified by the location of two opposite corners $\vec{a}$, $\vec{b}$ of the box. The objective is therefore to calculate the the intersection of the ray and the 3 pairs of planes ($[x_{min}, x_{max}]$,$[y_{min}, y_{max}]$,$[z_{min}, z_{max}]$) that define the volume.

$$t_{x_1} = \frac{\vec{a_x} - \vec{p_x}}{\vec{n_{r_x}}} \quad , \quad t_{x_2} = \frac{\vec{b_x} - \vec{p_x}}{\vec{n_{r_x}}}$$

$$t_{y_1} = \frac{\vec{a_y} - \vec{p_y}}{\vec{n_{r_y}}} \quad , \quad t_{y_2} = \frac{\vec{b_y} - \vec{p_y}}{\vec{n_{r_y}}}.$$

$$t_{z_1} = \frac{\vec{a_z} - \vec{p_z}}{\vec{n_{r_z}}} \quad , \quad t_{z_2} = \frac{\vec{b_z} - \vec{p_z}}{\vec{n_{r_z}}}$$

To find the intersection times t_{min}, t_{max} is to find the common intervals of the times $t_{x_1}, t_{x_2}, t_{y_1}, t_{y_2}, t_{z_1}, t_{z_2}$

$$t_{min} = \max(\min(t_{x_1}, t_{x_2}), \min(t_{y_1}, t_{y_2}), \min(t_{z_1}, t_{z_2}))$$

$$t_{max} = \min(\max(t_{x_1}, t_{x_2}), \max(t_{y_1}, t_{y_2}), \max(t_{z_1}, t_{z_2})).$$

If $t_{min} \leq t_{max}$, then the ray enters the volume at t_{min} and exits at t_{max}

This leads to the following algorithm for bounding box intersection:

```
Let Box minimum $B_l = (x_l,\ y_l,\ z_l)$
box maximum $B_h = (x_h,\ y_h,\ z_h)$
ray origin $R_o = (x_o,\ y_o,\ z_o)$
ray direction $R_d = (x_d,\ y_d,\ z_d)$
  $t_{near} = -\infty,\ t_{far} = \infty$
  if $x_d = 0$ then
    if $x_o < x_l$ or $x_o > x_h$ then
      return $false$
    end if
  else
```

```
    $t_1 \Leftarrow \frac{(x_l - x_o)}{x_d}$   \\time of intersection with minimum $X$ plane
    $t_2 \Leftarrow \frac{(x_h - x_o)}{x_d}$   \\time of intersection with maximum $X$ plane
    if $t_1 > t_2$ then
        swap$(t_1, t_2)$
    end if
    if $t_1 > t_{near}$ then
        $t_{near} \Leftarrow t_1$
    end if
    if $t_2 < t_{far}$ then
        $t_{far} \Leftarrow t_2$
    end if
    if $t_{near} > t_{far}$ then
        return $false$   \\box is missed
    end if
    if $t_{far} < 0$ then
        return $false$   \\box is behind ray
    end if
end if
if $y_d = 0$ then
    if $y_o < y_l$ or $y_o > y_h$ then
        return $false$
    end if
else
    $t_1 \Leftarrow \frac{(y_l - y_o)}{y_d}$   \\time of intersection with minimum $Y$ plane
    $t_2 \Leftarrow \frac{(y_h - y_o)}{y_d}$   \\time of intersection with maximum $Y$ plane
    if $t_1 > t_2$ then
        swap$(t_1, t_2)$
    end if
    if $t_1 > t_{near}$ then
        $t_{near} \Leftarrow t_1$
    end if
    if $t_2 < t_{far}$ then
        $t_{far} \Leftarrow t_2$
    end if
    if $t_{near} > t_{far}$ then
        return $false$   \\box is missed
    end if
    if $t_{far} < 0$ then
        return $false$   \\box is behind ray
    end if
end if
if $z_d = 0$ then
    if $z_o < z_l$ or $z_o > z_h$ then
        return $false$
    end if
else
    $t_1 \Leftarrow \frac{(z_l - z_o)}{y_d}$    \\time of intersection with minimum $Z$ plane
    $t_2 \Leftarrow \frac{(z_h - z_o)}{y_d}$    \\time of intersection with maximum $Z$ plane
    if $t_1 > t_2$ then
        swap$(t_1, t_2)$
    end if
    if $t_1 > t_{near}$ then
        $t_{near} \Leftarrow t_1$
    end if
    if $t_2 < t_{far}$ then
        $t_{far} \Leftarrow t_2$
    end if
    if $t_{near} > t_{far}$ then
        return $false$    \\box is missed
    end if
    if $t_{far} < 0$ then
        return $false$    \\box is behind ray
    end if
end if
```

5.2.2 Interpolation

After finding the ray intersection, the next step is to find the voxel intensity at each sample location along the ray. Unfortunately, the samples along the ray will not necessarily be at integer locations within the volume. Trilinear interpolation is therefore used to approximate voxel values along the ray.

5.2.3 Color Compositing

Given a transfer function and step size, the method of calculating the pixel color from the volume is called compositing. One compositing model is to take the average of all the sample voxel intensities along the ray. Using this model, the color $C(p)$ at the pixel location p was calculated as the color of the average intensity along the ray.

$$C(p) = f(\frac{\sum I(t)\Delta t}{n})$$

where f is the transfer function, $s(t)$ the signal intensity at the sample location t on the ray and n is the number of steps.

Another model used frequently in angiography is the Maximum Intensity Projection. With this model, the transfer function is applied to the maximum of the voxel intensities along the ray.

$$C(p) = f(\max_{t\in[0,T]} I(t))$$

Yet another method is to apply a threshold along the ray. The transfer function is applied to the value of the first voxel greater than or equal to the threshold α along the ray. This is first hit projection with compositing function

$$C(p) = \begin{cases} f(I(t)) & \exists t \in [0,t], I(t) \geq \alpha \\ 0 & \text{otherwise} \end{cases}$$

Another method is compositing based on absorption and emission. Let $C(p)$ be the colour of a given pixel p on the image plane. Then $C(p)$ is a superposition of the contribution of all the voxels $v(t)$ along the ray $r(t)$. $C(p)$ is therefore given as

$$C(p) = \int_{t_0}^{t_1} I(t)dt \tag{5.1}$$

Let every voxel along the ray radiate an energy $I(t)$ and absorb energy from other voxels along the ray with absorption coefficient $\tau(t)$, then for every change in distance along the ray, the change in intensity at position t is given by

$$\frac{dI(t)}{dt} = -\tau(t)I(t). \tag{5.2}$$

The emission and absorption values $I(t)$ and $\tau(t)$ are determined by a transfer function. Integrating equation 5.2 gives the contribution of each voxel along the ray

$$I(t) = Ke^{-\int_{t_0}^{t_1}\tau(t)dt}. \tag{5.3}$$

Given that the initial emission at point t_0 is $I(t_0)$, the equation above becomes

$$I(t) = I(t_0)e^{-\int_{t_0}^{t_1}\tau(t)dt}. \tag{5.4}$$

Substituting equation 5.4 into 5.1 results in the equation

$$C(p) = \int_{t_0}^{t_1} I(t_0)e^{-\int_{t_0}^{t_1}\tau(t)dt}dt. \tag{5.5}$$

This is the compositing function used in raycasting. The next step is to discretize the compositing function for implementation. Since t changes uniformly by the factor Δt along the ray $t(t)$, the integral in the compositing function can be replaced by the sum of the contributions of each voxel along the sample locations

$$C(p) = \sum_{i=0}^{N} I(i\Delta t) e^{-\sum_{j=0}^{i-1} \tau(j\Delta t)\Delta t} \Delta t, \tag{5.6}$$

where N is the number of sample points along the ray. The power sum of the inner exponential term is just the product of all the individual exponential terms

$$C(p) = \sum_{i=0}^{N} I(i\Delta t)(\prod_{j=0}^{i-1} e^{-\tau(j\Delta t)\Delta t})\Delta t. \tag{5.7}$$

Using taylor series expansion, the function $e^{-x} = 1-x+\frac{x^2}{2!}-\frac{x^3}{3!}+\cdots$. For very small x however, higher order terms tend to converge to zero, the function can therefore be approximated linearly by ignoring higher order terms as $e^{-x} = 1 - x$. Thus, the compositing equation changes to the approximation

$$C(p) = \sum_{i=0}^{N} I(i\Delta t)(\prod_{j=0}^{i-1}(1 - \tau(j\Delta t))\Delta t. \tag{5.8}$$

Rewriting $I(i\Delta t) = I_i$ and $\tau(j\Delta t) = \tau_j$ results in the formulation

$$C(p) = \sum_{i=0}^{N} I_i(\prod_{j=0}^{i-1}(1 - \tau_j)\Delta t \tag{5.9}$$

This is the function used to model the compositing function along the ray. If the accumulated colour up to position i along the ray is given as C_i and the energy radiated at point i is given as c_i, then the compositing function is implemented as a iterative function, which is the sum of the energy emitted at the point i and the superposition of the energies from other points the ray has already traversed.

$$C_i = c_i + (1 - \tau_i)C_{i+1}.$$

This is the back-to-front compositing equation. In this project, C_i is implemented as a convex combination of the absorption coefficient τ_i as

$$C_i = c_i\tau_i + (1 - \tau_i)C_{i+1}. \tag{5.10}$$

The above compositing equation results in a flat rendering of the volume. To provide depth information, the diffuse component of the Phong shading model was used. With diffuse reflection, it is assumed that the light coming into a surface is reflected evenly across the surface. Given a light source at the position $\vec{t}$ with intensity I_d and the normal to the surface denoted n, the diffuse reflection equation is given as

$$R_d = I_d cos\theta = I_d(\vec{l} \cdot \vec{n}).$$

Since the dot product can be negetive or positive, in some application the absolute value of the dot product is used. This solution however assumes the surface is two sided. For a one-sided volume as in the case of the volumes in use, a one-sided solution is employed. This is to substitute the dot product with the *max* function to arrive at the diffuse equation

$$R_d = I_d cos\theta = I_d\ max(0, \vec{l} \cdot \vec{n}).$$

Combining the diffuse reflection into the compositing equation derived above, we get

$$C_i = c_i\tau_i\ max(0, \vec{l} \cdot \vec{n}) + (1 - \tau_i)C_{i+1}.$$

In our volume, we replace the surface normal with the normalized gradient vector of the voxel at time where the gradient is given by

$$\nabla v(t) = [\frac{\partial v(t)}{\partial x}, \frac{\partial v(t)}{\partial y}, \frac{\partial v(t)}{\partial z}]$$

which is approximated using the central difference formula

$$\nabla v_i = \frac{[v(x_{i+1}, y, z) - v(x_{i-1}, y, z), v(x, y_{i+1}, z) - v(x, y_{i-1}, z), v(x, y, z_{i+1}) - v(x, y, z_{i-1})]}{2}.$$

All these steps are combined to give the final algorithm for volume ray-casting

```
for each pixel $p$ in the image plane $I$ do
    $r(t) \Leftarrow$ ray perpendicular to $I$ through $p$
    $t_1, t_2 \Leftarrow$ intersection of $r(t)$ with volume $f(x, y, z)$
    $C(p) \Leftarrow 0$    color of background pixel
    for $t$ from $t_2$ to $t_1$ reducing by stepsize do
        Calculate the colour contribution of $f(r(t))$ to the image
    end for
end for
```

5.3 Fourier Based Volume Rendering

Fourier based volume rendering is based on the idea used to develop tomographic images from projections as discussed in [13]. Considering that in x-ray computed tomography, a set of projections are given with the aim of recovering a $3-D$ representation of the structure, in fourier based volume rendering, the aim is to recover the projection information from a given $3-D$ structure which in this case is the $3-D$ volume. To get a picture of how this process works, consider a volume and its Fourier transform

$$F(u,v,w) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y,z)e^{-i2\pi(ux+vy+wz)}dxdydz. \tag{5.11}$$

Where $f(x,y,z)$ is the volume in consideration and $F(u,v,w)$ its Fourier transform. Assume now a slice of the Fourier transform is taken on the $(u-v)$-plane (where $w=0$). Then any point (u_0,v_0,w_0) is projected unto the point $(u_0,v_0,0)$. Equation 5.11 on this plane becomes

$$F(u,v,0) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y,z)e^{-i2\pi(ux+vy)}dxdydz. \tag{5.12}$$

This can be rewritten as

$$F(u,v,0) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\{\int_{-\infty}^{\infty} f(x,y,z)dz\}e^{-i2\pi(ux+vy)}dxdy. \tag{5.13}$$

Which is the $2-D$ fourier transform of the projection of $f(x,y,z)$ in the $z-$axis, the direction perpendicular to the original plane from which the slice was extracted. This is the Fourier Slice-Projection theorem which states that the inverse transform of a slice extracted from the frequency domain representation of a volume yields a projection of the volume in a direction perpendicular to the slice [17]. On the basis of this theorem, a projection of the volume can be calulated from the frequency domain transformation of the volume. This method is faster than the usual raycasting method because of how fast the FFT of a volume can be calculated. The other complexity is the extraction of a slice from the volume. The algorithm to create the fourier rendering is as follows:

1. Compute the $3-D$ fourier transform $\mathcal{F}(u,v,w)$ of the volume.
2. Compute the values of $\mathcal{F}(u,v,w)$ on a plane through the origin of $\mathcal{F}(u,v,w)$ by trilinear interpolation.
3. Perform the inverse $2-D$ Fourier transform of the slice to give an image of the $2-D$ projection of the volume.

The result of this algorithm is an x-ray image of the volume, where each pixel corresponds to the contribution of samples on a ray through the volume. No color or shading information is present in this image. However, it should be noted that the discrete fourier transform assumes a periodicity of the spatial function $f(x, y, z)$. The images produced using the above steps might therefore contain multiple images which are contributions from the periods of $f(x, y, z)$. The solution is to perform a zero padding of the volume before performing the fourier transform. To incorporate some directional shading into this method, it can be recalled that adding diffusion shading in raycasting is given by multiplying the contribution on each sample point on the ray by the reflection of a light source $\vec{L}$ with the surface normal $\vec{N}$

$$I_p = \int_{t_0}^{t_1} I_p f(r(t))\ max(0, \vec{N}(r(t)) \cdot \vec{L}) dt, \tag{5.14}$$

Where $I_p(r(t))$ is the intensity of the light from the direction $\vec{L}$ and $f(r(t))$ is the value of the volume along the ray, $\vec{N}(r(t))$ is the normal to the volume at the sample location $r(t)$ along the ray and $\vec{L}$ is the lighting direction. The function $max(0, \vec{N}(r(t))$ is nonlinear and finding its fourier transform is not easy. It can however be approximated by the function [11, 29]

$$max(0, \vec{N}(r(t)) = \frac{1}{2}(1 + (\vec{N} \cdot \vec{L})). \tag{5.15}$$

Equation 5.14 then becomes

$$\begin{aligned} I_p &= \int_{t_0}^{t_1} I_p f(r(t))\ \frac{1}{2}(1 + (\vec{N} \cdot \vec{L}))dt \\ &= \frac{1}{2}\int_{t_0}^{t_1} I_p f(r(t))dt + \frac{1}{2}\int_{t_0}^{t_1} I_p f(r(t))(\vec{N} \cdot \vec{L})dt, \end{aligned}$$

which can be decomposed into its constituent form as

$$\begin{aligned} I_p &= \frac{1}{2}\int_{t_0}^{t_1} I_p f(r(t))dt + \frac{1}{2}\int_{t_0}^{t_1} I_p f(r(t))N_x L_x dt \\ &+ \frac{1}{2}\int_{t_0}^{t_1} I_p f(r(t))N_y L_y dt + \frac{1}{2}\int_{t_0}^{t_1} I_p f(r(t))N_z L_z dt. \end{aligned}$$

Since the light source $\vec{L}$ is independent of the ray location, it can be taken out of the integral to arrive at:

$$\begin{aligned} I_p &= \frac{1}{2}\int_{t_0}^{t_1} I_p f(r(t))dt + \frac{1}{2}L_x\int_{t_0}^{t_1} I_p f(r(t))N_x dt \\ &+ \frac{1}{2}L_y\int_{t_0}^{t_1} I_p f(r(t))N_y dt + \frac{1}{2}L_z\int_{t_0}^{t_1} I_p f(r(t))N_z dt. \end{aligned}$$

Where $L_i,$ and N_i are the components of the light and normal in the $i-$axis respectively. The above equation sets up the projection equation for the volume and normals. Using the fourier projection-slice theorem, the algorithm can be setup as follows:

1. Create four copies of the Volume $f(x, y, z)$.
2. Multiply each of the first three copies of $f(x, y, z)$ created by a component $N_i, i = x,\ y,\ z$ to create the matrices $fN_x,\ fN_y,\ fN_z$.
3. Find the $3 - D$ Fourier transform of each of the matrices found in step 2 and the original matrix $f(x, y, z)$.
4. For a plane $P(u, v)$ through the origin extract a slice through each of the matrices created in step 3.
5. Find the $2 - D$ inverse Fourier transform of each of matrices from step 4 to get the matrices $fN_{2x},\ fN_{2y},\ fN_{2z},\ f_2$. The 2 is a reimder that these are $2 - D$ matrices.
6. Multiply f_2 by the constant $\frac{1}{2}$ and the matrices $fN_{2i},\ i = x, y, z$ by $\frac{1}{2}$ of the appropriate lighting component and add them together.

As in the case of the original image, each of the directinal derivatives is zero-padded to remove the periodicity effect from the final image.

5.4 Example Of Visualized Results From The Patlak Model

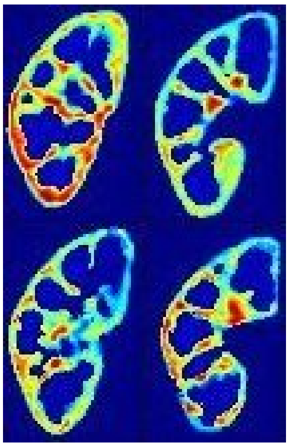

Figure 5.2: Single slices of c_2 volume parameter taken parallel to the xy plane

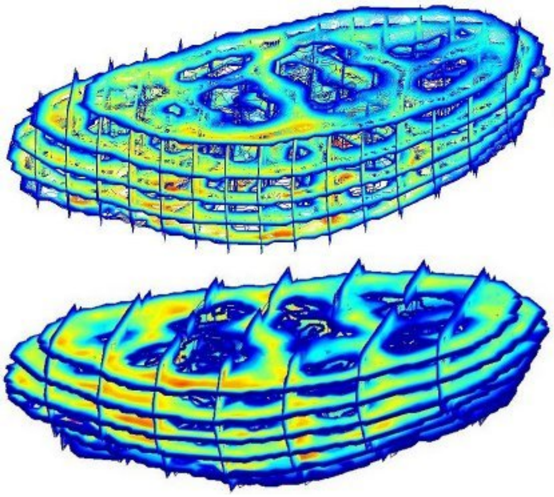

Figure 5.3: Multiple slices of the c_2 volume parameter displayed as slice contours. notice that the high intensity areas are consistent across slices

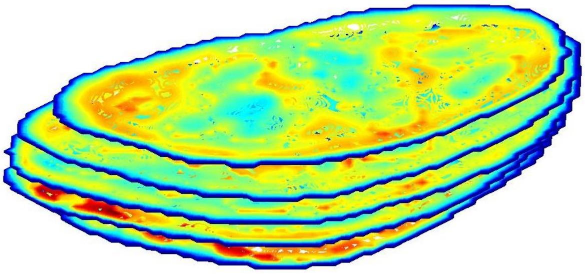

Figure 5.4: Multiple slices of the c_1 volume parameter displayed as slice contours.

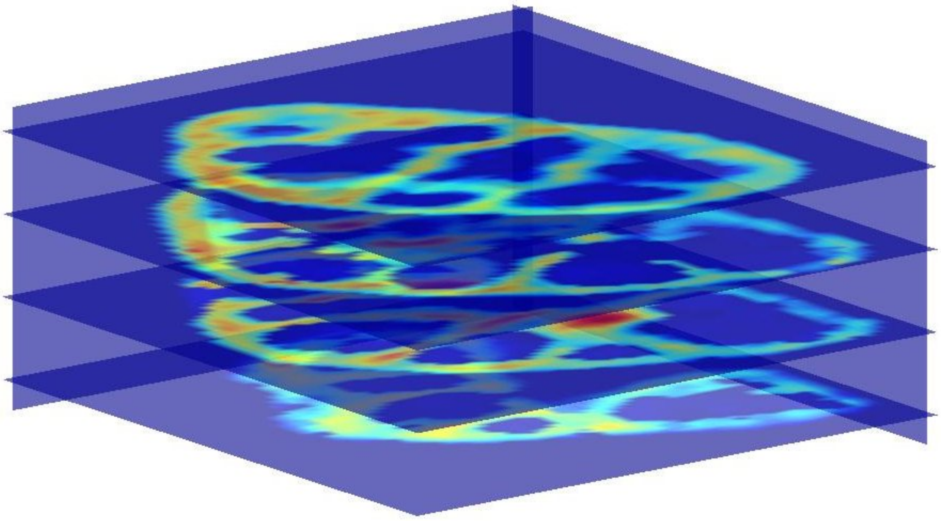

Figure 5.5: A multislice view of c_2

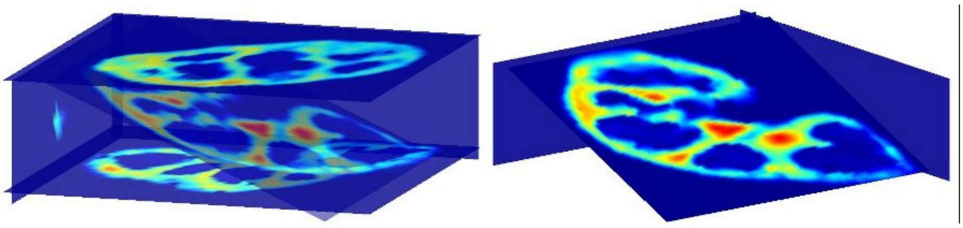

Figure 5.6: A multislice view of c_2 incorporating arbitrary slice planes

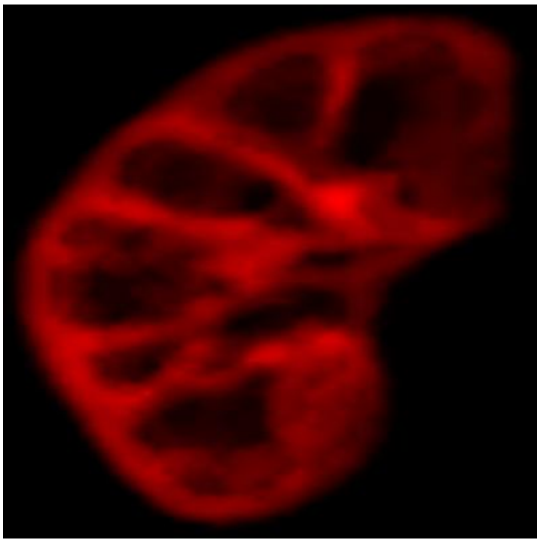

Figure 5.7: A raycast image of c_2 composited with averaging

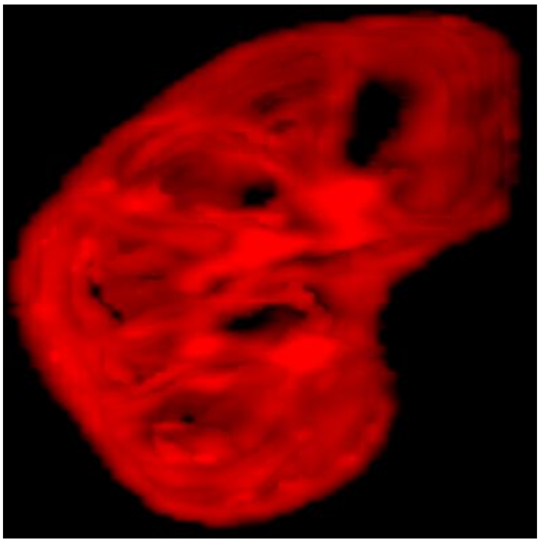

Figure 5.8: A raycast image of c_2 composited with maximum intensity projection

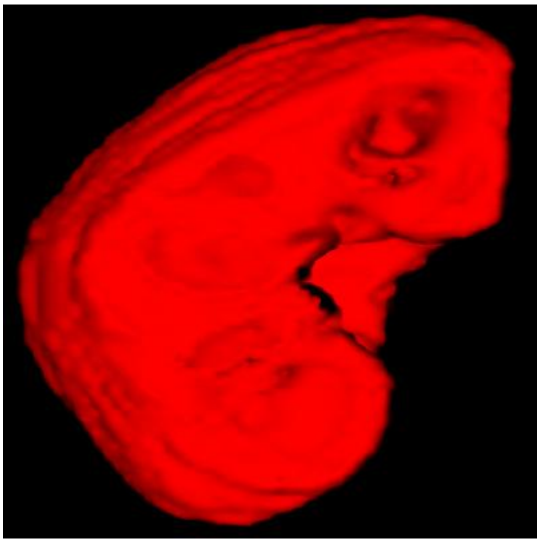

Figure 5.9: A raycast image of the vascular volume parameter (c_1) from Patlak

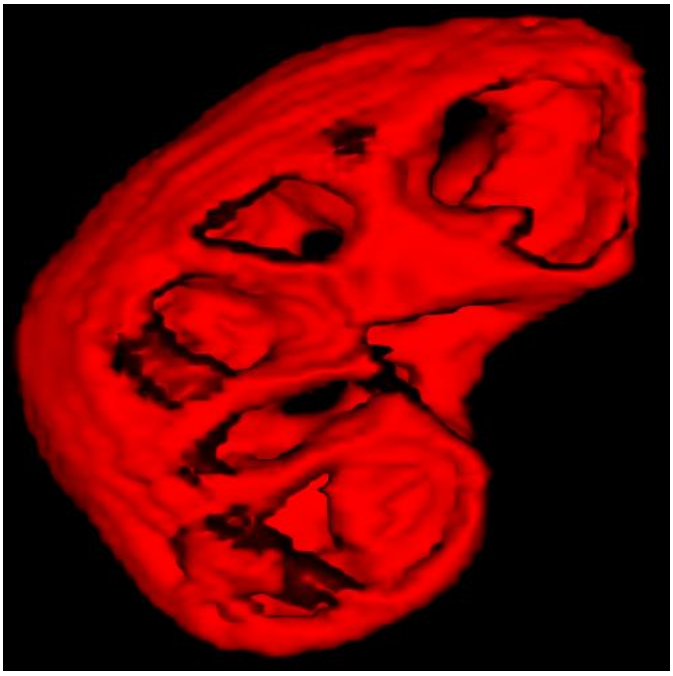

Figure 5.10: A raycast image of the tubular flow parameter (c_2) from Patlak

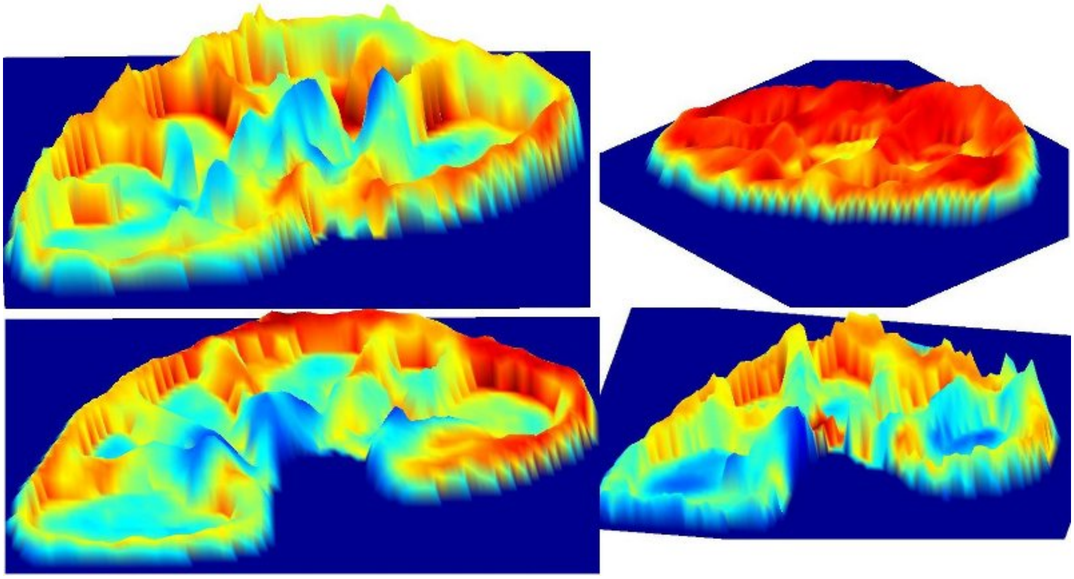

Figure 5.11: c_2 parameter result viewed as a heightmap

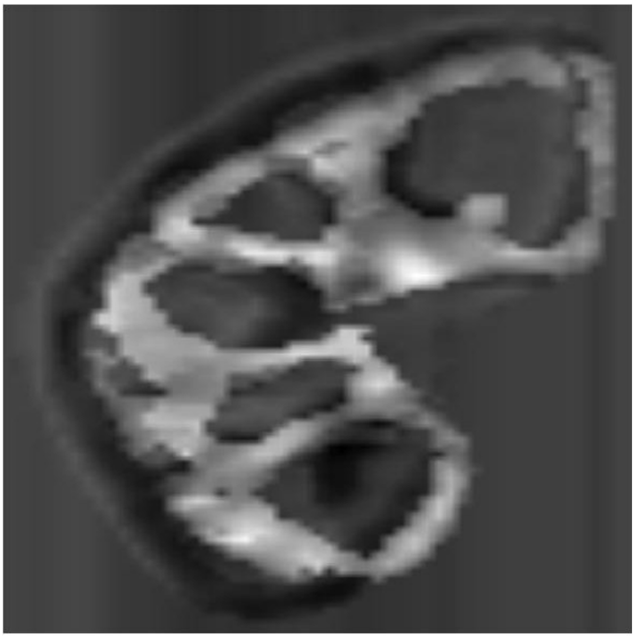

Figure 5.12: c_2 result as x-ray from Fourier based visualization

From these visualizations, the structure of the kidney is clearly seen. This means that results from a voxel-by-voxel visualization of the models can show the structure of the kidney. It can also be seen that the flow parameter c_2, (the analog of the GFR if we had used the concentration vs time curves) is high only along the cortex lines. It is very low or almost zero in the other parts of the kidney like the medula. On the other hand, figures 5.9 and 5.4 show that the vascular volume parameter c_1 shows relatively high values along the medula as well. These results also show where parameter values are too low instead of being high. This can be helpful in recognizing defective or failing kidneys.

Chapter 6

Conclusion

Renal parameter estimation using MRI images is a very interesting concept because it allows the user to acquire multiple parameters from a single test. For example, the ability to acquire five parameters from using the cortical compartment model makes it an attractive prospect and it would be interesting to see such applications in the hospitals. The results of this project however show that there are a number of factors that affect the result of this process. These include the segmentation and registration processes and image noise. These factors make it difficult to duplicate the results acquired from the tests. We cannot therefore prescribe this method as the sole method of renal parameter estimation but it can be used in collaboration with the traditional method of GFR estimation, which is the urine test.

After using the urine test to find the whole kidney GFR, the MRI models can be used to test how each single kidney contributes to the process while also producing other parameters including kidney volume.

Another factor that will affect the usage of this method is the computational cost of the models. From the numerical experiments performed, the Patlak model has the fastest execusion time. This is not surprising as the operations involved are integration (which is summation), matrix multiplication and SVD. This means it can be used if the aim is to access parameters fast. At the minimum, the other methods involve the use of nonlinear least squares while sourbron and the cortical compartment models also involve the use of the convolution operation which are all algorithmically expensive methods.

The voxelwise analyses of the kidney parameters also make it possible to analyse the intergrity of the kidney especially after a transplant since it will show which parts of the kidney cortex are not perfusing right. It will also make it possible to visualize any growth or tumour within the kidney which is perfusing in a way that is different from the tissues around it.

Appendices

Appendix A

Mathematical Preliminaries

A.1 Introduction

In the analysis of the renal parameter estimation models certain mathematical methods are used frequently. These are interpolation (and extrapolation) and least squares. This chapter presents a discussion of these concepts and their application to the rest of the thesis.

A.2 Interpolation

Given a discrete data set, interpolation is the process of creating new data points within the range of the data. This is very important because unlike continuous functions where it is easy to plug a data point and evaluate the function at that point, in discrete data sets, such information is not available for reconstructing the function at any point. There are however methods available for estimating the value of the function. In this thesis, two interpolation methods are used; linear and cubic spline interpolation. The interpolants considered are polynomial interpolants. The following theorem will be helpful in understanding the relationship between the number of input points, the interpolants and the uniqueness of the derived curve.

Theorem 1. *A polynomial of degree $n-1$ passing through n distinct points is unique.*

Proof. Let $p(t)$ and $q(t)$ be two different polynomials of degree $n-1$ passing through the same n distinct points. Define a new polynomial $r(t)$ as

$$r(t) = p(t) - q(t) \ .$$

Since $p(t)$ and $q(t)$ are different $r(t)$ is nonzero. $r(t)$ is of degree at most $n-1$ since both $p(t)$ and $q(t)$ are of degree at most $n-1$. $r(t)$ has n zeros. By the fundamental theorem of algebra however, a polynomial of degree $n-1$ must have exactly $n-1$ zeros. $r(t)$ can therefore have more than $n-1$ zeros only if it is the zero polynomial. This means

$$r(t) = p(t) - q(t) = 0 \Rightarrow p(t) = q(t)\ .$$

□

A.2.1 Linear Interpolation

Given two points (x_1, y_1), (x_2, y_2), linear interpolation finds a point (x, y) such that the three points lie on a straight line. The value of y along the unique straight line defined by (x_1, y_1), (x_2, y_2) is given by the equation

$$\frac{y - y_1}{x - x_1} = \frac{y_2 - y_1}{x_2 - x_1}.$$

Solving for y results in the equation

$$y = y_1 + (x - x_1)\frac{y_2 - y_1}{x_2 - x_1}.$$

Since in most cases MRI images are not taken on a uniform time scale, the data points must be interpolated on a uniform time scale to be used in this thesis.

A.2.2 Cubic Spline Interpolation

Given a dataset on sample points $(x_1, x_2, \cdots, x_n)$, in cubic spline interpolation, the aim is to fit the data to a piecewise function of the form

$$S(x) = \begin{cases} S_1(x), & x \in [x_1, x_2) \\ S_2(x), & x \in [x_2, x_3) \\ \cdots \\ S_{n-1}(x), & x \in [x_{n-1}, x_n), \end{cases} \tag{A.1}$$

where S_i is a polynomial of degree three of the form

$$S_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i \tag{A.2}$$

where $i = 1, 2, 3, \cdots, n-1$ with

$$S_i^{'}(x) = 3a_i(x - x_i)^2 + 2b_i(x - x_i) + c_i \tag{A.3}$$

and

$$S_i^{''}(x) = 6a_i(x - x_i) + 2b_i \tag{A.4}$$

being the first and second derivatives respectively and where $S(x)$ satisfies the following conditions:

1. $S(x_i) = f(x_i)$. That is $S(x)$ interpolates all the sample points.
2. $S(x)$ is continuous on $[x_1, x_n]$.
3. $S^{'}(x)$ is continuous on $[x_1, x_n]$.
4. $S^{''}(x)$ is continuous on $[x_1, x_n]$.

From property 1,

$$S(x_i) = y_i = a_i(x_i - x_i)^3 + b_i(x_i - x_i)^2 + c_i(x_i - x_i) + d_i$$
$$\Rightarrow y_i = d_i.$$

Property 2 implies that

$$S_{i+1}(x_i) = S_i(x_i)$$

$i = 1, 2, \cdots, n-1$ using the above with equation A.3,

$$\begin{aligned} S_i(x) &= d_i = S_{i+1}(x_i) = a_{i+1}(x_i - x_{i+1})^3 + b_{i+1}(x_i - x_{i+1})^2 \\ &+ c_{i+1}(x_i - x_{i+1}) + d_{i+1} \\ \Rightarrow d_i &= a_{i+1}h^3 + b_{i+1}h^2 + c_{i+1}h + d_{i+1} \end{aligned}$$

where $h = x_i - x_{i+1}$

$$\text{property 3} \Rightarrow S_i^{'}(x_i) = S_{i-1}^{'}(x_i)$$

from equation A.3

$$S_i^{'} = c_i \Rightarrow c_i = 3a_{i+1}h^2 + 2b_{i+1}h + c_{i+1}$$

$$\begin{aligned} \text{equation 4} &\Rightarrow S_i^{''}(x_i) = 2b_i \\ &\Rightarrow b_i = \frac{S_i^{''}(x_i)}{2} \end{aligned}$$

From equation A.4 and the continuity property $S_{i+1}(x_i) = S_i(x_i)$

$$\begin{aligned} 2b_{i+1} &= 6a_ih + 2b_i \\ a_i &= \frac{2b_{i+1} - 2b_i}{6h} \\ &= \frac{S_{i+1}^{''}(x_i) - S_i^{''}(x_i)}{6h} \end{aligned}$$

From the continuity property

$$\begin{aligned} c_i &= \frac{d_{i+1} - a_i h^3 - b_i h^2 - d_i}{h} \\ &= -a_i h^2 - b_i h - \frac{y_i - y_{i+1}}{h} \\ &= \frac{y_{i+1} - y_i}{h} - \frac{S''_{i+1}(x_i) + 2S''_i(x_i)}{6} h \end{aligned}$$

Some of work on the models were also tested with data resampled by cubic interpolation.

A.2.3 Trilinear Interpolation

Trilinear interpolation is a three dimensional extension of linear interpolation. In the visualization part of the work, the data volumetric data has to be resampled to be able to take data samples at any location within the volume. Trilinear interpolation was chosen because it produces much better results compared to the other alternative considered which was nearest neighbour interpolation though it is computationally more expensive. Given a cube with vertices C_{ijk} with $i, j, k = 0, 1$, if $\rho(u, v, w)$ is a point within the cube, then let

$$\begin{aligned} x &= u - \lfloor u \rfloor \\ y &= v - \lfloor v \rfloor \\ z &= w - \lfloor w \rfloor \end{aligned}$$

The idea is to interpolate the cube on all axes. One possibility is to first interpolate the cube four times using linear interpolation on the $z-$ axis, this results in

$$\begin{aligned} i_1 &= C_{000}(1 - z) + C_{001} z \\ i_1 &= C_{010}(1 - z) + C_{011} z \\ i_1 &= C_{100}(1 - z) + C_{101} z \\ i_1 &= C_{110}(1 - z) + C_{111} z \end{aligned}$$

The result of the above process can then be used to interpolate the cube two times on the $y-$ axis

$$\begin{aligned} j_1 &= i_1(1 - y) + i_2 y \\ j_2 &= i_3(1 - y) + i_4 y. \end{aligned}$$

Finally the latter process can be used to interpolate the cube once on the $x-$ axis to arrive at

$$\rho(u,v,w) = j_1(1-x) + j_2x.$$

Compacting the above processes into one reults in the trilinear interpolation formula

$$\begin{aligned}\rho(u,v,w) &= C_{000}(1-x)(1-y)(1-z) + C_{001}(1-x)(1-y)z \\ &+ C_{010}(1-x)y(1-z) + C_{011}(1-x)yz + C_{100}x(1-y)(1-z) \\ &+ C_{101}x(1-y)z + C_{110}xy(1-z) + C_{111}xyz.\end{aligned}$$

A.2.4 Extrapolation

Another mathematical method used in this paper is extrapolation. The idea here is to construct new data points outside a discrete set of known data points. One method of doing this is linear extrapolation. This is analogous to creating a tangent at the end of the sample data and extending the tangent beyond the limit of the data and picking off the values on the line. Given a point x to be extrapolated near the data points $(x_{k-1}, y_{k-1}), (x_k, y_k)$, then linear extrapolation gives

$$y(x) = y_{k-1} + \frac{x - x_{k-1}}{x_k - x_{k-1}}(y_k - y_{k-1}).$$

The downside to this method however is that if the extrapolated value is far from the sample points or should the sample data be nonlinear, a lot of errors will be introduced into the data. Another possibility is to extrapolate higher order polynomials. This can be achieved using least squares to approximate the function $f(x)$ that approximates the sample points and finding value of x on the function $f(x)$. Which is the method employed in matlab using polyval/polyfit.

A.3 Least Squares

Given a system of m equations in n unknowns, the problem of finding a solution $x \in \mathbb{C}^n$ that satisfies the equation $Ax = b$ where $A \in \mathbb{C}^{mxn}$ is the matrix of coefficients and $b \in \mathbb{C}^m$ is what least squares is concerned with. The system is said to be overdetermined if $m > n$. Let

$$r = Ax - b,\ r \in \mathbb{C}^m. \tag{A.5}$$

If the system of equations are linear in the independent variable, then r is linear otherwise r is nonlinear. The least squares problem is to find an x that minimizes the 2-norm of r. If

$$f(x) = ||r||_2 \ , \tag{A.6}$$

then the least squares problem is to find

$$\min_{x \in \mathbb{C}^n} f(x). \tag{A.7}$$

A.3.1 Linear Least Squares

Theorem 2. *The minimizer* x^* *of the set of equations* $Ax = b$ *must satisfy* $B^T Bx = B^T k$ *which is called the normal equation of the system.*

Proof.

$$\begin{aligned} r &= Ax - b \Rightarrow ||r||_2 = ||Ax - b||_2 \\ ||r||_2^2 &= r^T r = (Ax - b)^T (Ax - b) \\ &= x^T A^T Ax - x^T A^T b - b^T Ax + b^T b \\ &= x^T A^T Ax - 2x^T A^T b + b^T b, \end{aligned}$$

which gives r as a function of x. For x^* to be a critical point of the system, $\frac{\partial}{\partial x}$ must be equal to 0.

$$\begin{gathered} \tfrac{\partial}{\partial x}(x^T A^T Ax - 2x^T A^T b + b^T b) = 0 \\ -2\tfrac{\partial x^T}{\partial x} A^T Ax + x^T A^T A \tfrac{\partial x}{\partial x} - 2\tfrac{\partial x^T}{\partial x} A^T b = 0 \end{gathered}$$

which gives

$$2A^T Ax - 2A^T b = 0 \Rightarrow A^T Ax = A^T b \ .$$

For x to be a minimum point $\frac{\partial^2}{\partial x^2} > 0$

$$\tfrac{\partial^2}{\partial x^2}(x^T A^T Ax - 2x^T A^T b + b^T b) = A^T A \ .$$

But $x^T A^T Ax = ||Ax||_2^2$ is the square of the 2-norm of A and by the definition of a norm, $||Ax||_2^2 > 0$ provided $Ax \neq 0$ which is the case so long as $x \neq 0$ and A is not rank deficient. If the above conditions are satisfied, then $A^T A$ is fully positive definite and we therefore have a local minimum which is also a global minimum since $||r||_2^2$ is quadratic. □

If A is of full rank, the solution of the system is given as

$$x = (A^T A)^{-1} A^T b \; .$$

Although the above solution is fast and easy to compute, if A has perturbation errors or is close to singular, which is possible if within a certain range, the time series are close together and the change in attenuation is not very significant, then the normal equations can be ill-conditioned. In that case we can use a more robust method called Singular Value Decomposition.

Theorem 3. *Let A be an arbitrary $m \times n$ matrix with $m \geq n$. Then A can be written as $A = U\Sigma V^T$ where U is an $m \times n$ matrix such that $U^T U = I$, V is an $n \times n$ orthogonal matrix and Σ is an $n \times n$ diagonal matrix with entries $\sigma_1, \sigma_2, \cdots, \sigma_n$ with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$*

Using this method our matrix equation becomes

$$U\Sigma V^T x = b \Rightarrow x = V\Sigma^{-1} U^T b \; .$$

The entries of Σ^{-1} is just the reciprocal of the entries of Σ.

A.3.2 Nonlinear Least Squares

In nonlinear least squares, the aim is to solve the equation

$$\min_{x \in \mathbb{C}} f(x) \text{ where } f(x) = \frac{1}{2} ||r(x)||_2^2 \; .$$

This is equivalent to solving the same equation without the constant term. Assuming the $r_i(x)$ are twice differentiable, the Jacobian of $r(x)$ is

$$J(x) \in \mathbb{C}_{mxn} \text{ with } J(x)_{ij} = \frac{\partial r_i(x)}{\partial x_j} \; ,$$

where $i = 1, \cdots, m$ and $j = 1, \cdots, n$.
The Hessian is

$$H_i(x) = \nabla^2 r_i(x)$$

where

$$H_i(x)_{jk} = \frac{\partial^2 r_i(x)}{\partial x_j \partial x_k} \; .$$

Then the first derivative of $f(x)$ is given by

$$\nabla f(x) = \sum_{i=1}^{m} r_i(x) \cdot \nabla r_i(x) = J(x)^T r(x)$$

and the second derivative by

$$\nabla^2 f(x) = J(x)^T J(x) + \sum_{i=1}^{m} r_i(x) H_i(x) = J(x)^T J(x) + S(x) \ .$$

The necessary condition for x^* to be a local minimum of $f(x)$ is that its first derivative must be 0

$$\nabla f(x^*) = J(x^*)^T r(x^*) = 0 \ .$$

Using Taylor series expansion, $f(x)$ can expanded quadratically around x_c as

$$\begin{aligned} f(x + x_c) = f_c(x) &= f(x_c) + \nabla f(x_c)^T (x - x_c) \\ &+ \frac{1}{2}(x - x_c)^T \nabla^2 f(x_c)(x - x_c) \\ &= \frac{1}{2}||r(x_c)||_2^2 + J(x_c) r(x_c)^T (x - x_c) \\ &+ \frac{1}{2}(x - x_c)^T [J(x_c)^T J(x_c) + S(x_c)](x - x_c) \ . \end{aligned}$$

Then according to Newton's method, the zeros of $f'(x)$ can be approximated by

$$x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)},$$

which in this case reduces to

$$x^* = x_c - (J(x_c)^T J(x_c) + S(x_c))^{-1} J(x_c)^T r(x_c).$$

Using a linear approximation of $f(x)$ gives

$$f_c(x_c) = r(x_c)^T r(x_c) + J(x_c)(x - x_c) r(x_c)^T = r(x_c) + J(x_c)(x - x_c).$$

The idea of least squares then will be to minimize

$$\min_{x \in \mathbb{C}^n} ||r(x_c) + J(x_c)(x - x_c)||_2 \ , \tag{A.8}$$

which requires only first derivative information $r(x)$. Using Newton's method on this equation also gives the solution

$$x^* = x_c - (J(x_c)^T J(x_c))^{-1} J(x_c)^T r(x_c). \tag{A.9}$$

Equation A.8 can also be solved by applying linear least squares solution directly on it instead of using A.9. One other method used by Matlab and employed by the Levenberg-Maquardt method is to minimize the function

$$\min_{x \in \mathbb{C}^n} ||r(x_c) + J(x_c)(x_+ - x_c)||_2 \tag{A.10}$$

subject to the condition that

$$||x_+ - x_c||_2 \leq \delta_c, \tag{A.11}$$

where δ_c defines a trust region where the minimization function $f_c(x)$ can be trusted to accurately model the original function $f(x)$. The above formulation gives

$$x_+ = x_c - (J(x_c)^T J(x_c) + \mu_c I)^{-1} J(x_c)^T r(x_c), \tag{A.12}$$

where $\mu_c = 0$ if $\delta_c \geq ||(J(x_c)^T J(x_c))^{-1} J(x_c)^T r(x_c)||_2$ and $\mu_c > 0$ otherwise.

Bibliography

[1] *Matlab 7: 3D Visualization.*

[2] *Matlab Documentation.*

[3] Laurence Annet, Laurent Hermoye, Frank Peeters, Francois Jamar, Jean-Paul Dehoux, and Bernard E. Van Beers. Glomerular filtration rate: Assessment with dynamic contrast-enhanced mri and a cortical-compartment model in the rabbit kidney. *Journal of Magnetic Resonance Imaging*, 20(5):843–849, November 2004.

[4] Silke Aumann, Stefan O. Schoenberg, Armin Just, Karen Briley-Saebo, Atle Bjornerud, Michael Bock, and Gunnar Brix. Quantification of renal perfusion using an intravascular contrast agent (part 1): Results in a canine model. *Magnetic Resonance in Medicine*, 49(2):276–287, 2003.

[5] Ake Bjõrck. *Numerical Methods for Least Squares Problems.* SIAM, Philadelphia, Penn., 1996.

[6] David L. Buckley, Ala'a E. Shurrab, Ching M. Cheung, Andrew P. Jones, Hari Mamtora, and Philip A. Kalra. Measurement of single kidney function using dynamic contrast-enhanced mri: Comparison of two models in human subjects. *Journal of Magnetic Resonance Imaging*, 24(5):1117–1123, August 2006.

[7] A.A. Chan and S.J. Nelson. Simplified gamma-variate fitting of perfusion curves. *Biomedical Imaging: Nano to Macro, 2004. IEEE International Symposium on*, pages 1067–1070 Vol. 2, April 2004.

[8] George N. Coritsidis. Renal blood flow-glomerular filtration rate. stony brook university hospital and health sciences center. `http://www.uhmc.`

sunysb.edu/internalmed/nephro/webpages/Part_A.htm. 15 November, 2008.

[9] Elena Daghini, Laurent Juillard, John A. Haas, James D. Krier, Juan C. Romero, and Lilach O. Lerman. Comparison of mathematic models for assessment of glomerular filtration rate with electron-beam ct in pigs. *Radiology*, 242(2):417–424, February 2007.

[10] J. E. Dennis, Jr. and Robert B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations (Classics in Applied Mathematics, 16)*. Soc for Industrial & Applied Math, 1996.

[11] Alireza Entezari, Randy Scoggins, Torsten Möller, and Raghu Machiraju. Shading for fourier volume rendering. pages 131–138, 2002.

[12] J. H. Gillard, N. M. Antoun, N. G. Burnet, and J. D. Pickard. Reproducibility of quantitative ct perfusion imaging. *British Journal of Radiology*, 74(882):552–555, June 2001.

[13] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing, (3rd Edition)*. Prentice Hall, 2007.

[14] Nils Hackstein, Hendrik Kooijman, Stefan Tomaselli, and Wigbert S. Rau. Glomerular filtration rate measured using the patlak plot technique and contrast-enhanced dynamic mri with different amounts of gadolinium-dtpa. *Journal of Magnetic Resonance Imaging*, 22(3):406–414, 2005.

[15] Nils Hackstein, Cornelia Wiegand, Wigbert Stefan Rau, and Alexander Claus Langheinrich. Glomerular filtration rate measured by using triphasic helical ct with a two-point patlak plot technique. *Radiology*, 230:221–226, 2004.

[16] L. Hermoye, L. Annet, Ph. Lemmerling, F. Peeters, F. Jamar, P. Gianello, S. Van Huffel, and B.E. Van Beers. Calculation of the renal perfusion and glomerular filtration rate from the renal impulse response obtained with mri. *Magnetic Resonance in Medicine*, 51(5):1017–1025, 2004.

[17] Marc Levoy. Volume rendering using the fourier projection-slice theorem. In *Proceedings of the conference on Graphics interface '92*, pages 61–69, 1992.

[18] Mark T. Madsen. A simplified formulation of the gamma variate function. *Physics in Medicine and Biology*, 37(7):1597–1600, 1992.

[19] Tom Malzbender. Fourier volume rendering. *ACM Transactions on Graphics*, 12(3):233–250, July 1993.

[20] Paul Meier and Kenneth L. Zierler. On the theory of the indicator-dilution method for measurement of blood flow and volume. *J Appl Physiol*, 6:731–744, 1954.

[21] Iosif Mendichovszky, Michael Pedersen, Jørgen Frøkiær, Thomas Dissing, Nicolas Grenier, Peter Anderson, Kieran Mchugh, Qing Yang, and Isky Gordon. How accurate is dynamic contrast-enhanced mri in the assessment of renal glomerular filtration rate? a critical appraisal. *Journal of Magnetic Resonance Imaging*, 27(4):925–931, February 2008.

[22] Jussi Perkiö, Hannu J Aronen, Aki Kangasmäki, Yawu Liu, Jari Karonen, Sauli Savolainen, and Leif Østergaard. Evaluation of four postprocessing methods for determination of cerebral blood volume and mean transit time by dynamic susceptibility contrast. *Magnetic Resonance in Medicine*, 47:973–981, 2002.

[23] Burton D Rose and Post Theodore W. Chapter 2c: Determinants of gfr. `http://www.uptodate.com/patients/content/topic.do?topicKey=~wcZcV7zgQfEHFw`. 15 November, 2008.

[24] Burton D Rose and Post Theodore W. Chapter 7b: Exchange of water between plasma and interstitial fluid. `http://www.uptodate.com/patients/content/topic.do?topicKey=~MM.FYEZLPIspy6M`. 15 November, 2008.

[25] AM Smith, R Materne, and BE Van Beers. Quantitative measurement of blood perfusion, gfr and arterial vascular fraction in the kidney cortex. In *Proceedings of the 9th Annual Meeting of ISMRM, Glasgow, Scotland*, page 367, 2001.

[26] Steven P. Sourbron, Henrik J. Michaely, Maximilian F. Reiser, and Stefan O. Schoenberg. Mri-measurement of perfusion and glomerular filtration in the human kidney with a separable compartment model. *Investigative radiology*, 43(1):40–48, January 2008.

[27] Alexandru C. Telea. *Data Visualization: Principles and Practice.* A K Peters, 2008.

[28] Jr. Thompson, Howard K., C. Frank Starmer, Robert E. Whalen, and Henry D. Mcintosh. Indicator transit time considered as a gamma variate. *Circ Res*, 14(6):502–515, 1964.

[29] Takashi Totsuka and Marc Levoy. Frequency domain volume rendering. In *SIGGRAPH '93: Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 271–278. ACM, 1993.

[30] Lloyd N. Trefethen and David Bau III. *Numerical linear algebra.* Society for Industrial and Applied Mathematics, 1997.

[31] Evert-jan Ph.A. Vonken, Matthias J.P. van Osch, Chris J.G. Bakker, and Max A. Viergever. Measurement of cerebral perfusion with dual-echo multi-slice quantitative dynamic susceptibility contrast mri. *Journal of Magnetic Resonance Imaging*, 10:109–117, 1999.

[32] Robert M. Weisskoff, David Chesler, Jerrold L. Boxerman, and Bruce R. Rosen. Pitfalls in mr measurement of tissue blood flow with intravascular tracers: Which mean transit time? *Magnetic Resonance in Medicine*, 29(4):553–558, 1993.

[33] Jeff L Zhang, Henry Rusinek, Louisa Bokacheva, Lilach O Lerman, Qun Chen, Chekema Prince, Niels Oesingmann, Ting Song, and Vivian S Lee. Functional assessment of the kidney from magnetic resonance and computed tomography renography: impulse retention approach to a multicompartment model. *Magnetic resonance in medicine*, 59(2):278–88, 2008.