GLOBAL KINETIC IMAGING USING DYNAMIC POSITRON EMISSION TOMOGRAPHY DATA
by
Cristina Negoita

has been approved
July 2003

APPROVED:

Supervisory Committee

ACCEPTED:

Department Chair

Dean, Graduate College
This work studies the generation of global parametric images using dynamic positron emission tomography (PET) data. Unlike other imaging modalities, dynamic PET data suffers from low count statistics, leading to poor signal-to-noise ratios. This makes the solution of the inverse problem for estimation of the parameters difficult. PET is a useful diagnostic tool due to its ability to image biological functions \textit{in vivo}. Most of the work in dynamic PET imaging has focused on region of interest studies; averaged activity curves are extracted from regions of uniform tracer concentration and then used for parameter estimation. Global pixel-by-pixel kinetic images fully capture the spatial distribution of functional activity. This study focuses on the dynamics of one specific tracer, fluoro-deoxy-glucose (FDG), for which an initial value problem model has been previously validated.

In this dissertation, a new theoretical analysis of the FDG model is developed. Perturbation and sensitivity analyses are used to characterize the forward solution with respect to small variations in the kinetic parameters. Simulation studies confirm the analytical results thus derived. The numerical analysis of the generalized linear least squares algorithm used in parameter estimation is also presented. A constrained linear least squares method using a parameter constraint derived from data clustering is found to give the most accurate results. Spillover and partial volume effects (PVE) are studied; the impact of PVE corrections is limited on the accuracy of the kinetic estimates. In addition, this study proposes the use of nonlinear anisotropic diffusion filtering of kinetic images. Although nonlinear anisotropic diffusion improves parametric image quality, small abnormalities, potentially relevant to early diagnosis of disease, may be inadvertently smoothed.
To Domnul Gâdea,
poet and mathematician
ACKNOWLEDGMENTS

I would like to thank all my teachers and colleagues, my parents and friends, my husband and pets, for sharing with me their passion for life in general, and mathematics in particular. I thank my advisor and mentor, Professor Rosemary Renaut, for guiding me through the convoluted world of research in applied mathematics. She has patiently taught, corrected and encouraged me in my work, and offered me her candid friendship along the way. I thank Dr. Kewei Chen for introducing me to Positron Emission Tomography, and diligently teaching me the intricacies of this field. He is a resourceful and insightful researcher, and I am grateful for his guidance. My work has been partly supported throughout the past five years by the Arizona Alzheimer’s Disease Research Center. I greatly appreciate their investment in me, and hope the work presented here will soon be put to their use.
# TABLE OF CONTENTS

**LIST OF TABLES** ................................................................. viii

**LIST OF FIGURES** ............................................................. ix

**CHAPTER 1 Introduction** .................................................. 1
  1. Background - Positron Emission Tomography .......................... 1
     1.1. Dynamic Positron Emission Tomography .......................... 3
     1.2. Physical Factors Affecting PET Quantification ................. 4
     1.3. Signal to Noise Ratio Estimates .................................. 4
  2. Compartment Model for Fluoro-Deoxy-Glucose (FDG) Dynamics .... 5
  3. Overview ................................................................. 7

**CHAPTER 2 Theoretical Analysis of the Compartmental Model for FDG** 13
  1. Analysis of the Forward Problem ...................................... 13
     1.1. Non-dimensional Model Formulation .............................. 14
     1.2. Perturbation Analysis ............................................. 16
     1.3. Simulations ....................................................... 21
  2. Analysis of the Inverse Problem ...................................... 22
     2.1. Sensitivity Analysis ............................................... 22
     2.2. Simulations ....................................................... 32
  3. Conclusions and Discussion ........................................... 34

**CHAPTER 3 Computational Techniques** ................................ 47
  1. The Linear Problem and its Discrete Formulation .................. 47
     1.1. Laplace Formulation ............................................... 48
     1.2. Volterra Formulation .............................................. 49
  2. Error Analysis ......................................................... 50
     2.1. Interpolation Error ............................................... 51
     2.2. Simulations ....................................................... 52
     2.3. Measurement Error ................................................ 52
     2.4. Simulations ....................................................... 55
  3. Numerical Algorithms .................................................. 55
     3.1. Nonlinear Minimization Algorithm .............................. 56
     3.2. The Linear Least Squares Method - LS ........................ 58
     3.3. The Total Least Squares Method - TLS ........................ 60
     3.4. The Generalized Linear Least Squares Algorithm - GLLS ..... 61
  4. Simulations ............................................................. 70
  5. Conclusions and Discussion ........................................... 71
# Table of Contents

## CHAPTER 4  Physical Measurement Effects in PET Studies  
1. Spill-Over Effects ........................................... 76  
2. Spill-Over and Partial Volume Effects ...................... 79  
3. Discussion on Bound Constraints  .......................... 80  
3.1. Results - Spill-Over Effects ............................. 81  
3.2. Results - Spill-Over and Partial Volume Effects .......... 82  
4. Conclusions and Discussion .................................. 83  

## CHAPTER 5 Nonlinear Anisotropic Diffusion Filtering  
1. Problem Formulation ......................................... 92  
2. Nonlinear Anisotropic Diffusion  ............................ 93  
2.1. Results - Filtered Parametric Images ..................... 95  
2.2. Results - Parametric Filtered Tissue Activity Images .... 96  
3. Conclusions and Discussion .................................. 97  

## CHAPTER 6 Conclusions and Future Work ....................... 106  

## REFERENCES .................................................... 109  

## APPENDIX A THEORETICAL DERIVATIONS - MAPLE CODES .... 116  

## APPENDIX B NUMERICAL ALGORITHMS - MATLAB CODES .... 119
<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Analytically derived expressions for $C_i = \lim_{t \to -\infty} \int_0^t g_i(k, s) ds$ from Remarks 4, 5, 6, 7, when considering a solution to (2.7) with $k_4 \neq 0$. The forms for $c_1, c_2, c_3$ are given in (2.33).</td>
<td>33</td>
</tr>
<tr>
<td>2</td>
<td>Computed values for the uniform bounds $C_i$, given two sets of experimentally derived $k$ values, from [29], [46].</td>
<td>33</td>
</tr>
<tr>
<td>3</td>
<td>Numerical results obtained using the Laplace and Volterra formulations. Note, the accuracy of the Volterra estimates is comparable with those given by the Laplace, with at least one order of magnitude reduction in cost. The estimates given here are the least squares solutions obtained using a QR decomposition. The Volterra formulation does not depend on the size of the time step $dt$, used only in the double integral formulation, but we include it here to facilitate the cost comparison.</td>
<td>53</td>
</tr>
<tr>
<td>4</td>
<td>Relative error estimates for the Laplace and Volterra formulations. Here, $\hat{k}$ represents an estimate of the exact solution vector $k$.</td>
<td>53</td>
</tr>
<tr>
<td>5</td>
<td>Average and standard deviation measures computed over a sample of 1000 trials, to compare the results of the Laplace (L) and Volterra (V) formulations for the solution to the inverse problem. $K = K(X_{L,V})$ denotes the matrix condition number.</td>
<td>55</td>
</tr>
<tr>
<td>6</td>
<td>Relative error for the estimates presented in Table 5. Here, $\hat{k}$ represents the mean estimate of the exact solution vector $k$.</td>
<td>55</td>
</tr>
<tr>
<td>7</td>
<td>GLLS estimates obtained for Example 1.</td>
<td>69</td>
</tr>
<tr>
<td>8</td>
<td>GLLS estimates obtained for Example 2.</td>
<td>69</td>
</tr>
<tr>
<td>9</td>
<td>GLLS estimates obtained for Example 3.</td>
<td>70</td>
</tr>
<tr>
<td>10</td>
<td>Estimates obtained from each method, when there is no noise present in the data.</td>
<td>72</td>
</tr>
<tr>
<td>11</td>
<td>$k$ values published by Feng, et al. in [16].</td>
<td>81</td>
</tr>
<tr>
<td>12</td>
<td>$k$ values published by Piert, et al. in [46].</td>
<td>81</td>
</tr>
<tr>
<td>13</td>
<td>$k$ values obtained from fitting the clustered curves of the data under study.</td>
<td>82</td>
</tr>
<tr>
<td>14</td>
<td>Slope of regression line and R-Squared values comparing $K$ values using bounds from literature, Feng, [16], Piert, [46], and from clustering, [23].</td>
<td>98</td>
</tr>
<tr>
<td>15</td>
<td>Slope of regression line and R-Squared values comparing $K$ values using no filtering, a-posteriori ADF, and a-priori tissue activity curve (TAC) filtering.</td>
<td>99</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Release of positrons from the nucleus of a radio-active isotope.</td>
<td>2</td>
</tr>
<tr>
<td>2.</td>
<td>Sinogram image.</td>
<td>3</td>
</tr>
<tr>
<td>3.</td>
<td>Last time frame (frame 22, t = 45 minutes) of 31 slices of 2D sinograms of a human brain, reconstructed via filtered back projection.</td>
<td>8</td>
</tr>
<tr>
<td>4.</td>
<td>Time-space dynamics for slice number 16, over 22 frames.</td>
<td>9</td>
</tr>
<tr>
<td>5.</td>
<td>Signal-to-Noise Ratio (SNR) for slice number 16.</td>
<td>10</td>
</tr>
<tr>
<td>6.</td>
<td>SNR values for Figure 5 for early time.</td>
<td>10</td>
</tr>
<tr>
<td>7.</td>
<td>Clustered curves obtained from a human study, using a clustering algorithm described in [23].</td>
<td>11</td>
</tr>
<tr>
<td>8.</td>
<td>Signal-to-Noise Ratio for slice 16, using clustered curves derived following [23], see Figure 7.</td>
<td>11</td>
</tr>
<tr>
<td>9.</td>
<td>Compartment model structure for FDG metabolism.</td>
<td>11</td>
</tr>
<tr>
<td>10.</td>
<td>Comparison of input functions (u(t)), plasma time activity curves (PTAC), from real studies and simulations. Note, the scales are different depending on the model adopted.</td>
<td>12</td>
</tr>
<tr>
<td>11.</td>
<td>Comparison of (Y(\tau)), assuming (k_4 = .0068) and (Y_0(\tau)), assuming (k_4 = 0). Both (Y(\tau)), and (Y_0(\tau)) were obtained using a second order Runge-Kutta method for the numerical solution of (1.2), and a vector of parameter values, (k = [.102,.130,.062]) given in [29]. The final measurement time for this graph is (T = 25) min.</td>
<td>35</td>
</tr>
<tr>
<td>12.</td>
<td>As Figure 11, but to time (T = 45) min.</td>
<td>36</td>
</tr>
<tr>
<td>13.</td>
<td>Graph comparing the variation of (k_1) and (k_2) as functions of end-time, using two algorithms, both based on the Filtered Least Squares (FLS) method, one assuming (k_4 \neq 0), and one assuming (k_4 = 0). The values depicted by * are exact parameter values.</td>
<td>37</td>
</tr>
<tr>
<td>14.</td>
<td>Graph comparing the variation of (k_3) and (K = \frac{k_1 k_2}{k_2 + k_3}) as functions of end-time, using two algorithms, both based on FLS, one which assumes (k_4 \neq 0), and one assuming (k_4 = 0). The values depicted by * are exact parameter values.</td>
<td>38</td>
</tr>
<tr>
<td>15.</td>
<td>The functions (g_i(t) = \frac{\partial}{\partial k_i} f(k_i, t)) for the parameter vector (k_H = [.102,.13,.062,.0068]).</td>
<td>39</td>
</tr>
<tr>
<td>16.</td>
<td>The functions (g_i(t) = \frac{\partial}{\partial k_i} f(k_i, t)) for parameter vector (k_P = [.099,.141,.165,.013]), referenced in [46].</td>
<td>40</td>
</tr>
<tr>
<td>17.</td>
<td>Magnification of the graphs shown in Figure 16.</td>
<td>41</td>
</tr>
<tr>
<td>18.</td>
<td>The functions (N_i(t) = \int_0^t</td>
<td>g_i(s)</td>
</tr>
<tr>
<td>19.</td>
<td>The functions (N_i(t) = \int_0^t</td>
<td>g_i(s)</td>
</tr>
<tr>
<td>20.</td>
<td>Magnification of the graphs shown in Figure 19.</td>
<td>44</td>
</tr>
</tbody>
</table>
21. Comparison of the uniform bounds $C_i$, and the integral values $N_i(t)$ for each $i = 1, \ldots, 4$, using parameter vector $k_H = [0.102, 0.13, 0.062, 0.0068]$.  

22. Comparison of the uniform bounds $C_i$, and the integral values $N_i(t)$ for each $i = 1, \ldots, 4$, using parameter vector $k_P = [0.099, 0.141, 0.165, 0.013]$.  

23. Log plot of the relative solution error $\frac{\|y^{(k)} - y^{(k-1)}\|}{\|y^{(k)}\|}$ in Example 3.  

24. Log plot of the relative error $|Y - \hat{y}^{(k)}|$ in Example 3.  

25. Input and output functions used in simulation.  

26. Comparison of mean, std, bias and deviation over a sample size of 1000 runs, for various methods estimating the $k$ parameter with 1% noise in the output function.  

27. Comparison of mean, std, bias and deviation over a sample size of 1000 runs, for various methods estimating the $k$ parameter with 2% noise in the output function.  

28. Comparison of mean, std, bias and deviation over a sample size of 1000 runs, for various methods estimating the $k$ parameter with 5% noise in the output function.  

29. Comparison of mean, std, bias and deviation over a sample size of 1000 runs, for various methods estimating the $k$ parameter with 10% noise in the output function.  

30. Comparison of mean, std, bias and deviation over a sample size of 1000 runs, for various methods estimating the $k$ parameter with 20% noise in the output function.  

31. $k_1$ image using spill-over corrected model.  

32. $k_2$ image using spill-over corrected model.  

33. $k_3$ image using spill-over corrected model.  

34. $k_4$ image using spill-over corrected model.  

35. $m_{bt}$ image using spill-over corrected model.  

36. $K$ image using spill-over corrected model.  

37. $K$ image derived by using Patlak analysis.  

38. $k_1$ image using spill-over and partial volume effects corrected model.  

39. $k_2$ image using spill-over and partial volume effects corrected model.  

40. $k_3$ image using spill-over and partial volume effects corrected model.  

41. $k_4$ image using spill-over and partial volume effects corrected model.  

42. $m_{bt}$ image using spill-over and partial volume effects corrected model.  

43. $K$ image using spill-over and partial volume effects corrected model.
44. Pixel-wise comparison of parametric image of $K = \frac{k_1k_3}{k_2+k_3}$ estimated by both Patlak analysis and by pixel-by-pixel analysis, using a Constrained Filtered Least Squares (CFLS) algorithm with bounds from Feng [16], Piert [46], and clustering. The data was corrected for spill-over effects. The last graph shows the data from all three graphs in one figure.  

45. Pixel-wise comparison of parametric image of $K = \frac{k_1k_3}{k_2+k_3}$ estimated by both Patlak analysis and by pixel-by-pixel analysis, using CFLS algorithm and bounds from Feng [16], Piert [46], and clustering. The data was corrected for spill-over and partial volume effects. The last graph shows the data from all three graphs in one figure.  

46. Anisotropic diffusion filtered (ADF) image for $k_1$ using data corrected for spill-over effects.  

47. Anisotropic diffusion filtered image for $k_2$ using data corrected for spill-over effects.  

48. Anisotropic diffusion filtered image for $k_3$ using data corrected for spill-over effects.  

49. Anisotropic diffusion filtered image for $k_4$ using data corrected for spill-over effects.  

50. Anisotropic diffusion filtered image for $K$ using data corrected for spill-over effects.  

51. Anisotropic diffusion filtered Patlak $K$ image.  

52. Anisotropic diffusion filtered image for $k_1$ using filtered output data, corrected for spill-over and partial volume effects.  

53. Anisotropic diffusion filtered image for $k_2$ using filtered output data, corrected for spill-over and partial volume effects.  

54. Anisotropic diffusion filtered image for $k_3$ using filtered output data, corrected for spill-over and partial volume effects.  

55. Anisotropic diffusion filtered image for $k_4$ using filtered output data, corrected for spill-over and partial volume effects.  

56. Anisotropic diffusion filtered image for $m_B$ using filtered output data, corrected for spill-over and partial volume effects.  

57. Anisotropic diffusion filtered image for $K = \frac{k_1k_3}{k_2+k_3}$ using filtered output data, corrected for spill-over and partial volume effects.  

58. Anisotropic diffusion filtered image for $K = \frac{k_1k_3}{k_2+k_3}$ using Patlak analysis.  

59. Pixel-wise comparison of parametric image of $K = \frac{k_1k_3}{k_2+k_3}$ estimated by both Patlak analysis and by pixel-by-pixel analysis, using CFLS algorithm and bounds from Feng [16], Piert [46], and clustering. The data was corrected for spill-over and partial volume effects, but no filtering was used. The last graph shows the data from all three graphs in one figure.  

xi
60. Pixel-wise comparison of anisotropic filtered parametric image for $K = \frac{k_1 k_3}{k_2 + k_3}$ estimated by both Patlak analysis and by pixel-by-pixel analysis, using CFLS algorithm, with bounds derived from clustering. The images are derived using no filtering, parametric image filtering, and Time Activity Curve (TAC) a-priori filtering. The regression line estimates gave, respectively, a slope of 1.0308, 1.0277, 1.3743, and a R-Squared value 0.4497, 0.444, 0.6791.

61. Image for $\frac{k_3}{k_2}$, which is directly proportional to $K$, i.e., an increase in $\frac{k_3}{k_2}$ will cause an increase in $K$, assuming $k_1$ remains constant.
CHAPTER 1

Introduction

Unlike anatomical based imaging techniques such as magnetic resonance imaging (MRI) and X-ray computed tomography (CT), positron emission tomography (PET) is a radio-tracer functional imaging technique which allows in vivo visualization of biochemical and physiological processes [44]. Traditionally, these processes have been studied via invasive methods such as blood sampling or by sacrificing animals to localize tracer concentration in tissue [52].

Estimation of regional physiological parameters is useful for clinical diagnoses as well as for testing drug treatments. Using PET, one can follow a metabolic process of interest by imaging the concentration distribution of an appropriately-labeled tracer.

We are interested in using PET to image brain activity in patients with Alzheimer’s disease (AD). In AD studies, one way to measure disease progression is by measuring the glucose uptake in the brain. Studies [29, 45, 7] which determine a local cerebral metabolic rate (LCMR) of glucose uptake in a region of interest have proved successful in understanding AD progression.

More specific information may be obtained by estimating the individual kinetic parameters which describe glucose dynamics. In particular, it is believed that the individual kinetic parameters may be used for early detection of AD. Our study is focused towards understanding the spatial distribution of kinetic parameters in AD, as well as towards developing a precise measure for utilization in the early detection of AD.

1. Background - Positron Emission Tomography

To understand the nature of the data acquired by a PET camera, the physics of a radioactive decay process are first explained. Following data acquisition, a reconstruction process converts the raw data into a 2D-pixel or 3D-voxel image of the spatial distribution of tracer concentration. A dynamic sequence of 2D or 3D images is then obtained, and further processed into a parametric or kinetic image.

In a PET study, a small amount of a detectable positron-emitting substance is injected into the body. This positron labeled tracer is chosen accordingly to follow a certain physiological or biochemical process under study (glucose metabolism, blood flow, etc). In the process of being labeled with positrons, the tracer becomes a radioactive isotope. The
The basic idea behind PET is the measurement of the stabilization of a radioactive isotope through the process of positron emission.

Positrons are positively charged electrons. A deficiency of neutrons in the nucleus of an atom causes a substance to become radioactive. For example, oxygen-15 has 8 protons and 7 neutrons, while the non-radioactive counterpart, oxygen-16, has 8 protons and 8 neutrons. Positrons emitted from the nucleus of a radioactive isotope stabilize the nucleus by converting protons into neutrons. The released positrons travel a small distance, dependent on their energy (usually up to a few millimeters), thus expending their kinetic energy, after which they come to rest, Figure 1. A negatively charged electron is attracted by the resting positron and when in contact these two particles annihilate. The energy created in this annihilation process causes two photons to form at approximately $180^\circ$ apart (the variation near $180^\circ$ is due to Compton scatter). The energy of the photons (511 kilo electron volts) allows them to exit a body at the speed of light. Radiation detectors are placed inside the PET camera near the exit points of the two photons. The detectors are coupled such that they record an event only when they are both struck at the same time, by a photon of an appropriate energy level (thus reducing the chance of counting accidental coincidences). The signal detected by a pair of detectors represents the sum activity along the line connecting the two detectors. A ring of coupled detectors is used to record photons coming from all directions. The coincident signals thus measured are stored in a matrix form called a sinogram, Figure 2 - this is the raw data acquired by PET.

Through an image reconstruction process [50], each sinogram is transformed into an image of tracer concentration distribution. The reconstruction of a sinogram is an inverse problem, which can be numerically solved by various algorithms such as filtered back projection, [27], maximum-likelihood minimization [50], or other iterative reconstruction methods [37]. The reconstructed images used for the kinetic study presented in this thesis were generated via a filtered back projection. However, we do not focus on reconstruction methods in this work. Figure 3 shows a PET brain scan of 31-slices, from a real study, where each slice was reconstructed via a filtered back projection.
1.1. Dynamic Positron Emission Tomography. The process presented so far describes the acquisition and reconstruction for a single time interval. In dynamic PET images are obtained for a sequence of time intervals. Specifically, repeated scans are made on an interval $[0, T]$, at times $t_0 = 0, t_1, \ldots, t_m = T$. The time intervals $t_{i+1} - t_i$ are not fixed equally, but are specified by a given scanning protocol, see Figure 4 where the following scanning protocol was used:

$$t = [0, 1, 2167, 25, 2833, 3167, 35, 3833, 4167, 45, 55, 7167, 9, 1.25, 2, 3, 4.25, 5.75, 8.25, 12.5, 17.5, 25, 45] \text{ min}.$$  

For these time frames, different noise levels are exhibited; the signal-to-noise ratio (SNR) of the measured time-space activity is limited primarily by dose constraints of the injected radio-isotope, length of each time frame in the scanning protocol, physical properties of the tracer, and number of events recorded by detectors, [60].

The time dynamics of the tracer are modeled via a tracer kinetic model - usually an initial value problem in time. Due to the low SNR, particularly in early frames, region of interest (ROI) approaches have been used to characterize tracer kinetics [29]. If the ROI includes enough pixels, the SNR ratio is improved, and the data can then be used in nonlinear estimation procedures. A kinetic analysis of PET data performed on a pixel-by-pixel basis generates parametric images describing the spatial distribution of a biochemical or physiological process.
1.2. Physical Factors Affecting PET Quantification. The difficulty in analyzing PET data comes from its high level of noise. Several factors affect the spatial resolution of PET. Some are due to system physics and equipment limitations like the intrinsic detector resolution, positron range (the distance a positron travels before it annihilates), angulation error (the variation in the 180° angle joining the lines of travel of the two photons produced by annihilation), scatter and random coincidences, spill-over and partial-volume effects. Other sources of noise are introduced by the image reconstruction process [50], patient movement [44], and the compartment model as noted below, [18].

1.3. Signal to Noise Ratio Estimates. Let $\tilde{y}(t)$ be the pixel tracer concentration measured by the PET camera. Then $\tilde{y}(t) = y(t) + \epsilon(t)$, where $y(t)$ is the true concentration, and $\epsilon(t)$ is the noise introduced by the measurement process. Then, the mutual information of $y(t)$ and $\tilde{y}(t)$ is the average information that observations of $\tilde{y}(t)$ can provide about the true signal $y(t)$. Under certain assumptions about noise characteristics [47], one can find that the mutual information is an increasing function of the signal to noise ratio (SNR). The signal-to-noise ratio is defined as

$$SNR = \frac{\sigma_s}{\sigma_n}$$

(1.1)

where $\sigma_s$ and $\sigma_n$ are the variance of the signal and noise, respectively. In PET, the observed data is the true signal plus the noise, $y(t) + \epsilon(t)$. Early frames are dominated by noise, while in later frames, the noise is small. Traditionally, [9], in order to get an approximate value for the SNR, one computes the variance of the signal outside the brain, in the background region, and considers this to be the variance of the noise. To obtain the variance of the true signal, a region of interest is drawn in a high activity part of the brain, and the resulting variance is considered the variance of the signal. Using these criteria, we estimated the SNR as a function of time, for a 2D slice of the human brain from a dynamic set of data, see Figures 5 and 6.

Note that the SNR is very low over the initial time intervals, and increases gradually relative to the time interval length, after 12 minutes. The assumption in this calculation is that the noise is spatially uniform for a given time interval.

Observations made in other studies, [51], suggest that areas of high activity will tend to have a larger variance, whereas areas of low activity will have smaller variance. Absorption of tracer into tissue is nonuniform in the brain, in particular, gray matter tends to have a higher number of counts than white matter, and cerebral spinal fluid (CSF) is usually assumed to have no counts at all, hence a zero variance. To understand whether this is true for this set of real data, we clustered the data over time into 5 different clusters, resulting in 5 different time activity curves, illustrated in Figure 7, [23]. We considered these to be the true signals $y(t)$. For each curve, we computed the variance of the signal as the variance over a cluster, while the variance of the noise was assumed the same for all signals.

As shown in Figure 7, the activity count increases with cluster curve number, while, from Figure 8, the SNR decreases with respect to cluster curve number, except for cluster
1. This suggests that the variance for clustered curves with higher activity is lower than the variance of regions with lower activity. This contradicts the observations made in [51]. However, for the clustered curves 2 and 3, which represent regions with a higher activity count then curve 1, the SNR is also higher, hence the variance is higher, confirming Soares, et al. [51]. The same observations hold true when one compares curves 2 to 3 alone. Although the number of clustered curves used here is small, it suggests that results of Soares, et al. [51] are true in an averaged sense. We conclude with the observation that SNR is small for early time frames, and varies non-uniformly for a 2D signal, at each time interval.

2. **Compartment Model for Fluoro-Deoxy-Glucose (FDG) Dynamics**

For each physiological process under study, a specific radioactive tracer is developed which undergoes similar dynamics with its natural counterpart. The dynamics of each tracer are modeled mathematically. Certain assumptions and simplification are made in model development to obtain a working model of the problem. For Alzheimer’s studies, Fluoro-2-Deoxy-Glucose (FDG), an analogue of glucose, is used to measure glucose uptake, [29].

We consider the compartmental model for the dynamics of the FDG tracer, originally developed by Sokoloff [52]. Traditionally, one first determines the kinetic rates for FDG, then estimates the true kinetic rates for glucose, following the Michaelis-Menton relationship for competitive substrates, see [29].

At each pixel we consider the phosphorylation process for FDG. The model consists of three compartments: the FDG concentration in plasma, \( u(t) \), the FDG concentration in tissue, \( y_1(t) \), and the phosphorylated FDG (FDG-6-phosphate) concentration in tissue, \( y_2(t) \).

The FDG plasma time activity curve (PTAC), the model input \( u(t) \), can be derived either invasively, by blood sampling or non-invasively, from a region of interest in the reconstructed PET images, [63]. The PET scanner provides a measure of the combined FDG and phosphorylated FDG concentration in tissue, i.e. a measure of \( y(t) = y_1(t) + y_2(t) \). The basic assumptions of the model, [18], [29], are:

- Glucose metabolism in tissue is in steady state.
- Concentrations of glucose, FDG and FDG-6-P, are homogeneous in the compartments.
- Concentrations of FDG and FDG-6-P are small in comparison to their natural counterparts such that their presence does not affect the steady-state environment of glucose metabolism, and the kinetics of the system is first order.
- The transports of glucose and FDG to local brain tissue are not flow limited.
The compartment model illustrated in Figure 9 is described mathematically by the initial value problem, at the single cell, or pixel, level:

\[
\begin{align*}
\frac{dy_1}{dt} &= k_1 u(t) - (k_2 + k_3)y_1(t) + k_4 y_2(t), \\
\frac{dy_2}{dt} &= k_3 y_1(t) - k_4 y_2(t), \\
y_1(0) &= 0, \quad y_2(0) = 0,
\end{align*}
\]

The system rate constants are interpreted as follows:

- $k_1$ is the transport rate of FDG from the blood to the extra-vascular space,
- $k_2$ is the transport rate back of FDG from the extra-vascular space to the blood,
- $k_3$ is the phosphorylation rate of the intra-cellular FDG by hexokinase enzymes to FDG-6-phosphate, and
- $k_4$ is the de-phosphorylation rate of the intra-cellular FDG-6-phosphate back to FDG.

We are interested in estimating the reaction rates $k_i$, $i = 1, 2, 3, 4$, $k_i \in (0,1), \forall i$. If we assume, as it is standard, that $y(t)$ is a twice differentiable function \[13\] and $u(t)$ is differentiable, we can rewrite the initial value problem (1.2) as a second order differential equation in time,

\[
\begin{align*}
\ddot{y}(t) + a_1 \dot{y}(t) + a_2 y(t) &= b_1 \dot{u}(t) + b_2 u(t), \\
y(0) &= 0, \quad \dot{y}(0) = 0,
\end{align*}
\]

where

\[
[a_1, a_2, b_1, b_2] = [k_2 + k_3 + k_4, k_2 k_4, k_1, k_1 (k_3 + k_4)].
\]

In literature \[29\], the input function, which represents both a natural radioactive tracer decay and FDG metabolism, is generally assumed to have the form of a sum of exponentials, $\sum_{i=1}^{n} c_i e^{-d_i t}$, $d_i > 0$, $c_i$ at most a linear function of $t$, and $n \geq 2$. This is clearly a continuously differentiable function for all $t \geq 0$. Hence, assuming an input function that is continuously differentiable, the differentiability requirements on $y(t)$ follow from the requirements on $u(t)$.

Physically, the input-plasma data represents the time-dependent concentration of tracer arriving at the region of interest. Initially, there is no tracer in the system, therefore $u(0) = 0$. As $u(t)$ represents a continuous transport process, it is certainly a continuous function. Once the input uptake occurs, the concentration of the tracer in plasma exhibits a rapid decay towards 0. The differentiability of $u(t)$ is a result of the fact that transport in time by plasma occurs smoothly and not in sudden bursts of increased tracer concentration. Initially, however, as the tracer is injected in the patient, the tracer concentration function
has a jump from 0 to some positive value (note here we ignore the effects of delay in transport of the tracer from the place of injection to the measurement area). Thus, the differentiability of \( u(t) \) at \( t = 0 \) is not physically justifiable. Small errors are likely to be introduced by this assumption, given that we consider \( y(0) = 0 \) and \( \dot{y}(0) = 0 \). The continuity of the output-tissue time activity curve is not affected by the lack of differentiability of \( u(t) \) at the point \( t = 0 \), since \( y(t) \) is given by a convolution of \( u(t) \) with a continuous function of \( t \), see Chapter 2.

The graphs below, Figure 10, show the differences and similarities between an input function obtained from a real human study, and two others obtained from a simulation, described in detail in [14].

3. Overview

This dissertation gives a complete theoretical and numerical analysis of the Fluoro-Deoxy-Glucose tracer model utilized in PET for obtaining parametric images. The challenge of obtaining global images in PET comes from the poor signal to noise ratio of the acquired data. The numerical methods proposed here are robust to statistical noise, and provide reliable estimates. Data from human studies are used to generate individual parametric images.

Numerous methods have been proposed for generating global parametric images using dynamic PET data - we refer the reader to [7] for weighted integration methods, [55] for ridge regression methods, [11] for spectral analysis derived methods, [61] for spatial-temporal methods, [39] for wavelet analysis based methods, [20], [21], [22] for basis analysis, [54] for additive mixture models, [34] for cluster based methods, to name a few. Each method comes with advantages and disadvantages, and a concerted comparison has yet to be performed. We focus in this study on pixel-by-pixel image generation only.

Chapter 2 presents the theoretical analysis of the forward problem through the use of the nondimensional system, and examines the well-posedness of the inverse problem with respect to variations in parameters, and statistical noise. Simulations are used to confirm the analytical studies. Chapter 3 compares various numerical methods to solve the parameter estimation problem. An analysis for the convergence of the generalized linear least squares algorithm is given. Chapter 4 extends the methods to account for the spill-over and partial volume effects seen in practice. Chapter 5 introduces a nonlinear anisotropic diffusion filtering for the derived parametric images. Chapter 6 gives an account of the work presented and considers possible future studies.
Figure 3. Last time frame (frame 22, $t = 45$ minutes) of 31 slices of 2D sinograms of a human brain, reconstructed via filtered back projection.
Figure 4. Time-space dynamics for slice number 16, over 22 frames.
Figure 5. Signal-to-Noise Ratio (SNR) for slice number 16.

Figure 6. SNR values for Figure 5 for early time.
Figure 7. Clustered curves obtained from a human study, using a clustering algorithm described in [23].

Figure 8. Signal-to-Noise Ratio for slice 16, using clustered curves derived following [23], see Figure 7.

Figure 9. Compartment model structure for FDG metabolism.
Figure 10. Comparison of input functions $u(t)$, plasma time activity curves (PTAC), from real studies and simulations. Note, the scales are different depending on the model adopted.
CHAPTER 2

Theoretical Analysis of the Compartmental Model for FDG

We now develop a rigorous analysis of the kinetic model presented in Chapter 1. Historically, the deoxy-glucose compartment model was introduced in 1977 in the work of Sokoloff, [52], with the assumption that there is no hydrolysis of the phosphorylated tracer back into deoxy-glucose. This assumption, although shown in other clinical studies and simulations [18], [29] to be incorrect, has persisted in the development of numerical algorithms, widely used in clinical studies even today [41].

Here we pursue an analysis based on the solution to the initial value problem (1.3). We consider the well-posedness of the inverse problem and analyze the continuity of the inverse mapping leading to the kinetic parameter solution. Using perturbation analyses, we derive bounds on the difference between the solution assuming a nonzero hydrolysis of FDG-6-P and the solution assuming a null rate of hydrolysis. We verify that the inverse problem is well-posed with respect to variations in the kinetic parameters (however, ill-posed with respect to measurement noise).

1. Analysis of the Forward Problem

The system in (1.2) can be written in matrix form as

$$\dot{x} = Ax + b(t)$$

(2.1)

where

$$A = \begin{pmatrix} -(k_2 + k_3) & k_4 \\ k_3 & -k_4 \end{pmatrix}, \quad b(t) = \begin{pmatrix} k_1 u(t) \\ 0 \end{pmatrix}, \quad x(t) = \begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix}.$$  

(2.2)

This is a non-homogeneous linear differential equation, for which the solution is given in any standard book on differential equations.

**Theorem 1** (Perko, [42]) Let $\Phi(t)$ be any fundamental matrix solution of $\dot{x} = Ax$. Then the solution of the non-homogeneous linear system $\dot{x} = Ax + b(t)$ with initial condition $x(0) = x_0$ is unique and is given by

$$x(t) = \Phi(t)\Phi(0)^{-1}x_0 + \int_0^t \Phi(t)\Phi(\tau)^{-1}b(\tau)d\tau.$$  

(2.3)
Using (2.3), we can write the closed form solution of (1.2) as

\[ y(t) = y_1(t) + y_2(t) = \frac{k_1(k_3 + k_4 - \alpha_1)}{\alpha_2 - \alpha_1} \int_0^t u(s)e^{\alpha_1(t-s)}ds \\
- \frac{k_1(k_3 + k_4 - \alpha_2)}{\alpha_2 - \alpha_1} \int_0^t u(s)e^{\alpha_2(t-s)}ds, \tag{2.4} \]

where the roots

\[ \alpha_1 = \frac{(k_2 + k_3 + k_4) - \Delta}{2}, \quad \alpha_2 = \frac{(k_2 + k_3 + k_4) + \Delta}{2}, \text{ with} \]

\[ \Delta = \sqrt{(k_2 + k_3 + k_4)^2 - 4k_2k_4}, \]

are assumed to be non-equal and real, a reasonable assumption in FDG modeling based on standard physiological values for \( k_i \), see [46]. Equivalently, \( y(t) = \int_0^t f(t-s)u(s)ds \) where

\[ f(t) = \frac{k_1}{\alpha_2 - \alpha_1}[(k_3 + k_4 - \alpha_1)e^{-\alpha_1 t} - (k_3 + k_4 - \alpha_2)e^{-\alpha_2 t}] \tag{2.5} \]

\[ = \frac{k_1}{2\Delta}[(k_3 + k_4 - k_2 + \Delta)e^{-\alpha_1 t} - (k_3 + k_4 - k_2 - \Delta)e^{-\alpha_2 t}], \text{ with } k_i \geq 0. \tag{2.6} \]

Hence \( y(t) \) can be expressed as \( y(t) = (u * f)(t) \), where * is the convolution operation associated with the Laplace transform [62],

\[ y(t) = (u * f)(t) = \int_0^t f(t-s)u(s)ds = \int_0^t f(s)u(t-s)ds. \tag{2.7} \]

This definition assumes that \( u(t) \) and \( f(t) \) are both locally integrable functions on \([0, \infty)\) and that their Laplace integrals converge absolutely in some half-plane \( \text{Re}(s) > \rho \). Given \( y(t) \) and \( u(t) \), (2.7) defines an inverse problem for \( f(t) \), which depends on the kinetic parameters, \( f(t) = f(t; k) \).

**1.1. Non-dimensional Model Formulation.** Although clinical studies [29] have shown that the de-phosphorylation rate \( k_4 \) is nonzero, models like the Patlak analysis [41] assume \( k_4 = 0 \). To analyze the dependence of the exact solution (2.4) on the small de-phosphorylation rate \( k_4 \), and the difference between this solution and the one corresponding to \( k_4 = 0 \), we employ perturbation analysis techniques. This analysis assumes \( k_4 \), here denoted by \( \epsilon \), to be a small dimensionless parameter, and as such, the entire problem is expressed in dimensionless coordinates. The transformation of a problem into dimensionless form is usually not unique, and in most cases is dictated by the behavior one seeks to understand.

The model (1.2) uses values of input, output and parameters in certain units. The dimensions of the units are as follows

\[ \text{time } t \quad \text{min} \]
concentration $u$ counts (ml)$^{-1}$(min)$^{-1}$
concentration $y$ counts (ml tissue)$^{-1}$(min)$^{-1}$
parameter $k_1$ ml (ml tissue)$^{-1}$(min)$^{-1}$
parameters $k_2, k_3, k_4$ (min)$^{-1}$.

More specific units may be used depending on whether one counts the photon disintegrations in Becquerels (Bq) - 1 Bq is one disintegration per second, or milli-Curies (mCi) - 1 Curie is the quantity of radioactive material undergoing $3.7 \times 10^{10}$ disintegrations per second.

To transform this problem into its non-dimensionless version, we first let $k_1$ be an a-priori known value, $k_1 = K_1$. Note, all the initial values are zero, $y(0) = 0, \dot{y}(0) = 0$ and $u(0) = 0$, and for both input and output, unique maximum values are attained on the time interval of study. Hence, let $y^{\text{max}} = \max_{t>0}(y(t))$ and $u^{\text{max}} = \max_{t>0}(u(t))$. Define new nondimensional coordinates and parameters to be

$$
\tau := t \frac{K_1 u^{\text{max}}}{y^{\text{max}}},
$$
$$
U := \frac{u(t)}{u^{\text{max}}},
$$
$$
Y := \frac{y(t)}{y^{\text{max}}},
$$
$$
\bar{k}_1 := \frac{k_1}{K_1},
$$
$$
\bar{k}_2 := \frac{k_2 y^{\text{max}}}{K_1 u^{\text{max}}},
$$
$$
\bar{k}_3 := \frac{k_3 y^{\text{max}}}{K_1 u^{\text{max}}},
$$
$$
\bar{k}_4 := \frac{k_4 y^{\text{max}}}{K_1 u^{\text{max}}}.
$$

In the analysis of the forward problem, one knows precisely the value of $k_1$, and this in return could be used as $K_1$, which would give $\bar{k}_1 = 1$. However, for the inverse problem, this is exactly one of the parameters to be determined, hence it will be unknown. Rewriting the dimensionless problem one obtains

$$
\dot{Y}(\tau) + (\bar{k}_2 + \bar{k}_3 + \bar{k}_4) Y(\tau) + \bar{k}_2 \bar{k}_4 Y(\tau) = \bar{k}_1 \dot{U}(\tau) + \bar{k}_1 (\bar{k}_3 + \bar{k}_4) U(\tau),
$$
$$
Y(0) = 0, \quad \dot{Y}(0) = 0,
$$

which has exactly the same form as (1.2), and hence the exact solution is given by (2.4) with appropriate replacement of variables and parameters.

**Remark 1** Note, the sensitivity of the problem is not necessarily with respect to a small $k_4$ value, but rather a small value for $\bar{k}_4 = \frac{k_4 y^{\text{max}}}{K_1 u^{\text{max}}}$. In particular, the scale difference between the maximum value of the output to that of the input is relevant to the analysis of the problem. Since one needs to choose an a-priori value $K_1$, one can choose this to be on the order of $\frac{y^{\text{max}}}{u^{\text{max}}}$. Then $\bar{k}_4$ would be on the same order as the original $k_4$, and $\tau$ on the same order as $t$. 


1.2. Perturbation Analysis. In (2.8), denote \( \tilde{k}_4 \) by a small parameter \( \epsilon \), then collecting terms of order \( \epsilon \), rewrite it as

\[
\ddot{Y}(\tau) + (\tilde{k}_2 + \tilde{k}_3) \dot{Y}(\tau) = \epsilon \left[ \tilde{k}_1 U(\tau) - \dot{Y}(\tau) - \tilde{k}_2 Y(\tau) \right] + \tilde{k}_1 \dot{U}(\tau) + \tilde{k}_1 \tilde{k}_3 U(\tau). \tag{2.9}
\]

The following calculations are formal. Let \( Y(\tau, \epsilon) \) be an asymptotic expansion for \( Y(\tau) \), relative to the small parameter \( \epsilon \),

\[
Y(\tau, \epsilon) = Y_0(\tau) + \epsilon Y_1(\tau) + \epsilon^2 Y_2(\tau) + \epsilon^3 Y_3(\tau) + \ldots
\]

Substituting this in the initial value problem, one obtains

\[
\sum_{i=0}^{\infty} \epsilon^i \ddot{Y}_i(\tau) + (\tilde{k}_2 + \tilde{k}_3) \sum_{i=0}^{\infty} \epsilon^i \dot{Y}_i(\tau) = \epsilon \left[ \tilde{k}_1 U(\tau) - \sum_{i=0}^{\infty} \epsilon^i \dot{Y}_i(\tau) - \tilde{k}_2 \sum_{i=0}^{\infty} \epsilon^i Y_i(\tau) \right] + \tilde{k}_1 \dot{U}(\tau) + \tilde{k}_1 \tilde{k}_3 U(\tau),
\]

\[
\sum_{i=0}^{\infty} \epsilon^i Y_i(0) = 0, \text{ and } \sum_{i=0}^{\infty} \epsilon^i \dot{Y}_i(0) = 0. \tag{2.11}
\]

Equating powers of \( \epsilon \) yields

- **Order zero:**
  \[
  \ddot{Y}_0(\tau) + (\tilde{k}_2 + \tilde{k}_3) \dot{Y}_0(\tau) = \tilde{k}_1 \dot{U}(\tau) + \tilde{k}_1 \tilde{k}_3 U(\tau),
  Y_0(0) = 0, \dot{Y}_0(0) = 0;
  \]

- **Order one:**
  \[
  \ddot{Y}_1(\tau) + (\tilde{k}_2 + \tilde{k}_3) \dot{Y}_1(\tau) = \tilde{k}_1 U(\tau) - \left[ \dot{Y}_0(\tau) + \tilde{k}_2 Y_0(\tau) \right],
  Y_1(0) = 0, \dot{Y}_1(0) = 0;
  \]

- **Order \( i, i \geq 2 \):**
  \[
  \ddot{Y}_i(\tau) + (\tilde{k}_2 + \tilde{k}_3) \dot{Y}_i(\tau) = - \left[ \dot{Y}_{i-1}(\tau) + \tilde{k}_2 Y_{i-1}(\tau) \right],
  Y_i(0) = 0, \dot{Y}_i(0) = 0. \tag{2.12}
  \]

Note each initial value problem in (2.12) is a linear inhomogeneous equation, whose general solution can be formally expressed in terms of the inhomogeneity on the right hand side. Define

\[
  f_0(\tau) = \tilde{k}_1 \dot{U}(\tau) + \tilde{k}_1 \tilde{k}_3 U(\tau),
  f_1(\tau) = \tilde{k}_1 U(\tau) - \left[ \dot{Y}_0(\tau) + \tilde{k}_2 Y_0(\tau) \right],
  f_i(\tau) = - \left[ \dot{Y}_{i-1}(\tau) + \tilde{k}_2 Y_{i-1}(\tau) \right] \text{ for all } i \geq 2.
\]

The expression for each \( Y_i(\tau), i = 0, 1, \ldots \) is given by

\[
  Y_i(\tau) = \frac{1}{\tilde{k}_2 + \tilde{k}_3} \left[ \int_0^\tau f_i(s) ds - e^{-(\tilde{k}_2 + \tilde{k}_3)\tau} \int_0^\tau f_i(s) e^{(\tilde{k}_2 + \tilde{k}_3)s} ds \right]. \tag{2.13}
\]
Hence, the formal solution becomes

\[
Y(\tau, \epsilon) = \sum_{i=0}^{\infty} \epsilon^i Y_i(\tau)
\]

\[
= \frac{1}{k_2 + k_3} \left[ \sum_{i=0}^{\infty} \epsilon^i \int_0^\tau f_i(s) ds - \epsilon^{-(k_2+k_3)\tau} \sum_{i=0}^{\infty} \epsilon^i \int_0^\tau f_i(s) e^{(k_2+k_3)s} ds \right].
\]

Integrating by parts, given \(U(0) = 0\), and summing respectively, using the definition of the convolution operator, *, given in (2.7), one obtains

\[
Y(\tau, \epsilon) = \frac{\bar{k}_1 \bar{k}_3}{k_2 + k_3} \int_0^\tau U(s) ds + \frac{\bar{k}_1 \bar{k}_2}{k_2 + k_3} U(\tau) * e^{-(k_2+k_3)\tau}
\]

\[
- \frac{\epsilon}{k_2 + k_3} \left[ \bar{k}_2 \int_0^\tau Y_0(s) ds + \bar{k}_3 Y_0(\tau) * e^{-(k_2+k_3)\tau} \right]
\]

\[
+ \frac{\epsilon}{k_2 + k_3} \left[ \bar{k}_1 \int_0^\tau U(s) ds - \bar{k}_1 U(\tau) * e^{-(k_2+k_3)\tau} \right]
\]

\[
- \frac{1}{k_2 + k_3} \left[ \bar{k}_2 \sum_{i=2}^{\infty} \epsilon^i \int_0^\tau Y_{i-1}(s) ds + \bar{k}_3 \sum_{i=2}^{\infty} \epsilon^i Y_{i-1}(\tau) * e^{-(k_2+k_3)\tau} \right].
\]

Note the unperturbed problem,

\[
\tilde{X}(\tau) + (\bar{k}_2 + \bar{k}_3)\tilde{X}(\tau) = \bar{k}_1 \tilde{U}(\tau) + \bar{k}_1 \bar{k}_3 U(\tau),
\]

has a closed form solution given by

\[
X(\tau) = \frac{\bar{k}_1 \bar{k}_3}{k_2 + k_3} \int_0^\tau U(s) ds + \frac{\bar{k}_1 \bar{k}_2}{k_2 + k_3} U(\tau) * e^{-(k_2+k_3)\tau}.
\]

(2.14)

**Lemma 1** *In the limit, as \(\bar{k}_4 \to 0\), \(Y(\tau) \to X(\tau)\).*

**Proof:**

Taking the limit as \(\bar{k}_4 \to 0\) in (2.4) for \(Y(\tau)\) we have

\[
\lim_{k_4 \to 0} Y(\tau) = \bar{k}_1 \lim_{\bar{k}_4 \to 0} \frac{\bar{\Delta} + \bar{k}_3 + \bar{k}_4 - \bar{k}_2}{2\bar{\Delta}} U(\tau) * e^{-\bar{\alpha}_1(\tau)} + \frac{\bar{\Delta} - (\bar{k}_3 + \bar{k}_4) + \bar{k}_2}{2\bar{\Delta}} U(\tau) * e^{-\bar{\alpha}_2(\tau)}
\]

\[
= \bar{k}_1 \lim_{\bar{k}_4 \to 0} \left[ \frac{1}{2} + \frac{\bar{k}_3 + \bar{k}_4 - \bar{k}_2}{2\sqrt{(k_2 + k_3 + k_4)^2 - 4k_2k_4}} \right] \int_0^\tau U(s) e^{-\bar{\alpha}_1(\tau-s)} ds
\]

\[
+ \bar{k}_1 \lim_{\bar{k}_4 \to 0} \left[ \frac{1}{2} - \frac{\bar{k}_3 + \bar{k}_4 - \bar{k}_2}{2\sqrt{(k_2 + k_3 + k_4)^2 - 4k_2k_4}} \right] \int_0^\tau U(s) e^{-\bar{\alpha}_2(\tau-s)} ds
\]

\[
= \frac{\bar{k}_1 \bar{k}_3}{k_2 + k_3} \int_0^\tau U(s) ds + \frac{\bar{k}_1 \bar{k}_2}{k_2 + k_3} \int_0^\tau U(s) e^{-(k_2+k_3)(\tau-s)} ds
\]

which is exactly \(X(\tau)\), ie \(\lim_{\bar{k}_4 \to 0} Y(\tau) = X(\tau) = Y_0(\tau)\).
Remark 2 Note, while the exact solution of $Y(\tau)$ is a decaying exponential, the behaviour of $X(\tau)$ depends strongly on the form of $U(\tau)$, and in particular, on the behaviour of the integral $\int_0^\tau U(s)ds$. For example, if $\int_0^\tau U(s)ds$ is unbounded as $\tau \to \infty$, then $X(\tau)$ will be unbounded, while $Y(\tau)$ will remain bounded as $\tau \to \infty$. We conclude that the uniform convergence of $Y(\tau)$ to $X(\tau)$ as $\tilde{k}_1 \to 0$ depends on the form of the input function. In case $Y(\tau)$ converges to $X(\tau)$ non-uniformly, one has a singular perturbation problem, while for $Y(\tau)$ uniformly convergent to $X(\tau)$ as $\tau \to \infty$, one has a regular perturbation problem [38].

One would like to express each $Y_i(\tau)$ only in terms of $\tau$ and $U(\tau)$. This would facilitate understanding the behaviour of $Y_i(\tau)$ as $i \to \infty$. Let $E(\tau) = e^{-(k_2+k_3)\tau}$. Then

$$Y_i(\tau) = \frac{1}{k_2 + k_3} \left[ \int_0^\tau f_i(s)ds - (f_i * E)(\tau) \right].$$

Although $Y_0(\tau)$ can be computed relatively easy, as

$$Y_0(\tau) = \frac{\tilde{k}_1}{k_2 + k_3} \left[ \tilde{k}_3 \int_0^\tau U(s)ds + \tilde{k}_2 (U * E)(\tau) \right]$$

(2.15)

the expression for a general term $Y_n(\tau)$ becomes soon complicated by the fact that without an analytic form for $U(\tau)$ one can not provide a closed form solution for such a general term. Consequently we proceed by introducing some operator notation,

$$(IU)(\tau) = \int_0^\tau U(s)ds.$$

Then the square of this operator $I$ is defined as $(I^2U)(\tau) = I(IU)(\tau) = \int_0^\tau \int_0^\tau U(s)dsd\tau_1$, and by induction one obtains $(I^nU)(\tau) = \int_0^\tau \int_{t_1}^\tau \cdots \int_{t_{n-1}}^\tau U(s)dsd\tau_{n-1} \cdots d\tau_1$. A careful computation of $Y_1(\tau)$, using the fact that $((IU) * E)(\tau) = I(U * E)(\tau)$, gives

$$Y_1(\tau) = \frac{\tilde{k}_1}{k_2 + k_3} [IU - U * E] - \frac{\tilde{k}_1(k_2^2 + k_3^2)}{(k_2 + k_3)^2} (IU * E) - \frac{\tilde{k}_1k_2k_3}{(k_2 + k_3)^2} [I^2U + U * E^2]$$

where $(U * E^2)(\tau) = (U * E * E)(\tau) = \int_0^\tau \int_{t_1}^\tau U(s)ds e^{-(k_2+k_3)(\tau-s)}dsd\tau_1$. The general term of the asymptotic expansion given in (2.10) has the form

$$Y_n(\tau) = \sum_{i=0}^{n+1} \sum_{j=0}^{n+1-i} a_{ij} I^i U * E^j$$

(2.16)

for certain parameters $a_{ij}$, where $U * E^j = I^0U * E^j$, and $I^iU = I^iU * E^0$.

Lemma 2 The regular perturbation expansion given by (2.10) diverges as $\tau \to \infty$.

Proof:

Given $U(\tau) \geq 0$ for all $\tau$, as a consequence of the Riemann-Lebesgue Lemma, all terms of the form $I^iU * E^j$, with $j \geq 1$ converge to 0 as $\tau \to \infty$. Then

$$\lim_{\tau \to \infty} Y_n(\tau) = \lim_{\tau \to \infty} \sum_{i=1}^{n+1} a_{i0} I^iU$$
and since $U(\tau) \geq 0$ for all $\tau$, $\lim_{\tau \to \infty} I^i U \neq 0$, and we have that the limit
\[
\lim_{\tau \to \infty} \sum_{i=1}^{n+1} a_i I^i U
\]
diverges, ie $\lim_{\tau \to \infty} Y_n(\tau)$ diverges.

The problem (2.8), however, is usually solved on a restricted domain, where it is a regular perturbation problem. To obtain bounds on the approximate solution $X(\tau) = Y_0(\tau)$, one first looks at how well this solution satisfies an approximate initial value problem, ie
\[
\dot{Y}_0(\tau) + (k_2 + \tilde{k}_3 + \epsilon)Y_0(\tau) + \epsilon \tilde{k}_2 Y_0(\tau) = \tilde{k}_1 \dot{U}(\tau) + \tilde{k}_1 (\tilde{k}_3 + \epsilon) U(\tau) - \rho_0(\tau, \epsilon),
\]
\[
Y_0(0, \epsilon) = 0 - \gamma(\epsilon),
\]
\[
\dot{Y}_0(0, \epsilon) = 0 - \beta(\epsilon)
\] (2.17)

where $\gamma = \gamma(\epsilon)$, $\beta = \beta(\epsilon)$, $\rho_0 = \rho_0(\epsilon)$ are known residuals, generally assumed to be small for $\tau$ and $\epsilon$ such that $0 \leq \epsilon \tau \leq T$ and $0 \leq \epsilon \leq \epsilon_0$, for some given positive $T$ and $\epsilon_0$. Given the approximate solution $Y_0(\tau)$ in (2.15), the residuals are found to be
\[
\gamma(\epsilon) = \beta(\epsilon) = 0,
\]
\[
\rho_0(\tau, \epsilon) = \epsilon (\tilde{k}_1 U(\tau) - \dot{Y}_0(\tau) - \tilde{k}_2 Y_0(\tau))
\]
\[
= \epsilon \left[ \tilde{k}_1 U(\tau) - \tilde{k}_1 U(\tau) + \tilde{k}_1 \tilde{k}_2 (U \ast E)(\tau) - \tilde{k}_2 Y_0(\tau) \right]
\]
\[
= \epsilon \frac{\tilde{k}_1 \tilde{k}_2 \tilde{k}_3}{k_2 + k_3} [U \ast E - IU](\tau). \tag{2.18}
\]

Explicitly, the residual $\rho_0(\tau, \epsilon)$ is given by
\[
\rho_0(\tau, \epsilon) = \epsilon \frac{\tilde{k}_1 \tilde{k}_2 \tilde{k}_3}{k_2 + k_3} \int_0^\tau U(s) e^{-(k_2 + k_3)(\tau - s)} - U(s) \, ds \leq 0, \text{ for all } \tau,
\]
since $U(\tau)$ is a nonnegative function, $\tilde{k}_2 + \tilde{k}_3 > 0$ and $\tau \geq s$ for all $s$ and $\epsilon$ positive. Given this residual, the remainder $R_0(\tau, \epsilon)$ is defined as the difference between the exact solution and the approximation $Y_0(\tau)$,
\[
R_0(\tau, \epsilon) := Y(\tau) - Y_0(\tau). \tag{2.19}
\]

Using the residual thus defined, one finds $R_0(\tau, \epsilon)$ satisfies
\[
\dot{R}_0(\tau, \epsilon) + (k_2 + \tilde{k}_3) \dot{R}_0(\tau, \epsilon) = \rho_0(\tau, \epsilon) - \epsilon (\tilde{R}_0(\tau, \epsilon) + \tilde{k}_2 R_0(\tau, \epsilon)) \text{ for all } \tau > 0,
\]
\[
R_0(0, \epsilon) = \gamma(\epsilon) = 0,
\]
\[
\dot{R}_0(0, \epsilon) = \beta(\epsilon) = 0. \tag{2.20}
\]

**Lemma 3** Let $m, M$ be such that $U(s) \leq M e^{m s}$. Then, given the problem (2.8), with a solution of the form given in (2.4), and the approximation $Y_0(\tau)$ satisfying (2.17), for all $\tau$
and $\epsilon$ such that $0 \leq \tau \leq \frac{T}{\tau}$, $0 \leq \epsilon \leq \epsilon_0$, for fixed positive constants $T$, and $\epsilon_0$, there exist positive constants $B_{T, \epsilon}$ and $\epsilon_1$ such that

$$|Y(\tau) - Y_0(\tau)| \leq \frac{\epsilon k_1 k_2 \bar{k}_3}{k_2 + \bar{k}_3} MB_{T, \epsilon}, \quad (2.21)$$

for $0 < \epsilon \leq \epsilon_1$.

**Proof:** Using the equivalency of the initial value problem (2.20) with a Volterra integral equation of the second kind [62], one obtains

$$R_0(\tau, \epsilon) = \int_0^\tau (\tau - s) \rho_0(s) ds - \int_0^\tau R_0(s, \epsilon) [\epsilon \bar{k}_2 (\tau - s) + (\bar{k}_2 + \bar{k}_3 + \epsilon)] ds.$$

Note, $\int_0^\tau (\tau - s) \rho_0(s) ds \leq 0$, for all $\tau$. Consequently, we let

$$c(\tau) = -\int_0^\tau (\tau - s) \rho_0(s) ds \geq 0,$$

$$b_\tau(s, \epsilon) = \epsilon \bar{k}_2 (\tau - s) + (\bar{k}_2 + \bar{k}_3 + \epsilon) \geq 0,$$

for all $s \leq \tau$.

Applying the triangle inequality one obtains

$$|R_0(\tau, \epsilon)| \leq \int_0^\tau (\tau - s) \rho_0(s) ds + \int_0^\tau b_\tau(s, \epsilon) R_0(s, \epsilon) ds.$$

Therefore, since $c(\tau) \geq 0$, and $b_\tau(s, \epsilon) \geq 0$, the Gronwall’s inequality can be applied to give

$$|R_0(\tau, \epsilon)| \leq \int_0^\tau b_\tau(s, \epsilon) c(s) e^{\int_s^\tau b_\tau(t, \epsilon) dt} ds + c(\tau). \quad (2.22)$$

Let $f(s) = \int_s^\tau b_\tau(t, \epsilon) dt$. Then $f'(s) = -b_\tau(s, \epsilon)$, and an integration by parts formula for the integral in (2.22) gives

$$|R_0(\tau, \epsilon)| \leq \int_0^\tau f'(s) c(s) e^{\int_s^\tau f'(s) ds} + c(\tau)$$

$$= \int_0^\tau -f'(s) c(s) e^{\int_s^\tau f'(s) ds} + c(\tau)$$

$$= \int_0^\tau c'(s) e^{\int_s^\tau f'(s) ds} + c(\tau)$$

$$= \int_0^\tau c'(s) \left( \int_0^s \rho_0(t) dt \right) ds$$

since $c'(s) = -\int_0^s \rho_0(t) dt$. The explicit form of $f(s)$ is given by

$$f(s) = \int_s^\tau b_\tau(t, \epsilon) dt$$

$$= -\frac{\epsilon k_2 (\tau^2 - s^2)}{2} + \epsilon k_2 \tau (\tau - s) + (k_2 + k_3 + \epsilon) (\tau - s)$$

$$\leq \tau (\tau - s)(\epsilon k_2 \tau + (k_2 + k_3 + \epsilon)) = (\tau - s) b_\tau(0, \epsilon).$$
Therefore, we have $e^{f(s)} \leq e^{(\tau - s)b_r(0, \epsilon)} \leq e^{r b_r(0, \epsilon)}$, which implies

$$|R_0(\tau, \epsilon)| \leq -e^{r b_r(0, \epsilon)} \int_0^\tau \int_0^s \rho_0(t) dt ds.$$ 

Hence, we are now interested in finding a bound for $\int_0^\tau \int_0^s \rho_0(t) dt ds$. We are given that $U(s)$ is exponentially bounded, $U(s) \leq M e^{m s}$ for some known positive constants $M, m$. Note, for tracer dynamics, this assumption is satisfied by all input functions, since an input tracer function has a unique maximum achieved at early times, after which it follows an exponential decay. We consider first a bound on $-\rho_0(t, \epsilon) \geq 0$,

\[
-\rho_0(t, \epsilon) = -\epsilon \frac{k_1 k_2 k_3}{k_2 + k_3} \int_0^t U(s) \left( e^{-(k_2 + k_3)(t-s)} - 1 \right) ds \\
\leq \epsilon M \frac{k_1 k_2 k_3}{k_2 + k_3} \int_0^t e^{m s} \left( 1 - e^{-(k_2 + k_3)(t-s)} \right) ds \\
= \epsilon M \frac{k_1 k_2 k_3}{k_2 + k_3} \left[ e^{m t} - 1 - \frac{e^{m t} - e^{-(k_2 + k_3)t}}{m + k_2 + k_3} \right].
\]

Following this result, one can easily integrate this twice to find

$$|R_0(\tau, \epsilon)| \leq M e^{r b_r(0, \epsilon)} \frac{k_1 k_2 k_3}{k_2 + k_3} P(\tau),$$

where

\[
P(\tau) = \frac{e^{m \tau} - 1}{m^3} - \frac{\tau}{m^2} - \frac{\tau^2}{2m} - \frac{e^{m \tau} - 1}{m^2 (m + k_2 + k_3)} + \frac{\tau}{m(m + k_2 + k_3)} + \frac{e^{-(k_2 + k_3)\tau} - 1}{(k_2 + k_3)^2 (m + k_2 + k_3)} + \frac{\tau}{(k_2 + k_3)(m + k_2 + k_3)},
\]

and $\tau b_r(0, \epsilon) = \epsilon (k_2 + k_3) (\tau + \frac{k_2 + k_3}{\epsilon} + 1)$. Note this is not a uniform bound, although one can derive such a bound using the fact that the domain under consideration is bounded, $\epsilon \tau < T$, for some $\epsilon \leq \epsilon_0$, $T$ and $\epsilon_0$ given. For a uniform bound, define $B_{T, \epsilon} = \max_{0 \leq \tau \leq \frac{T}{\epsilon}} \{ e^{\tau b_r(0, \epsilon)} P(\tau) \}$ to obtain

$$|R_0(\tau, \epsilon)| \leq e^{\frac{k_1 k_2 k_3}{k_2 + k_3} B_{T, \epsilon}} M.$$ 

One chooses $\epsilon_1$ such that $\epsilon \leq \epsilon_1$ and $B_{T, \epsilon}$ is well defined, given $\tau \leq \frac{T}{\epsilon}$.

\section{Simulations.}

The following numerical studies support the analytical bounds derived in Section 1.2.

Note, the difference $|Y(\tau) - Y_0(\tau)|$ depends on $B_{T, \epsilon}$, and hence, on the length of the time interval $T$ over which tracer uptake is measured. As $T$ increases, $|Y(\tau) - Y_0(\tau)|$ also increases, as seen in Figure 12. In practice, the length of the time interval $T$ over which measurements are taken should be chosen such that an effective steady state is reached. The literature shows a varying time interval of imaging of the FDG tracer from 30 minutes to 120, with a preference for 60 minutes. The half-life of FDG is two hours, so ideally, one
would always choose 120 minutes of imaging time. However, the choice of $T$ will also depend on the underlying model parameters $k_1$ through $k_4$. To better understand the variation of the parameter vector $k$ with respect to the length of time of imaging, we solved for the solution vector $k$ for various end-times, using the image-derived input function shown in Figures 11 and 12, and assuming no noise in our data.

Using the filtered least squares (FLS) algorithm (to be introduced later in Chapter 3) and assuming $k_4 \neq 0$, we note $k_3$ converges to its best approximate value near time $T = 12$ minutes, Figure 14, while the other constants $k_1$ and $k_2$ are also best approximated as $T$ is smaller, Figure 13. These results confirm those in [18] where it is observed that an effective steady state is reached for $T \sim 10$ minutes. For the case in which FLS solves for the parameter vector assuming $k_4 = 0$, we observe a different behaviour of convergence, in particular the values for $k_1$, $k_2$ and $k_3$ start out large, and converge very quickly as $T$ increases, around $T = 5$ minutes.

**Remark 3** We conclude that for short time intervals $T$ over which measurements are made, one can closely approximate $Y(\tau)$ by $Y_0(\tau)$. The solution to the inverse problem assuming $k_4 = 0$ will not be far from the solution given by the same problem under the assumption $k_4 \neq 0$. However, as early data usually are highly contaminated by noise, these results will not hold for a real study, where the noise dominates the true values for $Y(\tau)$ for $\tau$ small.

2. Analysis of the Inverse Problem

We have seen in the previous section, equation (2.7), that the solution of the described kinetic problem is an inverse problem for $f(t)$. In this case we are interested in the parameters that determine the function $f(t)$, assuming knowledge of $u(t), y(t)$, the functional form of $f(t)$, (2.6) and conditions under which a unique solution can be found.

2.1. Sensitivity Analysis. For ease of notation in the following discussion we suppress the dependence on the parameter vector $k$ in $y(t) = y(k; t)$ and $f(t) = f(k; t)$.

**Theorem 2** (Davis, [12]) Suppose

1. $y(t)$ is a continuous function in $(0, T)$, where $0 \leq t \leq T$, with $y(0) = 0$, and a continuous derivative $\frac{d}{dt}y(t)$

2. $u(t, s) = u(t - s)$ is continuous for $0 \leq s \leq t \leq T$, and has a finite derivative $\frac{d}{dt}u(t - s)$ with regularly distributed discontinuities,

3. $u(0) \neq 0$.

Then equation (2.7) has one and only one continuous solution $f(t)$.
Note, however, that in the FDG model, the final condition of Theorem 2 is violated, and \( u(0) = 0 \). Suppose \( y(t) \) and \( u(t) \) satisfy conditions (1) and (2) above, then differentiating (2.7) we obtain
\[
\frac{d}{dt}y(t) = f(t)u(0) + \int_0^t f(s) \frac{d}{dt}u(t - s)ds
\] (2.23)
which, with \( u(0) = 0 \), simplifies to become
\[
\frac{d}{dt}y(t) = \int_0^t f(s) \frac{d}{dt}u(t - s)ds.
\] (2.24)

This is again a Volterra integral of the first kind. The restrictions of Theorem 2 now hold for this equation, namely \( \frac{d}{dt}y(t) \) is continuous, with \( \frac{d}{dt}y(t)|_{t=0} = 0 \), and \( \frac{d^2}{dt^2}y(t) \) continuous, and \( \frac{d}{dt}u(t - s) \) is also continuous, with a finite derivative, and \( \frac{d}{dt}u(t)|_{t=0} = 0 \). These conditions are a consequence of the problem posed in (1.3). Hence, the existence and uniqueness of the solution \( f(t) \) are guaranteed by Theorem 2.

We now consider if the problem (2.7) is well-posed.

**Definition 1** (Hadamard, [24]) Given a mapping \( A \), the inverse problem of solving \( A(f) = d \) for \( f \), given \( d \) is called well-posed if:
1. a solution exists for any data \( d \) in the data space,
2. the solution is unique in image space, and
3. the inverse mapping \( d \to f \) is continuous.

In our case, the mapping \( A(f) = \int_0^t f(s)u(t - s)ds \), with \( A(f) = y \). The last requirement in condition 3, continuous dependence of the solution on the data, is a necessary but not sufficient condition for the stability of the solution. Note, the first two requirements of existence and uniqueness of solutions are guaranteed for our problem by Theorem 2. If the inverse mapping \( y \to f \) is continuous, then the problem of solving \( y(t) = \int_0^t f(s)u(t - s)ds \) for \( f \), given its functional form (2.6), is well-posed.

**Lemma 4** Let \( y_0(t) = \int_0^t f_0(s)u(t - s)ds \), \( f_0(t) \) given by (2.6) with \( k_4 = 0 \), \( t \) finite, \( t \leq T \), where \( T \) is a positive constant, and \( k_2, k_3 \) fixed. Then, given \( \epsilon_1 > 0 \), there exists a \( \delta k_1 \geq 0 \) such that \( |k_1 - k_1| < \delta k_1 \) implies
\[
\delta y_1 = |y(\tilde{k}_1, t) - y(k_1, t)| < \epsilon_1.
\] (2.25)

**Proof:** Given that \( k_2, k_3 \) are held fixed, consider \( y_0(t) = y_0(k_1, t) \). Without loss of generality, let \( \tilde{k}_1 = k_1 + \delta k_1 \). Then, the Taylor series gives
\[
y(k_1 + \delta k_1, t) = y(k_1, t) + \delta k_1 y_{k_1}(k_1, t) + O(\delta k_1^2),
\]
where \( y_{k_1}(k_1, t) = \frac{\partial}{\partial k_1}(k_1, t) \). Given \( \epsilon_1 \), we need to show there exists a \( \delta k_1 \) which implies
\[
\delta y_1 = |y(k_1 + \delta k_1, t) - y(k_1, t)| \leq \epsilon_1.
\]
Since \( \delta y_1 \leq |\delta k_1||y_{k_1}(k, t)| \), it suffices to choose \( |\delta k_1| < \frac{\epsilon_1}{|y_{k_1}(k, t)|} \); whenever this bound is well defined. We have that

\[
y_0(t) = y_0(k, t) = \int_0^t f_0(k, s)u(t - s) \, ds = f_0(k, t) * u(t),
\]

where \( f_0(k, t) = \frac{k_1}{k_2 + k_3} (k_3 + k_2 e^{-(k_2 + k_3)t}) \), and \( k = [k_1, k_2, k_3] \). Let \( g_1(k, t) = \frac{\partial}{\partial k_1} f_0(k, t) \).

Then, it follows that

\[
\frac{\partial}{\partial k_1} y_0(k, t) = g_1(k, t) * u(t), \quad \text{and} \quad g_1(k, t) = \frac{1}{k_2 + k_3} (k_3 + k_2 e^{-(k_2 + k_3)t}) > 0.
\]

The input function \( u(t) \) is nonnegative and bounded for all \( t \geq 0, |u(t)| \leq M \). Then

\[
\left| \frac{\partial}{\partial k_1} y_0(k, t) \right| = \int_0^t g_1(k, s)u(t - s) \, ds \\
\leq M \int_0^t g_1(k, s) \, ds \\
= M \frac{1}{k_2 + k_3} \int_0^t k_3 + k_2 e^{-(k_2 + k_3)s} \, ds = MN_1(t)
\]

where

\[ N_1(t) = \frac{1}{k_2 + k_3} \left( k_3 t + k_2 \frac{1 - e^{-(k_2 + k_3)t}}{k_2 + k_3} \right). \]

Then \( \delta y_1 \leq \delta k_1 MN_1(t) \), and depending on \( t \), one can find \( \delta k_1 < \frac{\epsilon_1}{MN_1(t)} \). Note, a uniform bound can not be found since \( N_1(t) \) is unbounded as \( t \to \infty \), but for any given finite \( t \leq T \), the bound \( \frac{\epsilon_1}{MN_1(t)} \) is well defined, and hence \( \delta k_1 > 0 \) is well defined.

**Lemma 5** Let the assumptions in Lemma 4 be given, and assume \( k_1, k_3 \) fixed. Then, given \( \epsilon_2 > 0 \), there exists a \( \delta k_2 \geq 0 \) such that \( |\delta k_2| < \delta k_2 \) implies

\[
\delta y_2 = |y(\tilde{k}, k, t) - y(k, t)| < \epsilon_2.
\]

**Proof:** Following the outline of the proof given in Lemma 4, we seek to bound

\[
\frac{\partial}{\partial k_2} y_0(k, t) = \frac{\partial}{\partial k_2} f_0(k, t) * u(t).
\]

Let

\[
g_2(k, t) = \frac{\partial}{\partial k_2} f_0(k, t) = -\frac{k_1}{(k_2 + k_3)^2} \left[ k_3(1 - e^{-(k_2 + k_3)t}) + t(k_2 + k_3)e^{-(k_2 + k_3)t} \right] \leq 0.
\]
Then
\[
\left| \frac{\partial}{\partial k_2} y_0(k, t) \right| = - \int_0^t g_2(k, s) u(t - s) ds
\]
\[
\leq M \frac{k_1}{(k_2 + k_3)^2} \int_0^t k_3(1 - e^{-(k_2+k_3)s}) + s(k_2 + k_3)e^{-(k_2+k_3)s} ds
\]
\[
= MN_2(t),
\]
where
\[
N_2(t) = \frac{k_1}{(k_2 + k_3)^3} \left[ (k_3(t(k_2 + k_3) - k_3 + 1) - e^{-(k_2+k_3)t} (t(k_2 + k_3) - k_3 + 1) \right] \geq 0.
\]

Then, as before, one can find \( \delta k_2 < \frac{\epsilon_2}{MN_2(t)} \), where \( N_2(t) \) is well defined for any given finite \( t \leq T \).

**Lemma 6** Let the assumptions in Lemma 4 be given, and assume \( k_1, k_2 \) fixed. Then, given \( \epsilon_3 > 0 \), there exists a \( \delta k_3 \geq 0 \) such that \( |k_3 - k_3| < \delta k_3 \) implies
\[
\delta y_3 = |y(k_3, t) - y(k_3, t)| < \epsilon_3.
\]

**Proof:** Again, we follow the outline of the proof given in Lemma 4, and find a bound for
\[
\left| \frac{\partial}{\partial k_3} y_0(k, t) \right| = \frac{\partial}{\partial k_3} f_0(k, t) \ast u(t).
\]
Let
\[
g_3(k, t) = \frac{\partial}{\partial k_3} f_0(k, t)
\]
\[
= \frac{k_1k_2}{(k_2 + k_3)^2} \left[ 1 - \frac{1 + t(k_2 + k_3)}{e^{(k_2+k_3)t}} \right] \geq 0.
\]
Then
\[
\left| \frac{\partial}{\partial k_3} y_0(k, t) \right| = \int_0^t g_3(k, s) u(t - s) ds
\]
\[
\leq M \frac{k_1k_2}{(k_2 + k_3)^2} \int_0^t 1 - e^{-(k_2+k_3)s} - s(k_2 + k_3)e^{-(k_2+k_3)s} ds
\]
\[
= MN_3(t),
\]
where
\[
N_3(t) = \frac{k_1k_2}{(k_2 + k_3)^3} \left[ t(k_2 + k_3)(1 + e^{-(k_2+k_3)t}) - 2(1 - e^{-(k_2+k_3)t}) \right] \geq 0.
\]
Hence, there exists \( \delta k_3 < \frac{\epsilon_3}{MN_3(t)} \), where \( N_3(t) \) is well defined for any given finite \( t \leq T \).

**Theorem 3** The problem of solving equation \( y(t) = \int_0^t f(s) u(t - s) ds \) for \( f(t) \), given its functional form (2.6), and assuming \( t \) finite, \( 0 \leq t \leq T \), and \( k_4 = 0 \) is well-posed with respect to variations in the parameter vector \( k \).
**Proof:** The first two conditions follow from Theorem 2. From the Lemmas 4, 5, and 6, we conclude that the mapping \( y \rightarrow f \) is continuous, hence small perturbations in \( y(t) \) lead to small perturbations in the parameter vector \( k \), and the given inverse problem is well posed.

Note, for finite dimensional problems over \( \mathbb{R}^n \), it is well known that for linear operators, the inverse problem is always well posed [59]. The algebraic derivation presented here is mostly done to better understand the behaviour of \( y(k_i, t) \) as \( k_i \) vary.

We now consider a similar analysis for the case \( k_4 \neq 0 \).

**Lemma 7** Let \( y(t) = \int_0^t f(s)u(t-s)ds \), \( f(t) \) given by (2.6), and \( k_2, k_3, k_4 \) fixed positive constants. Then, given \( \epsilon_1 > 0 \), there exists a \( \delta k_1 \geq 0 \) such that \(|\tilde{k}_1 - k_1| < \delta k_1 \) implies

\[
\delta y_1 = |y(\tilde{k}_1, t) - y(k_1, t)| < \epsilon_1.
\]  

**Proof:** The outline follows exactly that given in Lemma 4, only here we do not assume \( k_4 = 0 \), and \( t \) is allowed to be infinite. We still seek to find a bound on

\[
y_{k_1}(k, t) = \frac{\partial}{\partial k_1} y(t) = \frac{\partial f}{\partial k_1}(t) * u(t) = g_1(k, t) * u(t)
\]  

where

\[
g_1(k, t) = \frac{(k_3 + k_4 - k_2 + \Delta)e^{-\alpha_1 t} - (k_3 + k_4 - k_2 - \Delta)e^{-\alpha_2 t}}{2\Delta}
\]  

since, from (2.6), \( \alpha_i \) and \( \Delta \) are independent of \( k_1 \). Note,

\[
\lim_{t \to \infty} g_1(k, t) = \lim_{t \to \infty} \frac{(k_3 + k_4 - k_2 + \Delta)e^{-\alpha_1 t} - (k_3 + k_4 - k_2 - \Delta)e^{-\alpha_2 t}}{2\Delta} = 0,
\]

\[
\lim_{t \to 0} g_1(k, t) = \lim_{t \to 0} \frac{(k_3 + k_4 - k_2 + \Delta)e^{-\alpha_1 t} - (k_3 + k_4 - k_2 - \Delta)e^{-\alpha_2 t}}{2\Delta} = 1,
\]

hence, \( g_1(k, t) \) monotonically decreases as \( t \geq 0 \) increases, and \( g_1(k, t) > 0 \). Given that \( u(t) \leq M, M > 0 \), we have

\[
|y_{k_1}(k, t)| = \left| \int_0^t g_1(k, s)u(t-s)ds \right| \\
\leq M \int_0^t g_1(k, s)ds \\
= M \int_0^t \frac{(k_3 + k_4 - k_2 + \Delta)e^{-\alpha_1 s} - (k_3 + k_4 - k_2 - \Delta)e^{-\alpha_2 s}}{2\Delta}ds \\
= MN_1(k, t)
\]
where
\[ N_1(k,t) = \frac{1}{2\Delta} \left[ \frac{k_3 + k_4 - k_2 + \Delta}{\alpha_1} \left( 1 - e^{-\alpha_1 t} \right) - \frac{k_3 + k_4 - k_2 - \Delta}{\alpha_2} \left( 1 - e^{-\alpha_2 t} \right) \right] \]
\[ \geq 0 \quad \forall t \geq 0. \]

Hence the perturbation in \( y \) with respect to \( k_1 \) is bounded by
\[ \delta y_1 \leq \delta k_1 M N_1(k,t). \]

Given \( \epsilon_1 \), choose \( \delta k_1 < \frac{\epsilon_1}{M N_1(k,t)} \).

**Remark 4** Note, the choice of \( \delta k_1 \) depends on time, assuming the solution vector \( k \) is constant. We can obtain a uniform bound by introducing the maximum norm for a given function, \( ||F(t)|| = \max_{t \geq 0} |F(t)| \). In this norm,
\[ ||\delta y_1|| \leq \delta k_1 M \max_{t \geq 0} N_1(k,t). \]

Note \( N_1(k,t) \) increases with time, and hence the maximum of the function \( N_1(k,t) \) is found by taking the limit
\[ \lim_{t \to \infty} N_1(k,t) = \lim_{t \to \infty} \frac{1}{2\Delta} \left[ \frac{k_3 + k_4 - k_2 + \Delta}{\alpha_1} \left( 1 - e^{-\alpha_1 t} \right) - \frac{k_3 + k_4 - k_2 - \Delta}{\alpha_2} \left( 1 - e^{-\alpha_2 t} \right) \right] \]
\[ = \frac{k_3 + k_4}{\alpha_1 \alpha_2} := C_1. \]

In terms of the vector \( k \) we have
\[ ||\delta y_1|| \leq (\delta k_1) M C_1. \quad (2.29) \]

Then one can choose \( \delta k_1 < \frac{\epsilon_1}{M C_1} \), and here \( \delta k_1 \) is independent of time.

**Lemma 8** Let \( y(t) = \int_0^t f(s) u(t - s) ds \), \( f(t) \) given by (2.6), and \( k_1, k_3, k_4 \) fixed positive constants. Then, given \( \epsilon_2 > 0 \), there exists a \( \delta k_2 \geq 0 \) such that \( |\tilde{k}_2 - k_2| < \delta k_2 \) implies
\[ \delta y_2 = |y(\tilde{k}_2, t) - y(k_2, t)| < \epsilon_2. \quad (2.30) \]

**Proof:** Again, initially the outline follows exactly that given in Lemma 4, with \( k_4 \neq 0 \), and \( t \) unbounded. We seek to find a bound on
\[ y_{k_2}(k,t) = \frac{\partial}{\partial k_2} y(t) \]
\[ = \frac{\partial f}{\partial k_2}(t) * u(t) \]
\[ = g_2(k,t) * u(t). \]
For an easy derivation, we introduce some notation, following the expression for \( f \) in (2.5),

\[
\begin{align*}
v_{\Delta} &= \frac{\partial \Delta}{\partial k_2} = \frac{k_2 + k_3 - k_4}{\Delta}, \\
A &= A(t) = (k_3 + k_4 - \alpha_1)e^{-\alpha_1 t} - (k_3 + k_4 - \alpha_2)e^{-\alpha_2 t}, \\
A^T &= A^T(t) = (k_3 + k_4 - \alpha_1)e^{-\alpha_1 t} + (k_3 + k_4 - \alpha_2)e^{-\alpha_2 t}.
\end{align*}
\]

Then, \( g_2(k, t) \) is found to be

\[
g_2(k, t) = \frac{-k_1 A v_{\Delta}}{\Delta^2} + \frac{k_1}{2\Delta} \left[ -tA + tv_{\Delta} A^T + v_{\Delta} (e^{-\alpha_1 t} + e^{-\alpha_2 t}) - (e^{-\alpha_1 t} - e^{-\alpha_2 t}) \right] \leq 0.
\]

Given that \( k_i \in (0, 1) \) for all \( i \), it can be shown that \( g_2(k, t) \leq 0 \) for all time \( t \geq 0 \). Also, in the limit as \( t \to \infty \), using L’Hospital and the continuity of the exponential function, we obtain

\[
\lim_{t \to \infty} g_2(k, t) = \lim_{t \to \infty} \frac{-k_1 A v_{\Delta}}{\Delta^2} + \frac{k_1}{2\Delta} \left[ -tA + tv_{\Delta} A^T + v_{\Delta} (e^{-\alpha_1 t} + e^{-\alpha_2 t}) - (e^{-\alpha_1 t} - e^{-\alpha_2 t}) \right] = 0,
\]

where we used the fact that \( \lim_{t \to \infty} \frac{1}{e^{\alpha t}} = 0 \) for each \( \alpha_i > 0 \). The limit for \( t \to 0 \) is

\[
\lim_{t \to 0} g_2(k, t) = \frac{k_1 v_{\Delta}}{\Delta^2} \Delta - \frac{k_1 v_{\Delta}}{\Delta} = 0.
\]

Then, as before,

\[
|y_{v_2}(k, t)| = -\int_0^t g_2(k, s) u(t - s) ds \\
\leq M \int_0^t \frac{k_1 A(s) v_{\Delta}}{\Delta^2} + \frac{k_1}{2\Delta} \left[ sA(s) - sv_{\Delta} A^T(s) - v_{\Delta} (e^{-\alpha_1 s} + e^{-\alpha_2 s}) + (e^{-\alpha_1 s} - e^{-\alpha_2 s}) \right] ds \\
= MN_2(k, t),
\]

where, given \( I_A = \int_0^t A(s) ds \), \( I_{tA} = \int_0^s A(s) ds \), and \( I_{tA^T} = \int_0^s A^T(s) ds \),

\[
N_2(k, t) = \frac{k_1 I_A v_{\Delta}}{\Delta^2} + \frac{k_1}{2\Delta} \left[ I_{tA} - v_{\Delta} I_{tA^T} + v_{\Delta} \left( \frac{e^{-\alpha_1 t} - 1}{\alpha_1} + \frac{e^{-\alpha_2 t} - 1}{\alpha_2} \right) \right] - \frac{k_1}{2\Delta} \left( \frac{e^{-\alpha_1 t} - 1}{\alpha_1} - \frac{e^{-\alpha_2 t} - 1}{\alpha_2} \right).
\]
Here, it is easy to derive the integrals that appear in \( N_2(k,t) \),

\[
I_A = (k_3 + k_4 - \alpha_1) \frac{1 - e^{-\alpha_1 t}}{\alpha_1} - (k_3 + k_4 - \alpha_2) \frac{1 - e^{-\alpha_2 t}}{\alpha_2},
\]

\[
I_{tA} = (k_3 + k_4 - \alpha_1) \left[ \frac{1 - e^{-\alpha_1 t}}{\alpha_1^2} - \frac{t e^{-\alpha_1 t}}{\alpha_1} \right] - (k_3 + k_4 - \alpha_2) \left[ \frac{1 - e^{-\alpha_2 t}}{\alpha_2^2} - \frac{t e^{-\alpha_2 t}}{\alpha_2} \right],
\]

\[
I_{tAT} = (k_3 + k_4 - \alpha_1) \left[ \frac{1 - e^{-\alpha_1 t}}{\alpha_1^2} - \frac{t e^{-\alpha_1 t}}{\alpha_1} \right] + (k_3 + k_4 - \alpha_2) \left[ \frac{1 - e^{-\alpha_2 t}}{\alpha_2^2} - \frac{t e^{-\alpha_2 t}}{\alpha_2} \right].
\]

Clearly, \( N_2(t) \geq 0 \) since it is the integral of a nonnegative quantity. With this \( N_2(t) \), which is well defined for all \( t \geq 0 \), one can choose \( \delta k_2 < \frac{\varepsilon_2}{MC_2} \), where \( \delta k_2 \) is time dependent.

**Remark 5** One can again look for a uniform bound for \( y_{k_2}(k,t) \). Given \( N_2(k,t) \geq 0 \), \( N_2(k,t) \) an increasing function of time, the maximum value is obtained in the limit, which is dictated by the limits of \( I_A, I_{tA} \) and \( I_{tAT} \). Hence, we first find

\[
\lim_{t \to \infty} I_A = (k_3 + k_4) \left( \frac{1}{\alpha_1} - \frac{1}{\alpha_2} \right) = \frac{\Delta(k_3 + k_4)}{k_2 k_4} := c_1,
\]

\[
\lim_{t \to \infty} I_{tA} = (k_3 + k_4) \left( \frac{1}{\alpha_1^2} - \frac{1}{\alpha_2^2} \right) - \left( \frac{1}{\alpha_1} - \frac{1}{\alpha_2} \right)
\]

\[
= \Delta \left( \frac{(k_3 + k_4)^2 + k_2(k_3 + k_4)}{k_2 k_4} \right) - \frac{\Delta}{k_2 k_4} := c_2,
\]

\[
\lim_{t \to \infty} I_{tAT} = (k_3 + k_4) \left( \frac{1}{\alpha_1^2} + \frac{1}{\alpha_2^2} \right) - \left( \frac{1}{\alpha_1} + \frac{1}{\alpha_2} \right)
\]

\[
= \frac{(k_2 + k_3 + k_4)^2 - 2k_2 k_4 (k_3 + k_4)}{k_2^2 k_4^2} - \frac{k_2 + k_3 + k_4}{k_2 k_4} := c_3,
\]

and derive

\[
\lim_{t \to \infty} N_2(k,t) = c_1 \frac{k_1 v_\Delta}{\Delta^2} + \frac{k_1}{2\Delta} \left[ c_2 - v_\Delta c_3 - v_\Delta \left( \frac{1}{\alpha_1} + \frac{1}{\alpha_2} \right) + \left( \frac{1}{\alpha_1} - \frac{1}{\alpha_2} \right) \right] := C_2
\]

using L’Hospital’s rule and the continuity of the exponential function. Given \( \varepsilon_2 \), the perturbation in \( y \) with respect to \( k_2 \) can be uniformly bounded,

\[
\delta k_2 < \frac{\varepsilon_2}{MC_2}.
\]

**Lemma 9** Let \( y(t) = \int_0^t f(s) u(t - s) ds \), \( f(t) \) given by (2.6), and \( k_1, k_2, k_4 \) fixed positive constants. Then, given \( \varepsilon_3 > 0 \), there exists a \( \delta k_3 \geq 0 \) such that \( |\bar{k}_3 - k_3| < \delta k_3 \) implies

\[
\delta y_3 = |y(\bar{k}_3,t) - y(k_3,t)| < \varepsilon_3.
\]
Proof: The initial outline follows the one given in Lemma 4. We seek to find a bound on

\[ y_{k_3}(k, t) = \frac{\partial}{\partial k_3} y(t) = \frac{\partial f}{\partial k_3} (t) * u(t) = g_3(k, t) * u(t). \]

Again, for an easy derivation, we introduce some notation,

\[ v_\Delta = \frac{\partial \Delta}{\partial k_3} = \frac{k_2 + k_3 + k_4}{\Delta}, \]

and \( A \) and \( A^T \) defined the same as in (2.31), from the proof of Lemma 8. Then \( g_3(k, t) \) is given by

\[ g_3(k, t) = \frac{-k_1 A v_\Delta}{\Delta^2} + \frac{k_1}{2 \Delta} \left[ -tA + tv_\Delta A^T + v_\Delta \left( e^{-\alpha_1 t} + e^{-\alpha_2 t} \right) \right] \geq 0. \]

Note the similarity with \( g_2(k, t) \), given in (2.32). Hence,

\[ \lim_{t \to \infty} g_3(k, t) = 0, \quad \lim_{t \to 0} g_3(k, t) = 0. \]

Therefore,

\[ |y_{k_3}(k, t)| = \int_0^t g_3(k, s) u(t - s) ds \leq M \int_0^t \frac{-k_1 A(s) v_\Delta}{\Delta^2} \]

\[ - \frac{k_1}{2 \Delta} \left[ sA(s) - sv_\Delta A^T(s) - v_\Delta \left( e^{-\alpha_1 s} + e^{-\alpha_2 s} \right) - \left( e^{-\alpha_1 s} - e^{-\alpha_2 s} \right) \right] ds \]

\[ = MN_3(k, t), \]

where, given \( I_A, I_{tA} \) and \( I_{AT} \) as in the proof of Lemma 8,

\[ N_3(k, t) = \frac{-k_1 I_A v_\Delta}{\Delta^2} - \frac{k_1}{2 \Delta} \left[ I_{tA} - v_\Delta I_{tAt} + v_\Delta \left( \frac{e^{-\alpha_1 t} - 1}{\alpha_1} + \frac{e^{-\alpha_2 t} - 1}{\alpha_2} \right) \right] \]

\[ - \frac{k_1}{2 \Delta} \left( \frac{e^{-\alpha_1 t} - 1}{\alpha_1} - \frac{e^{-\alpha_2 t} - 1}{\alpha_2} \right). \]

Again, \( N_3(t) \geq 0 \) as the integral of a nonnegative quantity, and one can choose

\[ \delta k_3 < \frac{\epsilon_3}{MN_3(k, t)}, \]

where \( \delta k_3 = \delta k_3(t) \).
Remark 6 A uniform bound is achieved by using the maximum norm,

\[ ||\delta y_3(k_3, t)|| \leq (\delta k_3)M \max_{t \geq 0} N_3(k, t). \]

As \( N_3(t) \) increases with time, the maximum is attained in the limit,

\[
\lim_{t \to \infty} N_3(k, t) = c_1 \frac{-k_1v_\Delta}{\Delta^2} - \frac{k_1}{2\Delta} \left[ c_2 - v_\Delta c_3 - v_\Delta \left( \frac{1}{\alpha_1} + \frac{1}{\alpha_2} \right) - \left( \frac{1}{\alpha_1} - \frac{1}{\alpha_2} \right) \right] := C_3 \tag{2.36}
\]

where \( c_1, c_2, c_3 \) are given in (2.33). Then, given \( \epsilon_3 \),

\[ \delta k_3 \leq \frac{\epsilon_3}{MC_3}, \]

where the bound on the left hand side is independent of time.

Lemma 10 Let \( y(t) = \int_0^t f(s)u(t - s)ds, f(t) \) given by (2.6), and \( k_1, k_2, k_3 \) fixed positive constants. Then, given \( \epsilon_4 > 0 \), there exists a \( \delta k_4 \geq 0 \) such that \( |\delta y_4| = |y(\tilde{k}_4, t) - y(k_4, t)| < \epsilon_4. \) \tag{2.37}

Proof: We follow the outline given in Lemma 4. Hence, we need to find a bound for

\[
y_{k_4}(k, t) = \frac{\partial}{\partial k_4} y(t)
= \frac{\partial f}{\partial k_4}(t) * u(t)
= g_4(k, t) * u(t).
\]

We introduce some notation,

\[ v_\Delta = \frac{\partial \Delta}{\partial k_4} = \frac{k_3 + k_4 - k_2}{\Delta}, \]

and let \( A \) and \( A^T \) be defined as in (2.31). Then \( g_4(k, t) \) is given by

\[ g_4(k, t) = \frac{-k_1 A v_\Delta}{\Delta^2} + \frac{k_1}{2\Delta} \left[ -tA + tv_\Delta A^T + v_\Delta (e^{-\alpha_1 t} + e^{-\alpha_2 t}) + (e^{-\alpha_1 t} - e^{-\alpha_2 t}) \right] \leq 0. \]

Computing the limits we find

\[ \lim_{t \to \infty} g_4(k, t) = 0, \]
\[ \lim_{t \to 0} g_4(k, t) = 0. \]

Therefore,

\[
|y_{k_4}(k, t)| = -\int_0^t g_4(k, s)u(t - s)ds
\leq M \int_0^t \frac{k_1 A(s) v_\Delta}{\Delta^2}
- \frac{k_1}{2\Delta} \left[ -sA(s) + sv_\Delta A^T(s) + v_\Delta (e^{-\alpha_1 s} + e^{-\alpha_2 s}) + (e^{-\alpha_1 s} - e^{-\alpha_2 s}) \right] ds
= MN_4(k, t),
\]
where
\[ N_4(k, t) = \frac{k_1 I_A v_t}{\Delta^2} - \frac{k_1}{2\Delta} \left[ -I_t A + v_t I_{tA} - v_t \left( \frac{e^{-\alpha_1 t} - 1}{\alpha_1} + \frac{e^{-\alpha_2 t} - 1}{\alpha_2} \right) \right] + \frac{k_1}{2\Delta} \left( \frac{e^{-\alpha_1 t} - 1}{\alpha_1} - \frac{e^{-\alpha_2 t} - 1}{\alpha_2} \right), \]
and \( I_A, I_{tA}, \) and \( I_{tA} \) are given in the proof of Lemma 8. Then (2.37) follows by choosing
\[ 0 < \delta k_4 < \frac{c_i}{MN_4(k, t)}, \]
where \( \delta k_4 \) is time dependent. \( \square \)

**Remark 7** A uniform bound for \( \delta k_4 \) is given by considering
\[ \lim_{t \to \infty} N_4(k, t) = c_1 \frac{k_1 v_t}{\Delta^2} - \frac{k_1}{2\Delta} \left[ -c_2 + v_t c_3 + v_t \left( \frac{1}{\alpha_1} + \frac{1}{\alpha_2} \right) + \left( \frac{1}{\alpha_1} - \frac{1}{\alpha_2} \right) \right] := C_4, \] (2.38)
where \( c_1, c_2, c_3 \) are given in (2.33). Then one can choose \( \delta k_4 < \frac{c_1}{MN_4(k, t)} \), and here \( \delta k_4 \) is independent of time.

**Theorem 4** The problem of solving equation \( y(t) = \int_0^t f(s) u(t - s) ds \) for \( f(t) \), given its functional form (2.6), where \( t \geq 0 \) is unbounded, is well-posed with respect to variations in the parameter vector \( k \).

**Proof:** From Theorem 2 and Lemmas 7, 8, 9, 10 it follows that the problem of solving (2.7) for \( k = [k_1, k_2, k_3, k_4] \) is well posed with respect to the parameter vector \( k \). \( \square \)

Note, even though the convolution solution to (2.7) is well-posed with respect to the parameters defining the function \( f(t; k) \), it is not well-posed with respect to perturbations in \( y(t) \) introduced by other means, ie random errors. This is well known for Fredholm integrals of first kind [56], for which the convolution (2.7) is an example. Hence, when solving for the parameter space \( k \), we are solving an ill-posed problem due to the noise introduced by the use of measured data \( y(t) \).

**2.2. Simulations.** The uniform bounds derived in Lemmas 7, 8, 9, 10 depend nonlinearly on the parameter vector \( k \), see Table 1.

We compare the form of the \( g_i(t), i = 1, \ldots, 3 \) for the two cases, one assuming \( k_4 = 0 \), and the other, \( k_4 \neq 0 \). This demonstrates the different behaviour of the solutions \( y_0(t) \), corresponding to \( k_4 = 0 \), and \( y(t) \), with respect to variations in \( k_i \). We use two sets of data, obtained experimentally, one in [29], \( k_H = [.102, .013, .062, .0068] \), and one in [46], \( k_P = [.099, .141, .165, .013] \). The results are illustrated in Figures 15, 16, 17. A comparison of the form of the \( N_i(t), i = 1, \ldots, 3 \) for \( k_4 = 0 \) and \( k_4 \neq 0 \) is presented in Figures 18, 19, 20.

The parameter vector \( k_P \) leads to comparatively smaller values for \( C_i \) then the parameter vector \( k_H \), especially for \( C_3 \) and \( C_4 \), which are smaller by a factor of almost 2. This seems to indicate that, assuming fixed transport parameters, both from plasma
Table 1. Analytically derived expressions for $C_i = \lim_{t \to \infty} \int_0^t g_i(k,s)ds$ from Remarks 4, 5, 6, 7, when considering a solution to (2.7) with $k_4 \neq 0$. The forms for $c_1, c_2, c_3$ are given in (2.33).

<table>
<thead>
<tr>
<th>$C_1$</th>
<th>( \frac{c_1 k_1 v_\Delta + k_1 v_\Delta}{\Delta} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_2$</td>
<td>( c_2 - v_\Delta c_3 - v_\Delta \left( \frac{1}{\alpha_1} + \frac{1}{\alpha_2} \right) + \left( \frac{1}{\alpha_1} - \frac{1}{\alpha_2} \right) ), where $v_\Delta = \frac{k_2 + k_3 - k_4}{\Delta}$</td>
</tr>
<tr>
<td>$C_3$</td>
<td>( c_2 - v_\Delta c_3 - v_\Delta \left( \frac{1}{\alpha_1} + \frac{1}{\alpha_2} \right) - \left( \frac{1}{\alpha_1} - \frac{1}{\alpha_2} \right) ), where $v_\Delta = \frac{k_2 + k_3 + k_4}{\Delta}$</td>
</tr>
<tr>
<td>$C_4$</td>
<td>( c_2 - v_\Delta c_3 - v_\Delta \left( \frac{1}{\alpha_1} + \frac{1}{\alpha_2} \right) - \left( \frac{1}{\alpha_1} - \frac{1}{\alpha_2} \right) ), where $v_\Delta = \frac{k_3 + k_4 - k_2}{\Delta}$</td>
</tr>
</tbody>
</table>

And back to it, higher phosphorylation and de-phosphorylation values imply smaller bound values $C_3$ and $C_4$, hence a smaller interval of variation for these respective parameters. Alternatively, small values for the phosphorylation and de-phosphorylation lead to larger intervals of variations, and therefore a potentially noisier spatial distribution.

To measure how small one needs to select $\delta k_i$ for a given parameter $k_i$, we computed values for the uniform bounds $C_i$, derived analytically - see Table 2. Figures 21 and 22 show the limiting behaviour of $N_i$ to $C_i$ for each $i = 1, \ldots, 4$.

Table 2. Computed values for the uniform bounds $C_i$, given two sets of experimentally derived $k$ values, from [29], [46].

<table>
<thead>
<tr>
<th>$k = [0.102, 0.13, 0.062, 0.0068]$</th>
<th>$k = [0.099, 0.141, 0.165, 0.013]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_1$</td>
<td>77.82</td>
</tr>
<tr>
<td>$C_2$</td>
<td>61.06</td>
</tr>
<tr>
<td>$C_3$</td>
<td>115.38</td>
</tr>
<tr>
<td>$C_4$</td>
<td>1052.31</td>
</tr>
</tbody>
</table>

From Table 2, we note that $[C_1, C_2, C_3, C_4] \simeq [O(10^2), O(10^2), O(10^2), O(10^3)]$. Recall, $C_i = \lim_{t \to \infty} \int_0^t |g_i(k(t-s))ds$ and, given $\epsilon_i$, there exists $\delta k_i < \frac{\epsilon_i}{MC_i}$ such that it implies $|\delta y_i| < \epsilon_i$. Let $\epsilon_i = \epsilon$ for all $i$, and assume $M$ to be order 1. Then, the vector $[\delta k_1, \delta k_2, \delta k_3, \delta k_4]$ is of order $\epsilon[10^{-2}, 10^{-2}, 10^{-2}, 10^{-3}]$.

**Remark 8** We conclude, in addition to well-posedness with respect to parameter variations, that tracer dynamics occur on two different scales, one due to parameters $k_1, k_2, k_3$, and another scale due exclusively to $k_4$ which varies slowly at small time, and also decays slower then the other parameters.
3. Conclusions and Discussion

The theoretical analysis of the forward problem points out the dependency of the non-dimensional solution $Y(\tau)$ on the small parameter $\bar{k}_4$, adjusted for the scale difference between the maximum value of the input $u(t)$ and maximum value of the output $y(t)$. Using a perturbation expansion, a bound on the difference between $Y(\tau)$ and its first order approximation $Y_0(\tau)$, assuming $\bar{k}_4 = 0$, is derived. The analytical results are supported by simulations which show that as the time interval of measurement increases, the difference $|Y(\tau) - Y_0(\tau)|$ increases. We conclude that for small time intervals, and appropriate parameter values, one can assume $Y(\tau)$ to be well estimated by $Y_0(\tau)$.

The inverse problem analysis shows that the problem (1.3) is well-posed with respect to variation in parameters. The time scales over which $k_i$, $i = 1, 2, 3$ vary is one order of magnitude greater than the time scale over which $k_4$ varies. The inverse problem of parameter estimation is therefore experienced on two time scales. This is in accord with biological observations. The ill-posedness of the problem (1.3) is due to the high level of noise in the data $y(t)$. 
Figure 11. Comparison of $Y(\tau)$, assuming $k_4 = 0.068$ and $Y_0(\tau)$, assuming $k_4 = 0$. Both $Y(\tau)$, and $Y_0(\tau)$ were obtained using a second order Runge-Kutta method for the numerical solution of (1.2), and a vector of parameter values, $k = [.102, .130, .062]$ given in [29]. The final measurement time for this graph is $T = 25$ min.
Figure 12. As Figure 11, but to time $T = 45$ min.
Figure 13. Graph comparing the variation of $k_1$ and $k_2$ as functions of end-time, using two algorithms, both based on the Filtered Least Squares (FLS) method, one assuming $k_4 \neq 0$, and one assuming $k_4 = 0$. The values depicted by * are exact parameter values.
Figure 14. Graph comparing the variation of $k_3$ and $K = \frac{k_1k_3}{k_2+k_3}$ as functions of end-time, using two algorithms, both based on FLS, one which assumes $k_4 \neq 0$, and one assuming $k_4 = 0$. The values depicted by * are exact parameter values.
Figure 15. The functions $g_i(t) = \frac{\partial}{\partial k_i} f(k_i, t)$ for the parameter vector $k_H = [.102, .13, .062, .0068]$. 
Figure 16. The functions $g_i(t) = \frac{\partial}{\partial k_i} f(k_i, t)$ for parameter vector $k_P = [.099, .141, .165, .013]$, referenced in [46].
Figure 17. Magnification of the graphs shown in Figure 16.
Figure 18. The functions $N_i(t) = \int_0^t |g_i(s)| ds$ for the parameter vector $k_H = [.102, .13, .062, .0068]$. 
Figure 19. The functions $N_i(t) = \int_0^t |g_i(s)| ds$ for parameter vector $k_P = [.099, .141, .165, .013]$, referenced in [46].
Figure 20. Magnification of the graphs shown in Figure 19.
Figure 21. Comparison of the uniform bounds $C_i$, and the integral values $N_i(t)$ for each $i = 1, \ldots, 4$, using parameter vector $k_H = [.102, .13, .062, .0068]$. 
Figure 22. Comparison of the uniform bounds $C_i$, and the integral values $N_i(t)$ for each $i = 1, \ldots, 4$, using parameter vector $k_P = [.099, .141, .165, .013]$. 
CHAPTER 3

Computational Techniques

In this chapter we consider numerical models for the solution to the inverse problem. We introduce a linear model, with two different formulations which lead to two different numerical approaches. A formal error analysis of the two formulations is presented. The more traditional nonlinear model is also introduced. Several least squares algorithms are compared in accuracy with the nonlinear model. The convergence properties of the generalized linear least squares algorithm (GLLS) for a general problem are considered. We present several numerical examples to support the analysis.

1. The Linear Problem and its Discrete Formulation

The form of the solution of the direct problem (2.4) leads to a nonlinear estimation of the parameters \( k_i, i = 2, \ldots, 4 \). For simplicity, we consider as example the solution to the inverse problem (2.7) assuming \( k_4 = 0 \). Then, following (2.14), and assuming exact values for \( y(t) \), one can linearly solve for \( k_1 \), as a function of \( k_2, k_3, t \),

\[
k_1 = \frac{(k_2 + k_3)y(t)}{k_3 \int_0^t u(s)ds + k_2 \int_0^t u(s)e^{-(k_2+k_3)(t-s)}ds}.
\]

Here, one can take \( t \) to be any positive value. However, as one tries to solve for the remaining two parameters \( k_2 \) and \( k_3 \) it is clear that their dependency on \( y(t) \) will be nonlinear due to the exponential form \( e^{-(k_2+k_3)t} \). Even if one assumes \( u(t) \) to be constant, \( u(t) = c \), which would facilitate an easy integration, the solution for \( k_2 \) and \( k_3 \) will be nonlinear,

\[
k_2e^{-(k_2+k_3)t} = k_2 + k_3(k_2 + k_3)t - \frac{y(t)(k_2 + k_3)^2}{k_1 c}.
\]

When one considers the over-determined system \( y(t) = (f * u)(t) \), \( f(t) = f(t; k) \) for a sequence of times \( \{t_i\}_{i=1}^m \) where \( m > 4 \), an approximate solution can be obtained by considering a minimization

\[
\min_k \sum_{i=1}^m ||y(t_i) - (f(k) * u)(t_i)||
\]

in some norm || · ||. By using a Laplace transform, we can modify the original problem (1.3) into a integral equation where the parameters \( a_1 = k_2 + k_3 + k_4, a_2 = k_2k_4, b_1 = k_1, b_2 = k_1(k_3 + k_4) \) appear linearly. Note, the parameters \( a_2 \) and \( b_2 \) remain nonlinear in \( k_2, k_4, \) and \( k_1, k_3, k_4 \), respectively.
1.1. Laplace Formulation. We derive the Laplace transform solution of the inverse problem (1.3) for the general $n^{th}$ order differential equation,

$$y^{(n)}(t) + a_1 y^{(n-1)}(t) + \cdots + a_n y(t) = b_1 u^{(n-1)}(t) + b_2 u^{(n-2)}(t) + \cdots + b_n u(t). \quad (3.1)$$

The Laplace transform of (3.1) is

$$s^n Y(s) - \sum_{i=1}^{n} s^{n-i} y^{(i-1)}(0) + a_1 \left[ s^{n-1} Y(s) - \sum_{i=2}^{n} s^{n-i} y^{(i-2)}(0) \right] + \cdots + a_n Y(s) = b_1 \left[ s^{n-1} U(s) - \sum_{i=2}^{n} s^{n-i} u^{(i-2)}(0) \right] + \cdots + b_n U(s) \quad (3.2)$$

where $y(0), \ldots, y^{(n-1)}(0)$ and $u(0), \ldots, u^{(n-2)}(0)$ are the initial conditions for the output, and input functions, respectively, and $Y(s)$ and $U(s)$ the Laplace transforms of $y(t)$ and $u(t)$,

$$Y(s) = (\mathcal{L}y)(s) = \int_{0}^{\infty} y(t)e^{-st} dt,$$

$$U(s) = (\mathcal{L}u)(s) = \int_{0}^{\infty} u(t)e^{-st} dt.$$

Rewrite (3.2) as

$$(s^n + a_1 s^{n-1} + \cdots + a_n) Y(s) = (b_1 s^{n-1} + b_2 s^{n-2} + \cdots + b_n) U(s) + v_1 s^{n-1} + \cdots + v_n \quad (3.3)$$

where $v_1, v_2, \ldots, v_n$ are the linear combinations of the input and output initial conditions, with $a_i, b_i, i = 1, 2, \ldots$ as their coefficients. We can write this equation more compactly as

$$A(s) Y(s) = B(s) U(s) + V(s) \quad (3.4)$$

with $V(s) = v_1 s^{(n-1)} + \cdots + v_n$, and $A(s), B(s)$ the polynomials in front of $Y(s)$ and $U(s)$ respectively. In most applications, the initial conditions are all zero. In cases where the initial conditions are not known, they can be considered as parameters to be determined. Divide both sides by $s^n$ to obtain

$$Y(s) = - \left[ \sum_{i=1}^{n} a_i s^{-i} \right] Y(s) + \left[ \sum_{i=1}^{n} b_i s^{-i} \right] U(s) + \sum_{i=1}^{n} v_i s^{-i} \quad (3.5)$$

We take the inverse Laplace transform of (3.5) to get back an expression in time-domain

$$y(t) = -a_1 \int_{0}^{t} y(\tau) d\tau - a_2 \int_{0}^{t} \int_{0}^{\tau_1} y(\tau) d\tau d\tau_1 - \cdots$$

$$- a_n \int_{0}^{t} \int_{0}^{\tau_1} \cdots \int_{0}^{\tau_{n-1}} y(\tau) d\tau d\tau_{n-1} \ldots d\tau_1$$

$$+ b_1 \int_{0}^{t} u(\tau) d\tau + b_2 \int_{0}^{t} \int_{0}^{\tau_1} u(\tau) d\tau d\tau_1 + \cdots$$

$$+ b_n \int_{0}^{t} \int_{0}^{\tau_1} \cdots \int_{0}^{\tau_{n-1}} u(\tau) d\tau d\tau_{n-1} \ldots d\tau_1$$

$$+ v_1 + \cdots + v_n \frac{t^{n-1}}{(n-1)!} \quad (3.6)$$
The data is sampled at discrete times $t_i$ where $i = 1, 2, \ldots, m$ with $m > 2n$ if all initial conditions are zero, otherwise $m > 3n$, where $n$ is the order of the differential equation. The $m$ equations resulting from this sampling, can be written in matrix form

$$Y = X_L \theta$$  \hspace{1cm} (3.7)

where $Y = [y(t_1), y(t_2), \ldots, y(t_m)]^T$, $\theta = [-a_1, -a_2, \ldots, -a_n, b_1, b_2, \ldots, b_n, v_1, \ldots, v_n]^T$, and $X_L$ is the coefficient matrix whose entries are integrals of the input and output functions. For simplicity of the derivation, we will assume all initial conditions are zero, but remark that a similar analysis results otherwise. A row $(X_L)_i$ of $X_L$ has the entries

$$(X_L)_{i,j} = \int_0^{t_i} \int_0^{\tau_1} \cdots \int_0^{\tau_{j-1}} y(\tau) \, d\tau \, d\tau_{j-1} \cdots d\tau_1, \quad \text{for } 1 \leq j \leq n$$

$$(X_L)_{i,j} = \int_0^{t_i} \int_0^{\tau_1} \cdots \int_0^{\tau_{j-1}} u(\tau) \, d\tau \, d\tau_{j-1} \cdots d\tau_1, \quad \text{for } n + 1 \leq j \leq 2n.$$

Assume the coefficient matrix $X_L$ is not rank deficient. The system is over-determined, with $m > 2n$. Usually, an over-determined system $X_L \theta = Y$ does not have an exact solution since $Y$ must be an element of range$(X_L)$ which is a proper subspace of $R^m$. In this case one can obtain a best approximate solution by minimizing the residual $r = Y - X_L \theta$ in some norm. We choose to minimize the problem in the least squares sense,

$$\min_{\theta \in R^{2n}} \|Y - X_L \theta\|_2^2 = \min_{\theta \in R^{2n}} \sum_{i=1}^{2n} (Y_i - (X_L)_i \theta)^2, \hspace{1cm} (3.8)$$

where $(X_L)_i \theta = \sum_{j=1}^{2n} (X_L)_{i,j} \theta_j$. Assuming $X_L$ has full rank, there is an unique solution $\theta_{ls}$.

For the FDG model $n = 2$ and the system reduces to minimizing $r = Y - X_L \theta$ where each row $i$ in $X_L$ has the form

$$(X_L)_i = \left[ \int_0^{t_i} y(\tau) \, d\tau, \int_0^{t_i} \int_0^{\tau_1} y(\tau) \, d\tau \, d\tau_1, \int_0^{t_i} \int_0^{\tau_1} u(\tau) \, d\tau \, d\tau_1, \int_0^{t_i} \int_0^{\tau_1} u(\tau) \, d\tau \, d\tau_1 \right], \hspace{1cm} (3.9)$$

and $\theta = [-a_1, -a_2, b_1, b_2]$.

### 1.2. Volterra Formulation.

Note that the initial value problem (1.3) is equivalent to a Volterra integral equation of the second kind (see Theorem 1.1.1 in [6]),

$$y(t) = h(t) - \int_0^t H(t, s) y(s) ds, \quad t \geq 0,$$

$$h(t) = \int_0^t (t-s)(b_1 u'(s) + b_2 u(s)) ds,$$

$$H(t, s) = a_1 + a_2 (t-s). \hspace{1cm} (3.10)$$

Then the solution given in (2.4) is the solution of this Volterra integral equation.
Note, integrating by parts, we can rewrite (3.10) as

\[ y(t) = \int_0^t H_u(t, s)u(s)ds - \int_0^t H_y(t, s)y(s)ds, \]

with polynomial kernels \( H_u(t, s) = b_1 + b_2(t - s) \) and \( H_y(t, s) = a_1 + a_2(t - s) \). Thus, the tracer concentration \( y(t) \) is equal to the difference between the total tracer amount entering system and the total tracer amount exiting system. This can be further exploited from the point of view of linear tracer kinetic modeling, to assign the polynomials \( H_u(t, s) \) and \( H_y(t, s) \) their respective parameters to account for system kinetics. Any general linear \( n^{th} \) order differential equation can be written in this form [6], which we refer to as the Volterra integral formulation. Given that tracer kinetics are modeled via linear compartment models, which lead to linear differential equations of degree equal to the number of compartments considered, one could model tracer kinetic via the Volterra integral formulation. Although a through study of this proposed modeling technique would require careful validation, we mention it here as an alternative to the classical differential equation modeling. This observation is first noted here with respect to tracer kinetic modeling, although mathematically this relationship is well known and understood, [40].

For the specific problem (3.10) we can obtain a different linear system by integrating by parts once and rearranging terms

\[ y(t) = -a_1 \int_0^t y(\tau)d\tau - a_2 \left[ t \int_0^t y(\tau)d\tau - \int_0^t \tau y(\tau)d\tau \right] + b_1 \int_0^t u(\tau)d\tau + b_2 \left[ t \int_0^t u(\tau)d\tau - \int_0^t \tau u(\tau)d\tau \right]. \]  

(3.11)

Then we obtain the linear system \( X_V \theta = Y \) and the matrix \( X_V \) has rows of the form

\[ (X_V)_i = \left[ \int_0^{t_i} y(\tau)d\tau, \int_0^{t_i} y(t_i - \tau)d\tau, \int_0^{t_i} u(\tau)d\tau, \int_0^{t_i} u(t_i - \tau)d\tau \right], \]  

(3.12)

where we note the difference between this expression and the one above, (3.9), is in replacing the evaluation of double integrals with entries \( y(\tau) \) by the evaluation of single integrals with entries of the form \( y(\tau)(t_i - \tau) \).

Another derivation of this result is obtained by rewriting the double integrals in (3.9) as

\[ \int_0^t \int_0^\tau y(\tau)d\tau d\tau_1 = \int_0^t \int_0^\tau y(\tau)d\tau_1 d\tau = \int_0^t y(\tau)\tau_1 \big|_0^\tau d\tau = \int_0^t y(\tau)(t - \tau)d\tau. \]  

(3.13)

We solve this problem by minimizing the residual \( r = Y - X_V \theta \) in the least squares sense.

2. Error Analysis

Our analysis thus far has not accounted for the measurement noise in the data,

\[ Y_i = Y(t_i) + e(t_i) \]
where \( Y(t_i) \) is the exact data and \( e(t_i) \) is the error at time \( t_i, i = 1, \ldots, m \). For dynamic PET data processing, the measurement error is due to several factors: photon counting in the process of photon emission, intrinsic tomograph resolution, image reconstruction processing (the recovery of the true activity distribution from a two dimensional Radon transformed data), length of each of the imaging time intervals, and total imaging time in relation to half-life of the radioactive tracer. Photon counting is assumed to have a Poisson distribution \([15], [16], [8]\). When the measurement counts are large, this error can be approximated to be Gaussian with 0 mean and unknown variance \( \sigma^2 \), see \([8]\).

In addition to this noise, errors are introduced from the calculation of the integrals which appear in the entries of \( X_L \) and \( X_V \). To complicate the problem further, the data \( y(t_i), i = 1, \ldots, m \) are non equally spaced. Note the errors introduced by the numerical estimation of the integrals in \( X_L, X_V \) appear only in the linear formulation of the problem. These truncation errors are minimal in comparison to the measurement errors, yet they do affect the accuracy of the solution.

We consider first the analysis of the problem when there is no measurement noise. This will give us an indication of the effect of the numerical integration on the final solution, for \( k_i, i = 1 \ldots, 4 \).

Most papers using the linear formulation of the problem \([9], [15]\) do not address the integration methods used in forming \( X_L \). Given the high noise levels in the real data, we cannot expect to be able to interpolate the function accurately \([40]\). The composite trapezoidal rule for non-equally spaced data does not require any further data interpolation, so we assume this is the method of choice. Note the Volterra formulation needs only the computation of single integrals. For the double integral formulation, however, which we also do via the composite trapezoidal rule, we must interpolate the data linearly on a fine, equally spaced grid of mesh size \( h = \frac{\min(t_{i+1} - t_i)}{2} \); we use the Matlab linear interpolation function \( \text{interp1} \).

2.1. Interpolation Error. Let \( \{t_i\}_{i=0,\ldots,m} \) be the measurement schedule used in a PET sampling. Then the function values \( y(t_i) = y_i, u(t_i) = u_i \) are obtained from radioactivity counts. The time intervals \( dt_j = t_{j+1} - t_j \) are non-equal and the composite trapezoidal rule for \( \int_0^{t_i} y(\tau)d\tau \) is given by

\[
I(y_i) = \frac{1}{2} \sum_{j=0}^{i-1} dt_j (y_j + y_{j+1})
\]

\[
= \frac{1}{2} [y_0 dt_0 + y_1 (dt_0 + dt_1) + \cdots + y_j (dt_{j-1} + dt_j) + \cdots + y_i dt_{i-1}]
\]

\[
= \sum_{j=0}^{i} \beta_j y_j,
\]

(3.14)
where $\beta_j = \frac{dt_{j-1} + dt_j}{2}, 0 < j < i$, $\beta_0 = \frac{dt_0}{2}$ and $\beta_i = \frac{dt_{i-1}}{2}$. The cumulative error obtained from approximating the exact integral by the composite trapezoidal rule is

$$T(y_i) = \int_0^{t_i} y(\tau) d\tau - \sum_{j=0}^{i} \beta_j y_j = -\sum_{j=0}^{i-1} \frac{(dt_j)^3}{12} y''(\xi_j)$$

(3.15)

for some $\xi_j \in (t_{j+1}, t_j)$ and $t_0 = 0$, see [31]. Clearly, the error term will always depend on the integrated function.

For the entries of the Laplace matrix $X_L$, we also need the truncation error for a double integral,

$$D(y_i) = \int_0^{t_i} \int_0^{t_i} y(\tau) d\tau d\tau_1 - \sum_{j=0}^{i} \beta_j \sum_{k=0}^{j} \beta_k y_k$$

$$= -\sum_{j=0}^{i} \beta_j \sum_{k=0}^{j-1} \frac{(dt_k)^3}{12} y''(\xi_k) - \sum_{j=0}^{i-1} \frac{(dt_j)^3}{12} y'(\xi_j)$$

where $\xi_k \in (t_{k+1}, t_k), \xi_j \in (t_{j+1}, t_j)$ and $\{t_k\}$ is a partition of the interval $[0, t_j]$. In the presence of no other measurement error, the entries in matrices $X_L$ and $X_V$ will only have truncation errors of this form.

### 2.2. Simulations.

Given an analytic form of the input $u(t)$, we can analytically solve for the output in terms of the given solution vector $k$. No measurement error is simulated in this study. We form matrices $X_L, X_V$ using analytic and numerical integration. The numerical estimation of the integrals is done using the composite trapezoidal rule. The least squares problem is solved using QR, for backwards stability. The results are shown in Table 3, for various time steps (here $\mathcal{K}(X_L,V)$ denotes the matrix condition number).

The results in Table 3 confirm that even in the absence of measurement error, the least squares estimates are in error. This error is due exclusively to the truncation error in the numerical estimation of the entries in both $X_L$ and $X_V$. Both the Laplace and Volterra formulations perform poorly, yet the Laplace formulation is more accurate than the Volterra formulation. However, the Volterra formulation is faster than the Laplace, by at least one order of magnitude. Moreover, no improvement is gained by using $dt < 10^{-1}$ for the double integral Laplace formulation. Relative error estimates are presented in Table 4. Note, the relative error for the vector $k$ in $\infty$ norm shows that number of correct significant digits in the estimate $\hat{k}$ is approximately 2 for the Laplace, and 1 for the Volterra.

### 2.3. Measurement Error.

Consider now the measurement error in the sampled data, $y(t_i) = y(t_i) + \epsilon(t_i)$ for $i = 1, \ldots, m$, where $y(t_i)$ are the exact data and $\epsilon(t_i)$ are the noise at times $t_i, i = 1, \ldots, m$. The sampling in the acquisition of PET data is such that the error is larger for small time, and smaller for larger times.

As note, the Laplace and Volterra formulations require the computation of single integrals of both the input and output functions $u(t)$ and $y(t)$, respectively. In presence of
Table 3. Numerical results obtained using the Laplace and Volterra formulations. Note, the accuracy of the Volterra estimates is comparable with those given by the Laplace, with at least one order of magnitude reduction in cost. The estimates given here are the least squares solutions obtained using a QR decomposition. The Volterra formulation does not depend on the size of the time step \( dt \), used only in the double integral formulation, but we include it here to facilitate the cost comparison.

<table>
<thead>
<tr>
<th>( dt )</th>
<th>( k_1 )</th>
<th>( k_2 )</th>
<th>( k_3 )</th>
<th>( k_4 )</th>
<th>( K(X_{L,V}) )</th>
<th>Cost(Hops)</th>
</tr>
</thead>
<tbody>
<tr>
<td>True values</td>
<td>0.1020</td>
<td>0.1300</td>
<td>0.0620</td>
<td>0.0068</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analytic estimate</td>
<td>0.1020</td>
<td>0.1300</td>
<td>0.0620</td>
<td>0.0068</td>
<td></td>
<td>986</td>
</tr>
<tr>
<td>( 10^{-1} ) Laplace estimate</td>
<td>0.1032</td>
<td>0.1410</td>
<td>0.0688</td>
<td>0.0089</td>
<td>1005</td>
<td>179252</td>
</tr>
<tr>
<td>Volterra estimate</td>
<td>0.1030</td>
<td>0.1440</td>
<td>0.0713</td>
<td>0.0092</td>
<td>1056</td>
<td>16228</td>
</tr>
<tr>
<td>( 10^{-2} ) Laplace estimate</td>
<td>0.1031</td>
<td>0.1411</td>
<td>0.0689</td>
<td>0.0089</td>
<td>1005</td>
<td>1153342</td>
</tr>
<tr>
<td>Volterra estimate</td>
<td>0.1030</td>
<td>0.1440</td>
<td>0.0713</td>
<td>0.0092</td>
<td>1055</td>
<td>16228</td>
</tr>
<tr>
<td>( 10^{-3} ) Laplace estimate</td>
<td>0.1031</td>
<td>0.1411</td>
<td>0.0689</td>
<td>0.0089</td>
<td>1005</td>
<td>11237692</td>
</tr>
<tr>
<td>Volterra estimate</td>
<td>0.1030</td>
<td>0.1440</td>
<td>0.0713</td>
<td>0.0092</td>
<td>1055</td>
<td>16228</td>
</tr>
</tbody>
</table>

Table 4. Relative error estimates for the Laplace and Volterra formulations. Here, \( \hat{k} \) represents an estimate of the exact solution vector \( k \).

| \( dt \) | \( |k_1 - \hat{k}_1| / k_1 \) | \( |k_2 - \hat{k}_2| / k_2 \) | \( |k_3 - \hat{k}_3| / k_3 \) | \( |k_4 - \hat{k}_4| / k_4 \) | \( \|k - \hat{k}\|_\infty / \|k\|_\infty \) |
|---------|----------------|----------------|----------------|----------------|----------------|
| \( 10^{-1} \) Laplace estimate | 0.0118 | 0.0846 | 0.1097 | 0.3088 | 0.0846 |
| Volterra estimate | 0.0098 | 0.1077 | 0.1500 | 0.3529 | 0.1077 |
| \( 10^{-2} \) Laplace estimate | 0.0109 | 0.0854 | 0.1113 | 0.3088 | 0.0854 |
| Volterra estimate | 0.0098 | 0.1077 | 0.1500 | 0.3529 | 0.1077 |
| \( 10^{-3} \) Laplace estimate | 0.0109 | 0.0854 | 0.1113 | 0.3088 | 0.0854 |
| Volterra estimate | 0.0098 | 0.1077 | 0.1500 | 0.3529 | 0.1077 |

measurement noise, the following error estimates are obtained

\[
\int_0^{t_i} y(\tau)d\tau - \sum_{j=0}^{i} \beta_j \bar{y}_j = T(y_i) - R(y_i)
\]  

(3.16)

where \( R(y_i) = \sum_{j=0}^{i} \beta_j \epsilon_j \) is the accumulated measurement error and \( T(y_i) \) is the truncation error derived in (3.15).

Let \( E_L \) and \( E_V \) be the error matrices for the Laplace and Volterra formulations, respectively. The global errors in the first column of \( E_L \) and \( E_V \) will have the form

\[
(E_L)_{i1} = (E_V)_{i1} = T(y_i) - R(y_i).
\]  

(3.17)
Let \( f(t) = \int_0^t y(\tau) d\tau \), with \( \mathcal{T}(t_j) = \int_0^{t_j} \varphi(\tau) d\tau \). Note, the error in \( f \) has the form
\[
v_j = \mathcal{T}(t_j) - f(t_j) = \int_0^{t_j} \varphi(\tau) d\tau - \int_0^{t_j} y(\tau) d\tau = \int_0^{t_j} (\varphi(\tau) - y(\tau)) d\tau = -[T(y_j) - R(y_j)].
\]
The second column in \( E_L \) has entries
\[
(E_L)_{i,2} = \int_0^{t_i} \int_0^{t_1} y(\tau) d\tau d\tau_1 - \sum_{j=0}^i \beta_j \sum_{k=0}^j \beta_k y_k
= T(f_i) - R(f_i)
= T(f_i) - \sum_{j=0}^i \beta_j v_j
= T(f_i) + \sum_{j=0}^i \beta_j T(y_j) - \sum_{j=0}^i \beta_j \beta_k \epsilon_k, \tag{3.18}
\]
where
\[
T(f_i) = -\sum_{j=0}^{i-1} \frac{(\Delta t_j)^3}{12} f''(\xi_j) = -\sum_{j=0}^{i-1} \frac{(\Delta t_j)^3}{12} y'(\xi_j), \quad \text{and} \quad T(y_j) = -\sum_{k=0}^{j-1} \frac{(\Delta t_k)^3}{12} y''(\xi_k), \tag{3.19}
\]
with \((t_k)\) a partition of the interval \([0, t_j]\). Comparing (3.17) and (3.19) it is clear that the second order integral requires smaller time steps for interpolation. In particular, a rescaled version for the errors \( T(y_i), R(y_i) \), by the length \( t_i \) of the interval gives
\[
(E_L)_{i,2} = \int_0^{t_i} \int_0^{t_1} y(\tau) d\tau d\tau_1 - \sum_{j=0}^i \beta_j \sum_{k=0}^j \beta_k y_k
= \frac{T(f_i)}{t_i} + \sum_{j=0}^i \frac{\beta_j}{t_j} \frac{T(y_j)}{t_j} - \sum_{j=0}^i \beta_j \beta_k \epsilon_k. \tag{3.20}
\]

To examine the errors in the second column of \( E_V \), let \( v^i(t) = y(t)(t_i - t) \). Then the entries in \((E_V)_{i,2}\) have the form
\[
(E_V)_{i,2} = \int_0^{t_i} y(\tau)(t_i - \tau) d\tau - \sum_{j=0}^i \beta_j y_j (t_i - t_j) - \sum_{j=0}^i \beta_j \epsilon_j (t_i - t_j)
= T(v^i) - R(v^i) = -\sum_{j=0}^{i-1} \frac{(\Delta t_j)^3}{12} [y''(\xi_j)(t_i - \xi_j) - 2y'(\xi_j)] - R(v^i).
\]

Given \( g(t) = \int_0^t u(\tau) d\tau \) and \( w^i(t) = u(t)(t_i - t) \), the third and fourth columns of \( X_L \) and \( X_V \) will have entries
\[
(E_L)_{i,3} = (E_V)_{i,3} = T(u_i) - R(u_i),
(E_L)_{i,4} = T(g_i) - R(g_i) = -\sum_{j=0}^{i-1} \frac{(\Delta t_j)^3}{12} u'(\xi_j) - R(g_i),
(E_V)_{i,4} = T(w^i) - R(w^i) = -\sum_{j=0}^{i-1} \frac{(\Delta t_j)^3}{12} [u''(\xi_j)(t_i - \xi_j) - 2u'(\xi_j)] - R(w^i).
\]
If no measurement errors are assumed for the input function \(u(t)\), then only truncation errors will appear.

Note that even in the absence of measurement noise, the errors obtained can be large due to the use of the composite trapezoidal rule on increasingly large intervals \(\Delta t_j\). In addition, when \(\Delta t_j\) is small, usually for small \(t_j\), the input function varies widely such that \(u'(t)\) can be very large and hence induce large errors in both \(X_L\) and \(X_V\).

2.4. Simulations. We present results obtained using noise contaminated output data. At each pixel, we assume a Gaussian noise distribution, with mean 0 and variance

\[ \sigma_i^2 = \alpha \frac{y_i(t_i)}{t_i - t_{i-1}}, \]

[9], where \(\alpha\) is referred to as the noise level,

\[ y_i = y_i + \sqrt{\alpha \frac{y_i}{t_i - t_{i-1}}}. \]  

(3.21)

Here, we use an analytic form of the input function, obtained from [14]. We take a sample of 1000 runs to obtain average and standard deviation measures for \(k\) values, using both a Laplace (L) and Volterra (V) integral formulation (here \(K = K(X_L,V)\) denotes the matrix condition number) - see Table 5.

Table 5. Average and standard deviation measures computed over a sample of 1000 trials, to compare the results of the Laplace (L) and Volterra (V) formulations for the solution to the inverse problem. \(K = K(X_L,V)\) denotes the matrix condition number.

<table>
<thead>
<tr>
<th>(\alpha)</th>
<th>(k_1)</th>
<th>(k_2)</th>
<th>(k_3)</th>
<th>(k_4)</th>
<th>(K)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.020</td>
<td>.1300</td>
<td>.0620</td>
<td>.0068</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.01 L</td>
<td>.1029 ( \pm 2.8(-4) )</td>
<td>.1399 ( \pm 4.6(-4) )</td>
<td>.0678 ( \pm 7.3(-4) )</td>
<td>.0086 ( \pm 1.9(-4) )</td>
<td>1005</td>
</tr>
<tr>
<td>.01 V</td>
<td>.1029 ( \pm 2.8(-4) )</td>
<td>.1428 ( \pm 4.9(-4) )</td>
<td>.0702 ( \pm 7.8(-4) )</td>
<td>.0088 ( \pm 1.4(-4) )</td>
<td>1055</td>
</tr>
<tr>
<td>1.0 L</td>
<td>1012 ( \pm 0.036 )</td>
<td>.1256 ( \pm 0.0125 )</td>
<td>.0584 ( \pm 0.0118 )</td>
<td>.0047 ( \pm 0.0058 )</td>
<td>1014</td>
</tr>
<tr>
<td>1.0 V</td>
<td>1011 ( \pm 0.036 )</td>
<td>.1278 ( \pm 0.0129 )</td>
<td>.0602 ( \pm 0.0122 )</td>
<td>.0049 ( \pm 0.0057 )</td>
<td>1065</td>
</tr>
<tr>
<td>2.0 L</td>
<td>.0948 ( \pm 0.0098 )</td>
<td>.1060 ( \pm 0.0341 )</td>
<td>.0403 ( \pm 0.0220 )</td>
<td>-.0116 ( \pm 0.0130 )</td>
<td>1037</td>
</tr>
<tr>
<td>2.0 V</td>
<td>.0949 ( \pm 0.0098 )</td>
<td>.1082 ( \pm 0.0354 )</td>
<td>.0421 ( \pm 0.0229 )</td>
<td>-.0103 ( \pm 0.0126 )</td>
<td>1093</td>
</tr>
</tbody>
</table>

As expected, the relative errors of the estimates under noisy conditions, shown in Table 6, are larger than those under no noise, as presented in Table 4. In terms of accuracy, the Laplace double integral formulation gives the same results as the Volterra formulation. It is clear that as the noise level increases, the error in the estimates also increases, with estimates for \(k_1\) as the most robust ones under noise. \(k_2\) and \(k_3\) may have at most about one correct significant digit as the noise increases from 10\% to 20\%, while \(k_4\) under 20\% noise will have no significant correct digits in its estimation.

3. Numerical Algorithms

We propose a number of algorithms to solve the pixel estimation problem. Both background as well as new analysis are presented in the following sections.
with Levenberg-Marquardt approximations are widely used in the field \[16, 29, 45, 7, 36, 53\].

Newton-Gauss methods along with Levenberg-Marquardt approximations are widely used in parametric imaging because of their accuracy and statistical reliability. For one pixel the computational cost of nonlinear methods is competitive, however they have mostly been employed on regions of interest (ROI). Newton-Gauss methods along with Levenberg-Marquardt approximations are widely used in the field \[16, 29, 45, 7, 36, 53\].

Table 6. Relative error for the estimates presented in Table 5. Here, $\hat{k}$ represents the mean estimate of the exact solution vector $k$.

| $\alpha$ | $|k_3-k_1|/k_1$ | $|k_2-k_3|/k_2$ | $|k_3-k_3|/k_3$ | $|k_4-k_4|/k_4$ | $\|\hat{k}-k\|_{\infty}$ |
|----------|----------------|----------------|----------------|----------------|-----------------|
| .01      | L 0.0088 (±0.0027) | 0.0762 (±0.0035) | 0.0935 (±0.0118) | 0.2647 (±0.0279) | 0.0762 (±0.0035) |
|          | V 0.0088 (±0.0027) | 0.0985 (±0.0038) | 0.1323 (±0.0126) | 0.2941 (±0.0206) | 0.0985 (±0.0038) |
| .1       | L 0.0353 (±0.0078) | 0.0962 (±0.0338) | 0.1903 (±0.0581) | 0.8529 (±0.3088) | 0.0965 (±0.0335) |
|          | V 0.0353 (±0.0088) | 0.0992 (±0.0169) | 0.1968 (±0.029)  | 0.8382 (±0.2794) | 0.0992 (±0.0169) |
| .2       | L 0.0961 (±0.0706) | 0.2623 (±0.1846) | 0.3548 (±0.350)  | 2.7059 (±1.911)  | 0.2623 (±0.1846) |
|          | V 0.0961 (±0.0696) | 0.2723 (±0.1677) | 0.3694 (±0.321)  | 2.5147 (±1.852)  | 0.2723 (±0.1677) |

3.1. Nonlinear Minimization Algorithm. Traditionally, nonlinear approximations have been used in parametric imaging because of their accuracy and statistical reliability. For one pixel the computational cost of nonlinear methods is competitive, however they become impractical for high resolution image wide parameter estimation. Consequently, they have mostly been employed on regions of interest (ROI). Newton-Gauss methods along with Levenberg-Marquardt approximations are widely used in the field \[16, 29, 45, 7, 36, 53\].

We implement the Levenberg-Marquardt (LM) method using the analytic form for the output function in the calculation of the gradient. For the simulation study, a plasma time activity curve proposed in \[14\] is used to generate the input function,

$$u(t) = (A_1 t - A_2 - A_3)e^{\lambda t} + A_2 e^{\lambda_2 t} + A_3 e^{\lambda_3 t}. \quad (3.22)$$

Most studies \[15\], \[64\], \[16\] assume no noise in the input function. Although an input plasma curve obtained from clinical studies will have some noise, this is in general not as high as the noise present in the output tissue activity curve. For clinical data, one can perform a nonlinear fit for the input function, based on the functional form given above, to obtain a smooth version of the plasma curve. This form is then used to obtain a functional form for the output function.

For FDG kinetics, the Levenberg-Marquardt algorithm minimizes the functional

$$g(k) := \sum_{i=1}^{m} |(f(k) * u)(t_i) - y_i|^2 \quad (3.23)$$

where $k$ is the parameter vector of interest, $(t_i)$ is the sampling schedule, $u_i = u(t_i)$ are the input data, and $y_i = y(t_i)$ are the tissue activity data. The function $f(t_i; k)$ is given by (2.6).

We follow the Levenberg-Marquardt algorithm presented in \[35\]. Let $g : R^n \to R$. Assuming $g(k)$ is a twice continuously differentiable function, and given $k_0$ an approximation to the local minimum of $g$, then in a neighborhood of $k_0$ we have the approximation

$$g(k) = g(k_0) + (k - k_0)^T \nabla g(k_0) + \frac{1}{2}(k - k_0)^T g''(k_0)(k - k_0) + \text{higher order terms}, \quad (3.24)$$
where
\[ g''(k_0) = \left( \frac{\partial^2 g}{\partial k_j \partial k_l} \right)_{j,l=1,...,n} \]
denotes the Hessian matrix of \( g \). Minimizing the right hand side of the Taylor approximation above, ignoring the higher order terms, one finds a new approximation of the minimum of \( g \),
\[ k^{(1)} = k^{(0)} - [g''(k^{(0)})]^{-1} \nabla g(k^{(0)}). \] (3.25)
Here \( k^{(0)} = k_0 \) is the initial approximation. We compute the derivatives
\[ \frac{\partial g}{\partial k_j}(k) = 2 \sum_{i=1}^{m} [(f(k) \ast u)(t_i) - y_i] \left( \frac{\partial f}{\partial k_j} \ast u \right)(t_i), \]
\[ \frac{\partial^2 g}{\partial k_j \partial k_l}(k) = 2 \sum_{i=1}^{m} \left[ \left( \frac{\partial f}{\partial k_j} \ast u \right)(t_i) \left( \frac{\partial f}{\partial k_l} \ast u \right)(t_i) + [(f(k) \ast u)(t_i) - y_i] \left( \frac{\partial^2 f}{\partial k_j \partial k_l} \ast u \right)(t_i) \right]. \]
The second order derivatives of the function \( f(k) \) are multiplied by the factor \((f(k) \ast u)(t_i) - y_i)\) which will become small in the process of minimizing \( g \). Hence, this term is ignored, and we form the matrix entries
\[ a_{jl} = 2 \sum_{i=1}^{m} \left( \frac{\partial f}{\partial k_j} \ast u \right)(t_i) \left( \frac{\partial f}{\partial k_l} \ast u \right)(t_i). \] (3.26)
The Levenberg-Marquardt combines the Newton iteration \( A(k^{(1)} - k^{(0)}) = -\nabla g(k_0) \) with the steepest descent method, \( k^{(1)} - k^{(0)} = -\lambda M \nabla g(k^{(0)}) \), with \( M \) a positive definite matrix, by introducing the \( n \times n \) matrix \( \tilde{A} \) with entries
\[ \tilde{a}_{jj} = (1 + \gamma)a_{jj}, \tilde{a}_{jl} = a_{jl}, j \neq l. \] (3.27)
Here \( \gamma \) is a positive parameter. Then the iteration becomes
\[ \tilde{A}(k^{(1)} - k^{(0)}) = b, \] (3.28)
where \( b = -\nabla g(k^{(0)}) \). For large values of \( \gamma \), the matrix \( \tilde{A} \) becomes diagonally dominant, and the iteration is close to a steepest descent. For small values of \( \gamma \) the iteration will turn into a Newton step.

Note, for PET data, the values \( y_i \) are in error, \( y_i = \bar{y}_i + e_i \) for each \( i = 1, ..., m \). The matrix \( \tilde{A} \) is in error only due to the estimates \( k \) and to possible errors in the input function \( u(t) \). Assuming no errors in the input curve, at each iteration we are actually solving
\[ \tilde{A}(k^{(1)} - k^{(0)}) = \tilde{b} + \epsilon \] (3.29)
where \( \epsilon_j = -2 \sum_{i=1}^{m} e_i \left( \frac{\partial f}{\partial k_j} \ast u \right)(t_i) \), \( j = 1, ..., n \). If \( \epsilon \) is a vector of 0 mean and \( \sigma^2 \) variance, the least squares solver provides the best unbiased solution to this problem, [4].
3.2. The Linear Least Squares Method - LS. The least squares method solves the minimization problem
\[ \theta_{ls} = \arg\min \| X\theta - Y \|^2_2. \] (3.30)

**Theorem 5 (Gauss-Markoff Theorem)** (Bjorck, [4]) Given the linear system \( X\theta = \hat{Y} \) where \( X \in \mathbb{R}^{m \times n} \) is a known matrix of full rank \( n \) and \( \hat{Y} = Y + \epsilon \) where \( \epsilon \) is a random vector with mean 0 and variance \( \sigma^2 \), then the best unbiased estimator is the least squares solution
\[ \theta_{ls} = (X^TX)^{-1}X^T\hat{Y}. \] (3.31)

Note this theorem guarantees a best unbiased estimator only in the case where the matrix \( X \) is not in error, or at most, the errors in \( X \) are independent. The errors in \( X_L \) and \( X_V \) are not independent, as one can see from the error analysis in Section 2. The parameters estimated from this method will consequently be biased.

We write explicitly the problem for the case when \( X \) is in error, \( (X + \Delta X)\theta = Y \), where \( \|\Delta X\| \leq \epsilon \) in some norm \( \|\cdot\| \).

We want to know how close is the computed least squares solution \( \hat{\theta}_{ls} \) to the exact solution \( \theta_{ls} \) from (3.8).

**Theorem 6 (Wedin)** (Higham, [28])
Let \( X \in \mathbb{R}^{m \times n} \) \((m \geq n)\) and \( X + \Delta X \) both be of full rank, and let
\[ \|Y - X\theta\|_2 = \min r = Y - X\theta, \]
\[ \|Y - (X + \Delta X)\hat{\theta}\|_2 = \min s = Y - (X + \Delta X)\hat{\theta}, \]
\[ \|\Delta X\|_2 \leq \epsilon\|X\|_2. \] (3.32)

Then, given that \( k_2(X)\epsilon < 1 \), we have
\[ \frac{\|\theta - \hat{\theta}\|_2}{\|\theta\|_2} \leq \frac{k_2(X)\epsilon}{1 - k_2(X)\epsilon} \left( 2 + (1 + k_2(X))\frac{\|r\|_2}{\|X\|_2}\frac{\|r\|_2}{\|\theta\|_2} \right), \] (3.33)
\[ \frac{\|r - s\|_2}{\|Y\|} \leq (1 + 2k_2(X))\epsilon. \] (3.34)

These bounds are approximately attainable.

We can interpret the result in (3.33) as saying that the sensitivity of the least squares solution is measured by the condition number \( k_2(X) \) if the residual is small, or by \( k_2(X)^2 \) otherwise.

One can solve the LS problem simply by solving the normal equations \( X^TX\theta_{ls} = X^TY \). However the normal equations give a computed solution \( \theta_{ls} \) whose relative error depends on the square of the condition number [28]. The condition number \( k_2(X) \) of the matrix \( X \) in our problem is on the order of \( 10^3 \). To provide a backwards stable solution to this least squares problem we use the QR method. In this case, the residual is small, i.e.
the QR method solves a nearby problem, and the sensitivity of the least squares solution is roughly proportional to $k^2(X)$. Note, the least squares problem $X\theta = Y$ minimizes the residual $f(\theta) = \|X\theta - Y\|_2$ with respect to $\theta$. However, $\theta$ depends on $k$ non-linearly, $\theta = \theta(k)$ and we want to find $k$ such that $f(\theta(k))$ is minimum.

**Lemma 11** The least squares solution $\theta_{ls}$ in (3.30) solves $\nabla_k f(k) = 0$.

**Proof:** Minimizing $f(\theta) = \|X\theta - Y\|_2^2$ in terms of $\theta$ gives

$$\nabla_\theta f(\theta) = 2X^T(X\theta - Y).$$

Minimizing $f$ with respect to $k$ gives

$$\nabla_k f(k) = 2JX^T(X\theta - Y),$$

where $J$ is the Jacobian

$$J = \begin{bmatrix}
0 & -1 & -1 & -1 \\
0 & -k_4 & 0 & -k_2 \\
1 & 0 & 0 & 0 \\
k_3 + k_4 & 0 & k_1 & k_1
\end{bmatrix}.$$ (3.35)

Then $\det J = -k_1k_2$ and as long as $k_1, k_2 \neq 0$, $J$ is nonsingular. Then $\nabla_k f(k) = J\nabla_\theta f(\theta)$, which implies $J^{-1}\nabla_k f(k) = \nabla_\theta f(\theta)$. Hence $\nabla_\theta f(\theta) = 0$ implies $\nabla_k f(k) = 0$. $\square$

When the vector $[a_1, a_2, b_1, b_2]$ is estimated, following (1.3),

$$[k_1, k_2, k_3, k_4] = [b_1, a_1 - b_2/b_1, a_1 - k_2 - k_4, a_2/k_2].$$

Due to noise, it is possible to obtain negative values for $k_i$ even if $a_1, a_2, b_1, b_2$ are all positive. In particular, we see that if $a_1 < b_2/b_1$ then $k_2$ will be negative. Similarly, $k_3$ can be negative if $a_1 < k_2 + k_4$.

Explicitly, the linear constraint on the parameters $k_i$, $0 < k_i < 1$ for all $i = 1, \ldots, 4$ translates into linear and nonlinear constraints for $a_1, a_2, b_1, b_2$:

$$0 < b_1 < 1, \quad b_2 < a_1b_1, \quad b_1(a_1 - 1) < b_2,$$

$$b_1a_2 < b_2(a_1 - b_2/b_1), \quad b_1b_2(a_1 + 1) - b_2^2 < b_1^2(a_1 + a_2),$$

$$a_2 > 0, \quad b_2 < b_1(a_1 - a_2).$$ (3.36)

These nonlinear constraints contribute to the difficulty of the problem, and suggest the use of constrained methods in the estimation problem.

A variant of the least squares algorithm is to solve the constrained least squares problem

$$\min_\theta \|X\theta - Y\|_2^2 \text{ subject to } \delta_{\min} \leq \theta \leq \delta_{\max}$$ (3.37)
for some a priori known bounds $\delta_{\min}, \delta_{\max}$, usually determined from some physical constraint on the problem or from prior results. Several methods are available for solving this problem. We implement this method via a Matlab routine which is described as a subspace trust region method based on the interior-reflective Newton’s method. We denote this method by CLS - constrained least squares. Alternatively, if one only considers the positivity constraint on the parameters $\theta_i, i = 1, ..., 4$, then the nonnegative least squares (NNLS) method can be used. This is again implemented using the Matlab routine \textit{lsqnonneg}.

3.3. The Total Least Squares Method - TLS. A more general model \cite{19} would account for errors in the system matrix, $X$, thus solving the problem

$$(X + E)\theta = Y + \epsilon. \tag{3.38}$$

The errors in both $E$ and $\epsilon$ are assumed to have mean 0 and variance $\sigma^2$. This is called the total least squares (TLS) problem. Note, it can be written in an alternative form as

$$(\tilde{X}, \tilde{Y})(\theta, -1)^T = 0. \tag{3.39}$$

where $\tilde{X} = (X + E), \tilde{Y} = Y + \epsilon$. The matrix $(\tilde{X}, \tilde{Y})$ is now rank deficient, with the right singular vector $(\theta, -1)^T$ corresponding to a 0 singular value. The TLS problem is then to find a perturbation matrix $(E, \epsilon)$ with a minimum Frobenius norm which lowers the rank of the matrix $(X, Y)$. Hence, the TLS solution is given by solving a minimization problem with constraint

$$\arg\min \| (E, \epsilon) \|_F, \quad (X + E)\theta = Y + \epsilon \tag{3.40}$$

where $\| \cdot \|_F$ is the Frobenius norm. The TLS solution can be obtained via singular value decomposition (SVD).

**Theorem 7** (Golub, \cite{19})

Let $(X, Y) = U\Sigma V^T, \Sigma = \text{diag}(\sigma_1, \ldots, \sigma_{n+1})$ with $U^T U = I_m, V^T V = I_{n+1}$, and

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{n+1} \geq 0 \tag{3.41}$$

be the singular values of $(X, Y)$. Let the singular values of $X$ be

$$\hat{\sigma}_1 \geq \hat{\sigma}_2 \geq \ldots \geq \hat{\sigma}_n > 0.$$

If $X$ has full rank and $\hat{\sigma}_n > \sigma_{n+1}$ then the TLS problem has a unique solution.

This is a standard result, and a proof of this can be found, for example, in \cite{58}, \cite{19}. If the system matrix $X$ does not have full rank, then the TLS solution is not unique. Van Huffel and Vandewalle \cite{58} claim that in typical applications an improvement of 10-15% in accuracy is observed in the solution given by TLS method as compared to LS. Unfortunately, the TLS problem is always worse conditioned then the LS problem \cite{4}. The TLS solution can also be found iteratively by considering the problem as non-linear minimization, in which case it is easier to impose a constraint.
3.4. The Generalized Linear Least Squares Algorithm - GLLS. The GLLS algorithm is introduced in [13]. GLLS is an extension of the generalized least squares (GLS) algorithm to non-uniformly sampled data.

Although GLLS has been used successfully in studies [13, 15, 9], there is no analytic proof of convergence of this algorithm. In addition, despite its derivation as an iterative algorithm, GLLS has mostly been used as a one step iteration - see for example [13], [15]. We briefly describe the method and discuss its convergence behaviour.

In the derivation of the Laplace formulation, equation (3.8) does not explicitly account for noise. To reflect the measurement noise $\epsilon$ we write

$$y = X\theta + \epsilon, \quad (3.42)$$

with $y \in \mathbb{R}^{m \times 1}$, $X \in \mathbb{R}^{m \times 2n}$, and $\theta \in \mathbb{R}^{2n \times 1}$. The least squares solution obtained from (3.42) will be biased. This is easily observed when expressing the measurement noise in $y(t)$ in the Laplace formulation

$$Y(s) = \frac{B(s)}{A(s)} U(s) + \frac{V(s)}{A(s)} + E(s), \quad (3.43)$$

where $E(s)$ is the Laplace transform of the white noise. GLLS attempts to whiten the noise by multiplying both sides by $\frac{A(s)}{A(s)}$ where $\hat{A}(s)$ is an estimate to $A(s)$.

$$\frac{A(s)}{A(s)} Y(s) = \frac{B(s)}{A(s)} U(s) + \frac{V(s)}{A(s)} + \frac{A(s)}{A(s)} E(s). \quad (3.44)$$

In the limit, if $\hat{A}(s) \to A(s)$, then the noise in (3.43) will be uncorrelated, and the least squares solution will be the best unbiased estimate.

Since PET data is non-uniformly sampled, it’s important to derive computable quantities $\frac{A(s)}{A(s)}$, $\frac{B(s)}{A(s)}$, and $\frac{V(s)}{A(s)}$. For simplicity of the derivation of these quantities, we ignore the last two terms in (3.44) for the following calculations. First note that

$$A(s) - \hat{A}(s) = \sum_{i=1}^{n} (a_i s^{n-i} - \hat{a}_i s^{n-i})$$

which implies

$$\frac{A(s)}{A(s)} Y(s) = Y(s) - \sum_{i=1}^{n} \hat{a}_i \frac{s^{n-i}}{A(s)} Y(s) + \sum_{i=1}^{n} a_i \frac{s^{n-i}}{A(s)} Y(s). \quad (3.45)$$

If we substitute this in (3.44) and separate the estimates $\hat{a}_i$ from $a_i$ we obtain

$$Y(s) - \sum_{i=1}^{n} \frac{\hat{a}_i s^{n-i}}{A(s)} Y(s) = - \sum_{i=1}^{n} a_i \frac{s^{n-i}}{A(s)} Y(s) + \sum_{i=1}^{n} \frac{b_i s^{n-i}}{A(s)} U(s) + \sum_{i=1}^{n} v_i \frac{s^{n-i}}{A(s)}. \quad (3.46)$$

If $\lambda_i$ are distinct nonzero roots of the polynomial $\hat{A}(s)$, then

$$\mathcal{L}^{-1}\left(\frac{s^{k}}{A(s)}\right) = \sum_{i=1}^{n} \frac{\lambda_i^k}{A(\lambda_i)} e^{\lambda_i t}. \quad (3.47)$$
Let \( \psi_i(t) = \frac{e^{\lambda_i t}y(t)}{A(\lambda_i)^T} \) and \( v_i(t) = \frac{e^{\lambda_i t}u(t)}{A(\lambda_i)^T} \). Assuming all initial conditions are 0, the inverse Laplace transform of (3.46) becomes

\[
g(t) - \sum_{j=1}^{n} \tilde{a}_j \sum_{i=1}^{n} \lambda_i^{n-j} \psi_i(t) = -\sum_{j=1}^{n} a_j \sum_{i=1}^{n} \lambda_i^{n-j} \psi_i(t) + \sum_{j=1}^{n} b_j \sum_{i=1}^{n} \lambda_i^{n-j} v_i(t). \tag{3.48}
\]

For the discrete case, this can be written as a matrix equation \( r = Z \theta \). For the case of repeated eigenvalues see [64].

For the initial estimate \( \hat{A}(s) \), GLLS uses the least squares solution obtained in (3.8). A general formulation of the GLLS algorithm follows. Note, for a \( n \)th order differential equation, we have \( 2n \) parameters to estimate, assuming zero initial conditions. For ease of notation, let \( 2n = p \). Then the coefficient matrix \( X \) in (3.7) will have size \( m \times p \), with \( m > p \).

**GLLS Algorithm:**

Given \( \theta^{(0)} = \theta_{LS}, \theta^{(0)} = [-\tilde{a}_1, -\tilde{a}_2, \ldots, -\tilde{a}_n, \hat{b}_1, \hat{b}_2, \ldots, \hat{b}_n]^T \)

\[
= [-a_1^{(0)}, -a_2^{(0)}, \ldots, -a_n^{(0)}, b_1^{(0)}, b_2^{(0)}, \ldots, b_n^{(0)}]^T
\]

for \( k = 1, 2, \ldots \),

find \( \lambda_i = \lambda_i^{(k-1)} \) for \( i = 1, \ldots, n \) roots of \( \hat{A}(s) = 0 \),

where \( A(k - 1)(s) = \hat{A}(s) = s^n + a_1^{(k-1)} s^{n-1} + \ldots + a_n^{(k-1)} \);

if \( \lambda_i \neq \lambda_j \) for all \( i, j = 1, \ldots, n \) then

calculate \( A'(\lambda_i) \)

for \( l = 1, \ldots, m \)

\[
z_l = \left[ \sum_{i=1}^{n} \lambda_i^{n-1} \psi_i(t_l), \ldots, \sum_{i=1}^{n} \psi_i(t_l), \sum_{i=1}^{n} \lambda_i^{n-1} v_i(t_l), \ldots, \sum_{i=1}^{n} v_i(t_l) \right];
\]

\[
r_l = \left[ Y_l - \sum_{j=1}^{n} a_j^{(k-1)} \sum_{i=1}^{n} \lambda_i^{n-j} \psi_i(t_l) \right];
\]

end;

form \( Z = (z_l); r = (r_l) \);

solve \( Z \theta^{(k)} = r \) for \( \theta^{(k)} \);

end

Note if \( \lambda_i = \lambda_j \) for all \( i, j \), then \( Z \) is not a full rank matrix. In fact a different derivation of the inverse Laplace formulation is necessary in this case, [64].

Here \( Y \) is the vector of discrete output measurements, and \( Y_l \) is the \( l^{th} \) entry in \( Y \). The stopping criterion is based on whether the relative error has reached a set tolerance, \( ||\theta^{(k)} - \theta^{(k-1)}|| < tol \) where \( tol \) is given.

We introduce the following notation: \( \psi_i(t_l) = \psi_{li} \) and similarly \( v_i(t_l) = v_{li} \). Under this notation, and using the fact that \( \theta = [-a_1, \ldots, -a_n, b_1, \ldots, b_n]^T \), each entry in \( r \) can be rewritten as
Let $\Lambda = [\lambda_1, \lambda_2, \cdots, \lambda_n]^T$, $\Lambda \in \mathbb{R}^{n \times 1}$. Then $\Lambda^q = [\lambda_1^q, \lambda_2^q, \cdots, \lambda_n^q]^T$ for $q = 0, 1, \ldots, n - 1$. Let $\Psi_i$ be the row vector of entries $\psi_{ij}$, $\Psi_i = [\psi_{i1}, \psi_{i2}, \ldots, \psi_{in}]$. Then $\sum_{j=1}^{n} \lambda_j^{n-1} \psi_{ij} = \Psi_i \Lambda^{n-1}$. We form matrix $\Psi = (\psi_{ij})$, $\Psi \in \mathbb{R}^{m \times n}$. Similarly, let $\Upsilon_i$ be the row vector of entries $\upsilon_{ij}$, $\Upsilon_i = [\upsilon_{i1}, \upsilon_{i2}, \ldots, \upsilon_{in}]$. We form matrix $\Upsilon = (\upsilon_{ij})$, $\Upsilon \in \mathbb{R}^{m \times n}$. Also, let

$$
\begin{align*}
\alpha &= [\theta_1, \theta_2, \ldots, \theta_n]^T = [-a_1, -a_2, \ldots, -a_n]^T, \alpha \in \mathbb{R}^{n \times 1}, \\
\beta &= [\theta_{n+1}, \theta_{n+2}, \ldots, \theta_{2n}]^T = [b_1, b_2, \ldots, b_n]^T, \beta \in \mathbb{R}^{n \times 1}, \\
V &= [\Lambda^{-1}, \Lambda^{n-2}, \ldots, \Lambda^0], V \in \mathbb{R}^{n \times n}, V \text{ is Vandermonde matrix,} \\
C &= [\Psi \Lambda^{-1}, \Psi \Lambda^{n-2}, \ldots, \Psi \Lambda^0] \\
&= \Psi V, C \in \mathbb{R}^{m \times n}, \\
D &= [\Upsilon \Lambda^{-1}, \Upsilon \Lambda^{n-2}, \ldots, \Upsilon \Lambda^0] \\
&= \Upsilon V, D \in \mathbb{R}^{m \times n}.
\end{align*}
$$

Thus we can rewrite $r$ as

$$r = Y + \theta_1 \Psi \Lambda^{-1} + \theta_2 \Psi \Lambda^{n-2} + \cdots + \theta_n \Psi \Lambda^0$$

$$= Y + [\Psi \Lambda^{-1}, \Psi \Lambda^{n-2}, \ldots, \Psi \Lambda^0] \alpha$$

$$= Y + C \alpha$$

$$= Y + \Psi V \alpha. \quad (3.50)$$

Note matrix $Z$

$$Z = [\Psi \Lambda^{-1}, \Psi \Lambda^{n-2}, \ldots, \Psi \Lambda^0, \Upsilon \Lambda^{-1}, \Upsilon \Lambda^{n-2}, \ldots, \Upsilon \Lambda^0]$$

$$= [C, D]$$

$$= [\Psi, \Upsilon] V, \text{ where } Z \in \mathbb{R}^{m \times 2n}.$$ 

In one step of the GLLS iteration we solve $r = Z\theta$ in the least squares sense, where $r = r(\theta^{(k-1)})$, $Z = Z(\theta^{(k-1)})$ and $\theta = \theta^{(k)}$, while $k$ here is the iteration number, i.e., we solve the least squares problem for $Z\theta = r$,

$$\sum_{i=1}^{n} \theta_i^{(k)} \Psi \Lambda^{-i} + \sum_{i=n+1}^{2n} \theta_i^{(k)} \Upsilon \Lambda^{2n-i} = Y + \sum_{i=1}^{n} \theta_i^{(k-1)} \Psi \Lambda^{n-i}. \quad (3.51)$$

Under the notation introduced, this is the same as solving the least squares problem for

$$[C, D] \begin{bmatrix} \alpha^{(k)} \\ \beta^{(k)} \end{bmatrix} = Y + [C, 0] \begin{bmatrix} \alpha^{(k-1)} \\ \beta^{(k-1)} \end{bmatrix}, \text{ or}$$

$$[\Psi, \Upsilon] V \begin{bmatrix} \alpha^{(k)} \\ \beta^{(k)} \end{bmatrix} = Y + [\Psi, 0] V \begin{bmatrix} \alpha^{(k-1)} \\ \beta^{(k-1)} \end{bmatrix},$$
where $\Psi = \Psi(\alpha^{(k-1)})$, $\Upsilon = \Upsilon(\alpha^{(k-1)})$ and $V = V(\alpha^{(k-1)})$. In particular, note that the left hand side of this equation is only dependent on the first $n$ entries of the solution vector $\theta$ at step $(k-1)$, which are given by $\alpha^{(k-1)}$.

We introduce the square of the residual

$$S(\theta) := ||Z\theta - r||_2^2 = \sum_{i=1}^m (Z\theta - r)_i^2.$$ 

(3.52)

In GLLS, at each iteration step we solve a least squares problem,

$$\min_{\theta \in \mathbb{R}^n} \|Z\theta - r\|_2,$$

therefore, we solve $\nabla_\theta S(\theta) = 0$. Traditionally, the least squares problem has fixed matrix entries in $Z$, and also in the vector $r$. Then, the solution to the minimization problem as written in (3.52), assuming full column rank for $Z$, is given by

$$\theta_{ls}^{(k)} = (Z^T Z)^{-1} Z^T r.$$ 

(3.52)

Consider explicitly the normal equations for the least squares problem, $(Z^T Z)\theta_{ls} = Z^T r$, under the notation introduced,

$$
\begin{bmatrix}
C^T \\
D^T \\
\end{bmatrix}
\begin{bmatrix}
C & D \\
\end{bmatrix}\theta_{ls}^{(k)} = 
\begin{bmatrix}
C^T \\
D^T \\
\end{bmatrix}
\begin{bmatrix}
Y + C\alpha_{ls}^{(k-1)} \\
\end{bmatrix} \Rightarrow 
\begin{bmatrix}
C^T C & C^T D \\
D^T C & D^T D \\
\end{bmatrix}
\begin{bmatrix}
\alpha^{(k)} \\
\beta^{(k)} \\
\end{bmatrix} = 
\begin{bmatrix}
C^T \\
D^T \\
\end{bmatrix}
\begin{bmatrix}
Y + C\alpha^{(k-1)} \\
\end{bmatrix}.
$$

(3.53)

Hence, we have

$$
C^T C\alpha^{(k)} + C^T D\beta^{(k)} = C^T Y + C^T C\alpha^{(k-1)},
$$

$$
D^T C\alpha^{(k)} + D^T D\beta^{(k)} = D^T Y + D^T C\alpha^{(k-1)}.
$$

(3.54)

At convergence, $\lim_{k \to \infty} \alpha^{(k)} = \alpha$, and we must have

$$
C^T D\beta = C^T Y, \text{ and } \quad D^T D\beta = D^T Y.
$$

(3.55)

This is equivalent to requiring $C^T \left[ I_m - D(D^T D)^{-1} D^T \right] Y = 0$, assuming $(D^T D)^{-1}$ exists. Given that $C = \Psi V$, and $D = \Upsilon V$, where $V$ is a Vandermonde matrix, hence nonsingular for $\lambda_i \neq \lambda_j$ (as previously assumed), we obtain

$$
\Psi^T \left[ I_m - \Upsilon(\Upsilon^T \Upsilon)^{-1} \Upsilon^T \right] Y = 0.
$$

The assumption that $(D^T D)^{-1}$ exists reduces to assuming $\Upsilon^T \Upsilon$ is invertible, which is reasonable considering the entries of the matrix $\Upsilon$. Note, $P_T = I_m - \Upsilon(\Upsilon^T \Upsilon)^{-1} \Upsilon^T$ is the orthogonal projection onto $\text{ran}(\Upsilon)^\perp$. Then $P_T Y$ is in the ran $(Y)^\perp$, therefore $\Upsilon^T (P_T Y) = 0$. But since $\Psi^T (P_T Y) = 0$, we have that $P_T Y$ is also in the ran$(\Psi)^\perp$. Using the fact that
ran(Ψ) = null(ΨT), \( \text{ran}(\Psi) = \text{null}(\Psi^T) \), we conclude that \( \text{null}(\Psi^T) \cap \text{null}(\Psi^T) \neq 0 \), in particular \( P_Y \in \text{null}(\Psi^T) \cap \text{null}(\Psi^T) \). Hence, at convergence, \( \beta = V^{-1}(\Psi^T)^{-1} \Psi^T Y \), \( \beta = \beta(\alpha) \).

We have thus proved the following result:

**Theorem 8** Assuming \((\Psi^T)^{-1}\) exists, the necessary conditions for the convergence of the GLLS algorithm are:

1. \( \beta_{\text{GLLS}} = \beta \) where \( \beta = V^{-1}(\Psi^T)^{-1} \Psi^T Y \), and

2. \( P_Y \in \text{null}(\Psi^T) \cap \text{null}(\Psi^T) \)

where \( \Psi = \Psi(\alpha) \) and \( \Psi = \Psi(\alpha) \) are fixed matrices, with \( \alpha \) as the limit \( \lim_{k \to \infty} \alpha_k = \alpha \).

Note, if \( \text{null}(\Psi^T) \neq 0 \), then \( \Psi^T \Psi \) is singular, and our assumption is contradicted, unless \( P_Y = 0 \) at convergence, and \( \text{null}(\Psi^T) = 0 \).

Consider now the original problem in Laplace domain, \((3.43)\). We seek a solution vector \( \theta \) which minimizes the error

\[
E(s) = Y(s) - \frac{B(s)}{A(s)} U(s).
\]

Applying the inverse Laplace transform, and rewriting the resulting problem as a minimization in the 2-norm we obtain

\[
\min_{\theta = [\alpha, \beta]} \|Y - D(\alpha)\beta\|_2. \tag{3.56}
\]

This can also be obtained by noting that

\[
S(\theta) = \sum_{l=1}^{m} (Z \theta - r)_l^2
\]

\[
= \sum_{l=1}^{m} \left( [C, D] \begin{bmatrix} \alpha \\ \beta \end{bmatrix} - (Y + C\alpha) \right)_l^2
\]

\[
= \sum_{l=1}^{m} (D\beta - Y)_l^2
\]

\[
= \|D\beta - Y\|_2^2,
\]

where \( D = D(\alpha) \). Since \( D(\alpha) = \Psi(\alpha)V(\alpha) \), we have

\[
\frac{\partial S}{\partial \alpha_i} = 2 \sum_{l=1}^{m} \left( \frac{\partial}{\partial \alpha_i} \Psi \right)_l V \beta + \Psi_l \left( \frac{\partial}{\partial \alpha_i} V \right) \beta \right) (D\beta - Y)_l
\]

\[
= 2 \left[ \left( \frac{\partial}{\partial \alpha_i} \Psi \right)_l V \beta + \Psi_l \left( \frac{\partial}{\partial \alpha_i} V \right) \beta \right]^T (\Psi^T \beta - Y),
\]

\[
\frac{\partial S}{\partial \beta_i} = 2 \sum_{l=1}^{m} (d_l)(D\beta - Y)_l
\]

\[
= 2D_i^T \left[ D\beta - Y \right],
\]
where \( D_l \) is the \( i^{th} \) column of \( D \), and

\[
\frac{\partial}{\partial \alpha_i} \mathcal{Y} = \begin{bmatrix}
\frac{\partial \mathcal{Y}_{I1}}{\partial \alpha_i}, & \frac{\partial \mathcal{Y}_{I2}}{\partial \alpha_i}, & \ldots, & \frac{\partial \mathcal{Y}_{In}}{\partial \alpha_i}
\end{bmatrix},
\]

\[
\frac{\partial}{\partial \alpha_i} \mathbf{V} = \begin{bmatrix}
\frac{\partial \lambda_1}{\partial \alpha_i}, & \frac{\partial \lambda_2}{\partial \alpha_i}, & \ldots, & \frac{\partial \lambda_{n-1}}{\partial \alpha_i}, 0
\\
\frac{\partial \lambda_1}{\partial \alpha_i}, & \frac{\partial \lambda_2}{\partial \alpha_i}, & \ldots, & \frac{\partial \lambda_{n-2}}{\partial \alpha_i}, 0
\\
\vdots & & & \\
\frac{\partial \lambda_1}{\partial \alpha_i}, & \frac{\partial \lambda_2}{\partial \alpha_i}, & \ldots, & \frac{\partial \lambda_{n}}{\partial \alpha_i}, 0
\end{bmatrix} \in \mathbb{R}^{n \times n}.
\]

Therefore, \( \nabla_\beta S = 2 \mathbf{V}^T \mathbf{Y} (\mathbf{Y} \mathbf{V}^T \beta - \mathbf{Y}) \). The expression for \( \frac{\partial S}{\partial \alpha_i} \) is more complicated. Note

\[
\frac{\partial}{\partial \alpha_i} \mathcal{Y} = \begin{bmatrix}
\text{diag} \left( t_1, \ldots, t_m \right) \mathbf{Y} - \mathbf{Y} \text{ diag} \left( f_1, \ldots, f_n \right) \end{bmatrix} \begin{bmatrix}
\frac{\partial \lambda_1}{\partial \alpha_i}, & \ldots, & \frac{\partial \lambda_n}{\partial \alpha_i}
\end{bmatrix}, \quad (3.57)
\]

where

\[
f_j = \frac{\partial}{\partial \lambda_j} A' (\lambda_j) = \frac{n(n-1)\lambda_j^{n-2} + \sum_{i=1}^{n-2} (n-i)(n-i-1)\alpha_i \lambda_j^{n-i-2}}{n\lambda_j^{n-1} + \sum_{i=1}^{n-1} (n-i)\alpha_i \lambda_j^{n-i-1}}.
\]

Similarly,

\[
\frac{\partial}{\partial \alpha_i} \mathbf{V} = \text{diag} \begin{bmatrix}
\frac{\partial \lambda_1}{\partial \alpha_i}, & \ldots, & \frac{\partial \lambda_n}{\partial \alpha_i}
\end{bmatrix} \mathbf{V}_{n-1} \text{ diag} \begin{bmatrix}
n-1, n-2, \ldots, 1, 0
\end{bmatrix},
\]

where

\[
\mathbf{V}_{n-1} = \begin{bmatrix}
\lambda_1^{n-2}, & \lambda_1^{n-3}, & \ldots, & 1, 0
\\
\lambda_2^{n-2}, & \lambda_2^{n-3}, & \ldots, & 1, 0
\\
\vdots & & & \\
\lambda_n^{n-2}, & \lambda_n^{n-3}, & \ldots, & 1, 0
\end{bmatrix} \in \mathbb{R}^{n \times n}.
\]

Let \( L_i = \begin{bmatrix}
\frac{\partial \lambda_1}{\partial \alpha_i}, & \ldots, & \frac{\partial \lambda_n}{\partial \alpha_i}
\end{bmatrix}, T = \begin{bmatrix}
t_1, \ldots, t_m
\end{bmatrix} \) and \( F_i = \begin{bmatrix}
f_1, \ldots, f_m
\end{bmatrix}. \)

Unfortunately, the nonlinearities with respect to \( \alpha_i \), for all \( i \), persist even under further expansion of terms,

\[
\frac{\partial S}{\partial \alpha_i} = 2 \begin{bmatrix} (T \mathbf{Y} - \mathbf{Y} F_i) L_i \mathbf{V}^T \beta + \mathbf{Y} L_i \mathbf{V}_{n-1} \end{bmatrix} \begin{bmatrix}
n-1, n-2, \ldots, 1, 0 \end{bmatrix} \beta^T (\mathbf{Y} \mathbf{V}^T \beta - \mathbf{Y}).
\]

We let \( J \in \mathbb{R}^{m \times n} \) be the matrix whose columns are of the form

\[
J_i = \begin{bmatrix} (T \mathbf{Y} - \mathbf{Y} F_i) L_i \mathbf{V}^T \beta + \mathbf{Y} L_i \mathbf{V}_{n-1} \end{bmatrix} \begin{bmatrix}
n-1, n-2, \ldots, 1, 0 \end{bmatrix} \beta^T.
\]

Then \( \nabla_\theta S = 2 [J, D]^T (D \beta - \mathbf{Y}) \). Note, the nonlinear minimization of \( S(\theta) \) with respect to \( \alpha \) is independent of the matrix \( \Psi \).
Consider the original problem of minimizing $S(\theta) = \|D(\alpha)\beta - Y\|_2^2$. GLLS considers as the initial solution the least squares estimate $\theta_{ls}$, given in (3.31). Thus, given $\theta_{ls}$, one could approximate $s(\theta) := D(\alpha)\beta - Y$ linearly in a neighborhood of this given point $\theta_{ls}$, by

$$\tilde{s}(\theta) = s(\theta_{ls}) + [J, D](\theta - \theta_{ls}).$$

One could further solve this via a linear least squares minimization, [4],

$$\min_\theta \| (D(\alpha_{ls})\beta_{ls} - Y) + [J, D](\theta - \theta_{ls}) \|_2^2$$

to obtain an improved solution $\theta$. The normal equations for this problem are

$$\theta = \theta_{ls} - \left( [J, D]^T [J, D] \right)^{-1} [J, D]^T(D\beta_{ls} - Y),$$

(3.58)

where on this right hand side, $D = D(\alpha_{ls}), J = J(\theta_{ls}), \beta = \beta_{ls}$. The necessary condition for $\theta = \theta^* = [\alpha^*, \beta^*]$ to be local minimum for $S$ is

$$\nabla_\theta S(\theta^*) = 2[J, D]^T(\theta^*)(D(\alpha^*)\beta^* - Y) = 0.$$ 

We can decouple (3.58) into

$$\alpha = \alpha_{ls} - (J^T J)^{-1} J^T(D\beta_{ls} - Y)$$

$$\beta = \beta_{ls} - (D^T D)^{-1} D^T(D\beta_{ls} - Y)$$

$$= (D^T D)^{-1} D^T Y.$$

Compare this with the expression derived for the GLLS algorithm in (3.54), where we now consider only the first iteration of GLLS, $k = 1$, where $\alpha^{(k-1)} = \alpha_{ls}$ and $\beta^{(k-1)} = \beta_{ls}$, and we let $\alpha = \alpha^{(1)}$,

$$\alpha = \alpha_{ls} - (C^T C)^{-1} C^T(D\beta_{ls} - Y),$$

$$\beta = (D^T D)^{-1} D^T C(\alpha_{ls} - \alpha) + (D^T D)^{-1} D^T Y.$$

One would deduce that, when solving for $\alpha$, GLLS attempts to replace the Jacobian $J$ with the matrix $C$. However, the entries derived for $J$ do not correspond, analytically, to those of $C$. For $\beta$, GLLS corrects the solution obtained through the nonlinear minimization, ie $\beta = (D^T D)^{-1} D^T Y$, by the term $(D^T D)^{-1} D^T C(\alpha_{ls} - \alpha)$. Note, a necessary condition for the convergence of GLLS is that $\beta = (D^T D)^{-1} D^T Y$ in the limit. We conclude that the GLLS algorithm is close to a Gauss-Newton type method, in which the Jacobian is approximated by the matrix $C$, and the problem is approximated by a problem linear in $\alpha$.

The method just derived, of Gauss-Newton type, has the advantage of not using second derivative information. In addition, linear problems are solved in just one iteration, while mildly nonlinear and nearly consistent problems show fast local convergence, [4]. For very nonlinear problems, or those which have large residuals, as potentially may be the case when the data has large noise, this method may not even be locally convergent [4]. Thus,
even if \( C \) were to estimate the entries in \( J \), it is possible that for pixels where the noise level is large, the method would not locally converge.

Alternatively, one could use a Levenberg-Marquardt nonlinear minimization, as presented in Section 3.1, to solve

\[
S''(\theta^{(k-1)}) \left( \theta^{(k)} - \theta^{(k-1)} \right) = -\nabla_{\theta} S(\theta^{(k-1)})
\]

for \( \theta^{(k)} \), where \( H := S''(\theta^{(k-1)}) \) is the Hessian matrix, usually approximated only by

\[
\frac{\partial^2 S}{\partial \theta_i \partial \theta_j} = 2 \sum_{l=1}^{m} \left[ \left( \frac{\partial S}{\partial \theta_i} \right) \left( \frac{\partial S}{\partial \theta_j} \right) \right]_l.
\]

Since \( \nabla_{\theta} S(\theta^{(k-1)}) = 2[J, D]^T (D\beta - Y) \), equation (3.59) becomes

\[
\theta^{(k)} = \theta^{(k-1)} - [S''(\theta^{(k-1)})]^{-1}[J, D]^T (D\beta - Y),
\]

where \( D = D(\alpha^{(k-1)}), J = J(\alpha^{(k-1)}, \beta^{(k-1)}) \).

We now proceed to present some numerical examples.

Given an integrable input function \( u(t) \) sampled non-uniformly, an output function \( y(t) \) is simulated given random parameters \( a_i, b_i, i = 1, 2 \), with the constraint that each parameter is positive. GLLS is used to solve the inverse problem. The initial least squares estimate is obtained by first using a cubic spline interpolation of \( u(t), y(t) \) to facilitate a direct integration by parts of the matrix entries (3.1). Two methods are used to approximate the convolutions \( \psi_i(t), \psi_i(t) \): one method uses the composite trapezoidal rule with uniform time stepping (Method 1), the second method uses integration by parts, given the cubic spline coefficients on each interval (Method 2). The time stepping chosen is on the order of \( 10^{-3} \). Note, the data for this experiment is only contaminated by the noise introduced by the numerical integration.

**Example 1** Given an analytic form of the input function, \( u(t) = 28.0975(e^{-0.857642t} - e^{-1.210864t}) \), \( [64] \), random values \( a_i, b_i, i = 1, 2 \) are generated in Matlab using the \( \text{randn} \) generating function, and an output function \( y(t) \) is computed as the solution to (3.1) with \( n = 2 \). GLLS is used to estimate \( a_i, b_i, i = 1, 2 \), see results in Table 7. In this example, the GLLS algorithm using the integration by parts given the cubic spline coefficients (Method 2), converges in one step to the original solution. However, the same algorithm shows a different behavior if the computation of the entries of the iteration matrix \( Z \) is performed using Method 1. The GLLS iteration is terminated if either 100 iterations are reached or the relative error of the solution vector is less than a set tolerance of \( 10^{-8} \). Although the value reached by the GLLS algorithm is close to the original value, it is further from the solution obtained using Method 2. A close analysis of error estimates of the least squares solution \([28]\) shows that they depend on the condition number of the matrix. In this example, the condition number of matrix \( Z \) reaches \( 10^{13} \) when Method 1 is used, compared to \( 10^2 \) for Method 2. One expects that under noisy conditions, this behavior is more prominent, and the estimates will become less accurate simply due to an ill conditioned variable iteration matrix.
Table 7. GLLS estimates obtained for Example 1.

<table>
<thead>
<tr>
<th></th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$b_1$</th>
<th>$b_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>True values</strong></td>
<td>1.6006</td>
<td>0.1988</td>
<td>0.0970</td>
<td>0.0678</td>
</tr>
<tr>
<td><strong>Method 1</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Initial Guess (LS)</td>
<td>1.4115</td>
<td>0.1682</td>
<td>0.0939</td>
<td>0.0578</td>
</tr>
<tr>
<td>First Iteration (GLLS)</td>
<td>1.4075</td>
<td>0.1631</td>
<td>0.0951</td>
<td>0.0567</td>
</tr>
<tr>
<td>GLLS estimate</td>
<td>1.4056</td>
<td>0.1606</td>
<td>0.0954</td>
<td>0.0558</td>
</tr>
<tr>
<td><strong>Method 2</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Initial Guess (LS)</td>
<td>1.4115</td>
<td>0.1682</td>
<td>0.0939</td>
<td>0.0578</td>
</tr>
<tr>
<td>First Iteration (GLLS)</td>
<td>1.6063</td>
<td>0.1994</td>
<td>0.0971</td>
<td>0.0680</td>
</tr>
<tr>
<td>GLLS estimate</td>
<td>1.6069</td>
<td>0.1995</td>
<td>0.0971</td>
<td>0.0681</td>
</tr>
</tbody>
</table>

**Example 2** We use the same generating input function as used in Example 1, with different initially chosen parameters, and present the results in Table 8. In this case, the estimates using both methods are further from the solution vector than one would desire. The condition number of the variable iteration matrix for the first iteration is on the order of $10^3$. This shows that even when the condition number of the iteration matrix is relatively small, one can get answers that are relatively far from the true solution.

Table 8. GLLS estimates obtained for Example 2.

<table>
<thead>
<tr>
<th></th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$b_1$</th>
<th>$b_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>True values</strong></td>
<td>1.0053</td>
<td>0.1225</td>
<td>0.1389</td>
<td>0.1115</td>
</tr>
<tr>
<td><strong>Method 1</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Initial Guess (LS)</td>
<td>0.2391</td>
<td>0.0121</td>
<td>0.1348</td>
<td>0.0113</td>
</tr>
<tr>
<td>First Iteration (GLLS)</td>
<td>1.1227</td>
<td>0.1393</td>
<td>0.1397</td>
<td>0.1269</td>
</tr>
<tr>
<td>GLLS estimate</td>
<td>1.1227</td>
<td>0.1392</td>
<td>0.1400</td>
<td>0.126</td>
</tr>
<tr>
<td><strong>Method 2</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Initial Guess (LS)</td>
<td>0.2391</td>
<td>0.0121</td>
<td>0.1348</td>
<td>0.0113</td>
</tr>
<tr>
<td>First Iteration (GLLS)</td>
<td>0.2893</td>
<td>0.0193</td>
<td>0.1351</td>
<td>0.0178</td>
</tr>
<tr>
<td>GLLS estimate</td>
<td>1.1006</td>
<td>0.1361</td>
<td>0.1392</td>
<td>0.1239</td>
</tr>
</tbody>
</table>

**Example 3** The same generating input function as in Example 1 was used, with different parameters $a_i, b_i$, $i = 1, 2$. The results are shown in Table 9.

In this case, the GLLS solution clearly diverges from the true solution. The condition number of the iteration matrix is relatively small, on the order of $10^6$. Both methods however approach the same solution, which suggests that the minimum of the functional $S$ in (3.52) is either not unique, or that minimizing in the Euclidean norm introduces additional solutions.
Table 9. GLLS estimates obtained for Example 3.

<table>
<thead>
<tr>
<th></th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$b_1$</th>
<th>$b_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>True values</strong></td>
<td>1.2614</td>
<td>0.0037</td>
<td>0.1630</td>
<td>0.2049</td>
</tr>
<tr>
<td><strong>Method 1</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Initial Guess (LS)</td>
<td>-0.0703</td>
<td>-0.0002</td>
<td>0.1627</td>
<td>-0.0119</td>
</tr>
<tr>
<td>First Iteration (GLLS)</td>
<td>-0.0669</td>
<td>-0.0002</td>
<td>0.1630</td>
<td>-0.0114</td>
</tr>
<tr>
<td>GLLS estimate</td>
<td>-0.1507</td>
<td>-0.0005</td>
<td>0.1630</td>
<td>-0.0251</td>
</tr>
<tr>
<td><strong>Method 2</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Initial Guess (LS)</td>
<td>-0.0703</td>
<td>-0.0002</td>
<td>0.1627</td>
<td>-0.0119</td>
</tr>
<tr>
<td>First Iteration (GLLS)</td>
<td>-0.0589</td>
<td>-0.0002</td>
<td>0.1627</td>
<td>-0.0101</td>
</tr>
<tr>
<td>GLLS estimate</td>
<td>-0.1273</td>
<td>-0.0004</td>
<td>0.1628</td>
<td>-0.0212</td>
</tr>
</tbody>
</table>

Figure 23. Log plot of the relative solution error $||\frac{\theta^{(k)}(k) - \theta^{(k-1)}}{\theta^{(k)}}||$ in Example 3.

Based on this analytical and numerical study, we use GLLS as a single step refinement to the least squares solution and call this method Filtered Least Squares (FLS). For comparison purposes, we also use this method iteratively and call it IFLS - iterative filtered least squares.

4. Simulations

We consider a sampling time consistent with a conventional plasma FDG tracer acquisition schedule (see [29]). In particular, in this study the length of the sampling is only 45 minutes, unlike other studies [16] in which a length of 120 minutes is considered. This fact greatly affects the accuracy of the estimates, especially in case of large noise. The input function is nonlinearly fitted to a function of the form, suggested and studied in [14],

$$u(t) = (A_1 t - A_2 - A_3) e^{\lambda_1 t} + A_2 e^{\lambda_2 t} + A_3 e^{\lambda_3 t}.$$  \hspace{1cm} (3.60)
To focus our attention on the effects of PET measurement noise on the estimation algorithms, we assume there is no noise in the plasma time activity input function. The output is generated by solving the direct problem, with a set of average parameters in grey matter obtained from a study by Huang [4], \( k_1 = 0.102 \text{ ml/(ml tissue)}^{-1}\text{min}^{-1}, k_2 = 0.13 \text{ min}^{-1}, k_3 = 0.062 \text{ min}^{-1}, k_4 = 0.0068 \text{ min}^{-1} \). For the linear formulation, the numerical evaluation of the integrals in matrix \( X = X_L \) is done by first interpolating the input and output data via a cubic spline and integrating the spline directly (using the Matlab function \texttt{spline}).

To simulate the noise in the output data, a Gaussian distribution is assumed, with mean 0 and variance

\[
\text{variance}(e(t_i)) = \alpha \frac{y(t_i)}{t_i - t_{i-1}},
\]

where \( \{t_i\}_{i=1, \ldots, m} \) is the sampling schedule, \( y(t_i) \) is the output data at time \( t_i \), and \( \alpha \) is the noise level. Note, the time interval is used because the sampling is non-uniform and there is greater variability over smaller intervals. Again, the Matlab function \texttt{randn} was used to generate random numbers with mean 0 and variance 1.

The simulated input data together with the output data are given in Figure 26. We perturb the output data and calculate the resulting statistics over a sample size of 1000 runs. For each method we plot the mean, bias, standard deviation and deviation, in this order. We let the relative error \( \rho_i := \left| \frac{k_{\text{estimate}} - k_{\text{exact}}}{k_{\text{exact}}} \right| \) for each \( i \). The cost of each method is given in number of flops. When there is no error in the data, we obtain the estimates shown in Table 10.

5. Conclusions and Discussion

For the case when no noise is present in either the input or output data, we make the following observations:

- The best method is the FLS, which is clearly a filter of the LS solution.
Table 10. Estimates obtained from each method, when there is no noise present in the data.

<table>
<thead>
<tr>
<th>Methods</th>
<th>$k_1$</th>
<th>$k_2$</th>
<th>$k_3$</th>
<th>$k_4$</th>
<th>$k_5$</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>true values</td>
<td>0.1020</td>
<td>0.1300</td>
<td>0.0620</td>
<td>0.0068</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LS</td>
<td>0.1027</td>
<td>0.0069</td>
<td>0.1326</td>
<td>0.0200</td>
<td>0.0644</td>
<td>0.0086</td>
</tr>
<tr>
<td>LM</td>
<td>0.1026</td>
<td>0.0059</td>
<td>0.1337</td>
<td>0.0285</td>
<td>0.0659</td>
<td>0.0629</td>
</tr>
<tr>
<td>CLS</td>
<td>0.1030</td>
<td>0.0098</td>
<td>0.1351</td>
<td>0.0392</td>
<td>0.0671</td>
<td>0.0823</td>
</tr>
<tr>
<td>NnLS</td>
<td>0.1027</td>
<td>0.0069</td>
<td>0.1326</td>
<td>0.0200</td>
<td>0.0644</td>
<td>0.0387</td>
</tr>
<tr>
<td>TLS</td>
<td>0.1027</td>
<td>0.0069</td>
<td>0.1326</td>
<td>0.0200</td>
<td>0.0644</td>
<td>0.0387</td>
</tr>
<tr>
<td>FLS</td>
<td>0.1021</td>
<td>0.0010</td>
<td>0.1307</td>
<td>0.0054</td>
<td>0.0630</td>
<td>0.0161</td>
</tr>
<tr>
<td>IFLS</td>
<td>0.1021</td>
<td>0.0010</td>
<td>0.1307</td>
<td>0.0054</td>
<td>0.0630</td>
<td>0.0161</td>
</tr>
</tbody>
</table>

Figure 25. Input and output functions used in simulation.

- The LS, NnLS and TLS give the same results, followed by the LM method which has a slightly larger bias.

- The CLS method overestimates all parameters, and consequently has the largest bias of all methods.

- The IFLS (iterative filtered least squares) performs only one iteration in the case of no noise, so the results are exactly as the ones given by FLS.

Under noisy conditions for the output function, we observe:

- The CLS (constrained least squares) method performs consistently better than all other methods.

- The NnLS method remains competitive with the CLS method.
Figure 26. Comparison of mean, std, bias and deviation over a sample size of 1000 runs, for various methods estimating the $k$ parameter with 1% noise in the output function.

- The LS, LM, FLS, and IFLS methods worsen as noise level increases.
- The TLS method performs consistently worse than all other methods.
- There seems to be no improvement in the accuracy of the estimates of the IFLS method over the FLS method.

In terms of cost, the TLS method is the cheapest, followed by LS, NnLS, and FLS, then CLS and IFLS, with the nonlinear LM method being the most expensive.
Figure 27. Comparison of mean, std, bias and deviation over a sample size of 1000 runs, for various methods estimating the $k$ parameter with 2% noise in the output function.

Figure 28. Comparison of mean, std, bias and deviation over a sample size of 1000 runs, for various methods estimating the $k$ parameter with 5% noise in the output function.
Figure 29. Comparison of mean, std, bias and deviation over a sample size of 1000 runs, for various methods estimating the $k$ parameter with 10% noise in the output function.

Figure 30. Comparison of mean, std, bias and deviation over a sample size of 1000 runs, for various methods estimating the $k$ parameter with 20% noise in the output function.
CHAPTER 4

Physical Measurement Effects in PET Studies

Real data suffers from noise introduced by spill-over and partial volume effects. Spill-over is usually bidirectional and accounts for a percentage of the tracer in plasma being counted as total tissue-tracer, as well as for some of the tracer in tissue being counted as plasma-tracer (due to limited spatial resolution). This leads to a blurring effect in the image. Spill-over coefficients can be introduced in the tracer model, and estimated as additional parameters. However, they can complicate the numerical identifiability of model parameters [17]. Consequently, one or more venous samples are made late in the study to improve their identification. Other methods for removing spill-over effects have been considered [18], but they suffer from being dependent on estimates of the imaged-object geometry as well as on other factors, like cardiac motion. Principal component analysis is also introduced to derive tissue-time activity curves free from spill-over effects [3], but we opt to consider the model dependent method which can be easily incorporated into the kinetic model (1.3).

Partial volume effects (PVE) occur due to the intrinsic spatial resolution of the PET camera; in objects smaller than the resolution of the tomograph, true tracer concentration will be underestimated. The pixel target tissue mixes with surrounding tissue, and this can lead to an inaccurate parameter estimation overall [57], [26], [49]. This is a significant concern for region of interest studies of the heart. Several methods for correcting for partial volume effects have been considered [1], [18]. However, this is currently an active area of research, and we focus here only on model dependent methods which can account for PVE.

1. Spill-Over Effects

Given the system in (1.2), one can adjust for the spill-over effects by considering

\[ y^m(t) = y(t) + m_{bt}u(t), \]

where \( y^m(t) \) is the measured tissue-tracer concentration, \( y(t) = y_1(t) + y_2(t) \) is the true tissue-tracer concentration, \( u(t) \) is the plasma-tracer concentration and \( m_{bt} \) is the spill-over coefficient from blood into tissue (via the interstitial space.) Note, the assumption here is that the spill-over coefficient \( m_{bt} \) is independent of time [18], and that there is no spill-over from tissue into blood (this will be considered later). Given this, the second order
differential model is now given by

\[ y^{\ddot{m}}(t) + a_1 y^{\dot{m}}(t) + a_2 y^{m}(t) = m_{bt} \ddot{u}(t) + b_1 \dot{u}(t) + b_2 u(t), \]

\[ y^{m}(0) = 0, \]

\[ y^{\dot{m}}(0) = 0 \] \hspace{1cm} (4.1)

where

\[ [a_1, a_2, b_1, b_2] = [k_2 + k_3 + k_4, k_2 k_4, k_1 + m_{bt}(k_2 + k_3 + k_4), k_1(k_3 + k_4) + m_{bt} k_2 k_4], \] \hspace{1cm} (4.2)

and note that the difference as compared to (1.3) lies in the introduction of the second order derivative for the input function \( u(t) \), and the modification in the coefficients for \( \ddot{u}(t) \) and \( u(t) \). Using the Volterra formulation, one can obtain a linear system \( X_V \theta = Y \) where \( X_V \) has rows of the form

\[ (X_V)_i = \left[ u(t), -\int_0^{t_i} y^{\dot{m}}(\tau)d\tau, -\int_0^{t_i} y^{m}(\tau)(t_i - \tau)d\tau, \int_0^{t_i} u(\tau)d\tau, \int_0^{t_i} u(\tau)(t_i - \tau)d\tau \right], \]

and

\[ \theta = [m_{bt}, k_2 + k_3 + k_4, k_2 k_4, k_1 + m_{bt}(k_2 + k_3 + k_4), k_1(k_3 + k_4) + m_{bt} k_2 k_4]. \] \hspace{1cm} (4.3)

The solution for each parameter \( k_i \) can be constructed,

\[ k_1 = \theta_4 - \theta_1 \theta_2, \]

\[ k_2 = \theta_2 - \frac{\theta_5 - \theta_1 \theta_3}{\theta_4 - \theta_1 \theta_2}, \]

\[ k_3 = \frac{\theta_5 - \theta_1 \theta_3}{\theta_4 - \theta_1 \theta_2} - \frac{\theta_3(\theta_4 - \theta_1 \theta_2)}{\theta_2(\theta_4 - \theta_1 \theta_2) + \theta_1 \theta_3 - \theta_5}, \]

\[ k_4 = \frac{\theta_3(\theta_4 - \theta_1 \theta_2)}{\theta_2(\theta_4 - \theta_1 \theta_2) + \theta_1 \theta_3 - \theta_5}. \]

Note, all parameters \( k_i, i = 1, \ldots, 4 \) are nonlinear functions of \( \theta_i, i = 1, \ldots, 5 \), in particular \( k_3 \) has the most complicated nonlinear dependence on \( \theta_i \). Also, all parameters \( k_i \) are nonlinearly affected by the presence of a spill-over term. A constrained least squares solution of the linear system \( X_V \theta = Y \) will provide some initial estimates for the solution vector \( k \), as well as for the spill-over coefficient.

The solution thus obtained can be filtered by the FLS algorithm, although some changes are necessary to adjust for the fact that, under a Laplace transform, the system in (4.1) becomes

\[ A(s)Y(s) = B(s)U(s) \] \hspace{1cm} (4.4)

where now \( B(s) = m_{bt}s^2 + b_1 s + b_2 \) is quadratic. Following the development of the GLLS algorithm, we rewrite the above equation as

\[ \frac{A(s) - \dot{A}(s)}{A(s)}Y(s) + Y(s) = \frac{B(s)}{A(s)}U(s), \]
where $\hat{A}(s) = s^2 + \hat{a}_1 s + \hat{a}_2$, with $\hat{a}_1, \hat{a}_2$ the initial parameter estimates. This can be further reduced,

$$Y(s) + \frac{s(a_1 - \hat{a}_1) + (a_2 - \hat{a}_2)}{A(s)} Y(s) = m_{bd} U(s) + \frac{s(b_1 - m_{bd}\hat{a}_1) + (b_2 - m_{bd}\hat{a}_2)}{A(s)} U(s) \Rightarrow$$

$$Y(s) - \hat{a}_1 \frac{s}{A(s)} Y(s) - \hat{a}_2 \frac{1}{A(s)} Y(s) = m_{bd} U(s) - a_1 \frac{s}{A(s)} Y(s) - a_2 \frac{1}{A(s)} Y(s) + (b_1 - m_{bd}\hat{a}_1) \frac{s}{A(s)} U(s) + (b_2 - m_{bd}\hat{a}_2) \frac{1}{A(s)} U(s).$$

Assuming the second order equation $\hat{A}(s) = 0$ has two nonzero, distinct solutions, the inverse Laplace transform is applied to give a result following (3.48),

$$y(t) - \sum_{j=1}^{n} \hat{a}_j \sum_{i=1}^{n} \lambda_i^{n-j} \psi_i(t) = m_{bd} u(t) - \sum_{j=1}^{n} a_j \sum_{i=1}^{n} \lambda_i^{n-j} \psi_i(t) + \sum_{j=1}^{n} (b_j - m_{bd}\hat{a}_j) \sum_{i=1}^{n} \lambda_i^{n-j} \psi_i(t).$$

(4.5)

The constrained linear least squares method is used to find the solution vector $\theta = \left[ m_{bd}, a_1, a_2, b_1 - m_{bd}\hat{a}_1, b_2 - m_{bd}\hat{a}_2 \right]$. Given these, a solution for the $k$ parameter vector can be easily resolved,

$$k_1 = \theta_4 - \theta_1(\theta_2 - \hat{\theta}_2),$$

$$k_2 = \frac{\theta_5 - \theta_1(\theta_3 - \hat{\theta}_3)}{\theta_4 - \theta_1(\theta_2 - \hat{\theta}_2)},$$

$$k_3 = \frac{\theta_5 - \theta_1(\theta_3 - \hat{\theta}_3)}{\theta_4 - \theta_1(\theta_2 - \hat{\theta}_2)} - \frac{\theta_3 \left[ \theta_4 - \theta_1(\theta_2 - \hat{\theta}_2) \right]}{\theta_2 \left[ \theta_4 - \theta_1(\theta_2 - \hat{\theta}_2) \right] + \theta_1(\theta_3 - \hat{\theta}_3) - \theta_5},$$

$$k_4 = \frac{\theta_3 \left[ \theta_4 - \theta_1(\theta_2 - \hat{\theta}_2) \right]}{\theta_2 \left[ \theta_4 - \theta_1(\theta_2 - \hat{\theta}_2) \right] + \theta_1(\theta_3 - \hat{\theta}_3) - \theta_5}.$$

We use this spill-over correction of the filtered least squares method to construct global parametric images.

Alternatively, one could assume the value of $m_{bd}$ as known, $\hat{m}_{bd} = m_{bd}$, and solve the following equation

$$Y(s) - \hat{m}_{bd} U(s) + \hat{m}_{bd}\hat{a}_1 \frac{s}{A(s)} U(s) + \hat{m}_{bd}\hat{a}_2 \frac{1}{A(s)} U(s) = \hat{a}_1 \frac{s}{A(s)} Y(s) - \hat{a}_2 \frac{1}{A(s)} Y(s) - a_1 \frac{s}{A(s)} Y(s) - a_2 \frac{1}{A(s)} Y(s) + b_1 \frac{s}{A(s)} U(s) + b_2 \frac{1}{A(s)} U(s).$$

Depending on the application, one also considers the spill-over that occurs from tissue into blood, such that

$$u^{m}(t) = u(t) + m_{bd} y(t),$$

$$y^{m}(t) = y(t) + m_{bd} u(t),$$
where \( u^m(t) \) is the measured plasma-tracer concentration, \( u(t) \) is the true (spill-over free) plasma-tracer concentration, \( y^m(t) \) is the measured tissue-tracer concentration, and \( y(t) \) is the true tissue-tracer concentration, \( y(t) \). For applications in cardiac studies over regions of interest, it is critical that one considers the effects of tracer spill-over from tissue into blood \([17], [9]\). For brain studies which use an image-derived input function \( u(t) \), the spill-over coefficient \( m_{tb} \) is estimated \([10]\) such that one obtains a spill-over corrected input function. In this case, clearly one needs to only account for the spill-over from blood into tissue, \( m_{bt} \).

The images shown in Section 3 below use a spill-over corrected input function such that only estimates for \( m_{bt} \) are derived, together with the kinetic parameters.

2. Spill-Over and Partial Volume Effects

Accounting for both spill-over and partial volume effects results in

\[
\begin{align*}
    u^m(t) &= r_u u(t) + m_{tb} y(t), \\
    y^m(t) &= r_y y(t) + m_{bt} u(t),
\end{align*}
\]

where \( r_u \) and \( r_y \) are recovery coefficients for \( u(t) \) and \( y(t) \) respectively, while the spill-over coefficients were described in Section 1 of this chapter. Note, since \( y(t) \) is the true tracer concentration in tissue, it still respects the convolution solution, \( y(t) = (f * u(t)) \). Given that in a experiment we only have the measured quantities \( u^m(t) \) and \( y^m(t) \), the second order differential equation obtained in terms of these quantities is

\[
\begin{align*}
    \ddot{y}^m(t) + a_1 \dot{y}^m(t) + a_2 y^m(t) &= \frac{m_{bt}}{r_u} \dot{u}^m(t) + b_1 u^m(t) + b_2 u^m(t), \\
    y^m(0) &= 0, \\
    \dot{y}^m(0) &= 0,
\end{align*}
\]

(4.6)

where

\[
\begin{align*}
    a_1 &= \frac{m_{tb}}{r_u} k_1 + (k_2 + k_3 + k_4), \\
    a_2 &= \frac{m_{tb}}{r_u} k_1 (k_3 + k_4) + (k_2 k_4), \\
    b_1 &= \frac{m_{bt}}{r_u} (k_2 + k_3 + k_4) + \frac{r_y}{r_u} k_1, \\
    b_2 &= \frac{m_{bt}}{r_u} (k_2 k_4) + \frac{r_y}{r_u} k_1 (k_3 + k_4).
\end{align*}
\]

A linear system can be constructed as before and one can solve for a solution vector \( \theta \). However, without any additional constraint, we have a solution vector of size 5, but a total of 8 unknowns. Hence, the system would have infinitely many solutions. Even by reducing the number of unknowns by two by solving for \( \frac{m_{bt}}{r_u}, \frac{m_{tb}}{r_u} \) and \( \frac{r_y}{r_u} \) as three unknowns, one would still be left with an under-determined system of 7 unknowns and only 5 equations. Alternatively, a linear constraint typically imposed for this problem \([18]\)

\[
\begin{align*}
    r_u + m_{tb} &= 1, \\
    r_y + m_{bt} &= 1,
\end{align*}
\]
reduces the number of unknown by 2, but still leaves the system under-determined. Without a further constraint, one could not solve this system uniquely.

Using an image-derived input function, corrected for both spill-over and partial volume effects, our system would simplify into a system similar to (4.6),

$$\ddot{y}^m(t) + a_1\dot{y}^m(t) + a_2y^m(t) = m_{bd}\ddot{u}^m(t) + b_1\dot{u}^m(t) + b_2u^m(t)$$

with coefficients

$$a_1 = k_2 + k_3 + k_4,$$
$$a_2 = k_3k_4,$$
$$b_1 = r_yk_1 + m_{bd}(k_2 + k_3 + k_4),$$
$$b_2 = r_yk_1(k_3 + k_4) + m_{bd}(k_2k_4).$$

Using the constraint $r_y + m_{bd} = 1$, this system is uniquely solvable.

3. Discussion on Bound Constraints

The kinetic images resulting from a pixel-by-pixel computation are highly contaminated by noise. An algorithm like FLS filters out only colored noise, but leaves behind white noise. The constrained methods tested above, CLS and NnLS suggest that the accuracy of the kinetic parameter $k$ at the pixel level will be improved by imposing reasonable bounds on the parameters. Hence, we consider combining the FLS algorithm with a CLS algorithm to obtain an improved solution. This hybrid algorithm, constrained filtered least squares (CFLS), uses a CLS method to obtain an initial estimate to the solution vector, which is further filtered via FLS, where the least squares solver is again a CLS algorithm. Hence, we need to generate a reasonable data constraint.

To derive bounds, we consider some results previously published in literature. Feng, [16] publishes values for $k$, using a region of interest analysis, and an unweighted nonlinear least squares estimation method. The FDG bolus dosage used in the clinical study was $5 - 10$ Bq, and the total time of imaging was 120 minutes. 23 regions of interest were chosen, with a mean number of pixels of 167 for each, and a pixel size of 2 mm $\times$ 2 mm. Those results are presented in Table 11. Piert, [46] publishes values for $k$, using a region of interest analysis, and the standard nonlinear least-squares analysis. In this study, the radioactivity of the FDG tracer injected was $10 \pm 1$ mCi. A dynamic sequence of 15 PET scans was obtained, for a total imaging time of 60 minutes. The regions of interest were drawn using a standard neuroanatomical atlas. His results are presented in Table 12.

Since neither of these articles derived values for the spill-over coefficient, we assume $m_{bd}$ varies between 0 and 1. Then, given the $k$ parameter values, one derives minimum and maximum bounds for the vector $\theta$.

In addition to the bounds derived from results given in literature [16], [46], we also use bounds derived from the given image, via averaged clustered curves [23], shown in Figure 7. Although clustering has been used by others in generating parametric images, for
Table 11. $k$ values published by Feng, et al. in [16].

<table>
<thead>
<tr>
<th>$k_1$</th>
<th>.2063</th>
<th>.1014</th>
<th>.1366</th>
<th>.1309</th>
<th>.0475</th>
<th>.0389</th>
<th>.1224</th>
<th>.1154</th>
</tr>
</thead>
<tbody>
<tr>
<td>.1299</td>
<td>.0785</td>
<td>.1332</td>
<td>.1377</td>
<td>.0485</td>
<td>.1028</td>
<td>.0854</td>
<td>.1310</td>
<td></td>
</tr>
<tr>
<td>.1318</td>
<td>.0919</td>
<td>.0725</td>
<td>.1284</td>
<td>.1223</td>
<td>.0380</td>
<td>.0436</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_2$</td>
<td>.8246</td>
<td>.3963</td>
<td>.6370</td>
<td>.4969</td>
<td>.1793</td>
<td>.2832</td>
<td>.4191</td>
<td>.2895</td>
</tr>
<tr>
<td>.3951</td>
<td>.2229</td>
<td>.4817</td>
<td>.4436</td>
<td>.1205</td>
<td>.9949</td>
<td>.3300</td>
<td>.5235</td>
<td></td>
</tr>
<tr>
<td>.4213</td>
<td>.2743</td>
<td>.2818</td>
<td>.0453</td>
<td>.4556</td>
<td>.1054</td>
<td>.4733</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_3$</td>
<td>.1208</td>
<td>.0896</td>
<td>.1273</td>
<td>.0982</td>
<td>.0466</td>
<td>.1101</td>
<td>.1179</td>
<td>.0670</td>
</tr>
<tr>
<td>.0880</td>
<td>.0324</td>
<td>.1060</td>
<td>.0853</td>
<td>.0394</td>
<td>.1863</td>
<td>.0586</td>
<td>.1289</td>
<td></td>
</tr>
<tr>
<td>.0799</td>
<td>.0588</td>
<td>.0737</td>
<td>.1477</td>
<td>.0996</td>
<td>.0331</td>
<td>.1824</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_4$</td>
<td>.01273</td>
<td>.01035</td>
<td>.01280</td>
<td>.01184</td>
<td>.0670</td>
<td>.0820</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.00719</td>
<td>.00063</td>
<td>.00464</td>
<td>.00737</td>
<td>.00423</td>
<td>.00616</td>
<td>.00898</td>
<td>.01210</td>
<td></td>
</tr>
<tr>
<td>.00984</td>
<td>.01092</td>
<td>.01775</td>
<td>.01292</td>
<td>.01165</td>
<td>.00790</td>
<td>.01806</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 12. $k$ values published by Piert, et al. in [46].

<table>
<thead>
<tr>
<th>$k_1$</th>
<th>.093</th>
<th>.099</th>
<th>.09</th>
<th>.131</th>
<th>.132</th>
<th>.134</th>
<th>.067</th>
<th>.082</th>
</tr>
</thead>
<tbody>
<tr>
<td>.080</td>
<td>.119</td>
<td>.106</td>
<td>.114</td>
<td>.129</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_2$</td>
<td>.148</td>
<td>.166</td>
<td>.141</td>
<td>.133</td>
<td>.16</td>
<td>.17</td>
<td>.161</td>
<td>.19</td>
</tr>
<tr>
<td>.151</td>
<td>.194</td>
<td>.176</td>
<td>.181</td>
<td>.204</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_3$</td>
<td>.179</td>
<td>.183</td>
<td>.165</td>
<td>.137</td>
<td>.188</td>
<td>.166</td>
<td>.122</td>
<td>.070</td>
</tr>
<tr>
<td>.101</td>
<td>.117</td>
<td>.124</td>
<td>.115</td>
<td>.088</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_4$</td>
<td>.017</td>
<td>.013</td>
<td>.013</td>
<td>.012</td>
<td>.016</td>
<td>.01</td>
<td>.013</td>
<td>.013</td>
</tr>
<tr>
<td>.01</td>
<td>.007</td>
<td>.012</td>
<td>.006</td>
<td>.009</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

example [34], [60], their approach is different from what we propose here. For example, Zhou [60] uses the cluster derived time activity curves to provide a basis for all solutions, then solves to generate linear combinations of these cluster time activity curves from the given vector space.

Using the FLS algorithm, with no a-priori bounds imposed on the individual parameters, we find values $k$ for each curve, Table 13. These are then used to obtain minimum and maximum bounds for the $\theta$ vector, (4.3), for a pixel-by-pixel image computation. To adjust for the variance of the output curves within the highest cluster, whose average is given by Curve 1, Figure 7, we add to our upper bound the variance in $\theta$ between Curve 1 and 2. The lower bound is adjusted by $\theta$ values computed for a clustered curve derived from all regions of no activity or almost no activity (i.e. regions outside the brain and the cerebral spinal fluid region).

3.1. Results - Spill-Over Effects. We compare 2-D images obtained using no constraint ($FLS_{nc}$), a constraint derived from literature Feng [16] ($CFLS_{Feng}$), and Piert [46] ($CFLS_{piert}$), and lastly the constraint derived from clustering, ($CFLS_{clst}$). The images
Table 13. $k$ values obtained from fitting the clustered curves of the data under study.

<table>
<thead>
<tr>
<th>Curve 1</th>
<th>Curve 2</th>
<th>Curve 3</th>
<th>Curve 4</th>
<th>Curve 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_1$</td>
<td>.0997</td>
<td>.1244</td>
<td>.1433</td>
<td>.1635</td>
</tr>
<tr>
<td>$k_2$</td>
<td>.2885</td>
<td>.2632</td>
<td>.2479</td>
<td>.2570</td>
</tr>
<tr>
<td>$k_3$</td>
<td>.2307</td>
<td>.2213</td>
<td>.2289</td>
<td>.2623</td>
</tr>
<tr>
<td>$k_4$</td>
<td>.0538</td>
<td>.0489</td>
<td>.0419</td>
<td>.0402</td>
</tr>
</tbody>
</table>

-derived are shown in Figures 31 through 36. To assess the accuracy of each method, we compare the image for $K = \frac{k_1 k_3}{k_2 + k_3}$ for each method with the image for $K$, Figure 37, obtained using Patlak linear regression [41]. The images derived in this and the next section were first segmented using a simple threshold, such that areas outside the brain, as well as the cerebral spinal fluid (CSF) region, were not processed and assumed to have no kinetic activity.

![Images of k values derived from different methods](image)

**Figure 31.** $k_1$ image using spill-over corrected model.

### 3.2. Results - Spill-Over and Partial Volume Effects

We derive images corrected for spill-over and partial volume effects, using a slice of brain from a human study, and bounds obtained from the values given in Tables 11, 12, 13. Figures 38 through 43 show these results. We note that the partial volume correction implemented here leads to very minor differences in the parametric images in comparison to the spill-over corrected images, Figures 31 through 36. Most clinical studies use magnetic resonance imaging (MRI) data to...
perform the partial volume corrections, as they become significant mostly for longitudinal studies.

4. Conclusions and Discussion

We note that constraint estimation techniques are necessary for obtaining meaningful images. The accuracy of the constraint is critical to the quality of the parametric images. Although some of these images are highly contaminated by noise, their usefulness lies in confirming the relative range of each of the parameters, regardless of the original bounds imposed. In addition, it shows that bounds act as a regularization constraint. Thus, the determination of accurate bounds, uniquely derived for each dynamic PET sequence, will provide increasingly accurate parametric images.

One can visually examine the nature of the noise present. The images derived using the bounds obtained from studies done by Feng, [16] show a speckle noise in comparison with images derived using bounds from [46] and from clustering. Speckle noise is known to be well represented by a gamma distribution [2]. Hence, even when noisy, an image may help in understanding the noise characteristics of parametric images, a subject still under research.

The difference between spill-over corrected images and those corrected in addition for partial volume effects, are small. The partial volume effects introduce a small scaling which cause the parameters values to increase slightly. For this study, we conclude that both spill-over and partial volume effects can be incorporated in our estimation. Partial
volume effects are more relevant for longitudinal studies where brain volume may change over time, and more accurate corrections may be derived by considering MRI data.

The estimated model macro-parameter \( K = \frac{k_1k_3}{k_2+k_3} \), obtained both from individual estimates and Patlak analysis, and the parameter \( k_4 \), were used to evaluate the accuracy of the numerical methods employed. In Figures 44, and 45, the correlation between estimates is 0.99.

For the spill-over corrected data, using the quantitative data from Figure 44, we note the following:

- Only \( CFLS_{\text{Piert}} \) and \( CFLS_{\text{clst}} \) have a linear regression slope near 1.
- The R-Squared for \( CFLS_{\text{Piert}} \) is \( \sim 1\frac{3}{4} \) as large as that for \( CFLS_{\text{clst}} \).
- \( CFLS_{\text{Piert}} \) and \( CFLS_{\text{clst}} \) are small, suggesting that only a small percentage of the data is explained by the regression line estimate.

For the spill-over and partial-volume corrected data, using the quantitative data from Figure 45, we note the following:

- Only \( CFLS_{\text{clst}} \) has a linear regression slope near 1.
- The R-Squared for \( CFLS_{\text{clst}} \) is small, .44.
- The \( CFLS_{\text{clst}} \) \( K \) estimates are higher then the Patlak derived \( K \) estimates, in agreement with other studies, [46].
Figure 34. $k_4$ image using spill-over corrected model.

The overall low R-Squared value points to the fact that the data has a higher variability, and is not, on average, close to the linear regression line. This may be due in part to the larger noise expected in a pixel-by-pixel data analysis, but may also be indicative of a larger variation in the true parameter values. Hence, we also look at the spatial distribution of the data for a qualitative interpretation:

- The $FLS_{nc}$ image and the $CFLS_{Feng}$ image show the most noise, for all parameters.
- The $FLS_{nc}$ image and the $CFLS_{Feng}$ image have a speckle-noise distribution.
- The Patlak-derived and pixel-derived $K$ images have a similar range, regardless of the method used.
- The $CFLS_{Piert}$ and $CFLS_{clst}$ $K$ images have a well defined spatial distribution.

The parametric image obtained for $k_4$ is best defined for the $CFLS_{clst}$ method. The $CFLS_{Piert}$ method is competitive with $CFLS_{clst}$, and certainly qualitatively better then the $k_4$ image derived by $FLS_{nc}$ and $CFLS_{Feng}$. Considering our results so far, we apply techniques from image restoration to improve parametric image quality in the next chapter.
Figure 35. $m_{bt}$ image using spill-over corrected model.

Figure 36. $K$ image using spill-over corrected model.
Figure 37. $K$ image derived by using Patlak analysis.

Figure 38. $k_1$ image using spill-over and partial volume effects corrected model.
Figure 39. $k_2$ image using spill-over and partial volume effects corrected model.

Figure 40. $k_3$ image using spill-over and partial volume effects corrected model.
Figure 41. $k_4$ image using spill-over and partial volume effects corrected model.

Figure 42. $m_{bt}$ image using spill-over and partial volume effects corrected model.
Figure 43. $K$ image using spill-over and partial volume effects corrected model.

Figure 44. Pixel-wise comparison of parametric image of $K = \frac{k_1 k_3}{k_2 + k_3}$ estimated by both Patlak analysis and by pixel-by-pixel analysis, using a Constrained Filtered Least Squares (CFLS) algorithm with bounds from Feng [16], Piert [46], and clustering. The data was corrected for spill-over effects. The last graph shows the data from all three graphs in one figure.
Figure 45. Pixel-wise comparison of parametric image of $K = \frac{k_1 k_2}{k_3 + k_4}$ estimated by both Patlak analysis and by pixel-by-pixel analysis, using CFLS algorithm and bounds from Feng [16], Piert [46], and clustering. The data was corrected for spill-over and partial volume effects. The last graph shows the data from all three graphs in one figure.
CHAPTER 5

Nonlinear Anisotropic Diffusion Filtering

In this chapter, we consider the application of image restoration techniques for parametric imaging. A formulation of the filtering problem for parametric imaging is first given. The main idea is to embed the original noisy image in a nonlinear diffusion process. We follow the robust anisotropic diffusion [5] derivation to choose the most appropriate nonlinear function for the diffusion coefficient. For the purpose of this study, anisotropic diffusion refers to nonlinear anisotropic diffusion.

The numerical method originally adopted for the solution of the nonlinear diffusion is the forward Euler’s method. A straightforward implementation of this method facilitates the generation of anisotropic filtered images. Certain parameters, however, need to be carefully chosen in the numerical implementation. We propose using information specific to the noise characteristics of the original image to allow for an automatic determination of the numerical parameters.

1. Problem Formulation

Let \( v : \Omega \in R^2 \rightarrow R \) denote the original true image, given by any one of the parametric images \( k_i \), and let \( v_0 \) be the observed image of the same parametric image (ie \( v_0 \) is a degradation of \( v \)). Then \( v_0 \) relates to \( v \) by

\[ v_0 = Rv + \eta \]

where \( \eta \) stands for additive noise and \( R \) is generally, a linear operator, usually representing a blur. The linear operator describes a deterministic phenomena usually caused by an image acquisition process, while the noise is a random process, caused by any signal transmission. Various methods for noise removal have been proposed, among others we mention linear filtering, stochastic modeling and variational/PDE approaches [2]. We consider here one of the PDE-based methods, and apply it to the problem of parametric imaging in PET. The problem of recovering \( v \) from \( v_0 \) is again an example of an inverse problem.

We remark that in order to apply this theory to our problem, we treat the parametric image \( v_0 \), obtained through a nonlinear process, as the observed image, and assume it can be decomposed into a blurred image \( Rv \) and noise. However, this blur is not caused by an acquisition process, as it is generally the case with restoration problems, but rather by the nonlinear reconstruction of kinetic images.
The basic idea of the PDE approach is to consider the restored image \( v \) as a version of the initial observed image \( v_0 \) at a special scale. The initial data \( v_0 \) is embedded in an evolution process, with the time variable representing a certain scale of the image. One of the most used PDEs in image restoration is the heat equation, which forms a low pass filter, equivalent to a Gaussian convolution,

\[
\frac{\partial v}{\partial t}(x,t) - \Delta v(x,t) = 0, \quad t \geq 0, x \in \mathbb{R}^2, \\
v(x,0) = v_0(x),
\]

where \( \Delta \) is the Laplacian operator, \( \Delta = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} \) for \( x = [x_1, x_2] \in \mathbb{R}^2 \). In the restoration of a parametric image, one is concerned with removing the noise in regions of uniform parameter values, while enhancing the edges between regions of different parameter values. This is clear from the images derived in Chapter 4 from a real patient. Also, it is generally accepted [44] that individual parameters \( k_i \) are constant within brain structures, i.e., one finds the same rate constants throughout the hippocampus, cerebellum, frontal cortex, etc. On average, for a functional image, one can assume that there are at least three distinct vector values \( k \), one for gray matter, one for white, and one for the cerebral spinal fluid (CSF). Hence, we want to design a diffusion process that enhances the edges between different functional regions, while smoothing within region of uniform functional activity, and still preserving small abnormalities. The Gaussian filter (5.1) treats all regions uniformly, and consequently, in the filtering process, it smears the edges between different functional regions.

2. Nonlinear Anisotropic Diffusion

Perona and Malik [43] considered a nonlinear diffusion model as a tool for multiscale description of images, image segmentation, edge detection, and image enhancement. The following partial differential initial boundary value problem is what we will refer to as the anisotropic diffusion model:

\[
\frac{\partial v}{\partial t} = \text{div}(c(|\nabla v|)\nabla v), \quad \text{in} \quad \Omega \times (0, T), \\
\frac{\partial v}{\partial N} = 0, \quad \text{on} \quad \partial \Omega \times (0, T), \\
v(x,y,0) = v_0(x,y) \quad \text{in} \quad \Omega,
\]

where \( N \) is the outward normal to the boundary \( \partial \Omega \). Here, the diffusion coefficient \( c(\cdot) \) is a nonnegative, smooth decreasing function of the magnitude of the local image gradient \( |\nabla v| = \sqrt{v_x^2 + v_y^2} \). To account for anisotropic diffusion, \( c(\cdot) \) is constrained to sharpen edges and smooth homogeneous regions within an image. Hence, near regions of intensity transitions, the choice of \( c(\cdot) \) should cause the model (5.2) to reduce to a backwards diffusion process, while in smooth areas it should reduce to a forward diffusion process.
In the nonlinear diffusion originally proposed, Perona and Malik [43] considered the nonlinear diffusion coefficient as
\[ c(s) = e^{-\left(\frac{s}{\sigma}\right)^2} \]  
and
\[ c(s) = \frac{1}{1 + \left(\frac{s}{\sigma}\right)^2} \]
where \( \sigma \) is a constant to be chosen depending on the particular application. The anisotropic diffusion model with these choices of \( c(s) \) has been shown to be ill-posed [33]. Hence, different diffusion coefficients must be chosen if one is to get a meaningful solution to the anisotropic diffusion model.

Rather then following the more general analytical development of this problem, we concentrate on its relationship to robust statistics, as presented in [5]. Considering the large noise levels generally seen in PET data, it is advantageous to consider the model (5.2) as equivalent to a robust estimation procedure which finds a piecewise constant image from a noisy input image. The problem of robust estimation,
\[ \min_v \int_{\Omega} \rho(|\nabla v|) d\Omega \]
is an optimization problem, where \( \rho(\cdot) \) is a robust norm. Qualitatively, a statistically robust norm is a norm that is insensitive to large data deviations, given a set of observations. For a more technical discussion on robust statistical procedures see [25], [30].

One can minimize this by gradient descent, see [5] for further references,
\[ \frac{\partial v(x, y, t)}{\partial t} = \text{div}(\rho'(|\nabla v|) \frac{\nabla v}{|\nabla v|}). \]  
Clearly, by choosing \( c(s) = \frac{\rho'(s)}{s} \), one finds the anisotropic diffusion model. This relationship has been used in the theoretical analysis of this problem, see [65], [66], [67], [68].

The discretization considered by Perona and Malik [43] is a forward Euler method, using a box method nearest-neighbor spatial discretization,
\[ v_{i,j}^{t+1} = v_{i,j}^t + \lambda [c_{i,\frac{1}{2},j} (v_{i+1,j}^t - v_{i,j}^t) + c_{i,j,\frac{1}{2}} (v_{i,j+1}^t - v_{i,j}^t) + c_{i,-\frac{1}{2},j} (v_{i-1,j}^t - v_{i,j}^t) + c_{i,j,-\frac{1}{2}} (v_{i,j-1}^t - v_{i,j}^t)]. \]
Here, the conduction coefficients are updated at every step,
\[ c_{i,\frac{1}{2},j} = c(\|\nabla v\|_{i+1,\frac{1}{2},j}) \sim c(|v_{i+1,j}^t - v_{i,j}^t|), \]
\[ c_{i,-\frac{1}{2},j} = c(\|\nabla v\|_{i-\frac{1}{2},j}) \sim c(|v_{i-1,j}^t - v_{i,j}^t|), \]
\[ c_{i,j,\frac{1}{2}} = c(\|\nabla v\|_{i,j+1}) \sim c(|v_{i,j+1}^t - v_{i,j}^t|), \]
\[ c_{i,j,-\frac{1}{2}} = c(\|\nabla v\|_{i,j-\frac{1}{2}}) \sim c(|v_{i,j-1}^t - v_{i,j}^t|). \]
Note, this is not the exact discretization of (5.2) since the norm of the gradient at each arc is crudely approximated by the absolute value of its projection along the direction of the arc,
Perona and Malik attempted other approximations of the gradient, but concluded the results were perceptually similar to the ones using this approximation. The constant $2R$ is a scalar, dictated by the stability requirement for this numerical scheme (CFL condition), for this problem $0 \leq \lambda \leq 1/2$. With the robust norm introduced above, this is just the gradient descent method,

$$
v_{i,j}^{t+1} = v_{i,j}^t + \lambda \left[ \rho'(v_{i+1,j} - v_{i,j}^t) + \rho'(v_{i,j+1} - v_{i,j}^t) + \rho'(v_{i-1,j} - v_{i,j}^t) + \rho'(v_{i,j-1} - v_{i,j}^t) \right]$$

(5.5)

for the discrete minimization problem, $\min_v \sum_k \sum_l \rho(v_l - v_k^t)$, where $k = (i,j)$ and $l$ corresponds to one of the 2D coordinates, $(i+1,j), (i,j+1), (i-1,j), (i,j-1)$.

Using the fact that $\rho'(s) = sc(s)$, we find that for the Perona and Malik diffusion function (5.3), $c(s) = e^{-(\frac{s}{\sigma})^2}, \rho'(s)$, is a linear function of the magnitude of the gradient. Hence, as the magnitude of the gradient gets larger near boundaries between areas on non-homogeneous parameter values, the influence given to outliers, $v_l - v_k$, by $\rho'(s)$ grows linearly, without bound. The result is a image that is blurred too much. A similar analysis is pursued for the second diffusion coefficient function in [5], [65]. Therefore, robust statistics arrive at the same conclusion as Kichenassamy, [33], which is that the problem (5.2) is ill-posed given the choices of $c(s)$ originally proposed by Perona and Malik, [43].

A variety of diffusion coefficients have been proposed [5], [32], [48], [65], sometimes without noting their relationship to robust statistics. The results in [5] compare a variety of functions $c(\cdot)$ from the point of view of robust statistics. Their conclusion is that Tukey’s biweight function is a best “robust” norm, whose influence function $\rho'(s)$ leads to a zero bias for large values corresponding to outliers in the data:

$$c(s; \sigma) = \begin{cases} 
0.5 \left[ 1 - \left( \frac{s}{\sigma} \right)^2 \right]^2, & |s| \leq \sigma, \\
0, & \text{else.}
\end{cases}$$

(5.6)

We use this function in our anisotropic diffusion filtering of parametric images. The parameter $\sigma$ that appears in the definition of $c(\cdot)$ must be chosen to be greater than the variance of the noise, but smaller than the size of true discontinuities in the image, to guarantee automatic convergence of the diffusion process [5].

### 2.1. Results - Filtered Parametric Images

We apply the anisotropic diffusion filter to the parametric images already derived. We use Tukey’s biweight function as the diffusion function. Theoretically, parameter $\sigma$ is obtained through trial and error as a compromise between the true size of discontinuities and the variance of noise, and such that automatic diffusion is achieved for all images presented in Figures 46, 47, 48, 49 and 50.

We note that if $\sigma$ is too small, then no diffusion is observed, and if $\sigma$ is too large, the image diffuses to a constant image. We want to iterate an image until convergence is reached, i.e. until the difference $||v^{t+1} - v^t||$ is smaller than a set tolerance, where here $t$ is the iteration number. To choose an appropriate $\sigma$, we estimate the true discontinuities in an image by considering the data processed as in Figures 31 through 36. We choose $\sigma$ as a percentage of the estimated discontinuities. For noisier images, as is the case when no
bounds are imposed, this percentage is smaller (15%), while for better quality images, this percentage is larger (95%). We obtain the images shown in Figures 46 through 51.

Figure 46. Anisotropic diffusion filtered (ADF) image for $k_1$ using data corrected for spill-over effects.

Remark 9 The anisotropic diffusion filtering improves some structures in certain images, like the parametric image for $k_1$ and $K$ using bounds from Feng, [16], while for other images, like the parametric image for $k_1$ using bounds from clustering, some structures look over-smoothed. Without knowing a-priori which functional structures are relevant to disease diagnosis, and how homogeneous a given region should be, it is hard to qualify whether anisotropic diffusion filtering should be employed uniformly throughout an image.

2.2. Results - Parametric Filtered Tissue Activity Images. One can also use the nonlinear anisotropic filter to smooth the time activity images for a given 2D slice. The smoothed versions can then be used to generate parametric images. Images shown in Figures 52 through 58 are obtained using this procedure. In this case, no prior segmentation of the outside of the brain region or the CSF region was performed. Note, the algorithm using bounds derived from clustering can detect the CSF region well in comparison with its respective counterparts, which use bounds derived from literature.

Remark 10 The parametric images derived using filtered tissue time activity data, Figures 52 through 55, show qualitatively overall less smoothness than the parametric images filtered
posteriori as in Figures 46 through 49. However, the Patlak derived parametric image $K$, Figure 58, shows a much smoothed structure than the Patlak derived image, Figure 51, which resembles closely the pixel derived $K$ image, Figure 50. In particular, Figure 58 is much smoother than its pixel derived counter part, Figure 57. We conclude that a-priori filtering of tissue time activity data to generate parametric images causes a greater loss of possibly relevant functional structures in comparison to the a-posteriori filtering of parametric images.

3. Conclusions and Discussion

A linear regression between Patlak derived $K$ image and pixel-by-pixel computation of $K$ is presented in Figure 59. First, the three methods of choosing bounds are compared to each other, Figure 59. The correlation between estimates is 0.99. We summarize the results in Table 14.

The estimates of $K$ from $CFLS$, regardless of the bounds used, are higher then the ones estimated by Patlak. This agrees with other observations already made in literature [29], [16], [60]. Given the fact that Patlak analysis assumes a $k_4 = 0$ value, and that some of the noise present outside the brain is also incorporated in these estimates, we conclude that $CFLS_{clst}$ is the best method for estimating parametric images. We also note that some of the higher values for the $k_4$ parametric image may be due to spill-over effects from tissue into blood. Although we do not account for this in our model, the image-derived input function should have been adjusted for this effect. Other studies, [17], point to the
Figure 48. Anisotropic diffusion filtered image for $k_3$ using data corrected for spill-over effects.

Table 14. Slope of regression line and R-Squared values comparing $K$ values using bounds from literature, Feng, [16], Piert, [46], and from clustering, [23].

<table>
<thead>
<tr>
<th>Regression Line Slope</th>
<th>R-Squared</th>
</tr>
</thead>
<tbody>
<tr>
<td>$CFLS_{Feng}$</td>
<td>.0026</td>
</tr>
<tr>
<td>$CFLS_{piert}$</td>
<td>.8603</td>
</tr>
<tr>
<td>$CFLS_{clst}$</td>
<td>1.020</td>
</tr>
</tbody>
</table>

fact that higher spill-over from tissue to blood leads to higher estimates for $k_4$, which in return lead to lower estimates for $K$. Since our values for global parametric $K$ image are higher then the those determined by Patlak analysis, we conclude that the higher values observed for $k_4$ are due to other factors, either noise or tissue-blood spill-over effects.

Following this analysis, we consider the effects of anisotropic filtering on $K$ images in Figure 60. We summarize the results in Table 15.

This data does not show conclusively if filtering improves the parametric image estimates. The regression line slope as well as the R-Squared value for $K$ when no filtering is used and a-posteriori ADF is used, are very close. Hence, we also consider the smoothness of an image to measure the efficacy of anisotropic diffusion for parametric imaging.

Although anisotropic diffusion filtering improves the quality of parametric images, for all methods, and for all parameters, the improvement is small, and not uniform for all parametric images $k_i$, $1 \leq i \leq 4$. However, we need to preserve certain abnormalities in
Figure 49. Anisotropic diffusion filtered image for $k_4$ using data corrected for spill-over effects.

Table 15. Slope of regression line and R-Squared values comparing $K$ values using no filtering, a-posteriori ADF, and a-priori tissue activity curve (TAC) filtering.

<table>
<thead>
<tr>
<th>Filter Type</th>
<th>Regression Line Slope</th>
<th>R-Squared</th>
</tr>
</thead>
<tbody>
<tr>
<td>no filtering</td>
<td>1.0308</td>
<td>.4497</td>
</tr>
<tr>
<td>a-posteriori ADF</td>
<td>1.0277</td>
<td>.444</td>
</tr>
<tr>
<td>a-priori TAC filtering</td>
<td>1.3743</td>
<td>.6791</td>
</tr>
</tbody>
</table>

our images to be able to identify region affected early in the stages of a disease. Without a-priori knowing what regions are expected to show functional differences, it is hard to assess whether smoothing parametric images offers a true improvement. Visual examination shows that a-posteriori anisotropic diffusion filtering of parametric images eliminates noise and enhances the spatial definition of edges.
Figure 50. Anisotropic diffusion filtered image for $K$ using data corrected for spill-over effects.

Figure 51. Anisotropic diffusion filtered Patlak $K$ image.
Figure 52. Anisotropic diffusion filtered image for $k_1$ using filtered output data, corrected for spill-over and partial volume effects.

Figure 53. Anisotropic diffusion filtered image for $k_2$ using filtered output data, corrected for spill-over and partial volume effects.
Figure 54. Anisotropic diffusion filtered image for $k_3$ using filtered output data, corrected for spill-over and partial volume effects.

Figure 55. Anisotropic diffusion filtered image for $k_4$ using filtered output data, corrected for spill-over and partial volume effects.
Figure 56. Anisotropic diffusion filtered image for $m_{bt}$ using filtered output data, corrected for spill-over and partial volume effects.

Figure 57. Anisotropic diffusion filtered image for $K = \frac{k_1 k_3}{k_2 + k_3}$ using filtered output data, corrected for spill-over and partial volume effects.
Figure 58. Anisotropic diffusion filtered image for $K = \frac{k_1 k_3}{k_2 + k_3}$ using Patlak analysis.

Figure 59. Pixel-wise comparison of parametric image of $K = \frac{k_1 k_3}{k_2 + k_3}$ estimated by both Patlak analysis and by pixel-by-pixel analysis, using CFLS algorithm and bounds from Feng [16], Piert [46], and clustering. The data was corrected for spill-over and partial volume effects, but no filtering was used. The last graph shows the data from all three graphs in one figure.
Figure 60. Pixel-wise comparison of anisotropic filtered parametric image for $K = \frac{k_1 k_3}{k_2 + k_3}$ estimated by both Patlak analysis and by pixel-by-pixel analysis, using CFLS algorithm, with bounds derived from clustering. The images are derived using no filtering, parametric image filtering, and Time Activity Curve (TAC) a-priori filtering. The regression line estimates gave, respectively, a slope of 1.0308, 1.0277, 1.3743, and a R-Squared value 0.4497, 0.444, 0.6791.
A thorough analysis of the FDG model, both theoretical as well as numerical, was undertaken in this work. The theoretical analysis considered here is unique in its application to dynamic PET parametric imaging. The theoretical results are confirmed experimentally as well as numerically through observations in previous studies. Specifically, the sensitivity of the model with respect to each parameter, and especially with respect to the small de-phosphorylation parameter, was presented, and two different decay time scales were observed. The time scale of decay for the de-phosphorylation parameter is an order of magnitude smaller than that of the other system parameters, in agreement with results from region of interest studies. A debatable point of ignoring the de-phosphorylation parameter was also considered. Singular perturbation analysis of the forward problem shows that the resulting solutions have the potential to differ drastically, depending on the form of the input function, the driving force of the system in our model. Recent studies, aware of the inaccuracies caused by ignoring the de-phosphorylation parameter, resort to assuming it fixed at a certain value [60]. This work makes no a-priori assumptions on any of the model parameters, providing the least biased results in comparison to other studies.

Following the relationship between linear differential equations and Volterra integrals, we introduce a slightly different matrix for the computation of the least squares solution. The Volterra matrix thus derived saves computational time, while preserving solution accuracy. Alternatively, the Volterra integral formulation can be interpreted to provide a new modeling approach to tracer kinetics. This new interpretation allows for an easy derivation of tracer models, especially when the number of compartments is large. Further studies for the construction of the Volterra kernels which explain system kinetics will be considered in future work.

Numerically, a number of methods were compared for the pixel-wise solution of the initial value problem modeling FDG dynamics. The convergence of the generalized linear least squares algorithm (GLLS) has not been previously considered in any other work we were able to identify. However, this algorithm has been used in many studies, both in simulations as well as for clinical data analysis. A derivation of the convergence behaviour was needed to understand an iteration stopping criteria which seemed to be chosen by many as an a-priori fixed iteration number. The analysis presented here shows that the convergence of the algorithm is parameter dependent, with some values leading to
convergence while others leading the algorithm to diverge from the true solution. Necessary conditions for convergence of GLLS were derived. For this work, the GLLS algorithm was adopted as a one step filtering procedure to eliminate the colored noise on the data, and was therefore renamed Filtered Least Squares (FLS) algorithm.

We applied our results to derive global parametric images in a human study. The introduction of parameter constraint in the numerical algorithm was shown to be critical for deriving qualitatively good images for real data studies. In particular, we use a data-specific constraint, derived from a cluster analysis, to obtain best estimates. Other work has employed cluster analysis to derive parametric images [60], without the use of pixel-by-pixel estimation. Our study is different from [60] first in not assuming an a-priori value for the de-phosphorylation parameter, thus allowing us to obtain a parametric image for this parameter, emphasizing its spatial variation. The results obtained by [60] also show that the remaining system parameters depend non-linearly on the de-phosphorylation parameter, hence the influence of a constant value for this parameter is not uniform in the estimation of the other parameters. Secondly, in using a clustering algorithm, other algorithm-defined parameters are introduced. Their optimization is critical to the quality of the results obtained, however methods to properly optimize these parameters still need to be developed [60]. In comparison, the methods used here do not require any optimal parameters in their calculations. Although clustering is employed to obtain parameter bounds, these results are only used to derive minimum and maximum bounds, and their accuracy is only important on average. Therefore, we conclude that a pixel-by-pixel estimation is less biased by a-priori assumptions when compared to cluster-derived global images.

The use of anisotropic diffusion filtering is also uniquely applied in this work. This filtering has been proposed and used successfully in other image restoration processes [2], yet in these applications the SNR ratio was much higher then that observed in PET data. The anisotropic diffusion filtering introduces a general framework for spatial filtering design based on robust statistics, rather than simple spatial constraint discussed in other works [60]. The flexibility of designing a nonlinear diffusion function can contribute to better filtering of the data, which can be modeled depending of the specific noise characteristics observed in human studies.

Further work is needed on validating our results in clinical studies to demonstrate the accuracy of the estimates derived. A quantitative characterization of the noise in the time activity curves as well as the speckle noise in the final parametric images can help identify specific optimal anisotropic filtering functions, and provide a full description of the statistics of the parametric imaging. A better understanding of the parametric images’ speckle noise should also lead to an improved estimate of an appropriate $\sigma$ parameter, able to discern between noise and true image discontinuities. A spatially non-uniform anisotropic filter may be then designed.

Finally, another goal of this work was to identify a new measure for the early detection of AD. Piert et. al [46] suggest that the phosphorylation rate $k_3$ is a good indicator for early detection of disease. We propose here a small improvement of this observation.
Noting that 

\[ K = \frac{k_1 k_3}{k_2 + k_3} = \frac{k_1}{k_3 + 1}. \]

we suggest that the ratio \( \frac{k_3}{k_2} \) should also be tested as an indicator of early detection of disease. Note that the amount of tracer phosphorylated, given as a rate by \( k_3 \), is affected by the amount of tracer leaving that same compartment. The value of \( K \) is affected therefore not only by increases or decreases observed in \( k_3 \), but rather by those observed in the ratio \( \frac{k_3}{k_2} \).

The ratio \( \frac{k_3}{k_2} \) is directly proportional with \( K \). Although the variation in \( k_2 \) is smaller than that observed in \( k_3 \), this new measure may provide a more sensitive marker in identifying regions affected early on.

![Image](image.png)

Figure 61. Image for \( \frac{k_3}{k_2} \), which is directly proportional to \( K \), ie an increase in \( \frac{k_3}{k_2} \) will cause an increase in \( K \), assuming \( k_1 \) remains constant.

A visual inspection of the new proposed measure in Figure 61 compared to the \( k_3 \) parametric image, Figure 48, shows regions of high phosphorylation activity in the frontal lobe that have only a moderate level of activity when using the \( \frac{k_3}{k_2} \) measure. Also, small regions of low phosphorylation activity in the posterior lobe in Figure 48 correspond to regions of relatively higher activity when measured with the new measure \( \frac{k_3}{k_2} \). We expect these differences to be significant for early disease diagnosis, yet they have to be clinically validated before one can give them an appropriate interpretation. In addition, this image shows a symmetry not present in any of the other parametric images, and hence we view this image as qualitatively better than the other parametric images.
REFERENCES


%The following codes can be used to check the computations presented in Chapter 2, Section 2.1, on Sensitivity Analysis.

% sensitivity analysis - assume k4 = 0
% this follows the analysis presented in Chapter 2

S := k1*k3/(k2+k3) + (k1*k2/(k2+k3))*exp(-(k2+k3)*t);

G1 := simplify(diff(S,k1));

G2 := simplify(diff(S,k2));

G3 := simplify(diff(S,k3));

N1 := int(G1,t=0..T);

N2 := int(G2,t=0..T);

N3 := int(G3,t=0..T);
% sensitivity analysis - assume k4 nonzero
% this follows the analysis presented in Chapter 2
S := k1*((k3+k4 - .5*(k2+k3+k4-sqrt((k2+k3+k4)^2 - 4*k2*k4)))*... 
   exp(-.5*(k2+k3+k4-sqrt((k2+k3+k4)^2 - 4*k2*k4))*t) +...
   (-k3-k4 + .5*(k2+k3+k4+sqrt((k2+k3+k4)^2 - 4*k2*k4)))*... 
   exp(-.5*(k2+k3+k4+sqrt((k2+k3+k4)^2 - 4*k2*k4))*t))/sqrt((k2+k3+k4)^2 - 4*k2*k4);

simplify(diff(S,k1));

g1 := simplify(diff(S,k2));

g3 := simplify(diff(S,k3));

g4 := simplify(diff(S,k4));

int(g1,t=0..T);

int(g2,t=0..T);

int(g3,t=0..T);

int(g4,t=0..T);
APPENDIX B
NUMERICAL ALGORITHMS - MATLAB CODES
% The following codes are referred to in Chapters 3 and 4.

% Code with spill over from the input function
% i.e. \( y = f @ \text{convolved } u + k5 \ast u, \)
% where \( k5 \) is an additional parameter, needed
% for real data processing
% \( U = \text{input} - \text{input data} - \text{plasma function} \)
% \( Y = \text{output} - \text{output data} - \text{tissue function} \)
% \( T = \text{time} - \text{needed in the integral calculation} \)

function [k, theta] = pix_kin_sp(T, U, Y, lb, ub, ...
    s_intgrU, s_intgr_tu, coefU);

% form matrix \( X \) via trapezoidal integration - Volterra formulation
dt = .001;

pp_y1 = spline(T, Y);
yint1 = trapsingInt(pp_y1, dt);

s_intgrY = [0; yint1(:)];

ty = T.*Y;

pp_ty = spline(T, ty);
tyint = trapsingInt(pp_ty, dt);

s_intgr_ty = [0; tyint(:)];

int2 = T.*s_intgrU(:) - s_intgr_tu(:);

int4 = T.*s_intgrY(:) - s_intgr_ty(:);

X = [U(:), s_intgrU(:), int2, -(s_intgrY(:)), -int4];

X = real(X); Y = real(Y);

options = optimset('LargeScale', 'off', 'disp', 'off');
theta_ls = lsqlin(X, Y, [], [], [], [], lb, ub, [], options);

[breaksY, coefY, ly, ky, dy] = unmkpp(pp_y1);

coefY = [coefY; zeros(1,4)];
\[ \theta = \text{fls}_\text{cs}(U, Y, T, \text{coefU, coefY,theta}_\text{ls}, \text{lb, ub}); \]

\[ k = \text{getK}_\text{fls} (\theta, \text{theta}_\text{ls}); \text{ return} \]
function theta = fls_cs(U, Y, T, coefU, coefY, theta_lls, lb, ub);

n = length(U);
a = [theta_lls(4); theta_lls(5)]; %output coef
lambda = rootsQuadr(a);
%calculate 1/A_prime at lambda1 and lambda2;
A_prime = 1./(2*lambda + a(1));

%calc. y1 and y2 from Laplace:
%y_i = A_prime(i)^-1 * (e^lambda_i*t_i) * 
%   integral(0,t_i, vector_to_Integrate)
%where vector_to_integrate = y(tau)*e^(-lambda_i*tau)
exp_l1 = exp(-lambda(1)*T);
exp_l2 = exp(-lambda(2)*T);

y_tk = [ (1./exp_l1).*ppexpInt1(T,coefY,lambda(1)), ... 
(1./exp_l2).*ppexpInt1(T,coefY,lambda(2))];

y_tk(1:n, 1) = (A_prime(1))*y_tk(1:n,1);
y_tk(1:n, 2) = (A_prime(2))*y_tk(1:n,2);

u_tk = [ (1./exp_l1).*ppexpInt1(T,coefU,lambda(1)), ... 
(1./exp_l2).*ppexpInt1(T,coefU,lambda(2))];

u_tk(1:n, 1) = (A_prime(1))*u_tk(1:n,1);
u_tk(1:n, 2) = (A_prime(2))*u_tk(1:n,2);

%calculate Z
z1 = lambda(1)*y_tk(1:n,1) + lambda(2)*y_tk(1:n,2);
z2 = y_tk(1:n,1) + y_tk(1:n,2);
z3 = lambda(1)*u_tk(1:n,1)+lambda(2)*u_tk(1:n,2);
z4 = u_tk(1:n,1) + u_tk(1:n,2);

rhsr = - (a(1)*(lambda(1)*y_tk(1:n,1) + lambda(2)*y_tk(1:n,2)) +... 
a(2)*(y_tk(1:n,1) + y_tk(1:n,2)));

r = Y + rhsr; r = real(r);
Z = real([ U, z3, z4, -z1, -z2]);

options = optimset('LargeScale', 'off', 'disp', 'off');

theta = lsqlin(Z, r, [], [], [], lb, ub, [], options); return
function k = getK_fls(theta, theta_old);

if nargin == 2
    k5 = theta(1);
else k5 = 0; theta_old = zeros(size(theta_new)); end

k1 = theta(2) - k5*(theta(4) - theta_old(4)); if k1 == 0
    k = zeros(6,1);
else
    %this is k1*(k3+k4)
    temp_th4 = theta(3) - k5*(theta(5) - theta_old(5));
    %exact calculation
    if k1*theta(4) == temp_th4
        k2 = 0; k3 = theta(4); k4 = 0;
        if theta(5) ~= 0 | theta(4) ~= theta(3)/k1
            disp('system is inconsistent'); end
    else
        k2 = theta(4) - temp_th4/k1;
        k4 = theta(5)/k2;
        k3 = temp_th4/k1 - k4;
    end

    if (k2+k3 == 0 | k2 == 0)
        K = k1; disp('K not defined - approximate by k1');
    else
        K = k1*k3/(k2+k3);
    end
    k = real([k1;k2;k3;k4;k5;K]);
end return