ISSN 1346-5597

NII National Institute of Informatics

NII Technical Report

Improvements to the Cluster Newton Method for Underdetermined Inverse Problems

Philippe J. Gaudreau, Ken Hayami, Yasunori Aoki,
Hassan Safouhi, and Akihiko Konagaya

NII-2013-002E
Nov. 2013

URL: http://www.nii.ac.jp/ Tokyo 101-8430, JAPAN +81-3-4212-2000 FAX: +81-3-3556-1916

Improvements to the Cluster Newton Method for Underdetermined Inverse Problems

Gaudreau, P.[1)], Hayami, K.[2) 1], Aoki, Y.[3)], Safouhi, H.[4)] and Konagaya, A.[5)]

1) University of Alberta
2) National Institute of Informatics
3) Uppsala University
4) University of Alberta / Campus Saint-Jean
5) Tokyo Institute of Technology

Abstract

The Cluster Newton method (CN method) has proved to be very efficient at finding multiple solutions to underdetermined inverse problems. In the case of pharmacokinetics, underdetermined inverse problems are often given extra constraints to restrain the variety of solutions. In this paper, we propose an algorithm based on the two parameters of the Beta distribution to find families of solution near a solution of interest. This allows for a much greater control of the variety of solutions that can be obtained with the CN method. In addition, this algorithm facilitates the task of obtaining pharmacologically feasible parameters. Moreover, we also make some improvements to the original CN method including an adaptive margin of error for the perturbation of the target values and the use of an analytical Jacobian in the resolution of the forward problem.

Keywords. Cluster Newton method; Underdetermined inverse problem; Beta distribution; Pharmacokinetics

1 Introduction

In the field of pharmacokinetics, underdetermined inverse problems occur frequently. This is not surprising considering that the data that can be collected does not often explain the complex mechanics of the human body. Through mathematical models, we are able to simulate these complex behaviors and gain valuable insight into the body's pharmacokinetics. Moreover, since the quantitative data that can be collected is generally far less than the number of parameters that govern its behavior, these problems are deemed underdetermined. In [1], Aoki et al. constructed a new algorithm, which they have coined the Cluster Newton method (CN method), capable of finding multiple solutions of an underdetermined inverse problem. They use the CN method to investigate Arikuma et al.'s pharmacokinetics model [2] for the anti-cancer drug CPT-11. The CN method proved to be significantly more robust and efficient than the Levenberg-Marquardt method [3] for solving separate inverse problems with different initial iterates. As is the case for pharmacokinetics, constraints are often given to restrict the variety of solutions that can be obtained for underdetermined inverse problems. As robust and efficient as the CN method proved to be, there was a need for improvement when it came to seeking specific sets of solutions given such constraints. For a complete review of the analysis performed on Arikuma et al.'s model, we refer the readers to [4].

[1]Corresponding author: hayami@nii.ac.jp

In this paper, we present several improvements to the CN method that increase its overall performance. In addition, we develop a new component to the algorithm that allows for complete control of the variety of solutions that can be obtained. This extension to the method greatly facilitates the task of obtaining pharmacologically feasible parameters. As case examples, we will consider two models concerning the anti-cancer drug CPT-11: Arikuma et al.'s pharmacokinetics model treated in [1] and Yoshida et al.'s model treated in [5].

2 Statement of the General Underdetermined Inverse Problem

We will now give a description of the underdetermined inverse problem that we are considering in this paper. Find $\boldsymbol{x}$ such that:

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{y}^*, \tag{1}$$

where $\boldsymbol{y}^*$ is a given constant vector in $\mathbb{R}^n$, $\boldsymbol{f}$ is a vector valued function from $\mathcal{X} \subset \mathbb{R}^m$ to $\mathbb{R}^n$ with $m > n$, and the solution of (1) is not unique. We assume the following properties concerning this inverse problem:

- The evaluation of $\boldsymbol{f}$ (solving the forward problem) is computationally expensive. Thus, we would like to minimize the number of function evaluations.
- The Jacobian of $\boldsymbol{f}$ is not explicitly known.

We denote the subset $\mathcal{X}_\epsilon^* \subset \mathcal{X}$ to be the set containing all values of $\mathcal{X}$ that approximatively satisfy (1) in the sense that the maximum norm relative residual is less than ϵ, i.e.,

$$\mathcal{X}_\epsilon^* := \{\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^m : \max_{i=1,\ldots,n} |(f_i(\boldsymbol{x}) - y_i^*)/y_i^*| < \epsilon\}. \tag{2}$$

The set $\mathcal{X}_\epsilon^*$ is often infinite and unbounded. However, we are only interested in a subset of this set $\mathcal{X}_\epsilon^*$, namely the subset that is relevant in the context of the problem, and corresponds to a range of reasonable physiological parameters. We assume that we know the following regarding the relevant values of $\boldsymbol{x} \in \mathcal{X}$:

- A rough lower bound and upper bound of $\boldsymbol{x}$ is known. We will denote these quantities as $\boldsymbol{x}^L$ and $\boldsymbol{x}^U$, respectively.

Consequently, we roughly know that the solutions we are interested in are somewhere near a "box" $\mathcal{X}^0 \subset \mathcal{X}$ defined as follows:

$$\mathcal{X}^0 := \left\{ \boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^m : \max_{i=1,\ldots,m} \left| \frac{x_i - x_i^L}{x_i^U - x_i^L} \right| \leq 1 \right\}. \tag{3}$$

In summary, we seek solutions that are close to this initial box $\mathcal{X}^0$. It is important to mention that a priori, it could be possible that $\mathcal{X}^0 \cap \mathcal{X}_\epsilon^* = \emptyset$.

We can now state the problem of interest mathematically. Find a set of l column vectors $\{\boldsymbol{x}_{.j}\}_{j=1}^l$ such that

$$\boldsymbol{x}_{.j} \in \mathcal{X}_\epsilon^* \qquad \text{and} \qquad ||\boldsymbol{x}_{.j} - \boldsymbol{x}_{.j}^{(0)}||_2 \approx \min_{\boldsymbol{x} \in \mathcal{X}_\epsilon^*} ||\boldsymbol{x} - \boldsymbol{x}_{.j}^{(0)}||_2, \tag{4}$$

for $j = 1, 2, \ldots, l$, where $\boldsymbol{x}_{.j}^{(0)}$ is a randomly chosen point in the box $\mathcal{X}^0$.

3 The Cluster Newton method

In this section, we will give a brief introduction to the CN method. For a more detailed explanation of the algorithm and examples of its application, we refer the reader to [1]. The algorithm is as follows:

Algorithm 1 : Cluster Newton method

- 1: Set up the initial points and the target values.

  1-1: Randomly choose initial points $\{\boldsymbol{x}_{.j}\}_{j=1}^{l}$ in the box $\mathcal{X}^0$. These initial points should be stored as an $m \times l$ matrix $\boldsymbol{X}^{(0)}$ where each column corresponds to a point $\boldsymbol{x}_{.j} \in \mathbb{R}^m$.

  1-2: Generate randomly perturbed target values $\{\hat{\boldsymbol{y}}_{.j}\}_{j=1}^{l}$ (to maintain well-posedness of step 2-3) near $\boldsymbol{y}^*$. We choose each value $\hat{\boldsymbol{y}}_{.j}$ such that

$$\max_{i=1,2,\ldots n} \left| \frac{\hat{y}_{ij} - y_i^*}{y_i^*} \right| < \eta, \tag{5}$$

  where $\eta \in (0,1)$ is a pre-assigned target accuracy. Lastly, we place these perturbed vector $\hat{\boldsymbol{y}}_{.j}$ into an $n \times l$ matrix $\hat{\mathbf{Y}}$.

- 2: For $k = 0, 1, 2, \ldots, K$

  2-1 : Solve the forward problem for each point (column vector) in $\boldsymbol{X}^{(k)}$, i.e.,

$$\mathbf{Y^{(k)}} = \boldsymbol{f}(\boldsymbol{X}^{(\mathbf{k})}), \tag{6}$$

  where each column vector of $\mathbf{Y^{(k)}}$ corresponds to the solution of the function $\boldsymbol{f}$ at each column vector of the matrix $\boldsymbol{X}^{(k)}$, that is:

$$\boldsymbol{y}_{.j}^{(k)} = \boldsymbol{f}(\boldsymbol{x}_{.j}^{(k)}), \quad \text{for} \quad j = 1, 2, \ldots l. \tag{7}$$

  2-2 : Construct a linear approximation of $\boldsymbol{f}$, i.e.,

$$\boldsymbol{f}(\boldsymbol{x}) \approx \mathbf{A^{(k)}}\boldsymbol{x} + \boldsymbol{y}_{\mathbf{o}}^{(\mathbf{k})}, \tag{8}$$

  by fitting a hyperplane to $\mathbf{Y}^{(k)}$. The slope matrix $\mathbf{A^{(k)}}$ and the constant vector $\boldsymbol{y}_o^{(k)}$ can be found as a least squares solution of an over-determined system of linear equations:

$$\min_{\mathbf{A^{(k)}} \in \mathbb{R}^{\mathbf{n}\times\mathbf{m}},\, \boldsymbol{y}_{\mathbf{o}}^{(\mathbf{k})} \in \mathbb{R}^{\mathbf{n}}} ||\mathbf{Y}^{(k)} - (\mathbf{A^{(k)}}\boldsymbol{X}^{(\mathbf{k})} + \boldsymbol{Y}_{\mathbf{o}}^{(\mathbf{k})})||_{\mathbf{F}}, \tag{9}$$

  where $\boldsymbol{Y}_o^{(k)}$ is an $n \times l$ matrix whose columns are all $\boldsymbol{y}_o^{(k)}$.

  2-3 : Find the update vector $\boldsymbol{s}_{.j}$ for all columns of $\boldsymbol{X}^{(k)}$ using the linear approximation, i.e., find a column vector $\boldsymbol{s}_{.j}$ such that:

$$\hat{\boldsymbol{y}}_{.j} = \mathbf{A}^{(k)}(\boldsymbol{x}_{.j}^{(k)} + \boldsymbol{s}_{.j}^{(k)}) + \boldsymbol{y}_o^{(k)}, \quad \text{for} \quad j = 1, 2, \ldots, l. \tag{10}$$

  It is clear from the structure of this problem that $\mathbf{A}$ will be a rectangular matrix with more columns than rows. Hence, the equations prescribed by (10) represent an underdetermined system of linear equations. Therefore, we cannot uniquely determine the vectors $\boldsymbol{s}_{.j}^{(k)}$ that

satisfy equation (10). To restrict our solutions to a unique solution, we choose the vectors $\boldsymbol{s}_{.j}^{(k)}$ with the shortest scaled length as follows. The vector $\boldsymbol{s}_{.j}^{(k)}$ written as a matrix $\mathbf{S}^{(k)}$ are chosen as the solution to the following minimum norm solution of an underdetermined system of linear equations:

$$\min_{\mathbf{S}^{(k)} \in \mathbb{R}^{m \times l}} ||(\mathrm{diag}\,(\hat{\boldsymbol{x}}))^{-1}\mathbf{S}^{(k)}||_{\mathbf{F}}, \tag{11}$$

$$\text{such that} \quad \hat{\mathbf{Y}} = \mathbf{A}^{(k)}(\mathbf{X}^{(k)} + \mathbf{S}^{(k)}) + \mathbf{Y}_o^{(k)}. \tag{12}$$

In equation (11), we use $\hat{\boldsymbol{x}} = \frac{1}{2}(\boldsymbol{x}^L + \boldsymbol{x}^U)$.

The scaling is used in equation (11) since the order of magnitudes of the values in the vector $\boldsymbol{s}_{.j}^{(k)}$ are different.

2-4 : Find new points approximating the solution manifold $\mathcal{X}_\epsilon^*$ by updating $\mathbf{X}^{(k)}$. If it is necessary, we shrink the length of the vector $\boldsymbol{s}_{.j}^{(k)}$ until the point $(\boldsymbol{x}_{.j}^{(k)} + \boldsymbol{s}_{.j}^{(k)})$ is in the domain of the function $\boldsymbol{f}$, i.e. ,

For $j = 1, 2, \ldots, l$

While $(\boldsymbol{x}_{.j}^{(k)} + \boldsymbol{s}_{.j}^{(k)}) \notin \mathcal{X}$,

$$\boldsymbol{s}_{.j}^{(k)} = \frac{1}{2}\boldsymbol{s}_{.j}^{(k)} \tag{13}$$

End while

End for

$$\mathbf{X}^{(k+1)} = \mathbf{X}^{(k)} + \mathbf{S}^{(k)} \tag{14}$$

End for

Now, we will define three quantities used by Aoki et al. [1] to measure the overall performance of the CN method. These quantities can be calculated at each iteration of the CN method.

Definition 1. The number of points that are in the domain of $\mathbf{f}$ is defined by the cardinality of the following set:

$$L_d = \{j = 1, 2, \ldots l \quad | \quad \mathbf{x}_{.j} \in \mathcal{X} \subset \mathbb{R}^m\}\,. \tag{15}$$

Otherwise stated, the number of elements in the set L_d, $|L_d|$, represents the number of points that are in the domain of $\mathbf{f}$.

Definition 2. The number of acceptable sets of parameters $\mathbf{x}_{.j}$ generated by the CN method is defined by the cardinality of the following set:

$$L_a = \{j \in L_d \quad | \quad \mathbf{x}_{.j} \in \mathcal{X}_\epsilon^*\}\,. \tag{16}$$

Otherwise stated, the number of elements in the set L_a, $|L_a|$, represents the number of column vectors $\mathbf{x}_{.j}$, for which $\mathbf{f}(\mathbf{x}_{.j})$ is within ϵ error of the target value $\boldsymbol{y}^*$.

For the remainder of this paper, we will use $\epsilon = 10\%$. It is also obvious from the the definition of these sets that $|L_a| \leq |L_d| \leq l$.

Definition 3. The residual associated with the points $\mathbf{x}_{\cdot j}$ that are pharmacologically feasible and generated by the CN method is defined by the Euclidean norm of the relative error between $\mathbf{Y}_{\cdot j}$ and $\mathbf{y}^*$:

$$r_j = \left(\sum_{i=1}^{n} \left| \frac{y_{ij} - y_i^*}{y_i^*} \right|^2 \right)^{1/2}, \quad j \in L_d. \tag{17}$$

The mean residual for the CN method denoted by $\hat{r}$, is simply defined as the average of all residuals r_j for points that are pharmacologically feasible:

$$\hat{r} = \frac{\sum\limits_{j \in L_d} r_j}{|L_d|} = \frac{\sum\limits_{j \in L_d} \left(\sum\limits_{i=1}^{n} \left| \frac{y_{ij} - y_i^*}{y_i^*} \right|^2 \right)^{1/2}}{|L_d|}. \tag{18}$$

4 PBPK models

In this paper, we will apply our algorithm to two different Physiologically Based Pharmacokinetic (PBPK) models concerning the pharmacokinetic effects of the drug CPT-11 in the hope of determining which model is better suited to reproduce the given experimental data. The first model, introduced by Arikuma et al. [2] is displayed in Figure 1. Here, GI stands for gastro-intestine. This

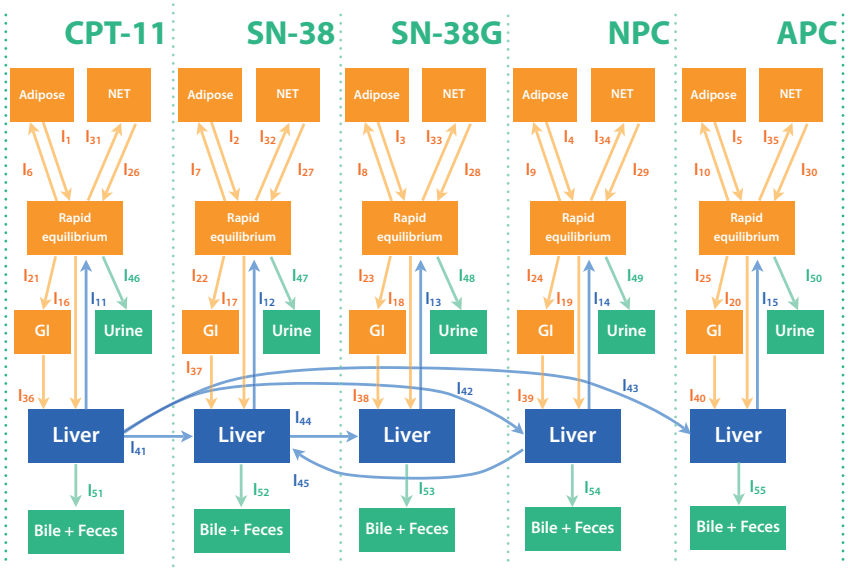

Figure 1: Arikuma et al.'s PBPK model.

particular model has four different types of pathways: the i.v. drip pathways, the blood flow pathways, the metabolic pathways, and the excretion pathways. These pathways quantitatively describe the flow rate of the compounds CPT-11, SN-38, SN-38G, NPC and APC in units of $[mg/min]$. They

are represented by the orange, blue and teal-green arrows labeled $\{l_i\}_{i=1}^{55}$ in Figure 1. There are in total 60 pharmacokinetic parameters $\{x_i\}_{i=1}^{60}$ associated with these pathways. Moreover, this pharmacokinetic model can be represented by the following stiff nonlinear system of differential equation:

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \mathbf{F}(\mathbf{u}, t, \boldsymbol{x}), \quad \mathbf{u}(0; \boldsymbol{x}) = \mathbf{0}, \tag{19}$$

where $\boldsymbol{x} \in \mathbb{R}^{60}$ are the pharmacokinetic parameters we wish to estimate, $t \in \mathbb{R}^{+}$ is time and $\mathbf{u}(t; \boldsymbol{x}) \in \mathbb{R}^{25}$ are the concentration amounts of the compounds CPT-11, SN-28, SN-38G, NPC and APC in different regions of the body at any time t when given a patient's pharmacokinetic profile $\boldsymbol{x}$. For instance, the ODE model for the concentration of SN-38G in Liver can be written as follows:

$$\begin{aligned}
\frac{\mathrm{d}u_{18}}{\mathrm{d}t} &= \left(\sum_{i \in \{18,38,44\}} l_i - \sum_{i \in \{13,53\}} l_i \right) / x_{57} \\
&= \left(x_{53} \cdot u_3(t) + \frac{x_{52}}{x_8} \cdot u_{13}(t) + \frac{x_{45} \cdot x_{50} \cdot x_{57}}{\frac{x_{40} \cdot x_{12}}{x_{22} \cdot u_{17}(t)} + 1} \right) / x_{57} \\
&\quad - \left(\frac{x_{52} + x_{53}}{x_{13}} \cdot u_{18}(t) + \frac{x_{33} \cdot x_{23}}{x_{13}} \cdot u_{18}(t) \right) / x_{57}.
\end{aligned} \tag{20}$$

The concentrations $\mathbf{u}_i$ are represented by the orange and blue boxes in Figure 1. For a more detailed explanation on the significance of each parameter and the general form of the function $\mathbf{F}(\mathbf{u}, t, \boldsymbol{x})$, we refer the reader to [1, 2]. The first five given values of $\boldsymbol{y}^* \in \mathbb{R}^{10}$ in equation (1) correspond to the accumulated concentrations in time of CPT-11, NPC, APC, SN-38 and SN-38G in Urine. The last 5 remaining entries correspond to the accumulated concentrations in time of CPT-11, NPC, APC, SN-38 and SN-38G in Bile and Feces. These excretion values are represented by the teal-green boxes in Figure 1. Moreover, these accumulated concentrations are given by the following expressions:

$$f_i(\boldsymbol{x}) = \begin{cases} \int_0^\infty x_{i+25}\, x_{i+20}\, u_i(s, \boldsymbol{x}) \mathrm{d}s & \text{for} \quad i = 1, \ldots 5, \\ \int_0^\infty (x_{i+25} x_{i+15}) / x_{i+5}\, u_{i+10}(s, \boldsymbol{x}) \mathrm{d}s & \text{for} \quad i = 6, \ldots 10. \end{cases} \tag{21}$$

Equation (21) corresponds to the function $\boldsymbol{f}$ in equation (1), although, for numerical purposes, the integration is performed from 0 to 13050 minutes. The domain for the function $\boldsymbol{f} : \mathcal{X} \subset \mathbb{R}^{60} \to \mathbb{R}^{10}$ is given by the following:

$$\mathcal{X} = \left\{ \boldsymbol{x} \in \mathbb{R}^{60} \,\middle|\, \boldsymbol{x} > 0 \text{ and } \sum_{j=55}^{58} x_j < 1000 \right\}, \tag{22}$$

where the inequality $\boldsymbol{x} > 0$ is taken entrywise.

The second PBPK model concerning the pharmacokinetics effects of the drug CPT-11 was introduced by Yoshida et al.'s in [5]. The model structure is slightly different from Arikuma et al.'s model, but some aspects remain the same. For example, in Yoshida et al.'s model, there is a return pathway between the intestines and the liver. This pathway is not included in Arikuma et al.'s model. The model still relies on four different types of pathways: the i.v. drip pathways, the blood flow pathways, the metabolic pathways and the excretion pathways. Similarly, these pathways quantitatively describe the flow rate of the compounds CPT-11, SN-38, SN-38G, NPC and APC in units of $[mg/min]$. On the other hand, there are in total 56 pharmacokinetic parameters $\{x_i\}_{i=1}^{56}$ associated with these

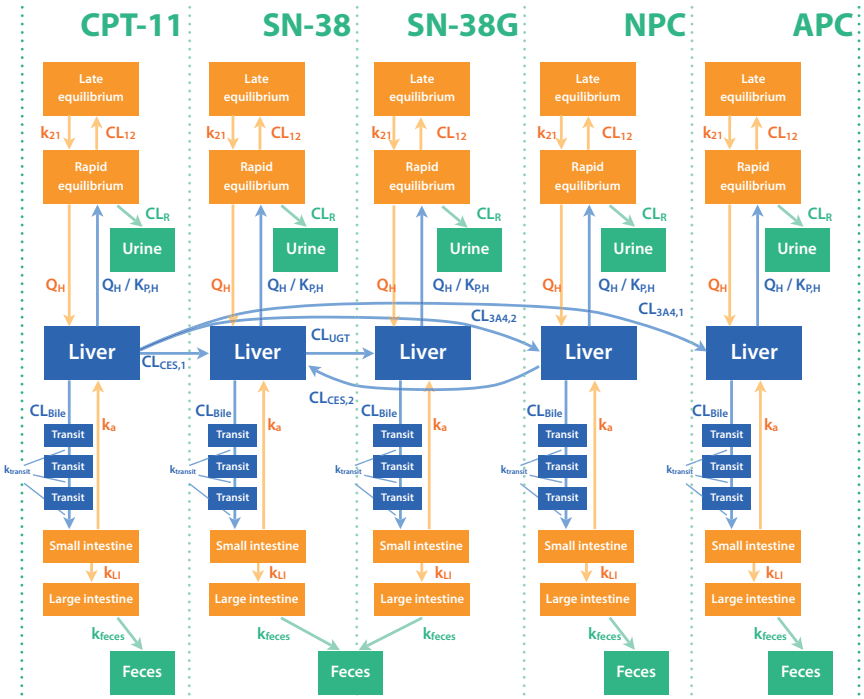

Figure 2: Yoshida et al.'s PBPK model.

pathways. The labeling of these parameters has no relation to the parameters in Arikuma et al.'s model. The model structure is presented in Figure 2. Similarly to Arikuma et al.'s model, Yoshida et al.'s model can also be represented by a stiff system of differential equation of the form:

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \mathbf{G}(\mathbf{u}, t, \boldsymbol{x}), \quad \mathbf{u}(0; \boldsymbol{x}) = \mathbf{0}, \tag{23}$$

where $\boldsymbol{x} \in \mathbb{R}^{56}$ are the pharmacokinetic parameters we wish to estimate, $t \in \mathbb{R}^{+}$ is time and $\mathbf{u}(t; \boldsymbol{x}) \in \mathbb{R}^{40}$ are the concentration amounts of the compounds CPT-11, SN-38, SN-38G, NPC and APC in different regions of the body at any time t given a patient's pharmacokinetic profile $\boldsymbol{x}$. Unlike Arikuma et al.'s model, the function $\mathbf{G}$ is linear in $\mathbf{u}$. The concentrations $\mathbf{u}$, corresponds to the orange and blue boxes in Figure 2. For a more detailed explanation on the significance of each parameter and the general form of the function $\mathbf{G}(\mathbf{u}, t, \boldsymbol{x})$, we refer the reader to [5]. The first 5 given values of $\boldsymbol{y}^{*} \in \mathbb{R}^{9}$ in equation (1) correspond to the accumulated concentrations in time of CPT-11, NPC, APC, SN-38 and SN-38G in Urine. The last 4 remaining entries correspond to the accumulated concentrations in time of CPT-11, NPC, APC and the mixture of SN-38 and SN-38G in Feces. These excretion values are represented by the teal-green boxes in Figure 2. Moreover, these excretion values are given by the following expressions:

$$f_i(\boldsymbol{x}) = \begin{cases} \int_0^{\infty} x_{5+i}\, u_i(s, \boldsymbol{x})\mathrm{d}s & \text{for} \quad i = 1, \dots 5 \\ \int_0^{\infty} x_{21}\, u_{51}(s, \boldsymbol{x})\mathrm{d}s & \text{for} \quad i = 6 \\ \int_0^{\infty} \{x_{22}\, u_{52}(s, \boldsymbol{x}) + x_{23}\, u_{53}(s, \boldsymbol{x})\}\mathrm{d}s & \text{for} \quad i = 7 \\ \int_0^{\infty} x_{16+i}\, u_{46+i}(s, \boldsymbol{x})\mathrm{d}s & \text{for} \quad i = 8, 9. \end{cases} \tag{24}$$

Equation (24) corresponds to the function $\boldsymbol{f}$ in equation (1), although, for numerical purposes, the integration is performed from 0 to 13050 minutes. The domain for the function $\boldsymbol{f} : \mathcal{X} \subset \mathbb{R}^{56} \to \mathbb{R}^{9}$

is given by the following:

$$\mathcal{X} = \left\{ \boldsymbol{x} \in \mathbb{R}^{56} \,\middle|\, \boldsymbol{x} > 0 \right\}, \tag{25}$$

where the inequality $\boldsymbol{x} > 0$ is taken entrywise.

5 The Beta Algorithm

In this section, we will present our Beta algorithm for finding multiple solutions near a solution of interest utilizing the two parameters of the Beta distribution and the CN method. First, we will define the beta distribution and explain its implication in the CN method. Secondly, we will discuss how to utilize the two parameters of the Beta distribution and the CN method to find solutions near a solution of interest.

5.1 The Beta distribution

To implement the CN method, one has to first distribute l column vectors $\{\boldsymbol{x}_{\cdot j}\}_{j=1}^{l}$ inside the initial box $\mathcal{X}^0$ defined by equation (3). The initial distribution of these points has a dramatic impact on the type of output solutions of the CN method. In [1], Aoki distributed these points using a uniform distribution. Although proving to be quite successful at finding a variety of solutions, the uniform distribution is not successful in treating extra constraints added to the problem. In practice, these added constraints often concern fitting experimental data with regard to the behavior of the function $\boldsymbol{u}(\boldsymbol{x}, t)$ in equation (33) described below. With this in mind, the Beta distribution is an excellent candidate as an initial distribution since it is an extension which allows for a non-uniform distribution. Now, we will give a short definition of the Beta distribution.

Definition 4. The Beta distribution denoted by $Beta(\alpha, \beta)$ is defined by the following probability density function:

$$f(x; \alpha, \beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha, \beta)}, \quad x \in [0, 1], \quad \alpha, \beta > 0, \tag{26}$$

where the normalization constant $B(\alpha, \beta)$ is the Beta function. The Beta function is naturally defined by the following integral:

$$\begin{aligned} B(\alpha, \beta) &= \int_0^1 x^{\alpha-1}(1-x)^{\beta-1} \mathrm{d}x \\ &= \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}, \end{aligned} \tag{27}$$

where $\Gamma(x)$ is the Gamma function.

Incidentally, if $\alpha = \beta = 1$, we recover the uniform distribution on the interval $[0, 1]$. In Figure 3, we plot the Beta probability density function given by equation (26) for different values of α and β. The value of x for which the Beta probability density function (shown in equation (26)) attains its maximum value (otherwise known as the Mode) is given by:

$$\text{Mode}(Beta(\alpha, \beta)) = \frac{\alpha - 1}{\alpha + \beta - 2}, \quad \text{for} \quad \alpha, \beta \geq 1 \text{ and } (\alpha, \beta) \neq (1, 1). \tag{28}$$

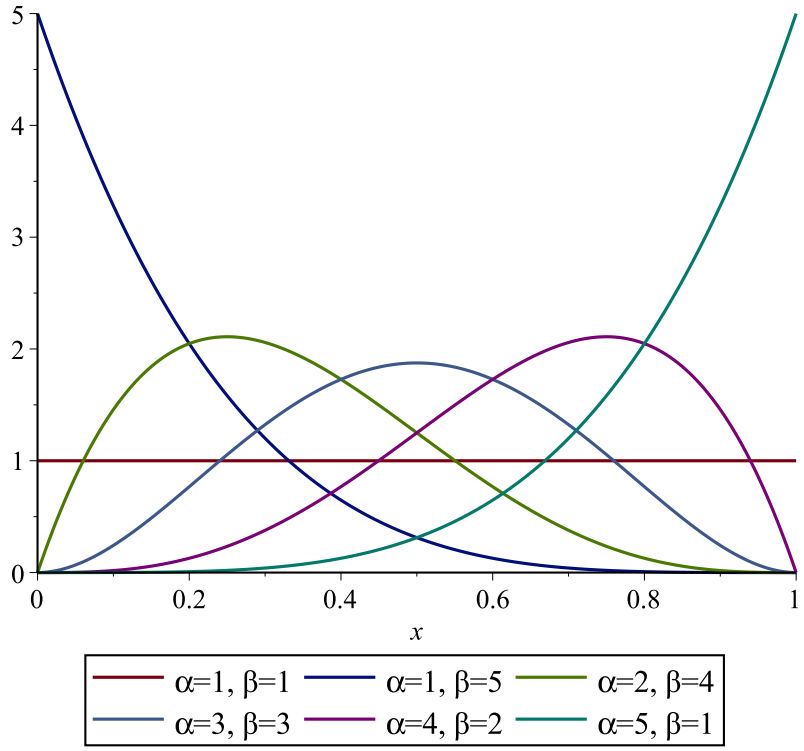

Figure 3: Different shapes of the Beta distribution for different values of α and β

The mean squared deviation from the mode (MSDM) of a Beta distribution is given by :

$$\begin{aligned} \text{MSDM}(Beta(\alpha,\beta)) &= E\left[(Beta(\alpha,\beta) - Mode(Beta(\alpha,\beta)))^2\right] \\ &= \int_0^1 \left(x - \frac{\alpha-1}{\alpha+\beta-2}\right)^2 f(x;\alpha,\beta)\mathrm{d}x \\ &= \frac{\alpha^2(\beta+1)+\alpha-6\alpha\beta+\beta+\beta^2(\alpha+1)}{(\alpha+\beta)(\alpha+\beta+1)(\alpha+\beta-2)^2}. \end{aligned} \tag{29}$$

As stated in step 1-1 of the CN method shown in section 3, we first need to randomly distribute the points $\boldsymbol{x}_{.j}$ inside the box $\mathcal{X}^0$ described in equation (3). In [1], the points $\boldsymbol{x}_{.j}$ were selected in the following way. First, an $m \times l$ random matrix whose entries are chosen uniformly on the interval $[0,1]$ is created. We shall denote this matrix by $\boldsymbol{U}(0,1)$. Secondly, using this matrix, the matrix $\boldsymbol{X}^{(0)}$ as described in step 1-1 is created as follows:

$$\boldsymbol{X}^{(0)} = \boldsymbol{x}^L \mathbf{1}_{1\times l} + \text{diag}\,(\boldsymbol{x}^U - \boldsymbol{x}^L)\,\boldsymbol{U}(0,1), \tag{30}$$

where $\mathbf{1}_{1\times l}$ is a row vector of dimension $1 \times l$ with all entries being 1. Since the Beta distribution is simply an extension of the uniform distribution, we propose the following initial distribution for the points $\boldsymbol{x}_{.j}$. First, we similarly create an $m \times l$ random matrix whose entries are chosen according to a $Beta(\alpha,\beta)$ on the interval $[0,1]$ with $\alpha \geq 1$ and $\beta \geq 1$. We shall denote this matrix by $\boldsymbol{Beta}(\alpha,\beta)$. Hence, our matrix $\boldsymbol{X}^{(0)}$ can be constructed in a similar fashion:

$$\boldsymbol{X}^{(0)} = \boldsymbol{x}^L \mathbf{1}_{1\times l} + \text{diag}\,(\boldsymbol{x}^U - \boldsymbol{x}^L)\boldsymbol{Beta}(\alpha,\beta). \tag{31}$$

Since the Beta distribution is defined on the interval $[0,1]$, we can see that each point $\boldsymbol{x}_{.j}$ is still bounded by $\boldsymbol{x}^L$ and $\boldsymbol{x}^U$. The parameters α and β introduce some bias within this range and cluster the points around the mode of the Beta distribution given by equation (28). Further, if we allow the Beta distribution for each parameter x_i, $i = 1, 2, \ldots, m$ to be different, we have

$$\boldsymbol{X}^{(0)} = \boldsymbol{x}^L \mathbf{1}_{1\times l} + \operatorname{diag}(\boldsymbol{x}^U - \boldsymbol{x}^L) \begin{bmatrix} \boldsymbol{Beta}(\alpha_1, \beta_1) \\ \boldsymbol{Beta}(\alpha_2, \beta_2) \\ \vdots \\ \boldsymbol{Beta}(\alpha_m, \beta_m) \end{bmatrix}, \tag{32}$$

where now $\boldsymbol{Beta}(\alpha_i, \beta_i)$, $i = 1, 2, \ldots, m$, is a random row vector of length l where entries are chosen according to the distribution $Beta(\alpha_i, \beta_i)$. It would be convenient to have an algorithm capable of obtaining the optimal value of α_i and β_i for each parameter x_i. In the following subsection, we will present an algorithm for obtaining the best value of α_i and β_i when an extra constraint is provided.

5.1.1 Finding a suitable initial distribution

Many models including Arikuma et al.'s model [2] and Yoshida et al.'s model [5] can be expressed by the following system of ordinary differential equations:

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = \mathbf{K}(\mathbf{u}, t, \boldsymbol{x}), \quad \mathbf{u}(0; \boldsymbol{x}) = \mathbf{0}, \tag{33}$$

where $\mathbf{u}(t, \boldsymbol{x}) \in \mathbb{R}^d$, $t \in \mathbb{R}$, $\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^m$ and $\mathbf{K} : (\mathbb{R}^d \times \mathbb{R} \times \mathcal{X}) \to \mathbb{R}^d$. Moreover, the function $\boldsymbol{f}$ in equation (1) is often given by a general equation of the form:

$$\boldsymbol{f}(\boldsymbol{x}) = \lim_{t \to a} \boldsymbol{g}(t, \boldsymbol{x}, \mathbf{u}), \tag{34}$$

for some well defined vector valued function $\boldsymbol{g} : \mathbb{R}^+ \times \mathcal{X} \times \mathbb{R}^d \to \mathbb{R}^n$.

Often, one can obtain experimental data concerning $\mathbf{u}(t, \boldsymbol{x})$ in equation (33) at different times $\{t_i\}_{i=1}^p$. Suppose we are given a set of mean values $\hat{\mathbf{h}}_i \in \mathbb{R}^q$ along with standard deviations $\mathbf{s}_i \in \mathbb{R}^q$ at each time t_i, $i = 1, 2, \ldots, p$.

Moreover, suppose we can model these values by a function $\mathbf{h}(t_i, \boldsymbol{x}, \mathbf{u})$ where $\mathbf{h} : \mathbb{R}^+ \times \mathcal{X} \times \mathbb{R}^d \to \mathbb{R}^q$ is a vector valued function depending on $\mathbf{u}(t, \boldsymbol{x})$. The explicit form for the function $\mathbf{h}(t_i, \boldsymbol{x}, \mathbf{u})$ for Arikuma et al.'s model and Yoshida et al.'s model are given by equation (51) and (52), respectively. Given this extra information, we wish to maximize the number of vectors $\boldsymbol{x}$ such that we approximatively cover the entire range:

$$\hat{\mathbf{h}}_i - \mathbf{s}_i \;\leq\; \mathbf{h}(t_i, \boldsymbol{x}, \mathbf{u}) \;\leq\; \hat{\mathbf{h}}_i + \mathbf{s}_i. \tag{35}$$

In this way, we can obtain an interval for all the elements of the vector $\boldsymbol{x}$. These intervals will correspond to the solution manifold given this extra constraint. After running the CN method of Algorithm 1 for K steps starting from a uniform distribution with the initial points $\{\boldsymbol{x}_{.j}\}_{j=1}^l$, one obtains $|L_d| \leq l$ new points $\tilde{\boldsymbol{x}}_{.j}$, $j = 1, 2, \ldots, |L_d|$. From these $|L_d|$ new points, we can determine the new range for each parameter x_i, $i = 1, 2, \ldots, m$. Let $\tilde{\boldsymbol{x}}_{.j} = (\tilde{x}_{1,j}, \ldots, \tilde{x}_{m,j})^T$. The new lower and upper bounds can be determined by the following expressions:

$$\tilde{x}_i^L \;=\; \min_{j \in L_d} \{\tilde{x}_{ij}\}, \quad i = 1, \ldots m, \tag{36}$$

$$\tilde{x}_i^U = \max_{j \in L_d} \{\tilde{x}_{ij}\}, \quad i = 1, \dots m, \tag{37}$$

where $\tilde{x}_i^L, \tilde{x}_i^U$ are the new lower bound and upper bound for each parameter x_i, respectively. In other words, the CN method moves the initial box to a new location.

$$\begin{aligned} \mathcal{X}^0 &= \left\{ \boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^m : \max_{i=1,\dots,m} \left| \frac{x_i - x_i^L}{x_i^U - x_i^L} \right| \leq 1 \right\} \\ \text{CN Method} &\Downarrow \\ \tilde{\mathcal{X}} &= \left\{ \tilde{\boldsymbol{x}} \in \mathcal{X} \subset \mathbb{R}^m : \max_{i=1,\dots,m} \left| \frac{\tilde{x}_i - \tilde{x}_i^L}{\tilde{x}_i^U - \tilde{x}_i^L} \right| \leq 1 \right\}. \end{aligned}$$

As an example, in Figure 4 we have shown visually through the use of a histogram how the range of parameter x_{32} in Arikuma et al.'s model changes after an implementation of the CN method. Furthermore, within the set of newly obtained points $\{\tilde{\boldsymbol{x}}_{.j}\}_{j=1}^{|L_d|}$, we can find a column vector $\boldsymbol{x}^\star$ with

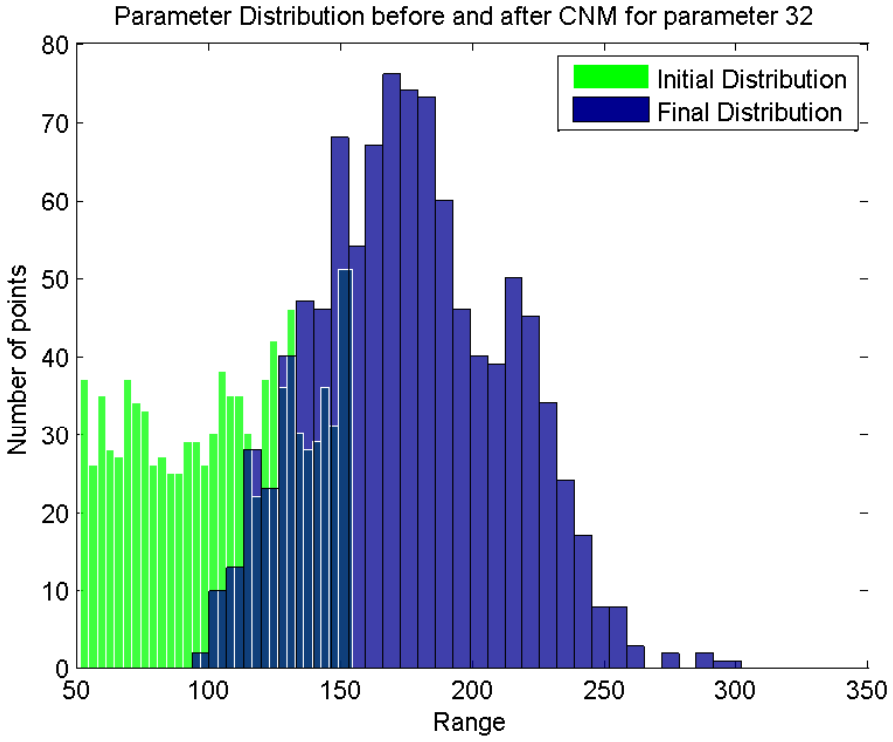

Figure 4: Initial and Final distribution of parameter x_{32} using an initial uniform distribution. The green columns correspond to the initial distribution of points before the implementation of the CN method. The blue columns correspond to the distribution of x_{32} after the implementation of the CN method.

the lowest weighted absolute norm of the error with respect to this experimental data. We define the weighted absolute norm of the error with respect to this experimental data simply as the weighted Euclidean norm of the difference between simulated and experimental data:

$$\rho_j = \left(\sum_{i=1}^{p} \sum_{k=1}^{q} \left| \frac{h_k(t_i, \boldsymbol{x}_{.j}, \mathbf{u}) - \hat{h}_{i,k}}{s_{i,k}} \right|^2 \right)^{1/2}, \quad j \in L_d. \tag{38}$$

Hence, the column vector $\boldsymbol{x}^\star$ with the lowest weighted absolute norm of the error with respect to this experimental data is given by:

$$\boldsymbol{x}^\star = \arg\min_{j \in L_d} \rho_j. \tag{39}$$

We can now apply the CN method a second time with our new initial box:

$$\tilde{\mathcal{X}} = \left\{ \tilde{\boldsymbol{x}} \in \mathcal{X} \subset \mathbb{R}^m : \max_{i=1,\ldots,m} \left| \frac{\tilde{x}_i - \tilde{x}_i^L}{\tilde{x}_i^U - \tilde{x}_i^L} \right| \leq 1 \right\}. \tag{40}$$

Within this box, we will set the value $\boldsymbol{x}^\star$ to be the mode of the new initial distribution. In this way, we can find a variety of solutions around the best fit solution $\boldsymbol{x}^\star$. We can set $\boldsymbol{x}^\star$ to be the mode of the new initial distribution by using the expression for the mode of the Beta distribution given in equation (28):

$$x_i^\star = \tilde{x}_i^L + (\tilde{x}_i^U - \tilde{x}_i^L) \left(\frac{\alpha_i - 1}{\alpha_i + \beta_i - 2} \right), \quad i = 1, \ldots m. \tag{41}$$

Here, we have two unknowns α_i and β_i. If we constrain our problem to

$$\alpha_i + \beta_i = D, \quad i = 1, \ldots, m, \tag{42}$$

where $D > 2$ is a constant, we have:

$$\alpha_i = 1 + \left(\frac{x_i^\star - \tilde{x}_i^L}{\tilde{x}_i^U - \tilde{x}_i^L} \right) (D - 2), \quad i = 1, \ldots m \tag{43}$$

$$\beta_i = D - \alpha_i, \quad i = 1, \ldots m. \tag{44}$$

To recapitulate, after applying this procedure, each parameter $x_i, i = 1, \ldots, m$ will have the following properties:

- $x_i \sim \tilde{x}_i^L + (\tilde{x}_i^U - \tilde{x}_i^L) Beta(\alpha_i, D - \alpha_i), \quad i = 1, \ldots, m, \quad (D > 2),$

  with $\alpha_i = 1 + \left(\frac{x_i^\star - \tilde{x}_i^L}{\tilde{x}_i^U - \tilde{x}_i^L} \right) (D - 2).$

- $Mode\,(x_i) = x_i^\star, \quad i = 1, \ldots, m,$

  where $x_i^\star$ is the i^{th} entry of the vector $\boldsymbol{x}^\star$ which corresponds the lowest weighted absolute error norm defined in equation (38).

The choice of the constant D has an enormous effect on the distribution of points around the mode. The mean squared deviation from the mode (MSDM) of a Beta distribution given by equation (29) is a good indicator of concentration of points around the mode of a Beta distribution . It is important to note that letting $D = 2$ in equations (43) and (44), we return to the uniform distribution where there is no mode. It would then be wise to select a value of D that would allow for some control over the MSDM. We will illustrate the effect of this value D on the MSDM by the following theorem.

Theorem 1. *Let x follow the following distribution: $x^L + (x^U - x^L) Beta(\alpha, D - \alpha)$, with $1 \leq \alpha \leq D - 1$, $D > 2$, $x^L < x^U$ and , $Mode(x) = x^\star$. Then we have the following inequality for the MSDM.*

$$l(D)(x^U - x^L)^2 \leq E\left[(x - x^\star)^2\right] \leq u(D)(x^U - x^L)^2, \tag{45}$$

where $E(g(x)) = \int_0^1 g(x) f(x;\alpha,\beta)dx$ *and*

$$l(D) = \left\{ \begin{array}{ll} \dfrac{1}{4(D+1)}, & 2 < D \leq 8 \\ \dfrac{2}{D(D+1)}, & D > 8 \end{array} \right\} \quad \textit{and} \quad u(D) = \left\{ \begin{array}{ll} \dfrac{2}{D(D+1)}, & 2 < D \leq 8 \\ \dfrac{1}{4(D+1)}, & D > 8 \end{array} \right\}. \tag{46}$$

Proof. The proof of this Theorem can be found in the Appendix. □

This inequality (45) can be seen visually in Figure 5. As we can see from equation (45), by choosing

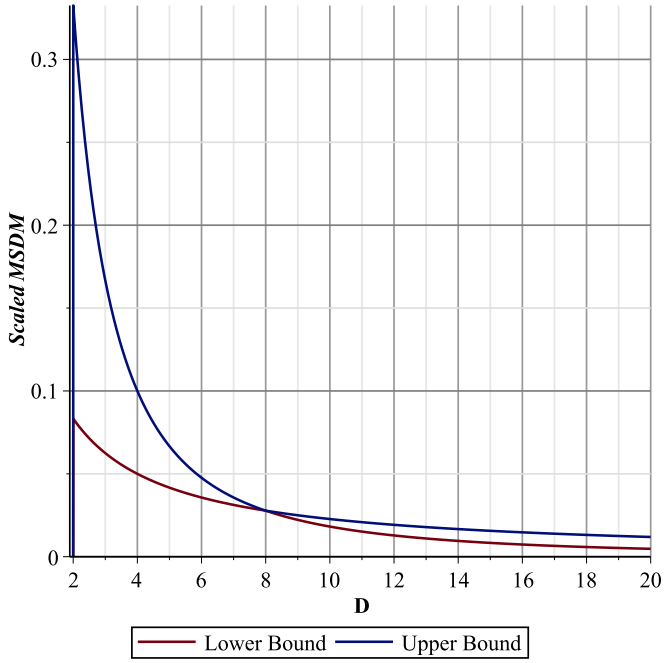

Figure 5: This figure displays the inequality of equation (45) . On the abscissa, we have the values of D. On the ordinate, we have the scaled MSDM : $E\left[(x-x^\star)^2\right]/(x^U-x^L)^2$

.

a value of $D \gtrsim 2$, the distribution for the parameters x_i will have a large MSDM with respect to its interval length squared. Consequently, the distribution of the parameters x_i will stray far away from our best fit solution $x_i^\star$. In this case, the new solutions found using the CN method with this new distribution would have a large variety and stray away from the best fit curve concerning the experimental data which we are trying to approximate. Conversely, if we pick a value of $D \gg 2$, the distribution for the parameters x_i will have a small MSDM with respect to its interval length squared. In turn, the distribution of every parameter x_i will stay near our best fit solution $x_i^\star$. In this case, the new solutions found using the CN method with this new distribution would have little variety around the best fit solution concerning the experimental data. The best value for the constant D would then depend on how much variety is wanted around the best fit solution $\boldsymbol{x}^\star$. In summary, we can rewrite the algorithm to obtain the best possible variety of solutions and range for every parameter x_i in the following way:

Algorithm 2: The Beta algorithm

- 1 - Set up the initial conditions.

$$\{\boldsymbol{x}_{.j}\}_{j=1}^{|L_d|} \leftarrow \text{Run CN method with uniform distribution in a predetermined range } (\boldsymbol{x}^L, \boldsymbol{x}^U) \tag{47}$$

- 2 - Find the new range for every parameter x_i

$$(\tilde{\boldsymbol{x}}^L, \tilde{\boldsymbol{x}}^U) \leftarrow \left(\min_{j\in L_d}\{\boldsymbol{x}_{.j}\}, \max_{j\in L_d}\{\boldsymbol{x}_{.j}\}\right) \tag{48}$$

- 3 - Find the best fit solution with respect to experimental data.

$$\boldsymbol{x}^\star = \arg\min_{j\in L_d} \rho_j. \tag{49}$$

- 4 - Choose a value for D heuristically based on the solution $\boldsymbol{x}^\star$. If the error ρ_j for $\boldsymbol{x}^\star$ is very large, in other words, if $\boldsymbol{x}^\star$ reproduces the experimental data poorly, it is better to choose a smaller value for D in the hope of obtaining a better approximation during the next run of the CN method. This will also lead to a larger variety of solutions. On the other hand, if the error ρ_j for $\boldsymbol{x}^\star$ is very small, in other words, if $\boldsymbol{x}^\star$ reproduces the experimental data satisfactorily, it is better to choose a larger value for D that will maximize the number of solutions satisfying equation (35).

- 5 - Run the Cluster Newton Method again with Beta distribution:

$$\{\tilde{\tilde{\boldsymbol{x}}}_{.j}\}_{j=1}^{|\tilde{L}_d|} \leftarrow \text{Implement CN method with } Beta(\alpha_i, D-\alpha_i) \text{ distribution in the range } (\tilde{\boldsymbol{x}}^L, \tilde{\boldsymbol{x}}^U), \tag{50}$$

  where $\alpha_i^{(m)} = 1 + \left(\dfrac{x_i^\star - \tilde{x}_i^L}{\tilde{x}_i^U - \tilde{x}_i^L}\right)(D-2)$.

- 6 - If satisfied, end ; else go to step 2.

It is important to note that the Beta algorithm can be applied multiple times if the number of solutions within the interval in equation (35) is unsatisfactory to the user. However, if the model does not reproduce the experimental data adequately, multiple iterations of the Beta algorithm will not help maximize the number of solutions satisfying equation (35).

6 Numerical Experiments

In this section, we will apply the Beta algorithm to Arikuma et al.'s model as well as Yoshida et al.'s model. The models were presented in section 4. We solved these ODEs using MATLAB 2012b stiff ODE solver ODE15s [6]. First, we will present the experimental data that will be used for both models. This data was obtained by Slatter et al. in [7]. The given values for the entries of the vector $\boldsymbol{y}^*$ are given in Table 1. This data can be approximated by equations (21) and (24) for Arikuma et al.'s model and Yoshida et al.'s model, respectively.

Table 1: Given value $\boldsymbol{y}^*$ for Arikuma et al.'s model and Yoshida et al.'s model.

Arikuma et al.'s model	$\boldsymbol{y}^*$	Yoshida et al.'s model	$\boldsymbol{y}^*$
CPT-11 in Urine	426.99	CPT-11 in Urine	426.99
SN-38 in Urine	8.19	SN-38 in Urine	8.19
SN-38G in Urine	57.56	SN-38G in Urine	57.56
NPC in Urine	2.66	NPC in Urine	2.66
APC in Urine	42.50	APC in Urine	42.50
CPT-11 in Bile/Feces	615.89	CPT-11 in Bile/Feces	615.89
SN-38 in Bile/Feces	157.07	SN-38 + SN-38G in Bile/Feces	162.21
SN-38G in Bile/Feces	5.14		
NPC in Bile/Feces	25.92	NPC in Bile/Feces	25.92
APC in Bile/Feces	158.02	APC in Bile/Feces	158.02

The additional experimental data provided for both models concerns the time course data of the total excretion profile of all compounds in Urine and in Bile/Feces. These mean values and standard deviations were also found in Slatter et al.'s work [7]. In the formalism of equation (35), we will denote the total excretion profile of all compounds in Urine at time t_i by $\hat{h}_{i,1}$, and its standard deviation error by $s_{i,1}$. Similarly, we will denote the total excretion profile of all compounds in Bile/Feces at time t_i by $\hat{h}_{i,2}$, and its standard deviation error by $s_{i,2}$. The data provided for both models are presented in Table 2. The data provided in Table 2 can be approximated by the following

Table 2: Experimental data provided for both Arikuma et al. and Yoshida et al.'s models.

t_i	$\hat{h}_{i,1}$	$s_{i,1}$	$\hat{h}_{i,2}$	$s_{i,2}$
90	100.00	66.66		
210	203.70	70.36		
330	255.55	85.18		
570	325.92	81.48		
810	374.07	92.59		
1530	425.92	96.29	114.81	196.29
2250	455.55	99.99		
2970	466.66	107.40	359.25	444.44
3690	474.07	111.11		
4410	481.48	107.40	651.85	351.85
5130	488.88	103.70		
5850	488.88	107.40	785.18	377.77
7290	492.59	111.11	833.33	374.07
8730	496.29	111.11	877.77	318.51
10170	503.70	107.40	977.77	118.51
11610			996.29	107.40
13050			996.29	107.40

functions $\mathbf{h}(t_i, \boldsymbol{x}, \mathbf{u})$ in the formalism of section 5.1.1.

In Arikuma et al.'s model, the function $\mathbf{h}(t_i, \boldsymbol{x}, \mathbf{u})$ in equation (35) is given by the following:

$$\mathbf{h}(t_i, \boldsymbol{x}, \mathbf{u}) = \begin{bmatrix} \displaystyle\int_0^{t_i} \sum_{j=1}^{5} x_{j+25}\, x_{j+20}\, u_j(s, \boldsymbol{x}) \mathrm{d}s \\ \displaystyle\int_0^{t_i} \sum_{j=6}^{10} (x_{j+25} x_{j+15})/x_{j+5}\, u_{j+10}(s, \boldsymbol{x}) \mathrm{d}s \end{bmatrix}. \tag{51}$$

In Yoshida et al.'s model, the function $\mathbf{h}(t_i, \boldsymbol{x}, \mathbf{u})$ in equation (35) is given by the following:

$$\mathbf{h}(t_i, \boldsymbol{x}, \mathbf{u}) = \begin{bmatrix} \displaystyle\int_0^{t_i} \sum_{j=1}^{5} x_{5+j}\, u_j(s, \boldsymbol{x}) \mathrm{d}s \\ \displaystyle\int_0^{t_i} \sum_{j=6}^{10} x_{15+j}\, u_{45+j}(s, \boldsymbol{x}) \mathrm{d}s \end{bmatrix}. \tag{52}$$

On a side note, the domains for the function $\boldsymbol{f}(\boldsymbol{x})$ in equation (1) is given by equation (22) and (25) for Arikuma et al.'s and Yoshida et al.'s model, respectively. As we can see, both models are constrained to the condition that every parameter be positive i.e. $\boldsymbol{x} > 0$. With this in mind, we applied the change of variable :

$$x_i = \exp(\hat{x}_i) > 0 \quad i = 1, \ldots, m, \tag{53}$$

to both models and used the CN method to find the parameters $\{\hat{x}_i\}_{i=1}^m$ instead, as in [1]. After the implementation of the CN method, we simply have to apply the exponential function to the parameters $\{\hat{x}_i\}_{i=1}^m$. This simple change of variable eliminates the need for this constraint in both models.

Although some information about the impact of the parameter D in the initial distribution of our parameters $\boldsymbol{x}$ was demonstrated in Theorem 1, we have not been able to find a method for finding the optimal value given the output of the Cluster Newton Method. However, we found heuristically that $D = 15$ gave the best results.

In the following figures, we will display the results of our Beta algorithm to both models. Figure 6 displays the results obtained by applying the Cluster Newton Method with a uniform distribution to Arikuma et al.'s model. The two figures on the left correspond to the functions $h_1(t, \boldsymbol{x}_{.j}, \mathbf{u}), j \in L_d$ and $h_2(t, \boldsymbol{x}_{.j}, \mathbf{u}), j \in L_d$ from equation (51). The figures on the right correspond to the three solutions that satisfied the lowest error with respect to experimental data defined in equation (38). We displayed the statistics from equations (15),(16),(17),(18) for this implementation of the CN method with uniform distribution in Table 3, where ϵ was set to 0.1 in (2).

As we can see from Table 3, the CN method with a uniform distribution applied to Arikuma et al.'s model yields very good approximations of its target values $\boldsymbol{y}^*$ with small residual values r_j and $\hat{r}$. In addition, we can see from Figure 6 that Arikuma et al.'s model can reproduce the total excretion of compounds in Urine quite well but fails to reproduce the total excretion of compounds in Bile/Feces. If Arikuma et al.'s model is an adequate model to describe the administration of the anti-cancer drug CPT-11, suitably the solutions near the best fit solution should converge towards a better fit solution.

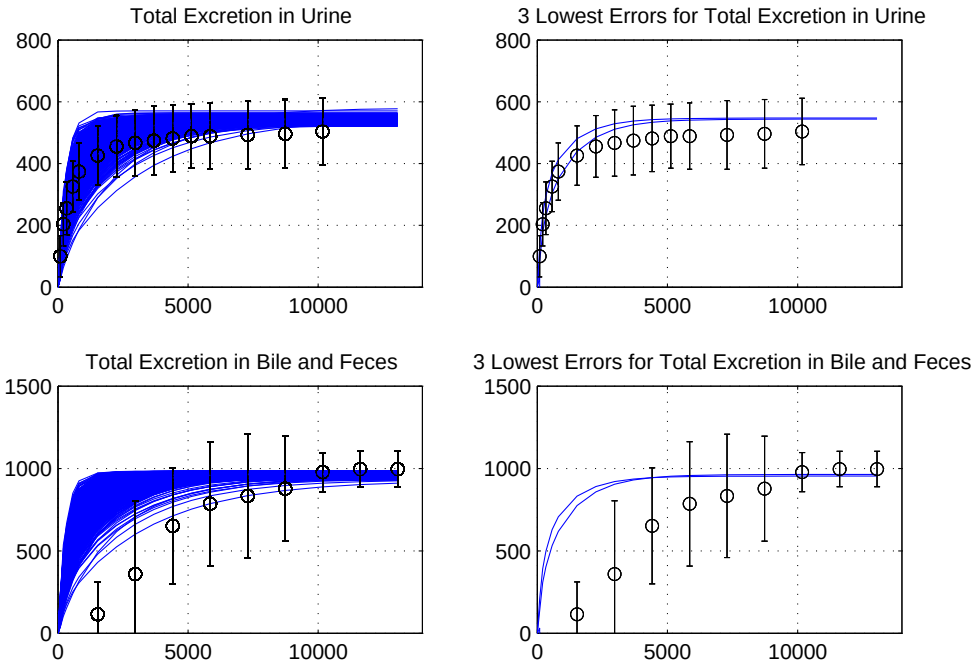

Figure 6: The total excretion of compounds in Urine and Bile/Feces as well as the three lowest error values for initial uniform distribution and 7 iterations of the CN method for Arikuma et al.'s model. The lowest error $\min_{j \in L_d} \rho_j$ was 3.49.

Table 3: Analysis of CN method with uniform distribution for Arikuma et al.'s model.

iteration step	$\lvert L_d \rvert$	$\lvert L_a \rvert$	$\hat{r}$	$\min_{j \in L_d} r_j$	$\max_{j \in L_d} r_j$	$\min_{j \in L_d} \rho_j$
1	1000	0	28.32	4.33	91.87	3.53
2	938	0	4.56	2.30	7.23	3.57
3	942	0	0.73	0.51	1.06	4.02
4	943	808	0.13	0.05	0.40	3.49
5	943	905	0.08	0.03	0.38	3.52
6	943	895	0.07	0.02	0.37	3.51
7	943	898	0.07	0.02	0.39	3.49

Implementing the Beta algorithm once with $D = 15$, we obtain the following solutions displayed in Figure 7. We displayed the statistics after one run of the Beta algorithm in Table 4. As we can see from this table, the Beta algorithm was successful in increasing the number of acceptable sets of parameters, $|L_a|$, generated by the CN method as well as lowering the mean residual error $\hat{r}$. However, by investigating Figure 7, it is clear that the Beta algorithm was not capable of obtaining better fit solutions to the total excretion of compounds in Bile/Feces. This could be due to a fallacy in the model which restricts the versatility of solutions Arikuma et al.'s model can admit.

We will now proceed similarly with Yoshida et al.'s model. Applying the CN method with a

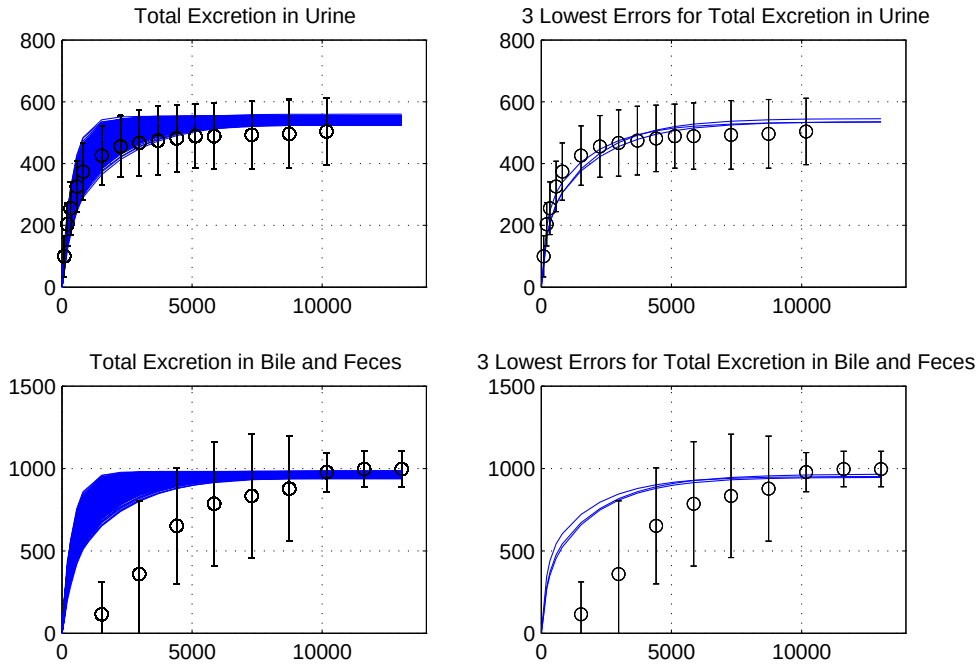

Figure 7: These figures display the Total excretion of compounds in Urine and Bile/Feces as well as the three lowest error values after one run of the Beta Algorithm for Arikuma et al's model. The lowest error $\min_{j \in L_d} \rho_j$ was 3.57.

Table 4: Analysis of CN method after one run of the Beta algorithm for Arikuma et al.'s model

iteration step	$\lvert L_d \rvert$	$\lvert L_a \rvert$	$\hat{r}$	$\min_{j \in L_d} r_j$	$\max_{j \in L_d} r_j$	$\min_{j \in L_d} \rho_j$
1	1000	0	1.25	0.21	4.49	3.59
2	1000	187	0.21	0.10	0.35	3.57
3	1000	996	0.13	0.07	0.19	3.57
4	1000	1000	0.07	0.03	0.11	3.56
5	1000	1000	0.05	0.02	0.09	3.56
6	1000	1000	0.04	0.02	0.09	3.53
7	1000	1000	0.04	0.02	0.08	3.57

uniform distribution, we obtain the following results displayed in Figure 8. Similarly, the two figures on the left correspond to the functions $h_1(t, \boldsymbol{x}_{.j}, \mathbf{u}), j \in L_d$ and $h_2(t, \boldsymbol{x}_{.j}, \mathbf{u}), j \in L_d$ from equation (52). The figures on the right correspond to the three solutions that satisfied the lowest error with respect to experimental data defined in equation (38). We displayed the statistics from equations (15),(16),(17),(18) for this implementation of the CN method with uniform distribution in Table 5. It is clear that Yoshida et al.'s model is not able to achieve 10% residual error with respect to the target value $\boldsymbol{y}^*$. However, by investigating Figure 8, it appears that Yoshida et al.'s model is better suited for finding solutions that reproduce the total excretion of compounds in Urine and Bile/Feces.

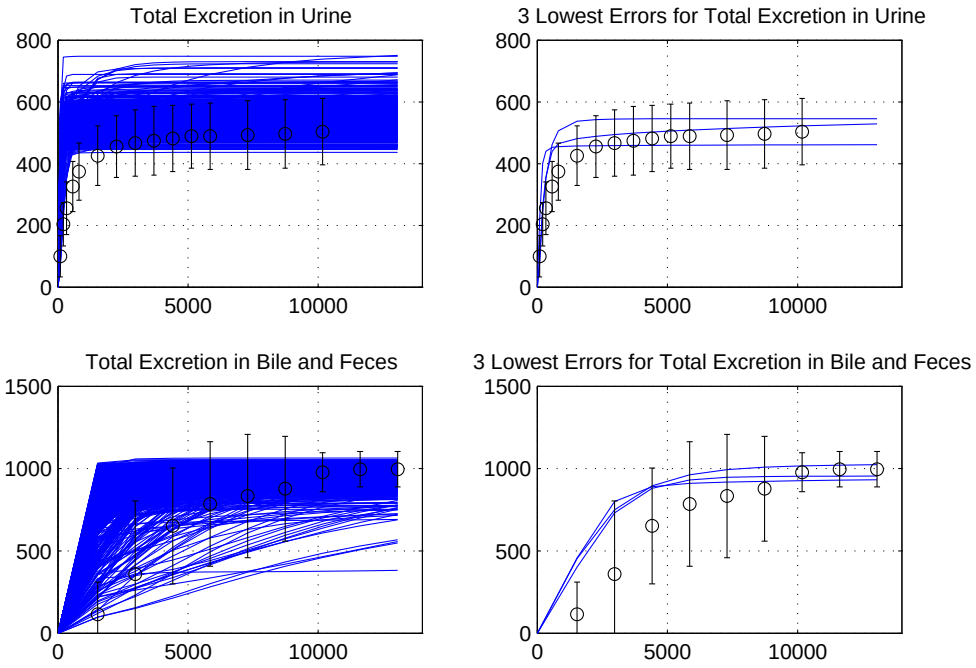

Figure 8: These figures display the Total excretion of compounds in Urine and Bile/Feces as well as the three lowest error values for initial uniform distribution and 7 iterations of the CN method for Yoshida et al.'s model. The lowest error $\min_{j \in L_d} \rho_j$ was 3.37.

Implementing the Beta Algorithm with $D = 15$, we obtain the following results displayed in Figure

Table 5: Analysis of CN method with uniform distribution for Yoshida et al.'s model

iteration step	$\lvert L_d \rvert$	$\lvert L_a \rvert$	$\hat{r}$	$\min_{j \in L_d} r_j$	$\max_{j \in L_d} r_j$	$\min_{j \in L_d} \rho_j$
1	1000	0	18.94	2.03	326.33	3.68
2	1000	0	2.74	1.33	80.64	3.02
3	1000	0	2.21	0.83	15.38	2.66
4	1000	0	1.35	0.33	6.49	3.49
5	1000	0	0.92	0.21	3.62	3.60
6	1000	2	0.77	0.14	3.49	3.48
7	1000	0	0.77	0.23	5.68	3.37

9. We displayed the statistics from equations (15),(16),(17),(18) after one run of the Beta Algorithm in Table 6. The Beta algorithm was successful in improving the number of parameter sets which could achieve 10% residual error with respect to the target value $\boldsymbol{y}^*$. Moreover, from Figure 9, we can see that the Beta algorithm did indeed find more solutions that fit better to the additional time course data of the total excretion of compounds in Urine and Bile/Feces.

Although Yoshida et al.'s model is not capable of obtaining a large number of parameters sets that

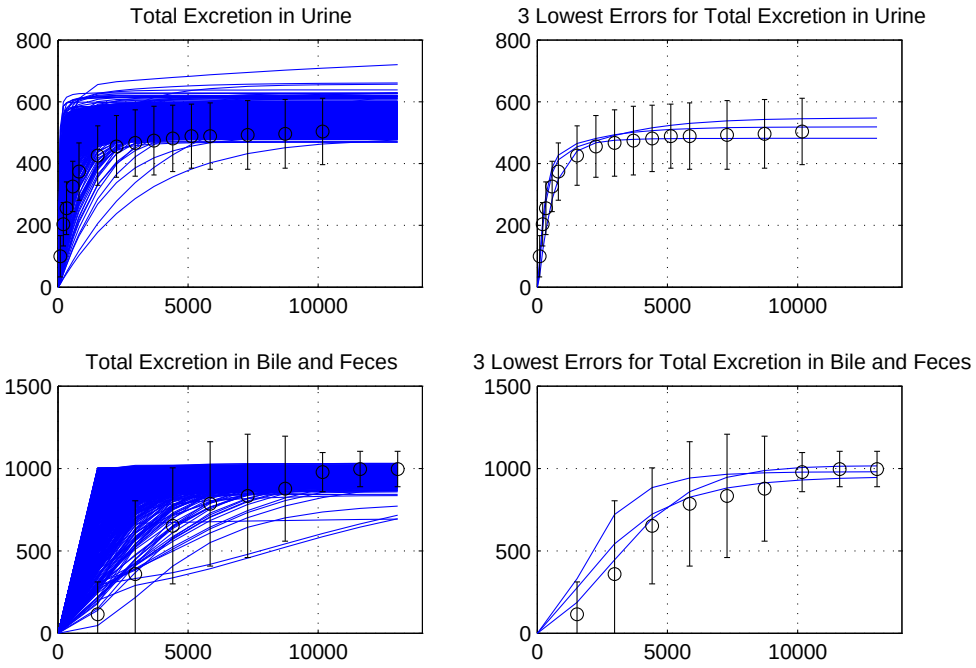

Figure 9: These figures display the Total excretion of compounds in Urine and Bile/Feces as well as the three lowest error values after one run of the Beta Algorithm. The lowest error $\min_{j \in L_d} \rho_j$ was 1.84.

Table 6: Analysis of CN method after one run of the Beta algorithm for Yoshida's model

iteration step	$\lvert L_d \rvert$	$\lvert L_a \rvert$	$\hat{r}$	$\min_{j \in L_d} r_j$	$\max_{j \in L_d} r_j$	$\min_{j \in L_d} \rho_j$
1	1000	0	1.96	0.73	29.34	1.76
2	1000	0	0.93	0.32	4.34	1.46
3	1000	4	0.56	0.14	2.20	1.74
4	1000	8	0.45	0.08	2.15	1.56
5	1000	12	0.42	0.10	2.10	2.29
6	1000	14	0.43	0.10	2.17	1.92
7	1000	9	0.50	0.12	2.15	1.84

achieve 10% residual error with respect to the target value $\boldsymbol{y}^*$, it is still the better model of the two. Since, we can't incorporate the standard deviation error on the target value $\boldsymbol{y}^*$, we are forcing the Cluster Newton method to find solutions around this point instead of a region around this point. As we have seen from Arikuma's model, even if the solutions come arbitrarily close to the target value $\boldsymbol{y}^*$, they may not simulate additional data very well. Moreover, in Yoshida et al.'s model, the Beta algorithm was capable of guiding many solutions around the additional experimental data concerning the total excretion of compounds both in Urine and Bile/Feces. This demonstrates the versatility of solutions Yoshida et al.'s model can output.

7 Conclusions

In this paper, we developed an algorithm capable of finding many solutions near a solution of interest utilizing the Cluster Newton Method and the Beta distribution. Our algorithm is based on concentrating the initial distribution of points for the CN method around the best fit solutions previously obtained with an originally uniformly distributed CN method. To reduce the number of free parameters, we introduced the constraint $\alpha_i + \beta_i = D > 2$ for $i = 1, \ldots m$. The choice of the parameter D has a significant effect on the mean squared deviation from the mode as was demonstrated by Theorem 1. It is the opinion of the authors that an algorithm capable of finding the optimal value for this constant given any data set would prove beneficial to this algorithm. However, for the purpose of this paper, we found heuristically that $D = 15$ was optimal.

We applied the Beta algorithm to two pharmacokinetic models concerning the cancer fighting drug CPT-11 and analyzed its effects. Arikuma et al.'s model was incapable of reproducing the experimental data concerning the total excretion of compounds in Urine and Bile/Feces even after using the Beta algorithm. On the other hand, Yoshida et al.'s model yielded a good diversity of solutions after the application of the Beta algorithm. This suggests that the Beta algorithm can achieve good results only if the model allows for versatility in its solutions and if the model correctly models the phenomenon. It is important to note that the uniform distribution was unable to obtain a large set of solutions near the experimental data. Utilizing the bias embedded in the Beta distribution, we were able to focus the output of the Cluster Newton Method near the experimental data in the case of Yoshida et al.'s model. Thus, the Beta algorithm is also a useful method for validating mathematical models.

Acknowledgment

This work was supported by the Grants-in-Aid for Scientific Research (C) of the Ministry of Education, Culture, Sports, Science and Technology, Japan. This work was also supported partly by a Grant-in-Aid for Scientific Research on Innovative Area Molecular Robotics (24104004) from the Ministry of Education, Culture, Sports, Science and Technology (MEXT), Japan.

8 Appendix

8.1 The Adaptive Margin

During the implementation of the CN method, there is a need to randomly perturb the target values $\boldsymbol{y}^\star$. Otherwise this least-squares problem becomes rank deficient. In [1], Aoki chose to construct the matrix $\hat{\mathbf{Y}}$ in equation (5) in the following way:

$$\hat{\mathbf{Y}} = \mathrm{diag}\,(\boldsymbol{y}^\star)\,\boldsymbol{U}(1-\eta, 1+\eta), \tag{54}$$

where $\boldsymbol{U}(1-\eta, 1+\eta)$ is a $10 \times l$ random matrix where each entry is randomly chosen from a uniform distribution on the interval $[1-\eta, 1+\eta]$ and $\eta \in (0, 1)$. In this formalism, the perturbations are the same regardless of the iteration step k. We implemented 10 iterations of the CN method for Arikuma et al.'s model with an initial uniform distribution 10 times for different values of η and took

the mean value of $|L_d|$, $|L_a|$ with $\epsilon = 10\%$ and the mean residual at each iteration. These values were defined in equations (15), (16) and (18), respectively. Figure 10 displays the mean value of $|L_d|$, $|L_a|$ and natural logarithm of the mean residual, $\log(\hat{r})$, at each iteration. At the first iteration, the

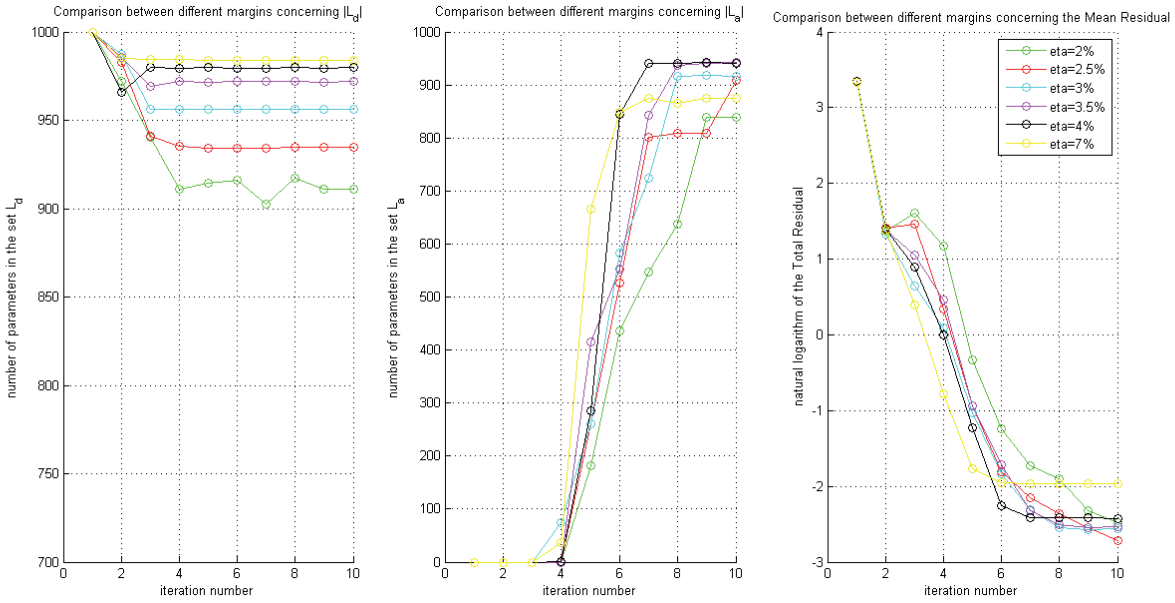

Figure 10: The leftmost figure displays the average number of elements in the set L_d; the middle figure displays the average number of elements in the set L_a and the rightmost figure displays the value of $\log(\hat{r})$ at the k^{th} iteration for different constant values of η.

parameters $\boldsymbol{x}$ might not belong to the set $\mathcal{X}_\epsilon^*$. In this sense, it would be wise to allow a broader perturbation η at the beginning and reduce the perturbations as the parameters $\boldsymbol{x}$ approach the real solution. As we can see from these figures, larger values of η improve the CN method for smaller iteration as they achieve lower residuals quicker and have a larger number of acceptable parameters. However, as the number of iterations increases, the smaller values of η reign supreme. With this in mind, we propose an adaptive margin; i.e. letting η be a function depending on the iteration step. By letting η vary at each iteration step, we found that this procedure yields a much faster convergence. After several different numerical trials, we found that

$$\eta(k) = 0.025 + 0.1/k!, \tag{55}$$

yields the best numerical results. With this selection for the margin of perturbations, only 6 iterations are needed to achieve optimal results. We presented some different choice of adaptive margins in Figure 11. It is important to note that although this particular adaptive margin proved to be optimal in our situation, it may not be optimal for different models. However, it is worth investigating when working with a new model.

8.2 Using an Analytic Jacobian when solving the Forward Problem

The model presented by Arikuma et al. [2] has the same form as the one shown in equation (19). To solve this stiff ODE system, the ODE package ODE15s in Matlab is required [6]. When using this

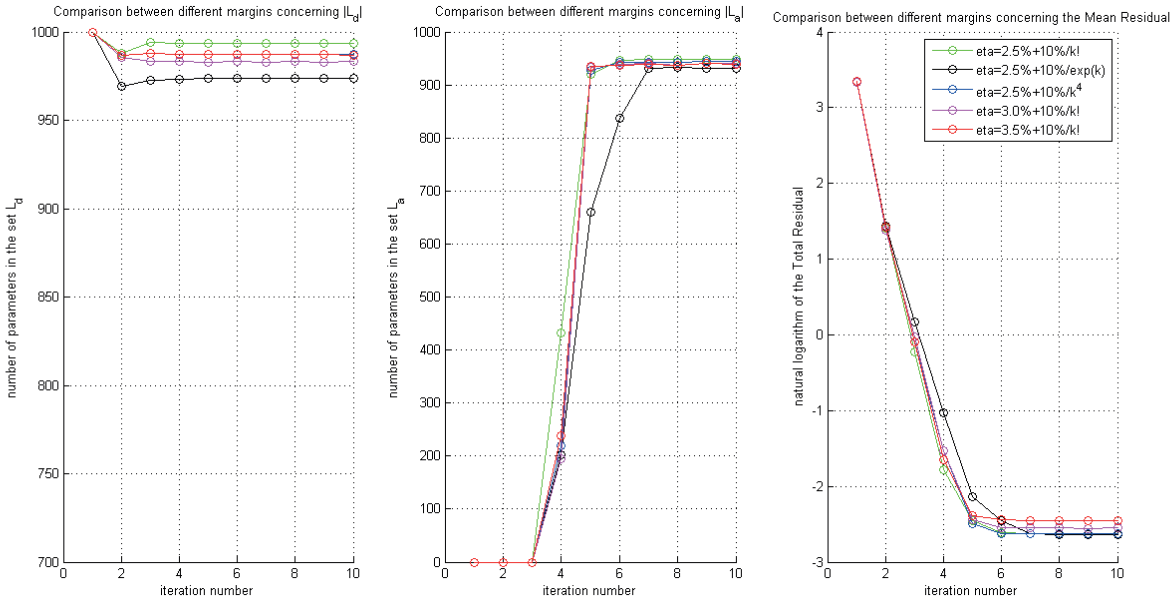

Figure 11: The leftmost figure displays the average number of elements in the set L_d; the middle figure displays the average number of elements in the set L_a and the rightmost figure displays the value of $\log(\hat{r})$ at the k^{th} iteration for different varying values of η.

ODE package, one has the option of explicitly providing the analytical Jacobian of $\mathbf{F}(\mathbf{u}, t, \boldsymbol{x})$ with respect to $\mathbf{u}$ or providing a known pattern of the Jacobian. This latter options is known as a Jpattern. If one cannot compute the Jacobian explicitly but has an idea which entries are non-zero, they can input a matrix into this ODE package in order for Matlab to numerically calculate these non-zero entries instead of the whole Jacobian. In Aoki et al.'s report, the Jpattern was used [1]. Although somewhat tedious to calculate, the Jacobian of $\mathbf{F}(\mathbf{u}, t, \boldsymbol{x})$ with respect to $\mathbf{u}$ can be calculated and coded efficiently as a sparse matrix. Figure 12 is a representation of the Jacobian for this system. Using 200 randomly selected column vectors $\{\boldsymbol{x}_{.j}\}_{j=1}^{200}$, we tested the efficiency of using both the Jpattern and analytical Jacobian. Figure 13 shows a box plot analysis of the ODE15s CPU time for these 200 different sets of parameters $\{\boldsymbol{x}_{.j}\}_{j=1}^{200}$. As we can see from this figure, the input of an analytical Jacobian inside the ODE15s solver greatly reduces the time it takes to solve the PBPK model ODE system. This is due to the fact that less function evaluations are needed in order to compute the Jacobian and approximate the optimal step size when using the analytical Jacobian. In addition, the use of an analytic Jacobian can improve the reliability and efficiency of the numerical integration.

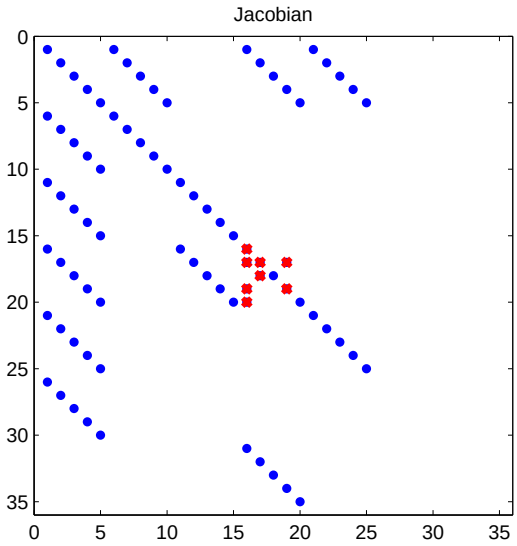

Figure 12: This figure shows the sparse structure of the Jacobian matrix of the PBPK model. There are 80 nonzero elements out of 1296. Hence, the density of this matrix is only 0.062. The blue entries are constant. That is, they are a function of $\boldsymbol{x}$ only. The red entries change with respect to time since they are a function of $\boldsymbol{u}(\boldsymbol{x}, t)$ and $\boldsymbol{x}$.

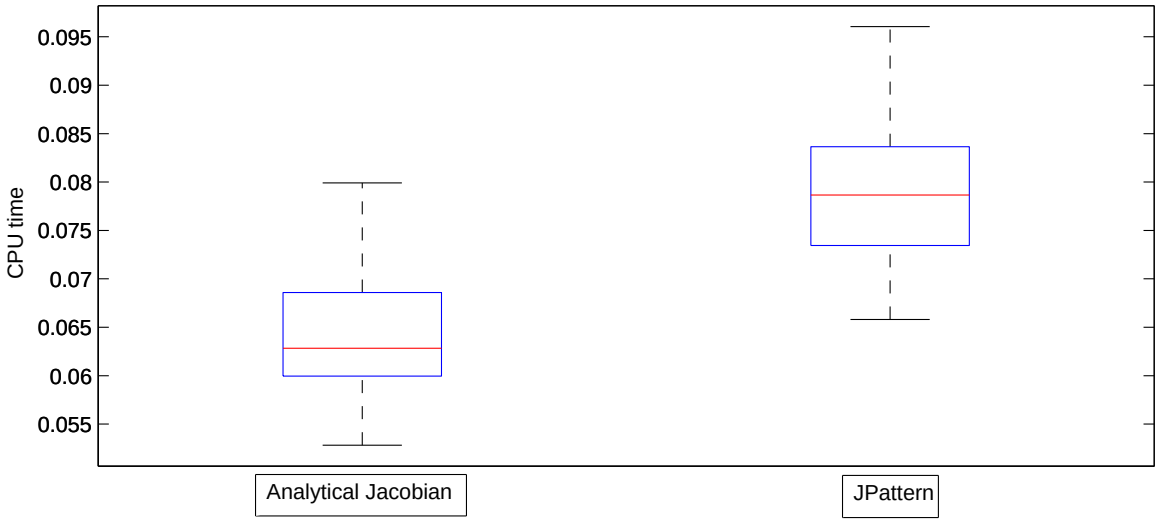

Figure 13: On the left hand side, we have the box plot for the ODE15s CPU time in seconds using the analytical Jacobian. On the right hand side we find the box plot ODE15s CPU time in seconds using a JPattern. The red line is the median CPU time, the outlines of the box represent the $25th$ and $75th$ percentiles. The top and bottom lines represent the max and min CPU time, respectively.

8.3 Proof of Theorem 1

We will now present the proof for Theorem 1.

Theorem. Let $x \sim x^L + (x^U - x^L)Beta(\alpha, D-\alpha)$ with $1 \leq \alpha \leq D-1$, $D > 2$, $x^L < x^U$ and , $Mode(x) = x^\star$. Then we have the following inequality for the MSDM.

$$l(D)(x^U - x^L)^2 \leq E\left[(x - x^\star)^2\right] \leq u(D)(x^U - x^L)^2, \tag{56}$$

where $E(g(x)) = \int_0^1 g(x)f(x;\alpha,\beta)\mathrm{d}x$ and

$$l(D) = \left\{ \begin{array}{ll} \frac{1}{4(D+1)}, & 2 < D \leq 8 \\ \frac{2}{D(D+1)}, & D > 8 \end{array} \right\} \quad \text{and} \quad u(D) = \left\{ \begin{array}{ll} \frac{2}{D(D+1)}, & 2 < D \leq 8 \\ \frac{1}{4(D+1)}, & D > 8 \end{array} \right\}. \tag{57}$$

Proof. Since $Mode(x) = x^\star$, it follows that:

$$\begin{aligned} x^\star &= Mode(x) \\ &= Mode(x^L + (x^U - x^L)Beta(\alpha, D-\alpha)) \\ &= x^L + (x^U - x^L)Mode(Beta(\alpha, D-\alpha)) \\ &= x^L + (x^U - x^L)\left(\frac{\alpha - 1}{D-2}\right) \\ \Rightarrow \quad \alpha &= 1 + \left(\frac{x^\star - x^L}{x^U - x^L}\right)(D-2). \end{aligned} \tag{58}$$

Moreover, since $x \sim x^L + (x^U - x^L)Beta(\alpha, D-\alpha)$, it follows that :

$$\frac{x - x^L}{x^U - x^L} \sim Beta(\alpha, D-\alpha). \tag{59}$$

The MSDM of a Beta distribution is given by equation (29). If we introduce the constraint that $\alpha + \beta = D$ with $D > 2$, we obtain the following:

$$E\left[\left(\frac{x - x^L}{x^U - x^L} - \frac{x^\star - x^L}{x^U - x^L}\right)^2\right] = \frac{\alpha^2(D - \alpha + 1) + \alpha - 6\alpha(D-\alpha) + (D-\alpha) + (D-\alpha)^2(\alpha+1)}{D\,(D+1)\,(D-2)^2}. \tag{60}$$

Taking in consideration that the operator $E(.)$ is a linear operator, simplifying the expression on the right hand side and completing the square in α, we obtain:

$$\frac{E\left[(x - x^\star)^2\right]}{(x^U - x^L)^2} = \left(\frac{8 - D}{D(D+1)(D-2)^2}\right)\left(\alpha - \frac{D}{2}\right)^2 + \frac{1}{4(D+1)}. \tag{61}$$

Inserting the value for $\alpha = 1 + \left(\frac{x^\star - x^L}{x^U - x^L}\right)(D-2)$ into our equation, we obtain:

$$\frac{E\left[(x - x^\star)^2\right]}{(x^U - x^L)^2} = \left(\frac{8 - D}{D(D+1)}\right)\left(\frac{x^\star - x^L}{x^U - x^L} - \frac{1}{2}\right)^2 + \frac{1}{4(D+1)}, \quad x^L \leq x^\star \leq x^U. \tag{62}$$

This is a parabola in $x^\star$. We have three cases to investigate concerning the bounds for $E\left[(x-x^\star)^2\right]$. When $\left(\frac{8-D}{D(D+1)}\right) > 0$, the parabola is concave upwards. When $\left(\frac{8-D}{D(D+1)}\right) = 0$, we have a straight line and when $\left(\frac{8-D}{D(D+1)}\right) < 0$, the parabola is concave downwards. When $2 < D < 8$, then $\left(\frac{8-D}{D(D+1)}\right) > 0$. Hence,

$$\frac{(x^U-x^L)^2}{4(D+1)} \leq E\left[(x-x^\star)^2\right] \leq \frac{2(x^U-x^L)^2}{D(D+1)}. \tag{63}$$

When $D = 8$, the MSDM is independent of the location of the mode and is given by:

$$E\left[(x-x^\star)^2\right] = \frac{(x^U-x^L)^2}{36}. \tag{64}$$

When $D > 8$, $\left(\frac{8-D}{D(D+1)}\right) < 0$. Hence, we obtain the inequality in (63) but reversed.

$$\frac{2(x^U-x^L)^2}{D(D+1)} \leq E\left[(x-x^\star)^2\right] \leq \frac{(x^U-x^L)^2}{4(D+1)}. \tag{65}$$

Combining these cases together, we obtain our desired result. □

References

[1] Y. Aoki, K. Hayami, H. De Sterck, and A. Konagaya, Cluster Newton Method for Sampling Multiple Solutions of an Underdetermined Inverse Problem: Parameter Identification for Pharmacokinetics. *SIAM Journal on Scientific Computing* (accepted for publication) (Preliminary version available as *NII Technical Report*, NII-2011-002E, National Institute of Informatics, Tokyo, 2011, at http://www.nii.ac.jp/TechReports//11-002E.html.)

[2] T. Arikuma, S. Yoshikawa, R. Azuma, K. Watanabe, K. Matsumura, and A. Konagaya, Drug interaction prediction using ontology-driven hypothetical assertion framework for pathway generation followed by numerical simulation. *BMC Bioinformatics*, 9(Suppl 6):S11, 2008.

[3] Å. Björck, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, 1996.

[4] A. Konagaya, Towards an In Silico Approach to Personalized Pharmacokinetics, Molecular Interactions, Prof. Aurelia Meghea (Ed.), ISBN: 978-953-51-0079-9, InTech, Available from: http://www.intechopen.com/books/molecular-interactions/towards-an-in-silico-approach-to-personalizedpharmacokinetics, 2012.

[5] K. Yoshida, K. Maeda, H. Kusuhara, and A. Konagaya, Estimation of feasible solution space using Cluster Newton Method: application to pharmacokinetic analysis of irinotecan with physiologically-based pharmacokinetic models, *BMC Systems Biology Supplement*, 7(Suppl 3):S3, 2013.

[6] L.F. Shampine and M.W. Reichelt, The MATLAB ODE suite, *SIAM Journal on Scientific Computing*, 18(1):1-22, 1997.

[7] J.G. Slatter, L.J. Schaaf, J.P. Sams, K.L. Feenstra, M.G. Johnson, P.A. Bombardt, K.S. Cathcart, M.T. Verburg, L.K. Pearson, L.D. Compton, L.L. Miller, D.S. Baker, C.V. Pesheck, and R.S. Lord III, Pharmacokinetics, Metabolism, and Excretion of Irinotecan (CPT-11) Following I.V. Infusion of [^{14}C]CPT-11 in Cancer Patients, *Drug Metabolism and Disposition*, 28:4, 1999.