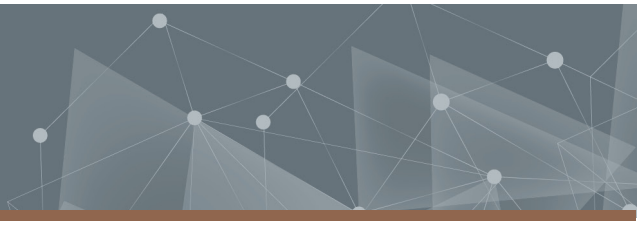
CHALMERS
UNIVERSITY OF TECHNOLOGY



Joint modeling of Longitudinal and Time-To-Event data

A Comparison of Joint versus Sequential Parameter Estimation

Master's thesis in Engineering Mathematics and Computational Science

SOPHIA AXILLUS

DEPARTMENT OF MATHEMATICAL SCIENCES

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2022
www.chalmers.se

Master's thesis 2022

Joint modeling of Longitudinal and Time-To-Event data

A Comparison of Joint versus Sequential Parameter Estimation

SOPHIA AXILLUS

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Mathematical Sciences
Division of Applied Mathematics and Statistics
Chalmers University of Technology
Gothenburg, Sweden 2022

Joint modeling of Longitudinal and Time-To-Event data
A Comparison of Joint versus Sequential Parameter Estimation
SOPHIA AXILLUS

Supervisor: Tim Cardilin, Fraunhofer Chalmers Centre
Examiner: Petter Mostad, Mathematical Sciences

Master's Thesis 2022
Department of Mathematical Sciences
Division of Applied Mathematics and Statistics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2022

Joint modeling of longitudinal and Time-To-Event data
A Comparison of Joint versus Sequential Parameter Estimation
SOPHIA AXILLUS
Department of Mathematical Sciences
Chalmers University of Technology

Abstract

Joint modeling is a technique used for parameter estimation in linked models of longitudinal and Time-To-Event (TTE) data. The goal of this is to reduce bias typically found when sequentially estimating related parameters by considering errors caused by both the models, the data, as well as between individuals (inter-individual-variability) simultaneously. The aim of this thesis was to distinguish scenarios when the joint model is suitable for use in the case of high frequent sampling.

To represent the longitudinal data, we apply a K-PD model to describe the effect of an inhibition of a measurable biomarker (response) with added random effects. This response is then linked to the TTE by using a parametric hazard equation for a given set of parameters. The set of parameters for these models are estimated with Maximum Likelihood Estimation for two approaches; a sequential and joint method. The sequential approach firstly estimates the parameters related to the K-PD model and then considers the individual simulated response as a covariate in the estimation of the TTE related parameters. In contrast, the joint model considers two contributions to the likelihood by including the TTE in order to get the full set of parameters.

The result of this is two algorithms based on the FOCE method. These algorithms are compared for several datasets with fixed parameter values during different conditions. By comparing metrics such as Relative Estimation Error and Relative Standard Error, we are able to show that the joint estimation approach provides less biased estimates for several different sampling frequencies. This is the case for most parameters but the difference is the largest for the parameters related to the TTE model. It is therefore concluded that for joint model frameworks using a joint parameter estimation should be considered. Moreover, we also show that the joint approach improves estimations when using a linked parametric hazard, especially in the case of high frequent sampling.

Keywords: joint model, NLME, survival analysis, FOCE, K-PD model.

Acknowledgements

First of all, I would like to thank my supervisor, Tim Cardilin from Fraunhofer-Chalmers Centre (FCC), for his engagement and guidance during this thesis. Tim's input and feedback has been invaluable during both the process of conducting as well as writing this thesis.

I would also like to express my sincere thanks to Mats Jirstrand from FCC, and my examiner Petter Mostad, for the guidance and advice.

Lastly, I would like to recognize all of the support and love I have continuously received from family and friends. Their belief in me kept my spirits strong throughout the process of conducting this thesis.

Thank you all!

Sophia Axillus, Gothenburg, June 2022

Nomenclature

Below is the nomenclature of indices and abbreviations that have been used throughout this thesis.

Indices

i	Individuals
j	Time points
k	Random effects

Abbreviations

FOCE	First Order Conditional Estimate
NLME	Nonlinear Mixed Effects Model
ODE	Ordinary Differential Equation
PD	Pharmacodynamics
PK	Pharmacokinetics
K-PD	(Pharmaco)kinetics-pharmacodynamics, simplification of a PK-PD model
TTE	Time-To-event
REE	Relative Estimation Error
RSE	Relative Standard error

Contents

1
Introduction

Mathematical modeling is a useful tool when drawing conclusions from experimental studies and can be used to find new interesting relationships. It has therefore several applications in pharmacology and drug development. For example; it can be used to conduct simulations or experiments which would have been too expensive, be it because of economical drawbacks or to spare the patient of invasive procedures, or in the case of comparing different drugs one might wish to account for many covariates influencing the result in order to draw the most objective conclusion [1]. Newer and more informative models have therefore been used and developed in order to understand the complexity of disease and treatment [1]. One such form of modeling is so called joint modeling, where one wishes to understand the dynamics behind a longitudinally measured response and connect it to an adverse event of sorts. This thesis, conducted in collaboration with the Fraunhofer Chalmers Centre (FCC), explores some of the challenges and advantages that joint modeling could offer to the field of mathematical modeling.

1.1 Background

General research in drug development often involves collecting both longitudinal and Time-To-Event (TTE) data. Longitudinal data are time series of sampled quantities, for example blood glucose levels every three months for several years, or lung peak expiratory flow (PEF) every day over a full year [2, 3]. The TTE data of this nature could be the time until a complication occurs, such as diabetic retinopathy, or the progression into a worse state of the disease, such as astma exacerbations [2, 3]. Joint modeling can be applicable in these cases, since the aim is to investigate an outcome of interest combined with a series of chosen measurements, which are believed to be involved with the outcome, in an effort to find the relationship between the two and make future predictions. An example of this is in clinical trials where a patient is tested frequently over a series of time with the aim of assessing the risk of an adverse event. This makes it possible to gain knowledge of the risk of the event as well as what could influence this risk, specific to each patient.

In drug development, a common approach is to model the relationship of body to drug response by applying a so called PK-PD framework for the longitudinal data and letting the event represent something the drug aims to prevent [1]. As for PK-PD modeling, the first part is in reference to pharmacokinetics (PK), where one wishes to model the system's response to the drug, and the second to represent

the pharmacodynamics (PD), which is the drug's effect on the system [1, 4]. The system used in PK models are commonly described as compartment models; where different compartments, e.g. blood stream or organ, interact with one another [1]. As for the PD, since the effect of the drug is of interest one could model either the stimulation or inhibition of a response which is assumed to remain in a steady state that is perturbed by the drug [4]. The resulting set of equations are a system of ordinary differential equations (ODEs) and typically needs to be solved numerically. To note, since no individual is the same it can also be useful to account for the variability between individuals by adding so called random effects, meaning that one or several model parameters is allowed to vary according to a specified probability distribution, which then turns the equations into a nonlinear mixed-effects (NLME) framework. Moreover, the longitudinal model can then be joined with the TTE by connecting the measured response to a parametric hazard function, which describes the rate of which an event is expected within the next time step.

How the model structure and relevant covariates is decided upon depends on what the intent is. In some cases it can be beneficial to use a simple model in order to keep it interpretative and in other cases that might not be an option because of the behaviour of the data. However, once the structure and covariates are chosen it will contain a set of unknown model parameters which need to be inferred by data. This process is called parameter estimation and gives information on what covariates influence the response variables the most and can also be used to learn if some of them are uninformative. Parameter estimation is typically performed by using Maximum Likelihood Estimation (MLE). The aim of MLE is to, given a model structure, find the parameters that maximizes the probability of the observed outcomes. Once this process is finished and a suitable model which describes the data is found one can use this to make predictions about new observations.

There are several ways of approaching the linkage between longitudinal and TTE data, such as estimating the longitudinal parameters separately and using the model predictions as a covariate in the TTE model [5]. However, since this approach does not take the simultaneous variability from both the individuals as well as the prediction error into account it is prone to bias [5]. Joint parameter estimation is therefore believed to correct for this bias by taking the inter-individual variability, model error as well as the estimated noise in the data, into account while estimating the TTE parameters [5, 6]. In practice, this means incorporating the two likelihoods into a combined one, which is then maximized to obtain the MLE [6]. Previous studies within joint modeling of different longitudinal models proved this hypothesis to be true and therefore this thesis aims to further explore the applicability of a joint parameter estimation for PK-PD modeling with regards to different sample frequencies [6].

Several applications of PK-PD modeling in various areas such as cardiovascular disease, oncology, and respiratory diseases has been studied extensively at FCC at the department of Systems and Data Analysis [3, 7, 8]. The department has also developed computational methods such as a Mathematica package for parameter

estimation and diagnostics in NLME models [9]. As of late, FCC has started studying the analysis of joint longitudinal and TTE data, including a first application to clinical data on asthma exacerbations [3]. Today, FCCs research in this area is particularly focused on the opportunities and challenges posed by frequently sampled longitudinal data together with TTE data.

1.2 Aim and objectives

This Master's thesis studies strategies of joint modeling of TTE data together with frequently sampled longitudinal data. Specifically, the aim is to be able to identify and distinguish scenarios where it is warranted to apply a joint parameter estimation approach in a mixed-effects framework, compared to the simpler modeling technique of a sequential parameter estimation, by developing an understanding of the relationship between estimation method and sampling frequency. This will be achieved by firstly simulating synthetic data with added random effects and secondly comparing sequential parameter estimation methods to joint approaches. The parameter estimation algorithms in question, for both cases, will need to be formulated and implemented as a part of this step. Lastly, this comparison needs to be analyzed in order to identify situations where a joint model framework is suitable and possible pitfalls to avoid.

2
Theory

The background knowledge needed to perform joint modeling is explained below. Firstly, the modeling framework needed to simulate data is presented. As for the longitudinal data, the kinetics of the system is presented as a PK-PD model, which describes the relationship between the drug and bodily response. To describe the TTE one can model the hazard, meaning the rate of which the individual could experience an event within the next time step. Secondly, the two different methods used to perform the parameter estimation are explained by deriving their respective likelihoods. These likelihoods do not have explicit solutions and therefore, as the third section, the First Order Conditional Estimation (FOCE) method is introduced. Lastly, an important measurement when dealing with larger datasets and means of parameters is one of variability. Hence, a way of calculating the precision of estimations is presented.

2.1 Drug response modeling in biological systems

Modeling of biological systems can look different depending on the scope of the problem [1]. Since this thesis focuses on models used for drug development, these will contain both a system representing the body and another representing the linked response. The section describes the PK-PD modeling framework and the modeling of TTE, in discrete time, as well as the mathematical framework of discrete NLME used for the synthetic data generation and parameter estimations.

2.1.1 PK-PD modeling

In order to explain the complexity existing in biological system it is common to use so called PK-PD models. PK models usually describe what the body does to the drug by modeling the absorption and removal of the drug through the system together with its interacting species [1]. In contrast, PD models describe how the body responds to the drug [4]. These models are constructed either in continuous time, using ODEs, or in discrete time, with difference equations, and describe how the system changes over time. PK-PD systems are commonly described via compartment models, where each compartment can be viewed as a box of fully mixed substance with a certain in- or outflow [10]. These boxes could represent biological objects such as organs, plasma or tissue. The compartments are also linked via proportionality constants of different orders, depending on the prior knowledge, and are then modelled together.

For a simple example, imagine the two compartment system illustrated in Figure 2.1. This scenario describes a patient receiving an oral dose of a drug that passes through to the gut before it is absorbed in to the blood and then removed with a specific rate.

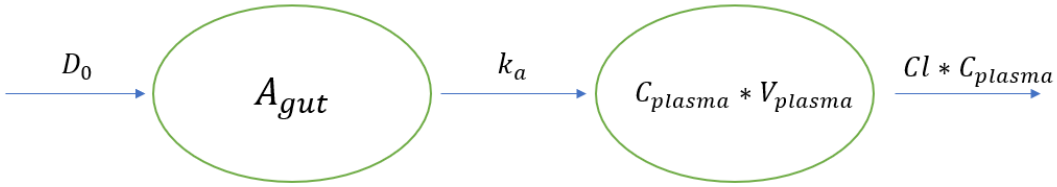


Figure 2.1: Two compartment model describing an oral dosage, D_0, passing through to the gut as an amount A_{gut} before being absorbed through the rate constant k_a to the blood stream with concentration C_{plasma} before being eliminated with the clearance rate Cl.

This two compartment case can be explained with ODEs to represent the flow between compartments. The set of equations becomes

$$\frac{\mathrm{d}A_{gut}}{\mathrm{d}t} = -k_a \cdot A_{gut}, \qquad A_{gut}(0) = D_0, \tag{2.1}$$

$$V_{plasma} \cdot \frac{\mathrm{d}C_{plasma}}{\mathrm{d}t} = k_a \cdot A_{gut} - Cl \cdot C_{plasma}, \qquad C_{plasma}(0) = 0, \tag{2.2}$$

where D_0 is the given dosage, A_{gut} the resulting amount of drug in the gut, k_a is the rate of which the drug is absorbed to the blood stream where it has the concentration C_{plasma}, before it is removed with the clearance rate Cl.

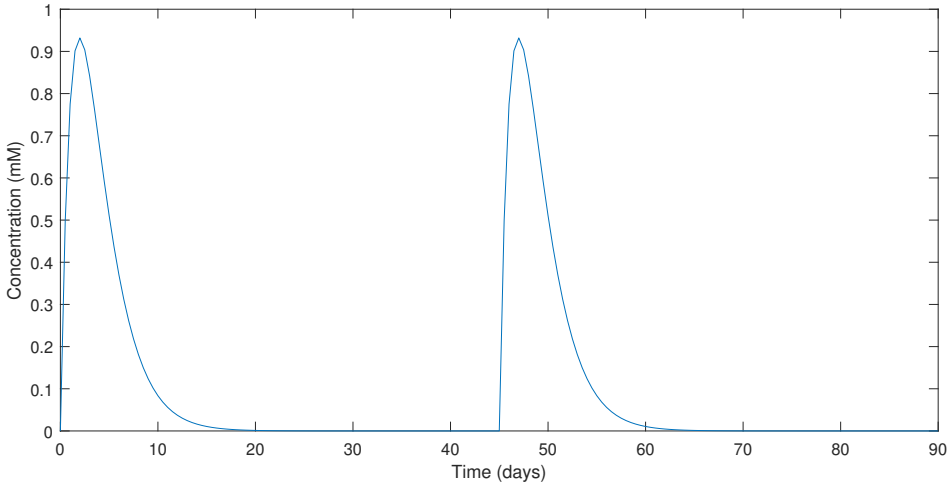
Now introduce a response variable, R, which is in a steady state with no drug present. Once the drug is added this will perturb the steady state and there will be a transient decrease or increase in response. These types of models are called turnover models and can be described using the equation

$$\frac{\mathrm{d}R}{\mathrm{d}t} = k_{in} - k_{out} \cdot R, \qquad R(0) = \frac{k_{in}}{k_{out}}, \tag{2.3}$$

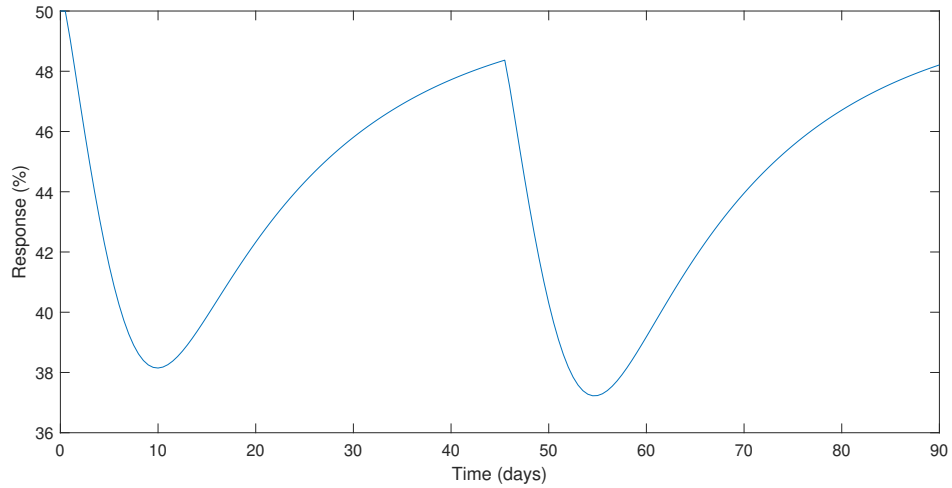
where k_{in} is a zero-order production and k_{out} the rate of the output [4]. The PK-PD dynamics described above can then be linked to a decreased response by either inhibition of the production (via k_{in}) or stimulation of the output (via k_{out}) of R. For example, an inhibition could be introduced as

$$k'_{in} = k_{in}\left(1 - \frac{I_{max} \cdot C_{plasma}}{IC_{50} + C_{plasma}}\right), \tag{2.4}$$

where I_{max} is the maximum inhibition response and IC_{50} the drug concentration causing 50% of the maximum inhibition. Equations 2.1-2.4 can then be simulated together to describe the dynamics of the body-drug relationship. An example of this is illustrated in Figure 2.2.



(a) pharmacokinetics



(b) Pharmacodynamics

Figure 2.2: Simulated PK-PD dynamics for the case described above where the individual was given a dose of a drug on days 0 and 45, with the intent of inhibiting the response, and observed over 90 days.

From simulations like this, one can then learn what influences the response and to what extent. More importantly, data like this can be used for further studies by linking it to the TTE in joint modeling.

2.1.2 TTE modeling

To model these events of interest one could make use of TTE data, which are the documented times for either one or several events occurring. Note that for the following sections the referenced time will be discrete, meaning that each time point $t_j = j \cdot \Delta t$ is dependent upon a sampling frequency, Δt. The case of one time events is called *Survival Analysis* and will be the focus of this thesis. The survival function $S(t_j)$

$$S(t_j|\boldsymbol{z}) = P(T > t_j|\boldsymbol{z}), \tag{2.5}$$

represents the probability that an event has not occurred at time t_j with the true value of the event time being represented as the stochastic variable T, given the explanatory variables $\boldsymbol{z}$ [11]. These explanatory variables can be both categorical, such as age or gender, and longitudinal, such as a measured response over time. Since the survival function does not cover the actual event time, but rather what happens before, it is useful to expand it with the hazard function , $h(t_j|\boldsymbol{z})$, which primary focus is the actual event [12]. The hazard equation describes the rate of which an event occur, or how large the risk of an event is for a time step of size Δt. It is defined as the following

$$h(t_j|\boldsymbol{z}) = P(T = t_j|T \geq t_j, \boldsymbol{z}), \tag{2.6}$$

where $h(t_j|\boldsymbol{z})$ is the probability of an an event occurring at time t_j given that the subject has not experienced an event until this point ($T \geq t_j$) and the explanatory variables $\boldsymbol{z}$ [11, 12].

The hazard and survival functions are related by a product of probabilities, see

$$S(t_j|\boldsymbol{z}) = \prod_{l=1}^{j} \Big(1 - h(t_l|\boldsymbol{z})\Big), \tag{2.7}$$

since each time step is independent of the next. However, because it is not possible to get an explicit expression for the hazard, as the true function of $S(t_j)$ is unknown, it is common to use a parametrization [11]. This parametrization can be approached in different ways, what is most important in the case of joint modeling is that it is linked to the response variables. A common way to describe the parametric hazard is as

$$h(t_j|\boldsymbol{z}) = f(\beta_0 + \boldsymbol{\beta z}), \tag{2.8}$$

where f is a strictly increasing function of the linear predictor $\beta_0 + \boldsymbol{\beta z}$, β_0 is the effect given the baseline variables and $\boldsymbol{\beta}$ the vector of the effects of each of the variables. Common functions f are, for example, the logistic discrete hazard equation and the grouped proportional hazards model (discrete Cox model), also called complementary log-log (clog-log), described below [12].

$$h_{log}(t_j|\boldsymbol{z}) = \frac{\exp(\beta_0 + \boldsymbol{\beta z})}{1 + \exp(\beta_0 + \boldsymbol{\beta z})}, \tag{2.9}$$

$$h_{clog-log}(t_j|\boldsymbol{z}) = 1 - \exp\Big(\text{-}\exp(\beta_0 + \boldsymbol{\beta z})\Big). \tag{2.10}$$

Note that both of Equations 2.9-2.10 take values in $[0, 1]$ since they represent probabilities.

Within survival analysis there is also an aspect of censoring, meaning that there are individuals in the study which the TTE is unknown for. There are different types of censoring, such as when an individual survives past the observational time, which is called right censoring, or it could be the case that the individual experiences an event prior to the study, in which case it is called left censoring. As for the scope of this thesis only right censoring will be considered. The result from a survival analysis is usually visualized using so called survival plots, see Figure 2.3.

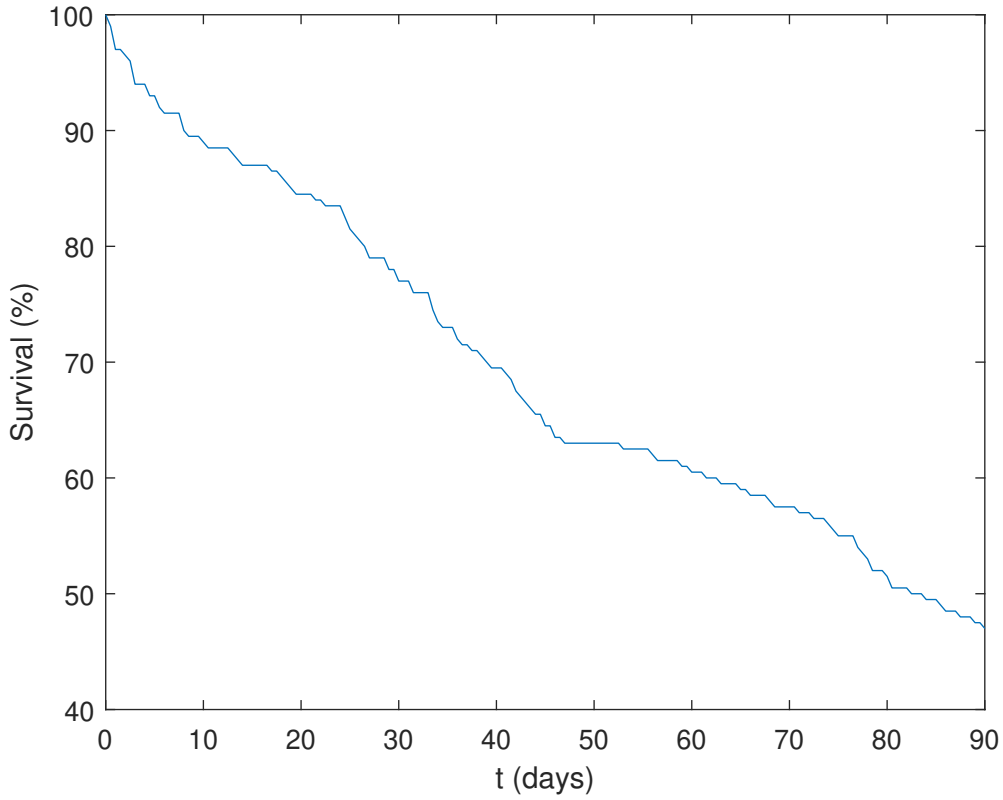


Figure 2.3: Survival curve for simulated TTE data for 200 individuals over 90 days with the PK-PD data described in Section 2.1.1.

These dynamics can the be linked and visualized together with the PK-PD data to get a clear overview of the combined system.

2.1.3 Nonlinear Mixed Effects Modeling (NLME)

The framework described in Section 2.1.1 consists of a set of ODEs and needs to be expanded in order to apply it to describe a heterogeneous population. This is because the model as it is only depends on the initial values, input, and the model parameters where the only source of variability is in the observational model. For

studies of populations there are more factors which need to be considered when building the model and estimating model parameters, which will be further explained in this section. However, as the aim is to link these with the framework described for TTE modeling in Section 2.1.2, this should be done for discrete time.

Since the approach presented in this thesis focuses on discrete time observations, consider a model in discrete time with the response variable $\boldsymbol{y}_i$, which is a matrix of dimension $M \text{x} \tau_i$ where M is the number of sampled quantities and τ_i the last observed time point for individual $i = 1, 2, ..., N$, describing the measured responses as the following

$$\boldsymbol{x}_i(t_j + \Delta t) = f\Big(\boldsymbol{x}_i(t_j, \boldsymbol{u}_i, \boldsymbol{\theta}_i), t_j, \boldsymbol{u}_i, \boldsymbol{\theta}_i\Big), \qquad \boldsymbol{x}_i(0, \boldsymbol{u}_i, \boldsymbol{\theta}_i) = \boldsymbol{x}_{i,0}(\boldsymbol{u}_i, \boldsymbol{\theta}_i), \quad (2.11)$$

$$\boldsymbol{y}_i(t_j + \Delta t) = g\Big(\boldsymbol{x}_i(t_j, \boldsymbol{u}_i, \boldsymbol{\theta}_i), t_j, \boldsymbol{u}_i, \boldsymbol{\theta}_i\Big) + \boldsymbol{e}, \qquad \boldsymbol{e} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}), \quad (2.12)$$

where for individual i at time point t_j, Δt is the step size, $\boldsymbol{x}_i(t_j)$ is a vector containing the set of state variables, $\boldsymbol{u}_i$ the input (dosage) for the set of parameters $\boldsymbol{\theta}_i$ and normally distributed errors $\boldsymbol{e}$ with mean zero and covariance matrix $\boldsymbol{\Sigma}$.

If the goal is to describe a full population one might run into inter-individual variability between subjects. A way to represent this variability is to introduce random effects on specific parameters. These random effects can be assumed to follow normal distribution to simplify calculations, but they can also be transformed if needed. For example, the random effects used in this thesis are assumed to follow a multivariate normal distribution with mean zero and covariance matrix $\boldsymbol{\Omega}$. The parameters would then change as following

$$\boldsymbol{\theta}_i = \boldsymbol{\theta} \cdot \exp(\boldsymbol{\eta}_i), \quad \boldsymbol{\eta}_i \sim \mathcal{N}(0, \boldsymbol{\Omega}), \quad (2.13)$$

where $\boldsymbol{\theta}$ is the vector of parameters shared among all individuals and $\boldsymbol{\eta}_i$ the random effects which differ for each individual. A nonlinear model containing this extended parameter vector of both fixed and random effects is called a nonlinear mixed effects model (NLME) and requires certain adjustments when estimating the parameters.

2.2 Parameter estimation

Parameter estimation is an important part in order to gain inference on the model. These estimations can be done using different approaches, such as by using some type of loss function, but in this thesis they are built upon Maximum Likelihood Estimation (MLEs). Since the likelihood function depends on the model and model assumptions, these are separated into three parts; one for the PK-PD model, one for the TTE model, and one for the joint approach.

The likelihood function describes the joint probability of observed data as a function of the parameters from the chosen model structure. Generally it can be described as, for a given discrete observation $\boldsymbol{y}_i$ and a discrete random variable Y for individual i

$$L_i(\boldsymbol{\theta}|\boldsymbol{y}_i) = f_{\boldsymbol{\theta}_i}(\boldsymbol{y}_i) = P_\theta(Y = \boldsymbol{y}_i), \quad (2.14)$$

where $f_{\theta i}(\boldsymbol{y}_i)$ is the probability mass function. If Y is a continuous random variable, f is the corresponding probability density. The likelihood is used as a measurement to describe "How likely the outcome $\boldsymbol{y}$ is given that the true value of the parameter is $\boldsymbol{\theta}$, when it describes the probability distribution of $\boldsymbol{y}$". The goal is to find the set of parameters which maximizes the likelihood and therefore the parameter estimation becomes a problem of optimization. This problem becomes easier by transforming the likelihood into a negative log-likelihood and minimizing said transform, since logarithms are strictly increasing. The reason for this is because for normally distributed, or other members of the exponential family, the log-transform of the probability distribution becomes easier to handle numerically.

2.2.1 Maximum likelihood estimation for NLME

For the models described Section 2.1.3, the likelihood estimation needs to be fitted for a mixed effects model with parameters $\boldsymbol{\theta}$ and random effects $\boldsymbol{\eta}_i$. However, since the values of the random effects are unknown and the goal is rather to find the distribution, it is ideal to marginalize the likelihood with regards to $\boldsymbol{\eta}_i$. Assume the random effects to be distributed as previously described

$$\boldsymbol{\eta}_i \sim \mathcal{N}(0, \boldsymbol{\Omega}), \tag{2.15}$$

where $\boldsymbol{\Omega}$ is the covariance matrix of dimension KxK for individual i with K random effects, with the variance of each random effect on its diagonal. The likelihood then takes the form

$$L(\boldsymbol{\theta}|\boldsymbol{y}) = \prod_{i=1}^{N} \int p(\boldsymbol{y}_i, \boldsymbol{\eta}_i|\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\eta}_i, \tag{2.16}$$

where $\boldsymbol{y}_i$ is the set of observations and $\boldsymbol{\theta}$ the set of parameters for the effects of the covariates as well as the standard deviation of $\boldsymbol{y}_i$ and the covariance matrix of the random effects $\boldsymbol{\Omega}$, for individual $i = 1, 2, ..., N$. By then applying Bayes' theorem and realizing that $\boldsymbol{\eta}_i$ is independent of $\boldsymbol{\theta}$, conditionally on $\boldsymbol{\Omega}$, and the equation takes the form

$$L(\boldsymbol{\theta}|\boldsymbol{y}) = \prod_{i=1}^{N} \int p(\boldsymbol{y}_i|\boldsymbol{\theta}, \boldsymbol{\eta_i}) p(\boldsymbol{\eta}_i|\boldsymbol{\Omega}) \mathrm{d}\boldsymbol{\eta_i}. \tag{2.17}$$

As for the PDFs, assume that the observations and random effects are normally distributed as follows

$$p(\boldsymbol{y}_i|\boldsymbol{\theta}, \boldsymbol{\eta}_i) = \frac{1}{(2\pi)^{M/2}|\boldsymbol{\Sigma}|^{1/2}} \cdot \exp(-\frac{1}{2} \sum_{j=1}^{\tau_i} \boldsymbol{\epsilon}_{ij}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\epsilon}_{ij}), \qquad \boldsymbol{\epsilon}_{ij} = \boldsymbol{y}_{ij} - \hat{\boldsymbol{y}}_{ij}, \tag{2.18}$$

$$p(\boldsymbol{\eta}_i|\boldsymbol{\Omega}) = \frac{1}{(2\pi)^{K/2}|\boldsymbol{\Omega}|^{1/2}} \cdot \exp(-\frac{1}{2} \boldsymbol{\eta}_i^T \Omega^{-1} \boldsymbol{\eta_i}), \tag{2.19}$$

where for subject i, M the number of outputs, K the number of random effects and $\boldsymbol{\epsilon}_{ij}$ the residuals for time point $j = 1, 2, ..., \tau_i$ for an output $m = 1, 2, ..., M$ and τ_i

represents the last observed time point. To simplify, the likelihood in Equation 2.17 can be rewritten by taking the logarithm and the final result becomes

$$L(\boldsymbol{\theta}|\boldsymbol{y}) = \prod_{i=1}^{N} \int \exp\Big(l_i(\boldsymbol{\eta_i})\Big) \mathrm{d}\boldsymbol{\eta_i}, \tag{2.20}$$

$$l_i(\boldsymbol{\eta_i}) = -\frac{1}{2} \sum_{j=1}^{\tau_i} \Big(\boldsymbol{\epsilon}_{ij}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\epsilon}_{ij} + \log(2\pi|\boldsymbol{\Sigma}|)\Big) - \frac{1}{2} \boldsymbol{\eta_i}^T \Omega^{-1} \boldsymbol{\eta_i} - \frac{1}{2} \log(2\pi|\boldsymbol{\Omega}|). \tag{2.21}$$

However, this integral does not have a closed form solution and needs to be approximated.

2.2.2 Maximum likelihood estimation for TTE model

To simplify derivations assume, without loss of generalization, $\Delta t = 1$ such that $t_j = j$. As described in Section 2.2, what needs to be accounted for is both the TTE data and possible censoring. Therefore introduce the variable δ_i, defined as

$$\delta_i = \begin{cases} 1 & \text{if } \tau_i < C, \\ 0 & \text{if } \tau_i \geq C, \end{cases} \tag{2.22}$$

where τ_i is the observed TTE for individual i and C the censoring time which is assumed to be the end of the study and therefore the same for all individuals. In other words, δ_i is one if an event occurs before the study ends and zero otherwise. Since this is in regard to discrete time the event is defined as the last time point where an observation was collected. Hence, if an event occurred in the time span $[j, j+1]$ then the TTE is $\tau_i = j$. The probability that an individual i survives until the time j, each time step being independent of the previous one, becomes a product of probabilities. By assuming right censored observations, meaning that the censoring time is the end of the study, the likelihood becomes

$$L_i(\boldsymbol{\theta}|\boldsymbol{y}_i, \tau_i) = h(\tau_i|\boldsymbol{y}_i)^{\delta_i} \cdot \Big(1 - h(\tau_i|\boldsymbol{y}_i)\Big)^{(1-\delta_i)} \cdot \prod_{j=1}^{\tau_i - 1} \Big(1 - h(j|\boldsymbol{y}_i)\Big), \tag{2.23}$$

where $h(\tau_i|\boldsymbol{y}_i)$ is the parametric hazard equation for the the observed response variable $\boldsymbol{y}_i$ at time τ_i and the last term represents the probability of the individual surviving each time step j. Taking the logarithm and product over all individuals the log-likelihood becomes

$$\begin{aligned} l(\boldsymbol{\theta}|\boldsymbol{y}, \boldsymbol{\tau}) = \log \prod_{i=1}^{N} L_i(\boldsymbol{\theta}|\boldsymbol{y}_i, \tau_i) = &\sum_{i=1}^{N} \Big(\delta_i \log\Big(h(\tau_i|\boldsymbol{y}_i)\Big) + (1-\delta_i) \log\Big(1 - h(\tau_i|\boldsymbol{y}_i)\Big)\Big) \\ &+ \sum_{i=1}^{N} \sum_{j=1}^{\tau_i - 1} \log\Big(1 - h(j|\boldsymbol{y}_i)\Big), \end{aligned} \tag{2.24}$$

where it is simply to minimize the negative of l to get the optimal values of the hazard parameters.

2.2.3 Maximum likelihood estimation for the joint model

In a similar manner to Section 2.2.1, where the likelihood was derived for the longitudinal model, one can get an expression for the likelihood of a joint model of both the longitudinal and TTE aspects. Again, as in the last section one can assume $\Delta t = 1$ to simplify derivations. Since the random effects, $\boldsymbol{\eta}_i$, are still present, the marginal likelihood is a useful tool and the likelihood becomes

$$L(\boldsymbol{\theta}|\boldsymbol{y}, \boldsymbol{\tau}) = \prod_{i=1}^{N} \int p(\boldsymbol{y}_i, \tau_i, \boldsymbol{\eta}_i|\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\eta}_i, \tag{2.25}$$

where $\boldsymbol{y_i}$ contains the sampled quantities, τ_i the recorded TTE for individual $i = 1, 2, ..., N$, and $\boldsymbol{\theta}$ is the set of population parameters for the joint model. The probabilities are now dependent on both the response variable as well as the TTE, conditionally on the parameters which has both the covariates for the NLME model as well as for the parametric hazard. In addition, since the TTE data is independent of the probability of the observed model there will be a third term added. By applying Bayes' rule to the last expression it follows

$$L(\boldsymbol{\theta}|\boldsymbol{y}, \boldsymbol{\tau}) = \prod_{i=1}^{N} \int p(\boldsymbol{y}_i|\boldsymbol{\theta}, \boldsymbol{\eta}_i) p(\tau_i|\boldsymbol{\theta}, \boldsymbol{\eta}_i) p(\boldsymbol{\eta}_i|\boldsymbol{\Omega}) \mathrm{d}\boldsymbol{\eta}_i, \tag{2.26}$$

since it has already been established that $\boldsymbol{\eta}_i$ is independent of $\boldsymbol{\theta}$, conditionally on the covariance matrix of the random effects, $\boldsymbol{\Omega}$. By taking the logarithm of Equation 2.26, it can be rewritten as

$$L(\boldsymbol{\theta}|\boldsymbol{y}, \boldsymbol{\tau}) = \prod_{i=1}^{N} \int \exp\Big(l_i(\boldsymbol{\eta}_i)\Big) \mathrm{d}\boldsymbol{\eta}_i, \tag{2.27}$$

$$\begin{aligned} l_i(\boldsymbol{\eta}_i) = \delta_i \log\Big(h(\tau_i|\boldsymbol{y}_i)\Big) + (1-\delta_i)\log\Big(1 - h(\tau_i|\boldsymbol{y}_i)\Big) + \sum_{j}^{\tau_i - 1} \log\Big(1 - h(j|\boldsymbol{y}_i)\Big) \\ - \frac{1}{2}\sum_{j}^{\tau_i} \Big(\log(2\pi|\boldsymbol{\Sigma}|) + \epsilon_{ij}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\epsilon}_{ij}\Big) - \frac{1}{2}\boldsymbol{\eta}_i \boldsymbol{\Omega}^{-1} \boldsymbol{\eta}_i - \frac{1}{2}\log(2\pi|\boldsymbol{\Omega}|), \end{aligned} \tag{2.28}$$

where h is the parametric hazard equation, $\boldsymbol{\epsilon}_{ij}$ the residuals for individual i at time $j = 1, 2, ..., \tau_i$, for the response with covariance matrix $\boldsymbol{\Sigma}$. Yet again, the expression derived in Equation 2.27 needs to be approximated in some way.

2.3 First Order Conditional Estimation

Since working with NLME models requires the application of marginal likelihoods there needs to be an approach to dealing with this when an explicit solution does not exist. One way of approximating integrals, like the one in Equation 2.27, is via Laplace's method with a first order approximation of the Hessian. This is called the FOCE method and uses two optimization problems to get the maximum likelihood estimations.

The aim in Laplace's method is to approximate the integrand by a Gaussian function. Since the longitudinal model is already based on normality assumptions this approximation is suitable. Firstly, approximate $l_i(\boldsymbol{\eta}_i)$ by a second order Taylor polynomial around its maximum $\boldsymbol{\eta}_i^*$ for each individual i

$$l_i(\boldsymbol{\eta_i}) \approx l_i(\boldsymbol{\eta_i}^*) + \frac{\mathrm{d}l_i(\boldsymbol{\eta_i}^*)}{\mathrm{d}\boldsymbol{\eta_i}}(\boldsymbol{\eta_i} - \boldsymbol{\eta_i}^*) + \frac{1}{2}\frac{\mathrm{d}^2 l_i(\boldsymbol{\eta_i}^*)}{\mathrm{d}\eta_{ik}\eta_{il}}(\boldsymbol{\eta_i} - \boldsymbol{\eta_i}^*)^2. \tag{2.29}$$

This will in turn set the second term (first order derivative) to zero. By applying this approach to the integral in Equation 2.27, it becomes

$$L(\boldsymbol{\theta}|\boldsymbol{y}, \boldsymbol{\tau}) = \prod_{i=1}^{N} \int \exp\left(l_i(\boldsymbol{\eta}_i^*) + \frac{1}{2}\frac{\mathrm{d}^2 l_i(\boldsymbol{\eta}_i^*)}{\mathrm{d}\eta_{ik}\eta_{il}}(\boldsymbol{\eta}_i - \boldsymbol{\eta}_i^*)^2\right) \mathrm{d}\boldsymbol{\eta}_i = \tag{2.30}$$

$$= \{\ \boldsymbol{\eta}_i^* \textit{ not a function of } \boldsymbol{\eta}_i\ \} =$$

$$= \prod_{i=1}^{N} \exp\Big(l_i(\boldsymbol{\eta}_i^*)\Big) \int \exp\left(\frac{1}{2}\frac{\mathrm{d}^2 l_i(\boldsymbol{\eta}_i^*)}{\mathrm{d}\eta_{ik}\eta_{il}}(\boldsymbol{\eta}_i - \boldsymbol{\eta}_i^*)^2\right) \mathrm{d}\boldsymbol{\eta}_i = \tag{2.31}$$

$$= \{\textit{Gaussian integral }\} =$$

$$= \prod_{i=1}^{N} \exp\Big(l_i(\boldsymbol{\eta}_i^*)\Big) \left|\frac{-\Delta l_i(\boldsymbol{\eta}_i^*)}{2\pi}\right|^{-\frac{1}{2}}, \tag{2.32}$$

where $|\Delta l_i(\boldsymbol{\eta}_i^*)|$ represents the determinant of the Hessian of the inner likelihood l_i. In the end, the optimal parameters are then found by maximizing Equation 2.32 with respect to $\boldsymbol{\theta}$. This results in two nested optimization problems where for each evaluation of the outer likelihood, $L(\boldsymbol{\theta}|\boldsymbol{y}, \tau_i)$, the MLE of the inner likelihood, $l_i(\boldsymbol{\eta}_i^*)$, needs to be calculated.

2.4 Precision in parameter estimations

When working with parameter estimations, in order to gain statistical inference, it is important to add a measurement of variability. A common way of approaching this is to measure numerical identifiability. The numerical identifiability aims to investigate, given the model structure and real data with noise, if it is possible to identify the true value of the estimations [13]. This often requires exploration of both the model structure as well as an analysis of the sensitivity of the parameter estimations [10].

One way of combining these two measurements is to use so called coefficients of variance, which gives an estimation of the dispersion [10]. A parameter with low dispersion is easily identified and vice versa. These coefficients can be calculated by noting that, if the problem is of minimization nature, the Hessian of the likelihood at the optimal parameter values is asymptotically equal to the so called Fisher's information matrix, F, which in its turn is asymptotically equal to the parametric covariance matrix [10]. See Cramér-Rao's bound [10]

$$\mathrm{COV}(\hat{\boldsymbol{\theta}}) \geq F^{-1}, \tag{2.33}$$

where the covariance matrix $\mathrm{COV}(\hat{\boldsymbol{\theta}})$ contains the parameter variances along its diagonal. Furthermore, the coefficients of variance can then be calculated as following.

$$\%RSE(\hat{\theta}_p) = 100 \cdot \frac{\sqrt{\mathrm{COV}(\hat{\theta}_p, \hat{\theta}_p)}}{\hat{\theta}_p}, \tag{2.34}$$

for a parameter θ_p, as a way to measure parameter Relative Standard Error (RSE) [10].

Since this thesis is based around synthetic data, one can also make use a more simple error measurement such as Relative Estimation Error (REE) defined as

$$\%REE(\hat{\theta}_p) = 100 \cdot \frac{\hat{\theta}_p - \theta_p}{\theta_p}, \tag{2.35}$$

for an estimation $\hat{\theta}_p$ of the true value θ_p. In comparison to Equation 2.34 this gives a measurement which cannot be inflated, or vice versa, for a bad estimation.

3
Methods

The methodology used in this thesis is based around simulations of a common K-PD model, with adjustments to the population, parameter values and sample frequency. The generated data is then fitted using two approaches to MLE in order to gain insight to what fits which datasets. This section will introduce both the model used for generating the synthetic data as well as the algorithms used to estimate the surrounding parameters. All of the code based around these methodologies were implemented in MATLAB R2021b and can be found in Appendix B.

3.1 Simulation of synthetic data

The design of this thesis consists of simulating data based on a parametrizations of the PK-PD model with added noise and then estimate the model parameters with the two different estimation methods. Therefore several datasets with differing qualitites needs to be simulated, which this section aims to explain.

3.1.1 Model structure

In some situations the data necessary for a full PK-PD model is not available. This could be the case in, for example, pediatrics where sample collecting is considered too invasive for the patient. An option is then to implement a so-called K-PD model structure, which means to approximate the PK part by a biophase and linked excretion rate (one-compartment model) [14]. This biophase refers to the hypothetical compartment where the drug is active. Figure 3.1 presents the dynamics, where the bolus drug input is represented by the state A. The biomarker, R is then inhibited via the infusion rate, IR, by altering the rate of the synthesis of R. The inhibition is assumed to be according to Michaelis-Menten and the set of equations are given as

$$\frac{\mathrm{d}A}{\mathrm{d}t} = -KDE \cdot A, \qquad A(0) = input(0), \tag{3.1}$$

$$\frac{\mathrm{d}R}{\mathrm{d}t} = k_{in}\left(1 - \frac{IR}{EDK_{50} + IR}\right) - k_{out} \cdot R, \qquad R(0) = \frac{k_{in}}{k_{out}}, \tag{3.2}$$

where KDE is the rate of which A is digested, input(t) represents the times when the patient takes the drug, EDK_{50} is the amount of which 50% of k_{in} is inhibited.

The infusion rate is defined as follows

$$IR = KDE \cdot A. \tag{3.3}$$

Since there is no explicit solution to Equations 3.1-3.2, one can use the forward Euler's method to approximate one. Assume a set of discrete time point $t_j = j \cdot \Delta t$, for a given sample frequency Δt, and the system becomes

$$A(t_j + \Delta t) = A(t_j) - \Delta t \cdot KDE \cdot A(t_j), \qquad A(0) = input(0), \tag{3.4}$$

$$\begin{aligned} R(t_j + \Delta t) = R(t_j) + \Delta t \cdot \Big(k_{in}\left(1 - \frac{IR(t_j)}{EDK_{50} + IR(t_j)}\right)k_{out} \cdot R(t_j)\Big), \\ R(0) = \frac{k_{in}}{k_{out}}. \end{aligned} \tag{3.5}$$

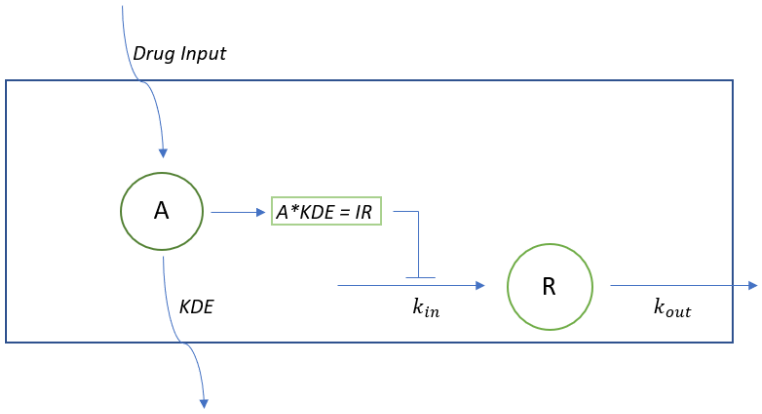


Figure 3.1: Illustration of the K-PD model. The dosage is ingested and active as the state A via the rate constant KDE, which effects the production of the biomarker R via the infusion rate IR. This biomarker is assumed to be in a steady-state of $\frac{k_{in}}{k_{out}}$ when no drug is present. The inhibition works by decreasing the production of R by decreasing the rate constant k_{in}.

The nature of this model causes the ratio of $\frac{k_{in}}{k_{out}}$ to be constant since the variables are correlated [14]. To avoid this issue the ratio will be set to a constant $m = 1$ and only k_{in} will be estimated, reducing the model in Equations 3.4-3.5 by one parameter.

As for the parametric hazard, this will be represented using Equation 2.10 and linked to the response as following

$$h(t|\boldsymbol{R}) = 1 - \exp\Big(\text{-}\exp(\alpha + \beta \boldsymbol{R})\Big), \tag{3.6}$$

where $\boldsymbol{R}$ is the matrix of all observations for every individual in each time point, α is the constant risk of an event and β the linked effect of the response. This means that a large β will cause a strong link between the response and the event. Since this hazard is calculated in discrete time and each time step can be seen as an interval, during the simulation the probability of an event occurring is determined by the value of the hazard at the previous time step.

3.1.2 Data generation

Simulation of synthetic data was done by iterating the dynamics described in Equation 3.4-3.5 with the forward Euler method for a given sampling frequency Δt. For each time step the hazard was calculated and the occurrence of an event was decided by a coin-flip approach. The code used for this purpose can be found in Appendix B.1. Let R_{ij} denote the simulated response for individual i at time t_j and let $\hat{R}_{ij}$ be the corresponding observation with additive noise. The data can therefore be described as

$$\hat{R}_{ij} = R_{ij} + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2), \tag{3.7}$$

where σ is the standard deviation which also needs to be estimated. The time series for the simulated data will then be data points at $t_j = j \cdot \Delta t$ for an individual i.

3.2 Parameter estimations

Below are the different methods of estimating the model parameters, presented in the manner of which they were executed to simulate results using the FOCE algorithm. The section starts by going through the derivations of the sequential approaches before moving onward to the joint model estimation. Note, the following derivations are with respect to the models mentioned in Section 3.1 meaning that there could be multiple random effects but only a single measured response variable. Also note that for the following derivations the sampling frequency Δt will be set to 1 for simplicity, such that $t_j = j = 1, 2, ..., \tau_i$ for an individual i.

3.2.1 Sequential parameter estimation

As previously described in Section 2.2, the aim is to approximate the marginal likelihood by expanding it using a Taylor polynomial of second order and make use of the fact that the integral turns into a Gaussian. The end result of this will be the FOCE algorithm which in turn will be linked to the estimation of the TTE parameters. For this section let

$$\boldsymbol{\theta} = \{\boldsymbol{\theta}_{KPD}, \boldsymbol{\theta}_{TTE}\}, \tag{3.8}$$

where $\boldsymbol{\theta}_{KPD}$ are the parameters related to the K-PD part of the dynamics and $\boldsymbol{\theta}_{TTE}$ the parameters used in the parametric hazard equation.

Estimating K-PD parameters

The process of estimating parameters in the longitudinal model involves two nested optimization problems. For each iteration of computing the outer likelihood, l^{KPD}, in Equation 2.32 for a fixed value of $\boldsymbol{\theta}_{KPD}$, the MLE of the inner likelihood, $l_i^{KPD}(\boldsymbol{\eta}_i^*)$ needs to be computed as following

$$\hat{\boldsymbol{\eta}_i}^* = \arg\max_{\boldsymbol{\eta}_i} l_i^{KPD}(\boldsymbol{\eta}_i | \boldsymbol{R}_i), \tag{3.9}$$

where $\boldsymbol{R}_i$ is the vector containing all observations of the biomarker concentration for individual i. As for the outer maximization problem this requires more computations. In order to describe the derivatives, the following rules were used for a

square, symmetric, real valued and positive definite matrix $\boldsymbol{A}$ and vector $\boldsymbol{b}$

$$\boldsymbol{y} = \boldsymbol{b^T A b} \Rightarrow \frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}x} = 2\boldsymbol{b^T A}\frac{\mathrm{d}\boldsymbol{b}}{\mathrm{d}x} + \boldsymbol{b^T}\frac{\mathrm{d}\boldsymbol{A}}{\mathrm{d}x}\boldsymbol{b},$$
$$\frac{\mathrm{d}\boldsymbol{A^{-1}}}{\mathrm{d}x} = -\boldsymbol{A^{-1}}\frac{\mathrm{d}\boldsymbol{A}}{\mathrm{d}x}\boldsymbol{A^{-1}}.$$

Since the response is now considered to be univariate, the likelihood described in Equation 2.21 becomes

$$l_i^{KPD}(\boldsymbol{\eta}_i) = -\frac{1}{2}\sum_{j=1}^{\tau_i}\left(\frac{\epsilon_{ij}^2}{\sigma^2} + \log(2\pi\sigma^2)\right) - \frac{1}{2}\boldsymbol{\eta}_i^T\boldsymbol{\Omega}^{-1}\boldsymbol{\eta}_i - \frac{1}{2}\log(2\pi|\boldsymbol{\Omega}|), \tag{3.10}$$

where ϵ_{ij} is the residual at time point j and τ_i the TTE for individual i, σ^2 is the variance of the response. The first order derivative is then described as following

$$\frac{\mathrm{d}l_i^{KPD}}{\mathrm{d}\eta_{ik}} = -\frac{1}{2}\sum_{j=1}^{\tau_i}\left(2\frac{\epsilon_{ij}}{\sigma^2}\frac{\mathrm{d}\epsilon_{ij}}{\mathrm{d}\eta_{ik}}\right) - \boldsymbol{\eta}_i^T\boldsymbol{\Omega^{-1}}\frac{\mathrm{d}\boldsymbol{\eta}_i}{\mathrm{d}\eta_{ik}}, \tag{3.11}$$

in regards to the random effect η_{ik}, where there is no interaction term included since σ is assumed to be independent of $\boldsymbol{\eta}$. The first term can be rewritten as

$$\begin{aligned}\frac{\mathrm{d}\epsilon_{ij}}{\mathrm{d}\eta_{ik}} &= \frac{\mathrm{d}\left(\hat{R}_{ij} - R_{ij}\right)}{\mathrm{d}\eta_{ik}} = \{R \text{ only effects } A \text{ via } IR\} = \\ &= -\left(\frac{\partial R_{ij}}{\partial \eta_{ik}} + \frac{\partial R_{ij}}{\partial IR_{ij}}\frac{\mathrm{d}IR_{ij}}{\mathrm{d}\eta_{ik}} + \frac{\partial R_{ij}}{\partial IR_{ij}}\frac{\partial IR_{ij}}{\partial A_{ij}}\frac{\mathrm{d}A_{ij}}{\mathrm{d}\eta_{ik}}\right),\end{aligned} \tag{3.12}$$

for the response variable R_{ij} and states IR_{ij} and A_{ij} at time point j for individual i. As for the second derivative with respect to another random effect l, the same approach is followed

$$\begin{aligned}\frac{\mathrm{d}^2 l_i^{KPD}}{\mathrm{d}\eta_{ik}\mathrm{d}\eta_{il}} = -\frac{1}{2}\sum_{j=1}^{\tau_i}\left(2\frac{\epsilon_{ij}}{\sigma^2}\frac{\mathrm{d}^2\epsilon_{ij}}{\mathrm{d}\eta_{ik}\mathrm{d}\eta_{il}} + 2\frac{\mathrm{d}\epsilon_{ij}}{\mathrm{d}\eta_{il}}\frac{1}{\sigma^2}\frac{\mathrm{d}\epsilon_{ij}}{\mathrm{d}\eta_{ik}}\right) - \\ \boldsymbol{\eta}_i\boldsymbol{\Omega^{-1}}\frac{\mathrm{d}^2\boldsymbol{\eta}_i}{\mathrm{d}\eta_{ik}\mathrm{d}\eta_{il}} - \frac{\mathrm{d}\boldsymbol{\eta}_i^T}{\mathrm{d}\eta_{il}}\boldsymbol{\Omega^{-1}}\frac{\mathrm{d}\boldsymbol{\eta}_i}{\mathrm{d}\eta_{ik}}.\end{aligned} \tag{3.13}$$

Since the aim is to approximate the Hessian, ignore the second order terms to get the first order estimation given by

$$H_{ikl} \approx -\sum_{j=1}^{\tau_i}\left(\frac{\mathrm{d}\epsilon_{ij}}{\mathrm{d}\eta_{il}}\frac{1}{\sigma^2}\frac{\mathrm{d}\epsilon_{ij}}{\mathrm{d}\eta_{ik}}\right) - \boldsymbol{\Omega}_{lk}^{-1}, \tag{3.14}$$

which then needs to be evaluated at $\boldsymbol{\eta}_i^*$. The outer maximization problem becomes

$$\hat{\boldsymbol{\theta}}_{KPD} = \underset{\boldsymbol{\theta}_{KPD}}{\arg\max}\, L^{KPD}(\boldsymbol{\theta}_{KPD}|\boldsymbol{R}) = \prod_{i=1}^{N}\exp\left(l_i^{KPD}(\boldsymbol{\eta}_i^*)\right)\left|\frac{-\boldsymbol{H}_i}{2\pi}\right|^{-\frac{1}{2}}, \tag{3.15}$$

where $\boldsymbol{R}$ is a matrix of the collected response in each time step for N individuals and $|\boldsymbol{H}_i|$ is the determinant of the approximation of the Hessian. By simplifying and rewriting it as the negative log-likelihood one gets

$$\hat{\boldsymbol{\theta}}_{KPD} = \underset{\boldsymbol{\theta}_{KPD}}{\arg\min} -l^{KPD}(\boldsymbol{\theta}_{KPD}|\boldsymbol{R}) = \sum_{i=1}^{N} -l_i^{KPD}(\boldsymbol{\eta}_i^*) + \frac{1}{2}\log\left|\frac{-\boldsymbol{H}_i}{2\pi}\right|. \tag{3.16}$$

Estimating TTE parameters

From Section 2.1.2, it is clear that the likelihood for the TTE model is given by Equation 2.23. Since the population parameters for the K-PD model, $\boldsymbol{\theta}$, as well as the fixed points for the random effects, $\boldsymbol{\eta}^*$, already are estimated one can predict the response in each time step. Doing so in combination with the given TTE data, the MLE estimates for the parametric hazard equation are found by maximizing Equation 2.23 as following

$$\begin{aligned}\hat{\boldsymbol{\theta}}_{TTE} = \max_{\boldsymbol{\theta}_{TTE}} L^{TTE}(\boldsymbol{\theta}_{TTE}|\boldsymbol{R},\boldsymbol{\tau}) = \max_{\boldsymbol{\theta}_{TTE}} \prod_{i=1}^{N} h(\tau_i|R_{i,\tau_i})^{\delta_i} \cdot \Big(1 - h(\tau_i|R_{i,\tau_i})\Big)^{(1-\delta_i)} \\ \cdot \prod_{j=1}^{\tau_i - 1} \Big(1 - h(j|R_{ij})\Big),\end{aligned} \tag{3.17}$$

where $\boldsymbol{R}$ is a matrix of each individual response in every time step and $\boldsymbol{\tau}$ the vector of TTE for N individuals.

Implementation for sequential estimation

As previously mentioned, by using the FOCE method there are essentially two optimization problems; an inner one where the aim is to find $\boldsymbol{\eta}_i^*$ by maximizing l_i for each individual, and an outer one which should maximize the full log-likelihood for the full population, l^{KPD}. The structure of this implementation is explained in Algorithm 1. The derivatives needed in Equation 3.10 will be approximated using explicit finite differences, see Appendix A.1. For the implementation of FOCE in MATLAB, see Appendix B.2, and for the likelihood of the TTE see Appendix B.3.

Algorithm 1 Sequential FOCE parameter estimation

1: Set start guess: $\boldsymbol{\theta}_{KPD} = \boldsymbol{\theta}_{KPD,0}$, $\boldsymbol{\theta}_{TTE} = \boldsymbol{\theta}_{TTE,0}$
2: Outer problem: $\hat{\boldsymbol{\theta}}_{KPD} = \min_{\boldsymbol{\theta}_{KPD}} -l^{KPD}(\boldsymbol{\theta}_{KPD})$
For every evaluation of $-l^{KPD}(\boldsymbol{\theta}_{KPD})$
3: **for all** individuals **do**
4: Set start guess: $\boldsymbol{\eta} = \boldsymbol{\eta}_0$
5: Inner problem: $\hat{\boldsymbol{\eta}}_i^* = \min_{\boldsymbol{\eta}_i} -l_i^{KPD}(\boldsymbol{\eta}_i)$
6: **end for**
7: Simulate: $\boldsymbol{R} = \boldsymbol{R}(\hat{\boldsymbol{\theta}}_{KPD}, \hat{\boldsymbol{\eta}}^*)$
8: Find: $\hat{\boldsymbol{\theta}}_{TTE} = \min_{\boldsymbol{\theta}_{TTE}} -l^{TTE}(\boldsymbol{\theta}_{TTE}|\boldsymbol{R},\boldsymbol{\tau})$

3.2.2 Joint parameter estimation

From following Section 2.2.3, the goal is to estimate the joint likelihood described in Equation 2.26. Once again, the FOCE algorithm is suitable to use since this integral has no explicit solution. The same derivation regarding the derivatives described in the previous section is therefore needed with regards to the inner and outer optimization problems. The joint inner likelihood for an individual i with TTE τ_i and a univariate response at time j, denoted R_{ij}, becomes

$$\begin{aligned} l_i(\boldsymbol{\eta}_i) = \delta_i \log(h(\tau_i|R_{i,\tau_i}) + (1-\delta_i)\log(1-h(\tau_i|R_{i,\tau_i})) \\ + \sum_{j=2}^{\tau_i - 1} \log(1-h(j|R_{ij})) + \sum_{j=1}^{\tau_i} \left(-\frac{1}{2}\log(2\pi\sigma^2) - \frac{\epsilon_{ij}^2}{2\sigma^2} \right) - \boldsymbol{\eta}_i^T \boldsymbol{\Omega^{-1}} \frac{\mathrm{d}\boldsymbol{\eta}_i}{\mathrm{d}\eta_{ik}}. \end{aligned} \tag{3.18}$$

The first derivative of this equation becomes

$$\begin{aligned} \frac{\mathrm{d}l_i}{\mathrm{d}\eta_{ik}} = \frac{\mathrm{d}h(\tau_i|R_{i,\tau_i})}{\mathrm{d}\eta_{ik}} \left(\frac{\delta_i}{h(\tau_i|R_{i,\tau_i})} - \frac{1-\delta_i}{1-h(\tau_i|R_{i,\tau_i})} \right) - \sum_{j=1}^{\tau_i - 1} \left(\frac{1-\delta_i}{1-h(j|R_{ij})} \frac{\mathrm{d}h(j|R_{ij})}{\mathrm{d}\eta_{ik}} \right) \\ - \frac{1}{2} \sum_{j=1}^{\tau_i} \left(\frac{2\epsilon_{ij}}{\sigma^2} \frac{\mathrm{d}\epsilon_{ij}}{\mathrm{d}\eta_{ik}} \right) - \boldsymbol{\eta}_i^T \boldsymbol{\Omega^{-1}} \frac{\mathrm{d}\boldsymbol{\eta}_i}{\mathrm{d}\eta_{ik}}, \end{aligned} \tag{3.19}$$

where the derivatives in regard to the hazard evaluates to

$$\begin{aligned} \frac{\mathrm{d}h(j|R_{ij})}{\mathrm{d}\eta_{ik}} &= \frac{\mathrm{d}h(j|R_{ij})}{\mathrm{d}\eta_{ik}} + \frac{\mathrm{d}h(j|R_{ij})}{\mathrm{d}R_{ij}} \frac{\mathrm{d}R_{ij}}{\mathrm{d}\eta_{ik}} + \frac{\mathrm{d}h(j|R_{ij})}{\mathrm{d}R_{ij}} \frac{\partial R_{ij}}{\partial \eta_{ik}} + \frac{h(j|R_{ij})}{\mathrm{d}R_{ij}} \frac{\partial R_{ij}}{\partial IR_{ij}} \frac{\mathrm{d}IR_{ij}}{\mathrm{d}\eta_{ik}} \\ &+ \frac{\mathrm{d}h(j|R_{ij})}{\mathrm{d}R_{ij}} \frac{\partial R_{ij}}{\partial IR_{ij}} \frac{\partial IR_{ij}}{\partial A_{ij}} \frac{\mathrm{d}A_{ij}}{\mathrm{d}\eta_{ik}} = \\ &= \frac{\mathrm{d}h(j|R_{ij})}{\mathrm{d}\eta_{ik}} + \frac{\mathrm{d}h(j|R_{ij})}{\mathrm{d}R_{ij}} \left(\frac{\partial R_{ij}}{\partial \eta_{ik}} + \frac{\partial R_{ij}}{\partial IR_{ij}} \frac{\mathrm{d}IR_{ij}}{\mathrm{d}\eta_{ik}} + \frac{\partial R_{ij}}{\partial IR_{ij}} \frac{\partial IR_{ij}}{\partial A_{ij}} \frac{\mathrm{d}A_{ij}}{\mathrm{d}\eta_{ik}} \right) = \\ &= \frac{\mathrm{d}h(j|R_{ij})}{\mathrm{d}\eta_{ik}} - \frac{\mathrm{d}h(j|R_{ij})}{\mathrm{d}R_{ij}} \frac{\mathrm{d}\epsilon_{ij}}{\mathrm{d}\eta_{ik}}. \end{aligned} \tag{3.20}$$

As for the second derivative, follow the steps which was explained in the previous section. The final expression can be used to approximate the Hessian and by ignoring the second order terms it follows

$$\begin{aligned} H_{ikl} \approx -\frac{\mathrm{d}h(\tau_i|R_{i,\tau_i})}{\mathrm{d}\eta_{ik}} \frac{\mathrm{d}h(\tau_i|R_{i,\tau_i})}{\mathrm{d}\eta_{il}} \left(\frac{1-\delta_i}{\left(1-h(\tau_i|R_{i,\tau_i})\right)^2} + \frac{\delta_i}{\left(h(\tau_i|R_{i,\tau_i})\right)^2} \right) \\ - \sum_{j=1}^{\tau_i - 1} \left(\frac{1}{\left(1-h(j|R_{ij})\right)^2} \frac{\mathrm{d}h(j|R_{ij})}{\mathrm{d}\eta_{ik}} \frac{\mathrm{d}h(j|R_{ij})}{\mathrm{d}\eta_{il}} \right) \\ - \sum_{j=1}^{\tau_i} \left(\frac{\mathrm{d}\epsilon_{ij}}{\mathrm{d}\eta_{il}}^T \frac{1}{\sigma^2} \frac{\mathrm{d}\epsilon_{ij}}{\mathrm{d}\eta_{ik}} \right) - \boldsymbol{\Omega}_{lk}^{-1}, \end{aligned} \tag{3.21}$$

which, as in the case of sequential estimation, needs to be evaluated at the maximum $\boldsymbol{\eta}^*$.

Implementation for joint estimation
The joint FOCE implementation is similar to its sequential counter part, but it does not include the last two steps and instead requires some more computations in order to get the derivatives. It is presented in Algorithm 2. As in Algorithm 1, both the derivatives with regards to the residuals (Equation 3.18) and hazard will be computed using finite differences, see Appendix A.1 and A.2. As for the implementation in MATLAB, see Appendix B.4.

Algorithm 2 Joint FOCE parameter estimation

1: Set start guess: $\boldsymbol{\theta} = \boldsymbol{\theta}_0$
2: Outer problem: $\hat{\boldsymbol{\theta}} = \min_{\boldsymbol{\theta}} -l(\boldsymbol{\theta})$
For every evaluation of $-l(\boldsymbol{\theta})$
3: **for all** individuals **do**
4: Set start guess: $\boldsymbol{\eta} = \boldsymbol{\eta}_0$
5: Inner problem: $\boldsymbol{\eta}_i^* = \min_{\boldsymbol{\eta}_i} -l_i(\boldsymbol{\eta}_i)$
6: **end for**

3.3 Optimization

In order to simplify the process of optimization, the negative log-likelihood was minimized using the function `fminunc` (Optimization Toolbox) in MATLAB [15]. The settings were to use the Quasi-Newton method to find the minimum of the objective function with an optimality tolerance of `1e-6`. The Quasi-Newton method is favorable when the Hessian is too expensive to compute at each iteration and instead relies on the approximation of the Hessian using the formula of Broyden, Fletcher, Goldfarb and Shanno (BFGS) [16–19].

4
Results

The results presented in the following sections will be comparing the two earlier mentioned approaches: sequential and joint parameter estimation. The first section gives a brief overview of the qualities of both the K-PD and TTE models, as well as one example of estimating the parameters. In order to compare the two approaches, a larger selection of datasets for different sampling frequencies is generated. This section also contains qualities of the sequential versus joint estimation algorithms with respect to REE and RSE, as well as information about the computational times.

4.1 Simulation of synthetic data

The synthetic data was simulated as described in Section 3.1 by using forward Euler, for details on implementation see Appendix B.1. The parameter values were chosen such that there would be a slow decline in response and clear linkage to the survival, see the selected values in Table 4.1. This slow decline is mainly caused by the relationship between KDE and k_{in}, and keeping the decrease slow will simplify the parameter estimations for less frequent sampled data.

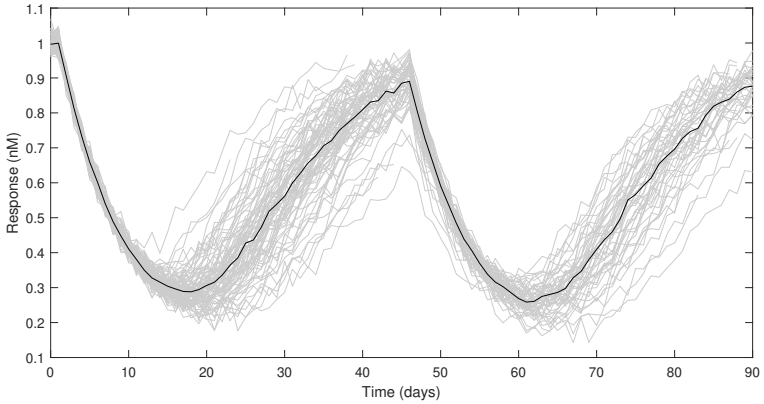
Table 4.1: Population estimations of a sequential and joint approach for an example dataset containing 100 individuals, given a dose of 50 mg at day 0 and day 45, and sampled once every day ($\Delta t = 1$) för a total of 90 days.

Parameter	Simulated	Start guess	Mean parameter estimation (RSE)	
$KDE[\text{d}^{-1}]$	0.2	0.1	0.2 (2%)	0.2 (2%)
$k_{in}[\text{nM.d}^{-1}]$	0.1	0.2	0.1 (0.2%)	0.1 (0.2%)
$EDK_{50}[\text{mg.d}^{-1}]$	0.1	0.2	0.1 (2%)	0.1 (2%)
α	-8.5	-10	-7.8 (7%)	-8.6 (7%)
β	6.2	5.0	5.0 (13%)	6.01 (12%)
σ[nM]	0.02	0.5	0.02 (1%)	0.02 (1%)
ω_{KDE}	0.2	0.8	0.2 (7%)	0.2 (7%)
			Sequential	Joint

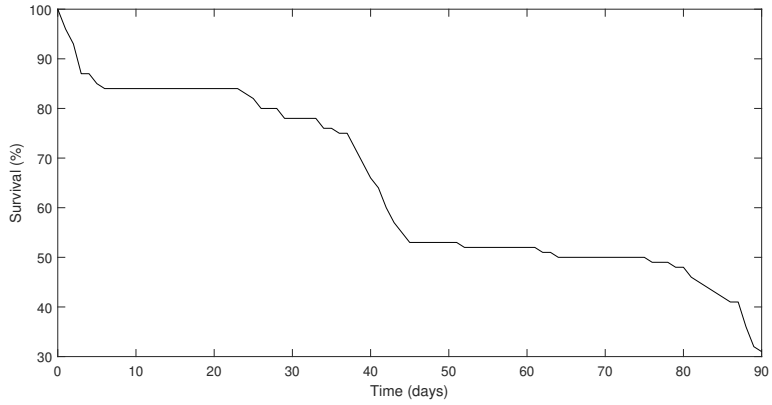
The dataset contained 100 individuals receiving a dose of 50 mg on days 0 and 45, where every individual was sampled once every day ($\Delta t = 1$) for 90 days. The behaviour of this dataset is presented as a response and survival curve in Figure 4.1. The individuals were given the second dosage just before the median response

reaches its steady state, which causes a more compressed survival curve. As mentioned, the hazard parameters were chosen such that the effect of the drug is clearly noticeable in the survival curve. See that the survival curve clearly slows down during time frames where the response is kept at its minimum. Also note that at the final day, 69 % of the individuals had an event occur to them.

The parameters were then estimated for this simulated dataset using both a sequential and joint approach and the results of this can be viewed in Table 4.1-4.2. Both approaches correctly identified the parameters for the longitudinal model but the joint estimation had larger success with regards to the hazard parameters α and β. In contrast, the sequential model identified β as its starting value and overestimated α. This issue is further presented in Appendix C.



(a) Pharmacodynamics



(b) Survival

Figure 4.1: Response and survival curves for a dataset consisting of 100 individuals, given a dose of 50 mg at day 0 and day 45 and sampled once a day ($\Delta t = 1$) over a 90 day period. The gray lines indicate the individual response curves and the black curve the median response.

As for the RSE, see Table 4.1, there is no significant difference between the estimations, even though it differs for β. For the K-PD parameters, the RSE is very small and between 0-2 %, which indicates excellent precision. The larger values of RSE is

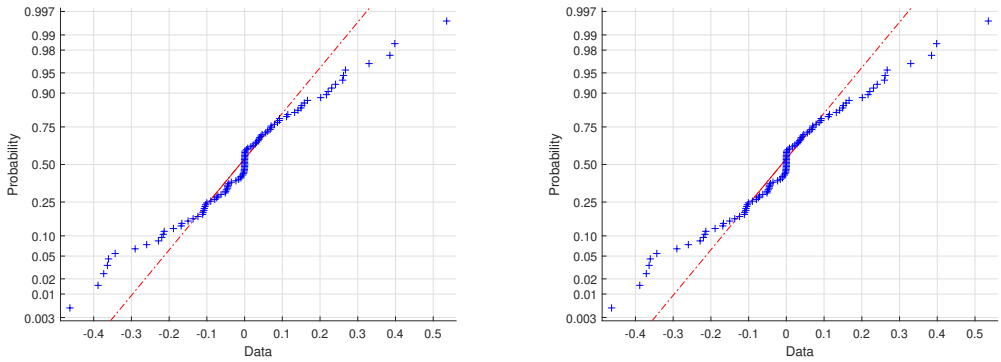
for α, β and ω_{KDE} with errors closer to 5-13%, which is still an acceptable range.

To check the normality assumption regarding the random effects, a normal plot of the optimal $\boldsymbol{\eta}^*$ values was created and is presented in Figure 4.2. From these plots it is noteworthy to see that both the sequential and joint model approach creates fat tails when fitting the random effects. In other words, the algorithms struggle to identify very large and very small random effects. Furthermore, see that the difference between the two approaches is minuscule.

For visualisation, eight randomized individual curves with both added fits were plotted and can be viewed in Figure 4.3. Again, both of the approaches performed well and correctly fit the individual data with regards to the K-PD model. This also goes for the individuals who had an event occur early in the study, meaning that not as many data points were available for fitting the inter-individual variability, ω_{KDE}.

Table 4.2: REE for a dataset consisting of 100 individuals who where sampled once every day for the sequential and joint modeling approaches.

Parameter	REE (%)	
$KDE[\text{d}^{-1}]$	-0.6	-0.6
$k_{in}[\text{nM.d}^{-1}]$	0.2	0.2
$EDK_{50}[\text{mg.d}^{-1}]$	0.2	0.2
α	-8.4	1.2
β	-19.3	-3.0
$\sigma[\text{nM}]$	0.5	0.5
ω_{KDE}	-12.9	-12.9
	Sequential	Joint



(a) **(b)**

Figure 4.2: Normal probability curves with respect to the random effects for the dataset of 100 individuals generated, using the parameters found in Table 4.1, for a (a) sequential and (b) joint modeling technique.

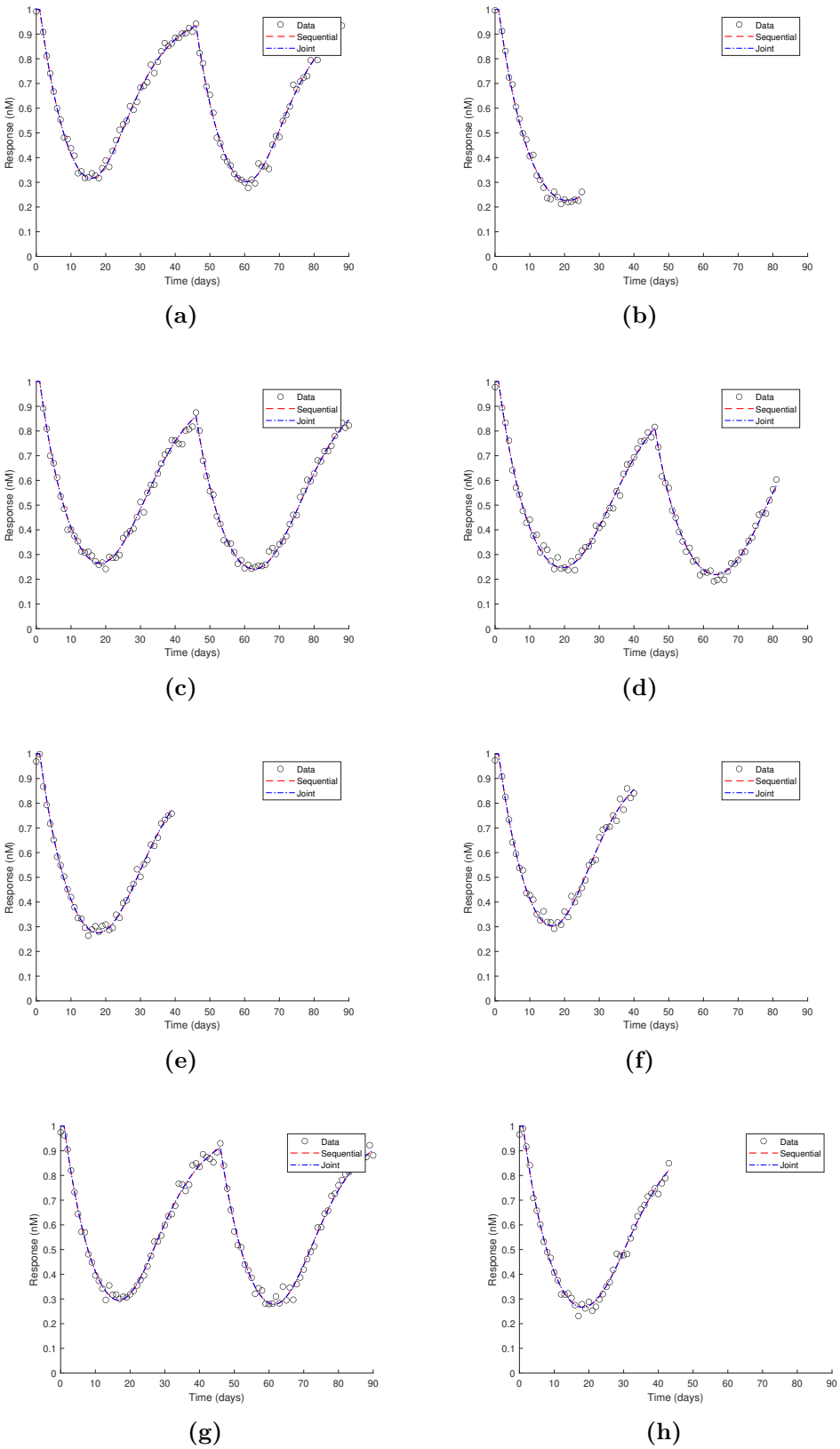


Figure 4.3: Fitted response curves for 8 individuals with a sequential (red) and joint (blue) modeling approach with population parameters found in Table 4.1.

4.2 Comparison of the effect of sampling frequency for the joint and sequential approaches

In order to investigate the effect of different sampling frequencies, 28 datasets with 100 individuals sampled at four different frequencies were computed for the following scenario: split the population in half such that 50% of the individuals do not receive treatment (control group). This was done with the aim of trying to estimate the baseline hazard better. Therefore the following section will compare the effect of increasing sample frequency with the investigated parameters presented in Table 4.3. The second section will include a section of similar analysis with datasets simulated for the case of everyone receiving treatment, as well as with the addition of comparative computational times between the two estimation approaches.

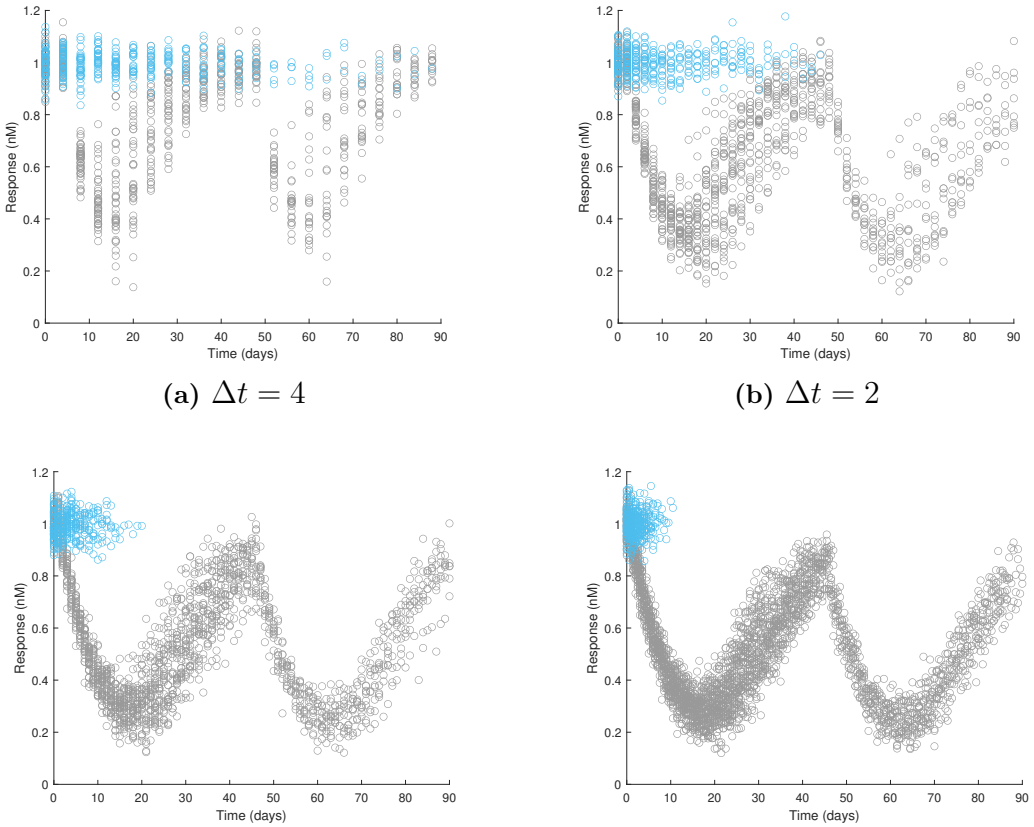
Table 4.3: Parameter values used for simulating data.

Parameter	Simulated
$KDE[\mathrm{d}^{-1}]$	0.2
$k_{in}[\mathrm{nM.d}^{-1}]$	0.1
$EDK_{50}[\mathrm{mg.d}^{-1}]$	0.1
α	-8.0
β	6.2
$\sigma[\mathrm{nM}]$	0.05
ω_{KDE}	0.2

The different sampling frequencies investigated were: once every four days ($\Delta t = 4$), once every other day ($\Delta t = 2$), once every day ($\Delta t = 1$) and lastly twice a day ($\Delta t = 0.5$). In order to avoid local minima the start guesses for each iterations were set to a random perturbation of 10% to the relative size of the true value of the parameter. The treatment group received two doses of 50 mg, one on day 0 and one on day 45, and all individuals were observed for 90 days.

Examples of the data for the four different sampling frequencies can be viewed in Figure 4.4. These figures show the response curve over time where the blue dots represent the control group and the grey dots the group who receive treatment. Note here that the data points which seem to be constant over time in $\Delta t = 2$, representing the control group, steadily disappears as these individuals drop out early with the resulting large hazard. As the hazard is relative to the step size (see Section 3.1.2), these individuals have events occur at a larger rate for smaller step sizes which turns this into the cluster seen around the beginning for $\Delta t = 1$ and $\Delta t = 0.5$. This can be interpreted as the least frequently sampled groups having a smaller linkage between the event and the measured response and vice versa fo the higher sampling rates. See that with the increasing sample frequencies it becomes easier to read out the shape of the K-PD dynamics, even with the noise present. As for the case with the least frequent sampling, every four days, it becomes harder to distinguish the response dynamics.

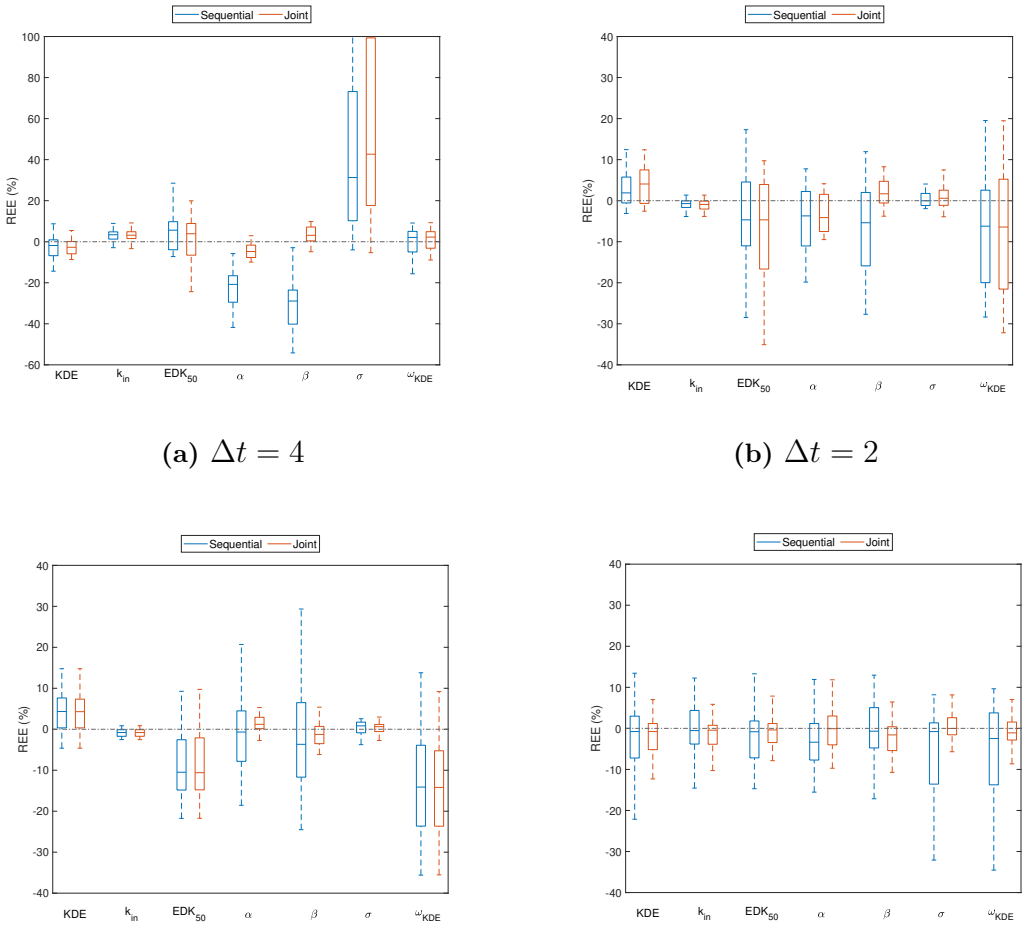
The parameter estimation results can be viewed in Figures 4.5-4.7, as well as additional material in Appendix D. In Figure 4.5, the REE is presented as boxplots where the bottom and top edges of the box represent the 25th to 75th quantiles for each of the sampling frequencies, whisker length was set to one. The joint approach performs better, meaning there is a smaller variation and mean of error, for most parameters in the case of high frequent sampling. Most noticeable is the improvement in estimating the hazard parameters α and β. Also note the difficulty that the algorithms have with the lowest sampling frequency, $\Delta t = 4$. However, even for this approach it would seem that the joint model technique created closer to true estimations for the TTE model. For the case with the highest sampling frequency, $\Delta t = 0.5$, both techniques estimate the true values quite well but with the difference that there is less bias in the joint approach. Furthermore, for the case of $\Delta t = 2$ and $\Delta t = 1$ see that there seems to be a difficulty in estimating the standard deviation of the random effects, ω_{KDE}, which is not present for the other two sampling frequencies. In contrast, $\Delta t = 4$ and $\Delta t = 0.5$ instead have a larger variation and mean when determining the standard error of the noise variable, σ.



(a) $\Delta t = 4$ **(b)** $\Delta t = 2$

(c) $\Delta t = 1$ **(d)** $\Delta t = 0.5$

Figure 4.4: Examples of data for the four different sampling frequencies. The data was simulated for 100 individuals, including a control group (colored blue).



(a) $\Delta t = 4$ **(b)** $\Delta t = 2$

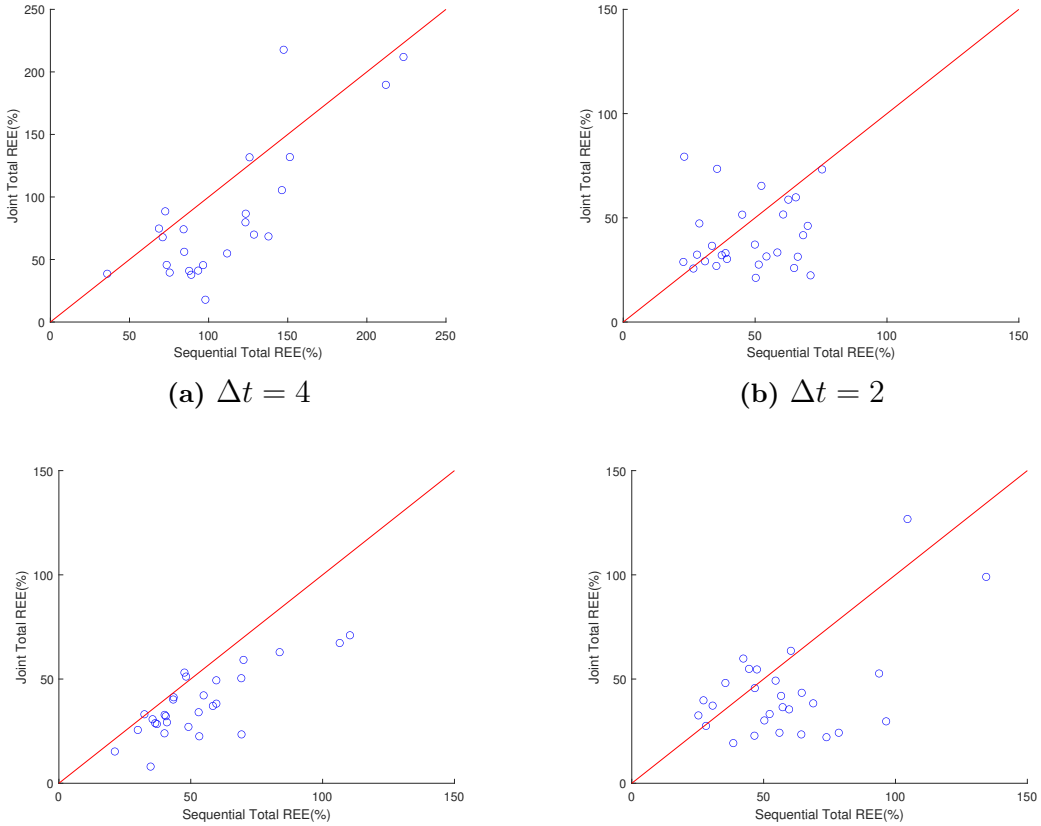
(c) $\Delta t = 1$ **(d)** $\Delta t = 0.5$

Figure 4.5: REE for the four different sampling frequencies for 28 iterations when using a joint (orange) and sequential (blue) modeling technique. The datasets were simulated for 100 individuals, with a control group of 50% and the rest receiving two doses of 50 mg.

Another view of the estimation errors is plotted in Figure 4.6, where each of the REE-values for every parameter is added together for the specific sampling frequency. The red line indicates the scenario where the sum of REE is equal for both approaches. For all four cases, the joint approach produces a lower REE in the majority of the datasets. There is also a slight trend in the data for $\Delta t = 4$, $\Delta t = 1$ and $\Delta t = 0.5$, suggesting that some datasets might have been more difficult fit for both models. There is the least difference between data points on each side of the red line (meaning the models perform more equally) for the most high frequent sampling, $\Delta t = 0.5$, but even so there are 8 points to the left of the line and 20 on the right, suggesting that joint performs better in estimation accuracy.

To further investigate the precision in the estimations, the RSE was calculated and can be viewed in Figure 4.7. However, since the optimizer did not always find an

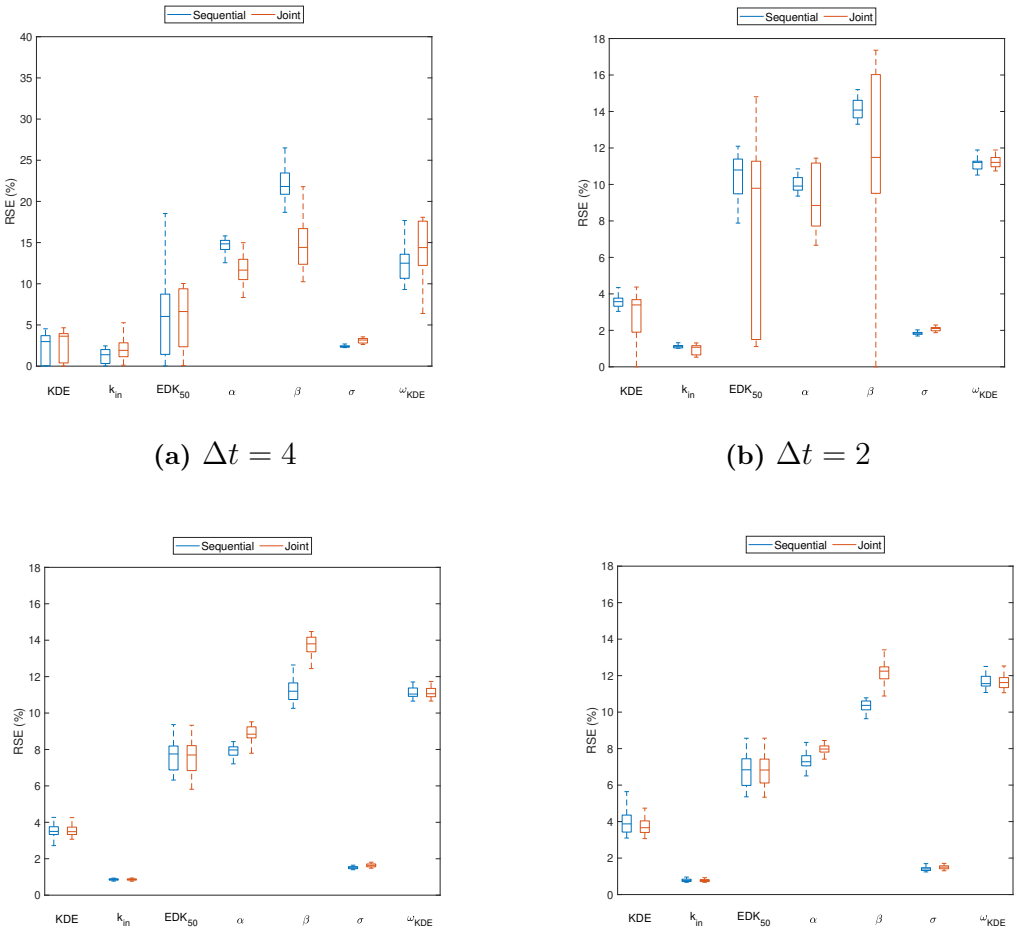
optimum the approximated Hessian in these estimated parameters was not positive definite. Hence, the relative standard errors could not be estimated and were therefore discarded for the following analysis. As for the values that could be calculated, then overall there are only slight differences in the joint and sequential estimation approaches. It is noticeable for α and β for the joint model in the case of higher frequent sampling, where there was a slight increase in variability. Furthermore, note the seemingly large variability in EDK_{50}, α and β for $\Delta t = 2$. Since the scaling is different to that of $\Delta t = 4$, this is only an illusion, as they are of comparable size in actuality.



(a) $\Delta t = 4$ **(b)** $\Delta t = 2$

(c) $\Delta t = 1$ **(d)** $\Delta t = 0.5$

Figure 4.6: Sum of the absolute value of REE for the four different sampling frequencies for 28 iterations. The red line indicates the case where both of the sums, for the joint and sequential model, are equal. In total, 2 outliers in (a) was removed for clarity in the plot.



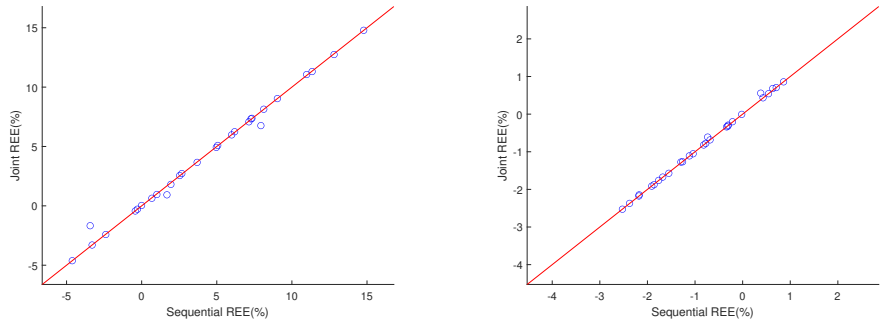
(a) $\Delta t = 4$ **(b)** $\Delta t = 2$

(c) $\Delta t = 1$ **(d)** $\Delta t = 0.5$

Figure 4.7: RSE for the four different sampling frequencies for 28 iterations. All repetition datasets were simulated for 100 individuals, where a control group of 50% and the rest receiving two doses of 50mg.

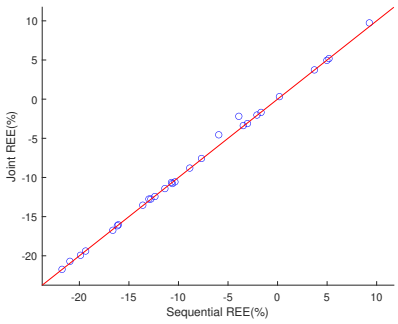
4.2.1 Correlations between REE and different parameters

To check if both approaches have difficulties with some datasets, the REE values were plotted against each other for every parameter and for each sampling frequency. Below is one example for $\Delta t = 1$. For this specific case it is clear that what causes this shift towards a larger total REE for the sequential approach lies within the estimations of α and β. As for the parameters related to the K-PD model, there is instead a strong correlation between datasets meaning that both the sequential and joint technique perform equally for these specific parameters.

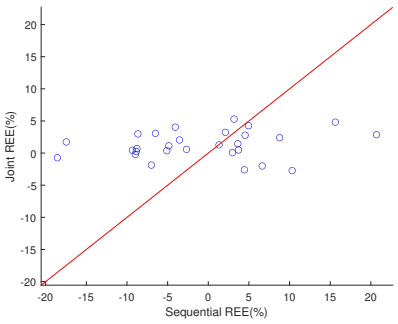


(a) KDE

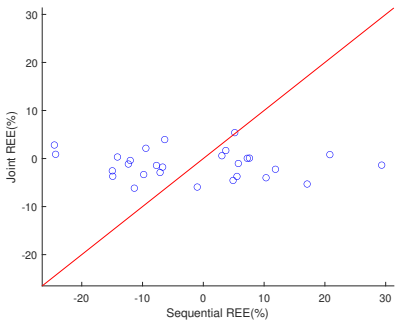
(b) k_{in}



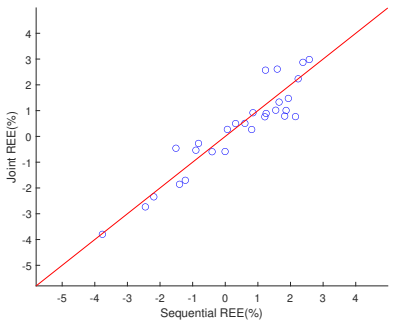
(c) EDK_{50}



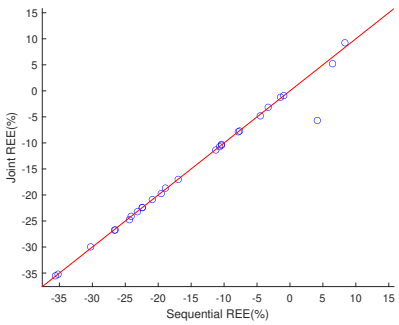
(d) α



(e) β



(f) σ

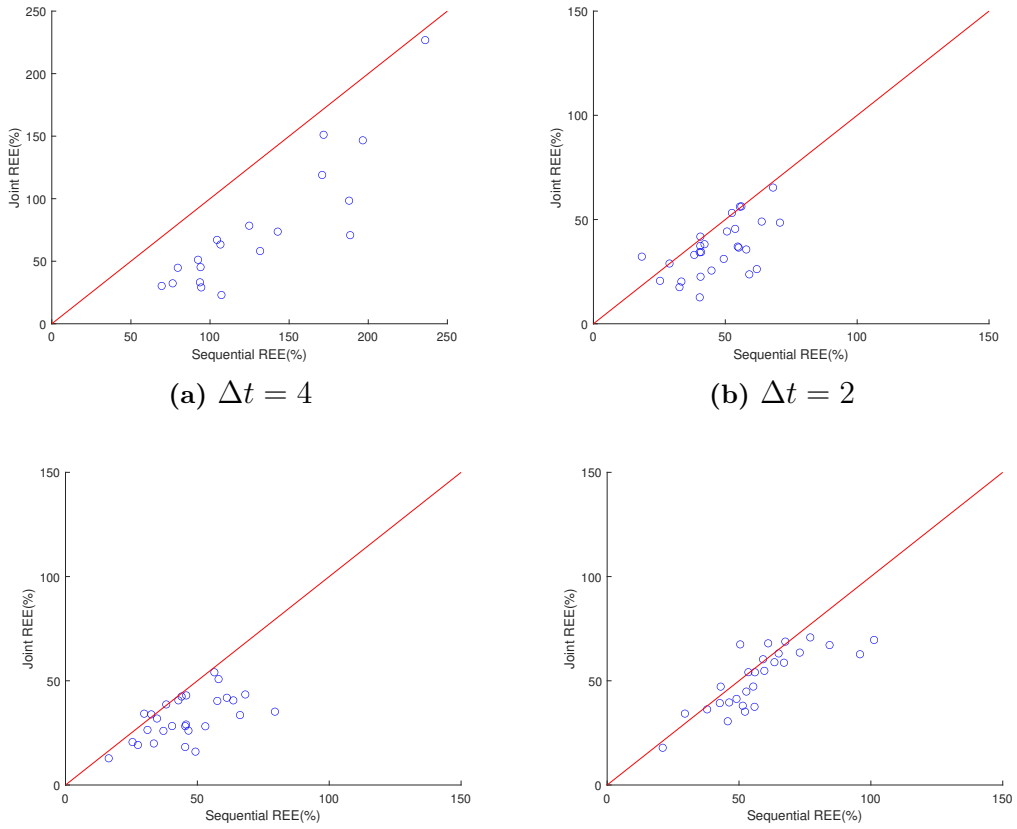


(g) ω_{KDE}

Figure 4.8: Total REE for the four different sampling frequencies for 28 iterations. The red line indicates the case where both of the sums, for the joint and sequential model, are equal.

4.2.2 The effect of sampling frequency for populations without a control group

The same analysis described in section 4.2 was carried out for a sample group of 100 individuals where everyone received treatment consisting of a drug of 50 mg on days 0 and 45. See Figure 4.9 for the total REE for each sampling frequency. Note that in the order of largest ($\Delta t = 4$) to smallest ($\Delta t = 0.5$), joint performs better in: 28, 23, 25 and 21 of the datasets. Overall, this showed similar results to that which was previously shown, meaning that for all cases the joint approach produces better estimations and overall lower total REE values.



(a) $\Delta t = 4$ **(b)** $\Delta t = 2$

(c) $\Delta t = 1$ **(d)** $\Delta t = 0.5$

Figure 4.9: Total REE for the four different sampling frequencies for 28 iterations. The red line indicates the case where both of the sums, for the joint and sequential model, are equal. In total, 7 outliers in (a) and one outlier in (c) were removed for clarity in the plots.

An equal analysis as explained above was also carried out the scenario of 100 individuals with no treatment group, for the case of further-from-true start guesses, presented in Appendix D. There were some exceptions from the previous results in the case of the sampling frequencies $\Delta t = 2$ and $\Delta t = 1$ where the sequential approach performed better. This occurred mainly for α and β, as the K-PD parameter estimations were comparatively similar between methods.

4.2.3 Computational times

The average computational times for both cases of control group versus no control group for both the sequential and joint model approaches are presented in Table 4.4. In total, parameters were estimated for 28 iterations for each sampling frequency. The sequential approach is about twice as fast as the joint model for lower frequent sampling, most noticeable is the difference for the case with a control group present. For 100 individuals, no control group, all computational times are larger compared to with a control. Overall it is quite clear that the joint parameter estimations is more time demanding. The difference between the two approaches also decrease with more dense sample frequency.

Table 4.4: Mean computational times for the four different sampling frequencies for both the sequential and joint modeling approaches for 100 individuals and averaged over 28 repeats. For the case with a control group present, 50 individuals received treatment.

Sampling frequency	Average time (standard dev.), 50% control group [minutes]		Average time (standard dev.), no control group [minutes]	
$\Delta t = 4$	4.0 (1.8)	2.3 (0.7)	5.6 (1.8)	3.6 (1.0)
$\Delta t = 2$	4.9 (0.8)	2.7 (0.7)	8.7 (1.2)	4.5 (1.1)
$\Delta t = 1$	8.3 (1.5)	4.6 (0.6)	14.2 (4.8)	7.4 (1.5)
$\Delta t = 0.5$	14.7 (2.3)	9.9 (4.4)	22.6 (3.2)	15.8 (9.1)
	Joint	Sequential	Joint	Sequential

5
Discussion

To revisit the aim of this thesis: identify scenarios where the joint parameter estimation is suitable and find possible pitfalls of implementation. The results do suggest that joint parameter estimation is superior, for this model structure of a joint K-PD and parametric hazard, to that of a sequential approach. Furthermore, specifically the difference in the two approaches is found in the hazard parameters where the joint approach has larger success in estimation without loosing too much precision. This was shown for several sampling frequencies and for settings of populations with and without a control group. However, both algorithms showed some struggle with identifying the qualities of the random effects. The following sections will discuss these main findings.

Section 4.1 showed that both algorithms correctly identified the K-PD parameters. The main difference was that the sequential approach underestimated both hazard parameters, which the joint approach could correctly estimate. This could be caused by a number of things. Firstly, if one looks at the baseline hazard at $t = 0$, which is determined by $\alpha + \beta$, both approaches seems to capture this quite well. In Appendix C, there were also signs that the hazard parameters could be correlated, causing different parameter values with the same baseline to give similar datasets, see Figure C.2. In combination with the fact that we are using predictions as covariates in the sequential approach, which could alter the likelihood landscape slightly, this could have caused bias, which has been previously shown [20].

Similar results are shown in Section 4.2, where a more systematic approach is implemented to get a clearer view of the estimation methods. See in Figure 4.6 that the joint approach produces a lower total REE in the majority of the datasets for all sampling frequencies. This is also clear from the individual boxplots in Figure 4.5, with the exception of issues in estimation σ and ω_{KDE} in some cases. The cause of either a bad estimation of σ or ω_{KDE}, with the basis of different sampling frequencies, could be that it becomes difficult to separate the two. As seen in Figure 4.4, for the lower sampled cases, we noted that the dynamics change quicker than the sampling. This could explain the difficulty that arises when estimating the noise. We do also see the same behaviour of the sequential estimation of α and β, for low sampled data, where it never captures the true value. This behaviour does however seem to correct itself when increasing the sampling frequency to $\Delta t = 0.5$. Another possible explanation is the issue with identifying more extreme values of the random effects. As was shown in Figure 4.2, both approaches struggle with identifying these values which will influence the estimation of their variance.

Previous studies that compares the joint approach to more simpler techniques has shown that the joint approach creates better estimates for the standard deviation of the random effects, ω, although be it for a different model and parametric hazard [6]. In comparison, we managed to show that the joint approach overall produce better estimates for that hazard parameters for all investigated sampling frequencies. However, in the case of the highest sampling frequency, $\Delta t = 0.5$, see Figure 4.5, the joint approach does produce significantly more stable estimations for both σ and ω_{KDE}. To conclude, this shows that the joint approach has promising results which could be applied to a multitude of models.

Important to note in studies of this sort, is that the amount of individuals needed and generated could influence the results. Since we could see in Section 4.2.3 that the joint model is overall more computationally expensive and slow, this limited the amount of simulations run. Because it is inherent that the K-PD model gets more data in total when working with smaller populations, causing an unfair amount of data for the hazard parameters. Future work should therefor include similar analyses for larger datasets, i.e more individuals, with more iterations to fully conclude that the joint estimation technique is indeed superior in the case of a K-PD-TTE model.

6
Conclusion

This thesis has concluded that there is some evidence that the joint likelihood approach produce better parameter estimations for a joint K-PD-TTE model, in comparison to a sequential method. More specifically this is seen in the hazard parameters but an overall improvement of stability in the estimations can be shown. Additionally, even though there is a slight decrease in precision, shown via larger coefficients of variance, these differences are not large enough to make an impact. However, as the joint model is more computationally complex there could be cases of more simple models where a sequential approach might suffice.

Overall this thesis shows that when possible, a joint modeling approach should be used for a K-PD modeling framework. Future work could focus on extending these results for a full PK-PD model, although it is advised to use caution and check for identifiability. Another important note for future work is to extend the data used and repetitions done to create more statistical inference of the differences between a joint and sequential parameter estimation framework.

Bibliography

1. Zou, H., Banerjee, P., Leung, S. S. Y. & Yan, X. Application of Pharmacokinetic-Pharmacodynamic Modeling in Drug Delivery: Development and Challenges. *Frontiers in Pharmacology* **11.** ISSN: 1663-9812 (2020).
2. Kebede, S., Tessema, Z., Balcha, S. & Ayele, T. *Joint Modeling of Time to Diabetic Retinopathy and Change in Fasting Blood Sugar Among Type 2 Diabetic Patients in University of Gondar Comprehensive Specialized Hospital, Northwest Ethiopia* Apr. 2021.
3. Leander, J., Jirstrand, M., Eriksson, U. G. & Palmér, R. A stochastic mixed effects model to assess treatment effects and fluctuations in home-measured peak expiratory flow and the association with exacerbation risk in asthma. *CPT Pharmacometrics Syst Pharmacol* (Nov. 2021).
4. Upton, R. N. & Mould, D. R. Basic concepts in population modeling, simulation, and model-based drug development: part 3-introduction to pharmacodynamic modeling methods. *CPT: pharmacometrics & systems pharmacology* **3,** e88–e88 (Jan. 2014).
5. Albert, P. S. & Shih, J. H. On estimating the relationship between longitudinal measurements and time-to-event data using a simple two-stage procedure. en. *Biometrics* **66,** 983–7, discussion 987–91 (Sept. 2010).
6. Desmée, S., Mentré, F., Veyrat-Follet, C. & Guedj, J. Nonlinear Mixed-Effect Models for Prostate-Specific Antigen Kinetics and Link with Survival in the Context of Metastatic Prostate Cancer: a Comparison by Simulation of Two-Stage and Joint Approaches. *The AAPS Journal* **17,** 691–699. ISSN: 1550-7416 (May 2015).
7. Wallman, M. *et al.* An Integrative Approach for Improved Assessment of Cardiovascular Safety Data. *Journal of Pharmacology and Experimental Therapeutics* **377,** 218–231. ISSN: 0022-3565 (2021).
8. Cardilin, T. *et al.* Exposure-response modeling improves selection of radiation and radiosensitizer combinations. *Journal of Pharmacokinetics and Pharmacodynamics,* 1–12 (Oct. 2021).
9. Leander, J., Almquist, J., Johnning, A., Larsson, J. & Jirstrand, M. Nonlinear Mixed Effects Modeling of Deterministic and Stochastic Dynamical Systems in Wolfram Mathematica. *IFAC-PapersOnLine* **54.** 19th IFAC Symposium on System Identification SYSID 2021, 409–414. ISSN: 2405-8963 (2021).
10. DiStefano III, J. *Dynamic Systems Biology and Simulation* ISBN: 9788578110796 (Academic Press, 2015).

11. Clark, T. G., Bradburn, M. J., Love, S. B. & Altman, D. G. Survival analysis part I: basic concepts and first analyses. en. *Br J Cancer* **89,** 232–238 (July 2003).
12. Tutz, G. & Schmid, M. Modeling Discrete Time-to-Event Data (Jan. 2016).
13. Raue, A. *et al.* Structural and practical identifiability analysis of partially observed dynamical models by exploiting the profile likelihood. *Bioinformatics* **25,** 1923–1929 (June 2009).
14. Jacqmin, P. *et al.* Modelling Response Time Profiles in the Absence of Drug Concentrations: Definition and Performance Evaluation of the K–PD Model. *Journal of Pharmacokinetics and Pharmacodynamics* **34,** 57–85. https://doi.org/10.1007/s10928-006-9035-z (2007).
15. MATLAB. *9.11.0.1837725 (R2021b) Update 2* (The MathWorks Inc., Natick, Massachusetts, 2021).
16. BROYDEN, C. G. The Convergence of a Class of Double-rank Minimization Algorithms 1. General Considerations. *IMA Journal of Applied Mathematics* **6,** 76–90. ISSN: 0272-4960. eprint: https://academic.oup.com/imamat/article-pdf/6/1/76/2233756/6-1-76.pdf (Mar. 1970).
17. Fletcher, R. A new approach to variable metric algorithms. *The Computer Journal* **13,** 317–322. ISSN: 0010-4620. eprint: https://academic.oup.com/comjnl/article-pdf/13/3/317/988678/130317.pdf (Jan. 1970).
18. Goldfarb, D. A family of variable-metric methods derived by variational means. *Mathematics of computation* **24,** 23–26 (1970).
19. Shanno, D. F. Conditioning of Quasi-Newton Methods for Function Minimization. *Mathematics of Computation* **24,** 647–656. ISSN: 00255718, 10886842. http://www.jstor.org/stable/2004840 (2022) (1970).
20. Albert, P. S. & Shih, J. H. On Estimating the Relationship between Longitudinal Measurements and Time-to-Event Data Using a Simple Two-Stage Procedure. *Biometrics* **66,** 983–987. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1541-0420.2009.01324_1.x. https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1541-0420.2009.01324_1.x (2010).

A

Derivation of sensitivities

The sections below will present the derivation of the sensitivities used for Algorithm 1 and 2. As for the derivatives where finite differences where applied, the step size h was set to `1e-6`.

A.1 Residual derivatives

This section describes explicitly how the derivatives in Eq. 3.12 are computed. Below are the two residual derivatives presented in regards to the two different longitudinal models. The only small difference between them is if the parameterization of the inhibition via the state IR is applied or not. The model presented in Section 3.1.1 in Eq. 3.4-3.5. The derivatives becomes as following

$$\frac{\mathrm{d}\epsilon_{ij}}{\mathrm{d}\eta_{ik}} = \frac{\mathrm{d}\left(\hat{R}_{ij} - R_{ij}\right)}{\mathrm{d}\eta_{ik}} = -\left(\frac{\partial R_{ij}}{\partial \eta_{ik}} + \frac{\partial R_{ij}}{\partial IR_{ij}}\frac{\mathrm{d}IR_{ij}}{\mathrm{d}\eta_{ik}} + \frac{\partial R_{ij}}{\partial IR_{ij}}\frac{\partial IR_{ij}}{\partial A_{ij}}\frac{\mathrm{d}A_{ij}}{\mathrm{d}\eta_{ik}}\right). \tag{A.1}$$

More specifically each term is given by

$$\frac{\partial R_{ij}}{\partial \eta_{ik}} = \frac{\partial R_{ij}(\boldsymbol{\eta}_i)}{\partial \eta_{ik}} \approx \frac{R_{ij}(\boldsymbol{\eta}_i + h) - R_{ij}(\boldsymbol{\eta}_i)}{h}, \tag{A.2}$$

$$\begin{aligned}
\frac{\partial R_{ij}}{\partial IR_{ij}} \cdot \frac{\mathrm{d}IR_{ij}}{\mathrm{d}\eta_{ik}} &\approx \frac{\left(R_{ij} - R_{i,j-1}\right)/\Delta t}{\left(IR_{ij} - IR_{i,j-1}\right)/\Delta t} \cdot \frac{\mathrm{d}IR_{ij}}{\mathrm{d}\eta_{ik}} = \\
&= -\frac{k_{in}\Big(1 - IR(t_{j-1})/\Big(EDK_{50} + IR(t_{j-1})\Big) - k_{out} \cdot R(t_{j-1})}{KDE^2 \cdot A_{i,j-1}} \cdot KDE \cdot A_{i,j-1} \cdot \boldsymbol{v}_{KDE} = \\
&= -\frac{k_{in}\Big(1 - IR(t_{j-1})/\Big(EDK_{50} + IR(t_{j-1})\Big) - k_{out} \cdot R(t_{j-1})}{KDE} \cdot \boldsymbol{v}_{KDE},
\end{aligned} \tag{A.3}$$

$$\frac{\partial R_{ij}}{\partial IR_{ij}} \cdot \frac{\partial IR_{ij}}{\partial A_{ij}} \cdot \frac{\mathrm{d}A_{ij}}{\mathrm{d}\eta_{ik}} \approx \frac{\Big(R_{ij} - R_{i,j-1}\Big)/\Delta t}{\Big(IR_{ij} - IR_{i,j-1}\Big)/\Delta t} \cdot \frac{\Big(IR_{ij} - IR_{i,j-1}\Big)/\Delta t}{\Big(A_{ij} - A_{i,j-1}\Big)/\Delta t} \cdot \frac{\mathrm{d}A_{ij}}{\mathrm{d}\eta_{ik}} =$$

$$= -\frac{k_{in}\Big(1 - IR(t_{j-1})/\Big(EDK_{50} + IR(t_{j-1})\Big) - k_{out} \cdot R(t_{j-1})}{KDE^2 \cdot A_{i,j-1}}$$

$$\cdot \frac{-KDE^2 \cdot A_{i,j-1}}{-KDE \cdot A_{i,j-1}} \cdot \frac{A_{ij}(\boldsymbol{\eta}_i + h) - A_{ij}(\boldsymbol{\eta}_i)}{h} =$$

$$= -\frac{\mathrm{d}k_{in}\Big(1 - IR(t_{j-1})/\Big(EDK_{50} + IR(t_{j-1})\Big) - k_{out} \cdot R(t_{j-1})}{KDE \cdot A_{i,j-1}}$$

$$\cdot \frac{A_{ij}(\boldsymbol{\eta}_i + h) - A_{ij}(\boldsymbol{\eta}_i)}{h}, \tag{A.4}$$

where $\boldsymbol{v}_{KDE}$ is 1 if and only if the random effect is in regards to KDE, otherwise it is zero.

A.2 Hazard derivatives

The derivatives in regards to the hazard can be computed in a similar manner. See the following equation which was derived in Section 3.2.2 using the chain rule

$$\frac{\mathrm{d}h(j|R_{ij})}{\mathrm{d}\eta_{ik}} = \frac{\mathrm{d}h(j|R_{ij})}{\mathrm{d}\eta_{ik}} + \frac{\mathrm{d}h(j|R_{ij})}{\mathrm{d}R_{ij}} \left(\frac{\partial R_{ij}}{\partial \eta_{ik}} + \frac{\partial R_{ij}}{\partial IR_{ij}} \frac{\mathrm{d}IR_{ij}}{\mathrm{d}\eta_{ik}} + \frac{\partial R_{ij}}{\partial IR_{ij}} \frac{\partial IR_{ij}}{\partial A_{ij}} \frac{A_{ij}}{\mathrm{d}\eta_{ik}} + \frac{\partial R_{ij}}{\partial A_{ij}} \frac{A_{ij}}{\mathrm{d}\eta_{ik}} \right) =$$

$$= \{R \text{ not a function of } A\} =$$

$$= \frac{\mathrm{d}h(j|R_{ij})}{\mathrm{d}\eta_{ik}} + \frac{\mathrm{d}h(j|R_{ij})}{\mathrm{d}R_{ij}} \cdot \left(\frac{\partial R_{ij}}{\partial \eta_{ik}} + \frac{\partial R_{ij}}{\partial IR_{ij}} \frac{\mathrm{d}IR_{ij}}{\mathrm{d}\eta_{ik}} + \frac{\partial R_{ij}}{\partial IR_{ij}} \frac{\partial IR_{ij}}{\partial A_{ij}} \frac{\mathrm{d}A_{ij}}{\mathrm{d}\eta_{ik}} \right) =$$

$$= \frac{\mathrm{d}h(j|R_{ij})}{\mathrm{d}\eta_{ik}} - \frac{\mathrm{d}h(j|R_{ij})}{\mathrm{d}R_{ij}} \frac{\mathrm{d}\epsilon_{ij}}{\mathrm{d}\eta_{ik}} \tag{A.5}$$

Since no random effects applied to the hazard parameters was investigated, the first term is set to zero. As for the second term, we can approximate the derivative by explicit finite differences

$$\frac{\mathrm{d}h(j|R_{ij})}{\mathrm{d}R_{ij}} \approx \frac{h(j|R_{ij} + h) - h(j|R_{ij})}{h} \tag{A.6}$$

The final expression becomes

$$\frac{\mathrm{d}h(j|R_{ij})}{\mathrm{d}\eta_{ik}} \approx -\frac{h(j|R_{ij} + h) - h(j|R_{ij})}{h} \cdot \frac{\mathrm{d}\epsilon_{ij}}{\mathrm{d}\eta_{ik}}. \tag{A.7}$$

B

Code

The following sections will contain the code for the major functions used in this thesis. All the results are based on iterations and modifications of inputs for the functions below.

B.1 Synthetic data simulation

```
function [u, h, survival] = simulate_data(N, N_states, t, dt, U0, ...
    dudt, param, Ndosage, dosage, control, h_eq, include_death)
% Simulates data for a K-PD-TTE model for a given model structure and
% hazard using forward Euler

% inputs: N: number of individuals, N_states = number of states,
%         t: time vector, dt: time, U0: start values [N,k],
%         dudt: set of equations describing K-PD dynamics,
%         param: can be individual or population parameters,
%         dosage: dose of drug (mg), control: Nx1 of binary inputs (1 -
%         individual is given the drug, zero otherwise),
%         h_eq: chosen hazard equation, include_death: true if events
%         should be simulated, false otherwise.

% outputs: u: Nxlength(t)xk matrix containing the solutions for k states,
%             (1: A, 2: R, 3:IR )
%          h: values of the hazard
%          survival: how many subjects experience an event in each time
%          step

% time and dosing
```

```
22 input = repmat([dosage, zeros(1,floor(length(t)/Ndosage)-1)
       ], [1, Ndosage]);
23
24 % starting values
25 u = zeros(N, length(t), N_states);
26 u(:,1,:) = U0;
27 h = zeros(N, length(t));
28 saved_deaths = zeros(length(t), N);
29 survival = zeros(length(t), 1);
30
31 h(:, 1) = h_eq(u(:,1,2), param);
32
33 %Forward Euler
34     for i = 1:length(t)-1
35         % Step forward
36         if input(i) ~= 0
37             u(:, i,1) = u(:, i,1) + dosage.*control;
38         end
39         step = reshape(dudt(u,i, param,U0), [N, N_states]);
40         u(:,i + 1,1:(N_states-1)) = reshape(u(:, i,1:
               N_states-1), N, N_states-1) + dt*step(:,1:
               N_states-1);
41         step_plus = reshape(dudt(u,i+1, param,U0), [N,
               N_states]);
42
43         % Update
44         u(:,i + 1, end) = step_plus(:,end);
45         h(:,i+1) = h_eq(u(:,i+1,2), param);
46
47         if include_death == true
48         % Who died
49         r = rand(N,1);
50         death_ind = find(h(1:N,i+1) > r);
51         saved_deaths(i+1,1:length(death_ind)) = death_ind;
52         u(death_ind,(i+1):end,:) = NaN(size(death_ind, 1),
               length(t)-i, N_states);
53         h(death_ind,(i+1):end) = NaN(size(death_ind, 1),
               length(t)-i);
54
55         survival(i+1)=length(unique([nonzeros(saved_deaths
               (1:i+1,:)); nonzeros(saved_deaths(1:i,:))]));
56         end
57
58     end
59 end
```

B.2 Longitudinal parameter estimation using FOCE

```
function [loglikelihood, eta_fp] = FOCE(param, eta_guess,
    dudt, dxdt, R_data, t, ...
     N_states, dt, Ndosage, dosage, control_input, R,
        chosen_param, h_eq, event_time)
% FOCE algorithm, computes the negative loglikelihood of a
    Nonlinear Mixed
% Effects Model by finding the fixed points of the random
    effects and
% approximating the marginal likelihood in this point
%
% input: params : a px1 vector containing the parameters
    belonging to the
%                 KPD model.
%        eta_guess : scalar, starting guess for eta.
%        dudt : ODE equations of the KPD system.
%        dxdt : equations expressing the residual
    derivatives.
%        R_data : response variable data collected.
%        t : time vector representing when the data was
    sampled.
%        N_states : the number of elements present in the
    KPD model.
%        dt : time step for simulating data.
%        Ndosage : time vector of when the drug was
    administrated.
%        dosage : scalar, represents how large the dose is.
%        control_input: Nx1 vector of who receives the drug
    (1 if true,
%        zero otherwise)
%        U0 : initial values for simulating the data.
%        chosen_param : 1xn_p vector, represents which
    parameters have
%        added random effects.
%        h_eq : chosen paramaterized hazard equation for the
     simulated
%        data.
%        event_time : Nx1 vector of the index representing
    the interval
%        where each individual had an event occur, if
    event_time = t(end)
%        then event did not occur.
%
% output: loglikelihood : the negative loglikelihood of the
    NLME model.
```

```
30 %         eta_fp : the fixed points of eta for a given set
       of parameters.
31
32      if length(chosen_param) > 1
33          omega = [param(end-3)^2, param(end-2); param(end-1),
                param(end)^2];
34          omega_inv = inv(omega);
35      else
36          omega = param(end,:).^2;
37          omega_inv = inv(omega);
38      end
39      sigma = param(end-length(chosen_param)*length(
           chosen_param));
40      res_sq = @(eta, R_data, param, control) Res_sq(param,eta
           , R_data, dudt, t, ...
41          N_states, dt, Ndosage, dosage, control, chosen_param
               , R, h_eq);
42
43      inner_likeli = @(eta, R_data, param, control) 0.5*sum(
           res_sq(eta,R_data, param, control)./(sigma.^2) + log
           (2*pi*det(sigma.^2)) ...
44          , 2,'omitnan')+0.5*transpose(eta).*param(end,:)
               .^(-2).*eta ...
45          + 0.5*log(2*pi*det(param(end,:).^2));
46
47      %%%% Inner optimization
48      opt = optimoptions('fminunc','display','none');
49      for ii = 1:size(R_data,1)
50          eta_opt = @(eta) inner_likeli(eta, R_data(ii,:),
               param, control_input(ii));
51          try
52              %eta_opt(eta_guess)
53              eta_fp(:,ii) = fminunc(eta_opt, eta_guess, opt);
54          catch
55              eta_opt(eta_guess)
56              disp('Warning: error in FOCE')
57              eta_fp(:,ii) = eta_guess.*ones(length(
                   chosen_param), 1);
58          end
59      end
60
61      %%%% Compute sensitivities
62      param_vec = zeros(size(param));
63      param_vec(1,:) = 1; %KDE
64      new_param = param.*ones(size(param,1),size(R_data,1));
65      new_param(chosen_param,:) = new_param(chosen_param,:).*
```

```
        exp(eta_fp);
66      if sum(chosen_param == 2)==1
67          R_opt =  R*exp(eta_fp(chosen_param == 2,:)');
68      else
69          R_opt = R*ones(size(R_data,1),1);
70      end
71  [u0, ~, ~] = simulate_data(size(R_data,1), N_states, t,
        dt, ...
72              [zeros(size(R_data,1),1), R_opt, zeros(size(
                    R_data,1),1)], dudt, ...
73              [new_param(1:3,:); [0, 0]'.*ones(2,size(
                    R_data,1))],...
74              Ndosage, dosage, control_input, h_eq, false)
                    ;
75
76   if length(chosen_param) > 1
77       for ll = 1:length(chosen_param)
78           delta = zeros(size(eta_fp));
79           delta(ll,:) = 1e-6;
80           new_param_delta = param.*ones(size(param,1),
                    size(R_data,1));
81           new_param_delta(chosen_param,:) =
                    new_param_delta(chosen_param,:).*exp(eta_fp(
                    ll,:) + delta);
82          %Ratio = [R*ones(size(R_data,1),1), R*exp(
                    eta_fp(ll,:) + delta), R*ones(size(R_data,1)
                    ,1)];
83           if chosen_param(ii) == 2
84               R_opt =  R*exp(eta_fp(chosen_param == 2,:)'+
                        h(ii,:)');
85           elseif sum(chosen_param == 2)==1 &&
                    chosen_param(ii) ~= 2
86               R_opt =  R*exp(eta_fp(chosen_param == 2,:)
                        ');
87           else
88               R_opt = R*ones(size(R_data,1),1);
89           end
90           [u_delta, ~, ~] = simulate_data(size(R_data,1),
                    N_states, t, dt, [zeros(size(R_data,1),1),
                    R_opt, zeros(size(R_data,1),1)], dudt, ...
91               [new_param_delta(1:3,:); [0, 0]'.*ones(2,
                        size(R_data,1))], ...
92               Ndosage, dosage, control_input, h_eq, false
                        );
93          for ii = 1:size(R_data,1)
94              u_delta(ii, (event_time(ii)+1):end,2) = nan;
```

```
95                  u0(ii, (event_time(ii)+1):end, 2) = nan;
96              end
97              ddeta(:,:,ll) = dxdt(u0, u_delta, 1e-6,
                    new_param, param_vec(chosen_param(ll)), [
                    zeros(size(R_data,1),1), R_opt, zeros(size(
                    R_data,1),1)]);
98
99          end
100     else
101         delta = 1e-6;
102         new_param_delta = param.*ones(size(param,1),size(
                R_data,1));
103         new_param_delta(chosen_param,:) = new_param_delta(
                chosen_param,:).*exp(eta_fp + delta);
104         if sum(chosen_param == 2)==1
105                 R_opt = R*exp(eta_fp(chosen_param == 2,:)'+
                        delta');
106         else
107                 R_opt = R*ones(size(R_data,1),1);
108         end
109         [u_delta, ~, ~] = simulate_data(size(R_data,1),
                N_states, t, dt, [zeros(size(R_data,1),1), R_opt,
                 zeros(size(R_data,1),1)], dudt, ...
110              [new_param_delta(1:end-length(omega)+1,:); [0,
                     0]'.*ones(2,size(R_data,1))],...
111              Ndosage, dosage, control_input, h_eq, false);
112         for ii = 1:size(R_data,1)
113             u_delta(ii, (event_time(ii)+1):end,2) = nan;
114             u0(ii, (event_time(ii)+1):end, 2) = nan;
115         end
116         ddeta = dxdt(u0, u_delta, delta, new_param,
                param_vec(chosen_param), [zeros(size(R_data,1),1)
                ,R_opt, zeros(size(R_data,1),1)]);
117     end
118
119     %%%% Compute Hessian
120     if length(chosen_param) > 1
121         for ll = 1:size(R_data,1)
122             for jj = 1:length(chosen_param)
123                 for kk = 1:length(chosen_param)
124                     h(jj,kk) =-sum(ddeta(ll,:,jj).*ddeta(ll
                            ,:,kk).*inv(sigma.^2),2,'omitnan') -
                             omega_inv(jj,kk);
125                 end
126             end
127             %H(:,:,ll) = h;
```

```
128             H_det(ll) = det(-h/(2*pi));
129             [~, flag] = chol(-h);
                                                                    %
                   Check for a non-negative definite matrix
130             if flag ~= 0
131                 disp('Hessian not negative definite, opt out
                        ')
132                 %break
133             end
134         end
135     else
136         H = -sum(ddeta.^2.*inv(sigma^2),2,'omitnan') - param(
                end,:).^(-2);
137             if H > 0
138                 disp('Hessian not negative definite, opt out
                        ')
139             end
140         for ll = 1:size(R_data,1)
141             H_det(ll) = det(-H(ll)/(2*pi));
142         end
143     end
144     %%%% Compute loglikelihood
145     loglikelihood =  sum(0.5*sum(res_sq(eta_fp,R_data, param
            , control_input)./(sigma^2) + log(2*pi*det(sigma.^2))
            ...
146                 , 2,'omitnan')'+0.5*row_prod(eta_fp,param(end,:)
                        .^(-2)) ...
147             + 0.5*log(2*pi*det(param(end,:).^2)) ...
148             + 0.5*log(H_det), 2, 'omitnan');
149
150     %%%% Local functions
151     function residuals = Res_sq(param, eta, R_data, dudt, t,
            ...
152     N_states, dt, Ndosage, dosage, control, chosen_param, R,
            h_eq)
153     % Calculates the sum of square errors of the data
            compared to the
154     % simulated model
155         new_param = param.*ones(size(param,1),size(R_data,1)
                );
156         new_param(chosen_param,:) = new_param(chosen_param
                ,:).*exp(eta);
157         if sum(chosen_param == 2)==1
158                 R_opt =  R*exp(eta(chosen_param == 2,:))';
159         else
160                 R_opt = R*ones(size(R_data,1),1);
```

```
161             end
162             [u, ~, ~] = simulate_data(size(R_data, 1), N_states,
                    t, dt,...
163                 [zeros(size(R_data,1),1), R_opt, zeros(size(
                        R_data,1),1)], dudt, ...
164                 [new_param(1:3,:); [0, 0]'.*ones(2,size(R_data
                        ,1))], Ndosage, dosage, control, h_eq, false)
                        ;
165             residuals = (R_data-u(:,:, 2)).^2;
166         end
167
168         function ssum = row_prod(A,B)
169         % Computes rowsum of two matrices A and B
170             for ii = 1:size(A,2)
171                 ssum(ii) = transpose(A(:,ii))*B*A(:,ii);
172             end
173         end
174     end
```

B.3 Sequential hazard parameters

```
1  function loglikelihood = likelihood_haz(param, R, event_time
       , delta, h_eq)
2  % Calculates the negative log likelihood for the hazard for
       a given
3  % parametrization of the hazard, can be changed to logit if
       one pleases by
4  % altering the h-function.
5
6  % input: param - a px1 vector of the parameters belonging to
        the hazard
7  %        R - data containing the response from the KPD
       simulation (with deaths)
8  %        event_time - last recorded data point before event
9  %        delta - binary, 0 for no event happening to ind. i,
        1 if event
10 %        occured
11 %        h_eq - chosen equation for the hazard equation
12 % output: scalar negative log likelihood
13
14     loglikelihood = -(sum(delta.*log(h_eq(diag(R(:,
           event_time)), param)) +...
15             (1-delta).*log(1-h_eq(diag(R(:, event_time)),
                   param)) +...
16              prob_surv(R, param, h_eq, event_time)'));
```

```
17
18      function ind_sum = prob_surv(R, param, h_eq, event_time)
19          for jj = 1:size(R,1)
20              ind_sum(jj) = sum(log(1-h_eq(R(jj,1:event_time(
                    jj)-1), param)));
21          end
22      end
23
24  end
```

B.4 Joint parameter estimation with FOCE

```
1  function [loglikelihood, eta_fp] = JFOCE(param,eta_guess,
       dudt, dxdt,dhdt, R_data, t, ...
2       N_states, dt, Ndosage, dosage, control_input, R, h_eq,
            chosen_param, event_time, delta)
3  % Joint FOCE algorithm, computes the negative loglikelihood
       of
4  % a joint K-PD-TTE model.
5  %
6  % input: param : a px1 vector containing the parameters
       belonging to the
7  %                  K-PD + hazard model.
8  %          eta_guess : scalar, starting guess for eta.
9  %          dudt : ODE equations of the KPD system.
10 %          dxdt : equations expressing the residual
       derivatives.
11 %          dhdt : equations representing the hazard likelihood
        derivatives.
12 %          R_data : response variable data collected.
13 %          t : time vector representing when the data was
       sampled.
14 %          N_states : the number of elements present in the K-
       PD model.
15 %          dt : time step for simulating data.
16 %          Ndosage : time vector of when the drug was
       administrated.
17 %          dosage : scalar, represents how large the dose is (
       mg).
18 %          control_input : Nx1 vector of 0/1 values describing
        who
19 %          receives the drug.
20 %          U0 : initial values for simulating the model data
21 %          h_eq : chosen paramaterized hazard equation for the
22 %          simulated data.
```

```
23 %         chosen_param : 1xn_p vector, represents which
   parameters have
24 %         added random effects.
25 %         event_time : Nx1 vector of the index representing
   the interval
26 %         where each individual had an event occur, if
   event_time = t(end)
27 %         then event did not occur.
28 %         delta : Nx1 vector of binary values (0 - no event,
   1 - event)
29 %         indicating which individuals had en event occur
   during the
30 %         observed time.
31 %
32 % output: loglikelihood : the negative loglikelihood of the
33 %          joint K-PD-TTE model.
34 %          eta_fp : the fixed points of eta for a given set
   of parameters.
35     if length(chosen_param) > 1
36         omega = [param(end-3)^2, param(end-2); param(end-1),
               param(end)^2];
37     else
38         omega = param(end,:)^2;
39     end
40     sigma = param(end-length(chosen_param)*length(
       chosen_param));
41
42     res_sq = @(eta, R_data, param, control) Res_sq(param,eta
       , R_data, dudt, t, ...
43         N_states, dt, Ndosage, dosage, control, chosen_param
               , R, h_eq);
44     sim_data = @(param, eta, data, ind, control) simulate_R(
       param, eta, data,...
45         dudt, t, N_states, dt, Ndosage, dosage, control,
               chosen_param, R,...
46         h_eq, ind);
47
48     inner_likeli = @(eta, data, param, event_time, delta,
       control) ...
49         -(delta.*log(h_eq(diag(sim_data(param, eta, data,
               event_time, control)),...
50         param)) +(1-delta).*log(1-h_eq(diag(sim_data(param,
               eta, data,...
51         event_time, control)), param)) +survival_prod(
               sim_data(param, eta,...
52         data, 1:1:length(t), control), param, h_eq,
```

```
            event_time)) +...
53          0.5*sum(res_sq(eta,data, param, control)./(sigma^2)
               + log(2*pi*sigma^2) ...
54          , 2,'omitnan')+0.5*row_prod(eta,inv(omega)) + 0.5*
               log(det(2*pi*omega));
55
56      %%%%%%% Inner optimization
57      opt = optimoptions('fminunc','display','none', '
           OptimalityTolerance', 1e-6);
58      for ii = 1:size(R_data,1)
59          eta_opt = @(eta) inner_likeli(eta, R_data(ii,:),
               param, event_time(ii),...
60          delta(ii), control_input(ii));
61          try
62            eta_fp(:,ii) = fminunc(eta_opt, eta_guess, opt);
63          catch
64            eta_fp(:,ii) = eta_guess;
65            disp('Warning: error in JFOCE')
66          end
67      end
68
69       %%%%%%% Compute sensitivities
70       h = 1e-6;
71       param_vec = zeros(size(param));
72       param_vec(1,:) = 1; % Specific for KDE
73       new_param = param.*ones(size(param,1),size(R_data,1));
74       new_param(chosen_param,:) = new_param(chosen_param,:).*
           exp(eta_fp);
75       if sum(chosen_param == 2)==1
76                R_opt =  R*exp(eta_fp(chosen_param == 2,:)');
77       else
78                R_opt = R*ones(size(R_data,1),1);
79       end
80       [u0, ~, ~] = simulate_data(size(R_data,1), N_states, t,
           dt, ...
81                     [zeros(size(R_data,1),1), R_opt, zeros(size(
                          R_data,1),1)], ...
82                     dudt, new_param, Ndosage, dosage,
                          control_input, h_eq, false);
83        if length(chosen_param) > 1  % For several random
             effects
84            for ii = 1:length(chosen_param)
85                h = zeros(size(eta_fp));
86                h(ii,:) = 1e-6;
87                new_param_delta = param.*ones(size(param,1),
                     size(R_data,1));
```

```
88                  new_param_delta(chosen_param,:) =
                        new_param_delta(chosen_param,:).*exp(eta_fp
                        + h);
89
90                  if chosen_param(ii) == 2
91                      R_opt =  R*exp(eta_fp(chosen_param == 2,:)'+
                            h(ii,:)');
92                  elseif sum(chosen_param == 2)==1 &&
                        chosen_param(ii) ~= 2
93                       R_opt =  R*exp(eta_fp(chosen_param == 2,:)
                            ');
94                  else
95                      R_opt = R*ones(size(R_data,1),1);
96                  end
97
98                  [u_delta, ~, ~] = simulate_data(size(R_data,1),
                        N_states, t, dt, ...
99                              [zeros(size(R_data,1),1), R_opt,
                                zeros(size(R_data,1),1)],...
100                             dudt, new_param_delta, Ndosage,
                                dosage, control_input, h_eq,
                                false);
101
102                 for jj = 1:size(R_data,1)
103                     u_delta(jj, (event_time(jj)+1):end,2) = nan
                            ;
104                     u0(jj, (event_time(jj)+1):end, 2) = nan;
105                 end
106
107                 ddeta(:,:,ii) = dxdt(u0, u_delta, h(ii),
                        new_param, param_vec(chosen_param(ii)),...
108                      [zeros(size(R_data,1),1), R_opt, zeros(size
                            (R_data,1),1)]);
109                 ddlambda(:,:,ii) = dhdt(u0, u_delta, h(ii),
                        param, ddeta(:,:,ii));
110             end
111         else
112             new_param_delta = param.*ones(size(param,1),size(
                    R_data,1));  % For one random effect
113             new_param_delta(chosen_param,:) = new_param_delta(
                    chosen_param,:).*exp(eta_fp + h);
114             if sum(chosen_param == 2)==1
115                     R_opt =  R*exp(eta_fp(chosen_param == 2,:)'+
                            h');
116             else
117                     R_opt = R*ones(size(R_data,1),1);
```

```
118         end
119         [u_delta, ~, ~] = simulate_data(size(R_data,1),
                N_states, t, dt, ...
120             [zeros(size(R_data,1),1), R_opt, zeros(size(
                    R_data,1),1)], dudt, ...
121             new_param_delta, Ndosage, dosage, control_input,
                     h_eq, false);
122         for ii = 1:size(R_data,1)
123             u_delta(ii, (event_time(ii)+1):end,2) = nan;
124             u0(ii, (event_time(ii)+1):end, 2) = nan;
125         end
126         ddeta = dxdt(u0, u_delta, h, new_param, param_vec(
                chosen_param),...
127             [zeros(size(R_data,1),1), R_opt, zeros(size(
                    R_data,1),1)]);
128         ddlambda = dhdt(u0, u_delta, h, param, ddeta);
129     end
130
131     %%%%%% Compute Hessian
132     omega_inv = inv(omega);
133     if length(chosen_param) > 1
134         count = 0;
135         test_param = new_param;
136         for ll = 1:size(R_data,1)
137             count = count + 1;
138             for jj = 1:length(chosen_param)
139                 for kk = 1:length(chosen_param)
140                     hess(jj,kk) = -ddlambda(ll,event_time(
                            ll),jj).*ddlambda(ll,event_time(ll),
                            kk).*((1-delta(ll))./(1-h_eq(R_data(
                            ll,event_time(ll)), test_param(:,ll)
                            )).^2 + delta(ll)./(h_eq(R_data(ll,
                            event_time(ll)), test_param(:,ll))
                            .^2)) ...
141                         -dlambda_sum(R_data(ll,:), param,
                                h_eq, event_time(ll), ddlambda(
                                ll,:,kk), ddlambda(ll,:,jj)) '...
142                         -sum(ddeta(ll,:,jj).*ddeta(ll,:,kk)
                                .*inv(sigma^2),2,'omitnan') -
                                omega_inv(jj,kk);
143                 end
144             end
145             H_det(ll) = det(-hess/(2*pi));
146             [~, flag] = chol(-hess);
                                                        %
                Check for a non-negative definite matrix
```

```
147             if flag ~= 0
148                 disp('Hessian not negative definite, opt out
                        ')
149             end
150         end
151     else
152         H = -diag(ddlambda(:,event_time)).^2.*((1-delta)
                ./(1-h_eq(diag(R_data(:,event_time)), param)).^2
                +...
153             delta./(h_eq(diag(R_data(:,event_time)), param)
                    .^2)) -...
154             dlambda_sum(R_data, param, h_eq, event_time,
                    ddlambda, ddlambda)'-...
155             sum(ddeta.^2*inv(sigma^2),2,'omitnan') -
                    omega_inv;
156
157         for ii = 1:size(R_data,1)
158             H_det(ii) = det(-H(ii)/(2*pi));
159         end
160     end
161
162     %%%%%% Compute loglikelihood
163     loglikelihood =  sum(0.5*sum(res_sq(eta_fp,R_data, param
            , control_input)./(sigma^2) + log(2*pi*sigma^2) ...
164             , 2,'omitnan')'- transpose(delta.*log(h_eq(diag(
                    sim_data(param, eta_fp, R_data, event_time,
                    control_input)), param))) -...
165             transpose((1-delta).*log(1-h_eq(diag(sim_data(
                    param, eta_fp, R_data, event_time,
                    control_input)), param))) -...
166             survival_prod(sim_data(param, eta_fp, R_data,
                    1:1:length(t), control_input), param, h_eq,
                    event_time) +0.5*row_prod(eta_fp,omega_inv)
                    +...
167             0.5*log(det(2*pi*omega)) + 0.5*log(H_det), 2, '
                    omitnan');
168
169     %%%%%% Functions used
170     function sim_data = simulate_R(param, eta, R_data, dudt,
            t, ...
171     N_states, dt, Ndosage, dosage, control, chosen_param, R,
            h_eq, ind)
172     % Calculates the sum of square errors of the data
            compared to the
173     % simulated model
174         new_param = param.*ones(size(param,1),size(R_data,1)
```

```
175             );
                new_param(chosen_param,:) = new_param(chosen_param
                    ,:).*exp(eta);
176             if sum(chosen_param == 2)==1
177                     R_opt =  R*exp(eta(chosen_param == 2,:))';
178             else
179                     R_opt = R*ones(size(R_data,1),1);
180             end
181             [u, ~, ~] = simulate_data(size(R_data, 1), N_states,
                    t, dt, [zeros(size(R_data,1),1), R_opt, zeros(
                    size(R_data,1),1)], dudt, ...
182                     new_param, Ndosage, dosage, control, h_eq,
                        false);
183             sim_data = u(:,ind,2);
184     end
185
186     function residuals = Res_sq(param, eta, R_data, dudt, t,
            ...
187     N_states, dt, Ndosage, dosage, control, chosen_param, R,
            h_eq)
188     % Calculates the sum of square errors of the data
            compared to the
189     % simulated model
190             new_param = param.*ones(size(param,1),size(R_data,1)
                    );
191             new_param(chosen_param,:) = new_param(chosen_param
                    ,:).*exp(eta);
192             if sum(chosen_param == 2)==1
193                     R_opt =  R*exp(eta(chosen_param == 2,:))';
194             else
195                     R_opt = R*ones(size(R_data,1),1);
196             end
197             [u, ~, ~] = simulate_data(size(R_data, 1), N_states,
                    t, dt, [zeros(size(R_data,1),1), R_opt, zeros(
                    size(R_data,1),1)], dudt, ...
198                     new_param, Ndosage, dosage, control, h_eq,
                        false);
199             residuals = (R_data-u(:,:, 2)).^2;
200     end
201
202     function rowprod = row_prod(A,B)
203     % Computes rowsum of two matrices A and B
204             for ii = 1:size(A,2)
205                     rowprod(ii) = transpose(A(:,ii))*B*A(:,ii);
206             end
207     end
```

```
208
209
210         function ind_sum = survival_prod(data, param, h_eq,
                event_time)
211         % Calculates the probability of survival until time
                tau_i-1
212             for jj = 1:size(data,1)
213                 ind_sum(jj) = sum(log(1-h_eq(data(jj,1:
                        event_time(jj)-1), param)));
214             end
215         end
216
217     function to_sum = dlambda_sum(data, param, h_eq,
            event_time, dlambda1, dlambda2)
218     % Calculates the sum in the likelihood for TTE
219             for ii = 1:size(data,1)
220                 to_sum(ii) = sum(dlambda1(ii,1:event_time(ii)-1)
                        .*...
221                                 dlambda2(ii,1:event_time(ii)-1).*...
222                                 1./((1-h_eq(data(ii,1:event_time(ii)-1)
                                        , param)).^2));
223             end
224     end
225 end
```

C

Possible issues regarding identifiability

To further investigate the possibility of existing problems with the identifiability in the parametric hazard, a set of different values α and β were studied more closely. In total 40 different values of β in the range of $[30, 60]$, with a matched α which kept the baseline hazard within a value of $|\alpha + \beta| < 3$, was investigated. See Figure C.1 for the range of chosen parameter values. One dataset with 500 individuals who where sampled once a day was simulated for each parameter combination.

Since the survival curves contain most of the data necessary when estimating the parameters, with regards to the TTE likelihood, they are a good visual tool to view the dataset. Therefore the different survival curves were plotted for a set random seed, in order for the curves to be comparable. The result of this is presented in Figure C.2, where each color represent one value of the baseline hazard with a spacing of 0.5. This means that the curves at the top, and the bottom, represent the extreme baseline values of +3 and -3. Note the clustering that occurs for every baseline value. Each cluster contains at least one value of β in the full range of $[30, 60]$, which still generate the same results in regards of how many individuals have an event occur for each time point. These clusters also tend to become more and more compact with more extreme baselines.

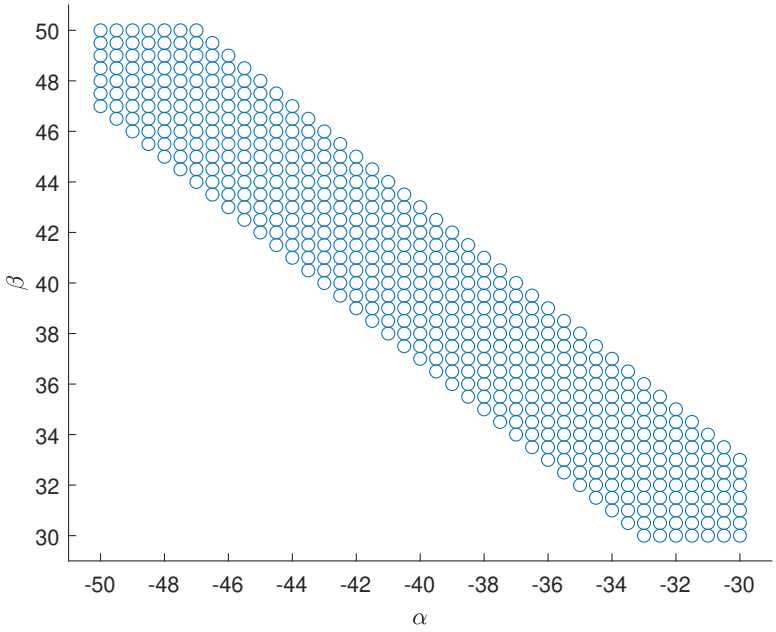


Figure C.1: Discretization of the different values of α and β which keeps the baseline hazard in the range of $[0.05, 0.99]$.

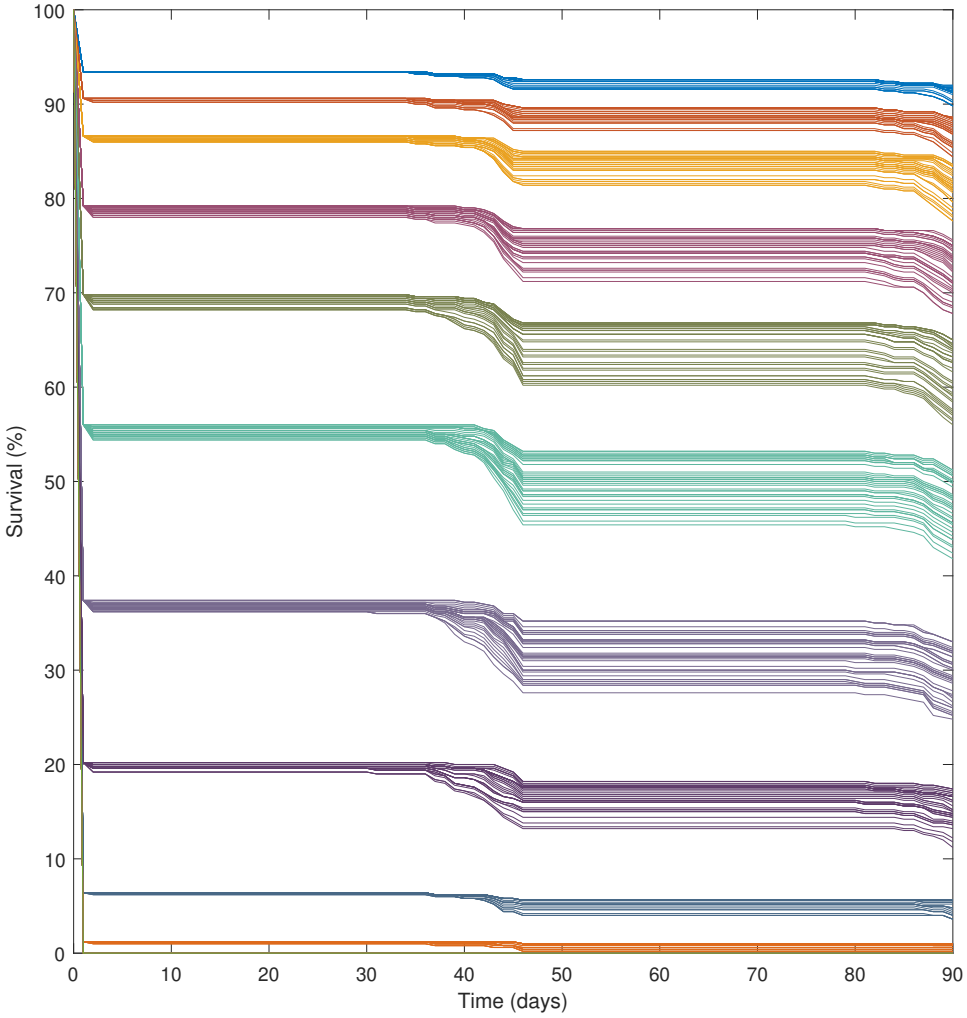


Figure C.2: Simulated survival curves for the parameter combinations seen in Figure C.1 for 500 individuals who where sampled once a day for 90 days. Each color represents a different baseline hazard such that $|\alpha + \beta|$ was kept below 3.

D

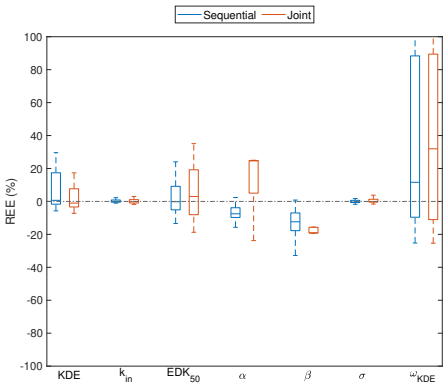
Sampling frequencies - complementary results

To test out start guesses further away, the same analysis carried out as in Section 4.2 was carried out but for 18 repeats and 100 individuals where everyone received the same dosage of 50 mg twice over 90 days. The simulated parameter values as well as the start guess is displayed in Table D.1.

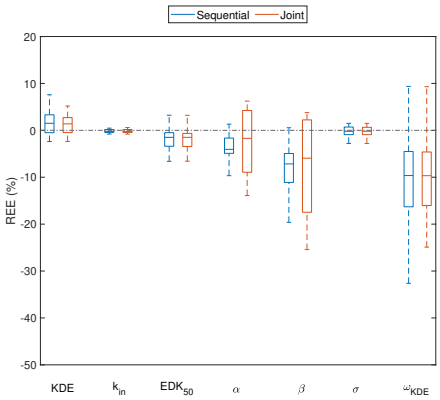
‘ The different sampling frequencies were: $\Delta t = 2$, $\Delta t = 1$ and $\Delta t = 0.5$, and the results are displayed as the REE values for each parameter in Figure D.1 and as the total REE in Figure D.2. For $\Delta t = 2$, joint performs better in 3 of the datasets, for $\Delta t = 1$ in 8 of the datasets and for $\Delta t = 0.5$ in 16 of the datasets.

Table D.1: Parameter values used for simulating data.

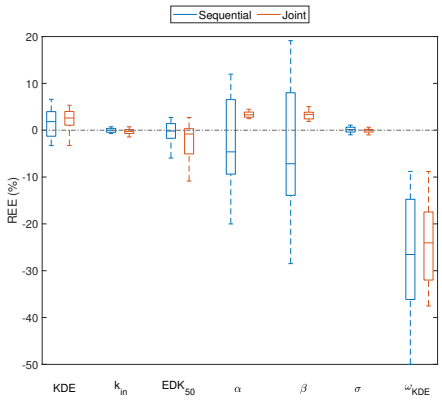
Parameter	Simulated	Start guess
$KDE[\mathrm{d}^{-1}]$	0.2	0.1
$k_{in}[\mathrm{nM.d}^{-1}]$	0.1	0.2
$EDK_{50}[\mathrm{mg.d}^{-1}]$	0.1	0.2
α	-8.0	-10
β	6.2	5
$\sigma[\mathrm{nM}]$	0.05	0.2
ω_{KDE}	0.2	0.1



(a) $\Delta t = 2$

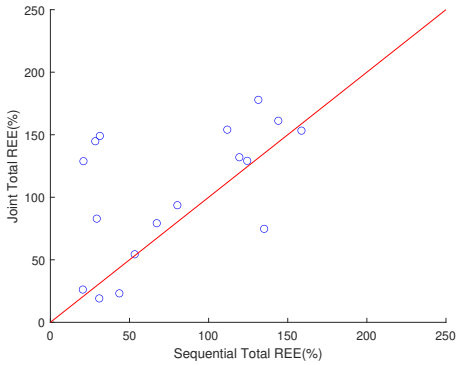


(b) $\Delta t = 1$

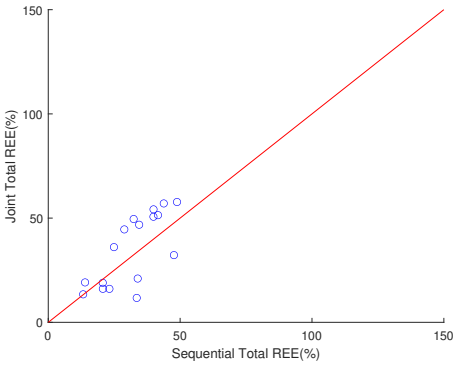


(c) $\Delta t = 0.5$

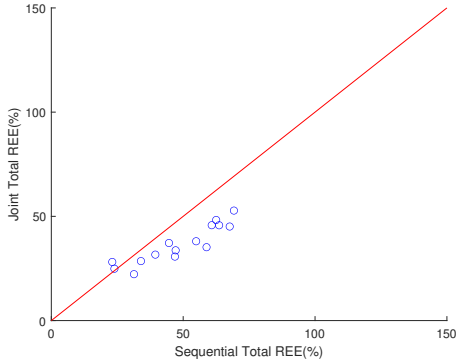
Figure D.1: REE for the two estimation approaches, joint (orange) and sequential (blue), for three different sampling frequencies for 18 iterations. The repetition datasets were simulated for 100 individuals where everyone received the same treatment of two doses of 50 mg.



(a) $\Delta t = 2$



(b) $\Delta t = 1$



(c) $\Delta t = 0.5$

Figure D.2: Total REE for the three different sampling frequencies for 18 iterations. The red line indicates the case where both of the sums, for the joint and sequential model, are equal.

DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY