CHALMERS
UNIVERSITY OF TECHNOLOGY

Sensitivity-based Gradients for Parameter Estimation in Nonlinear Mixed Effects Models with Deterministic and Stochastic Dynamics

Master's Thesis in Complex Adaptive Systems

HELGA KRISTÍN ÓLAFSDÓTTIR

Department of Signals and Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2016

Master's thesis 2016:EX072/2016

Sensitivity-based Gradients for Parameter Estimation in Nonlinear Mixed Effects Models with Deterministic and Stochastic Dynamics

HELGA KRISTÍN ÓLAFSDÓTTIR

Department of Signals and Systems
Chalmers University of Technology
Gothenburg, Sweden 2016

Sensitivity-based Gradients for Parameter
Estimation in Nonlinear Mixed Effects Models
with Deterministic and Stochastic Dynamics
HELGA KRISTÍN ÓLAFSDÓTTIR

Supervisor: Joachim Almquist, Fraunhofer-Chalmers Research Centre
Examiner: Jonas Sjöberg, Signals and Systems

Master's Thesis 2016:EX072/2016
Department of Signals and Systems
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2016

Sensitivity-based Gradients for Parameter
Estimation in Nonlinear Mixed Effects Models
with Deterministic and Stochastic Dynamics
HELGA KRISTÍN ÓLAFSDÓTTIR
Department of Signals and Systems
Chalmers University of Technology

Abstract

Nonlinear Mixed Effects (NLME) modelling has for a long time been used for modelling individuals of a population that behave by the same qualitative mechanisms but with quantitative differences between individuals. This kind of modelling increases for example knowledge on the effects of drugs on the body, which is important in drug development when deciding dosing regiments.

An important part of NLME modelling involves estimating the parameters of the model for a given dataset. This is often done using gradient based methods, where the gradients are traditionally approximated using finite differences. This approximation might cause longer execution times and numerical problems leading to failure at estimating parameters.

This thesis investigates the robustness of parameter estimation using the method S-FOCE, where gradients of the optimisation algorithm are computed exactly instead of using numerical approximations. This is done for NLME models with deterministic dynamics. The comparison was performed by estimating parameters of simulated data from pharmacokinetic and pharmacodynamic models using both a parameter estimation program that uses the exact gradients and an industry standard parameter estimation software, NONMEM, that partly uses finite difference approximations to compute the gradients. This thesis also shows how the S-FOCE method could be extended for a general NLME model with stochastic dynamics. This involves deriving the first and second order sensitivities of the Extended Kalman Filter.

The results show that for a simple model with no failure in parameter estimation using finite differences, the S-FOCE method performs equally well. However, models where the finite difference method had lower success frequency, the S-FOCE method suggests significant improvement in robustness in terms of the frequency of successful estimates of parameters and their uncertainties as. The quality of the uncertainty estimates was similar for all methods.

Keywords: Nonlinear Mixed Effects, sensitivity equations, pharmacokinetics, pharmacodynamics, NONMEM, parameter estimation, Population Likelihood, First Order Conditional Estimation

Acknowledgements

I would like to thank my supervisor, Joachim Almquist at Fraunhofer-Chalmers Research Centre (FCC) for his great support, guidance and discussions on the topic and my examiner Dr. Jonas Sjöberg, Professor at Signals and Systems at Chalmers, for his constructive comments and advice. My gratitude also extends to Mats Jirstrand at FCC for his technical expertise. Finally, I want to thank FCC for the opportunity to work on this thesis and their provided facility.

Helga Kristín Ólafsdóttir, Gothenburg, June 2016

Contents

List of Figures

List of Tables

1 Introduction

This chapter introduces the background and motivation for this thesis, declares the main contributions and gives an overview of the thesis.

1.1 Background and Motivation

Many processes occurring within individuals of a population operate by the same qualitative mechanisms in every individual but with quantitative aspects differing from one individual to another. These processes could for example be how drug gets absorbed and later eliminated from the body or how an individual responds to insulin when treated for diabetes. Nonlinear mixed effects (NLME) modelling [13] is a powerful tool for quantifying the existence of such population variability. NLME models describe individual observations that depend on a continuously changing state of each individual. The dynamics of the state are described using differential equations.

NLME modelling has for a long time played a dominant role in fields like clinical pharmacology when modelling processes called population pharmacokinetics (PK) and pharmacodynamics (PD), which describe what happens to drugs in the body and what response the drug triggers in the body, respectively. There, understanding what happens from dose to response gives important information when deciding suitable dose and dosing regiments. Although widely used within pharmacology, the method is also applicable to other fields as well and novel applications are still being found, as illustrated by its recent use in single-cell systems biology studies [1, 10].

In order to model the behaviour of the population, the non-varying parameters describing the population as well as the parameters describing individual variability have to be estimated. Parameter estimation in NLME models is a challenging task, often suffering from both long run-times and numerical instability, and it is frequently the bottleneck of the modelling work-flow. Using gradient-based methods requires computing derivatives of a function that is not explicitly defined. The traditional way is to compute the derivatives using finite difference approximations. The use of gradient-based optimisation with exact gradients computed using so called sensitivity-equations instead of computing the gradients using the finite difference approach has shown great potential for addressing the above issues [2]. The exact derivative of a function that is the solution of a differential equation is then obtained by creating a new differential equation that has the derivative as a solution.

For random variables that depend on a set of parameters, the likelihood is the probability of a set of observations of the random variables, given certain parameter values. This can be used as a measure of how well certain parameters fit an observed set of data. One approach to parameter estimation in NLME models is to maximise the population likelihood. This likelihood does not have a closed form solution. In order to estimate the parameters, approximations have to be made and the problem solved numerically. During this study, the likelihood was approximated using First Order Conditional Estimation (FOCE). In [2] it was shown that for gradient based optimisation of the FOCE approximation, the gradient precision and accuracy could be improved, along with computational time, by using sensitivity equations to compute the gradients. This parameter estimation method will be referred to as S-FOCE. The robustness of the use of S-FOCE for parameter estimation is however still to be analysed. This will be done by comparing the robustness of S-FOCE with the industry standard software NONMEM [4]. Figure 1.1 displays the two different types of parameter estimation methods that will be compared.

The robustness of parameter estimation is measured in terms of success and failure of obtaining both a parameter estimate and its uncertainties. The parameter estimation can fail during two main steps. The first failure occurs during the estimation

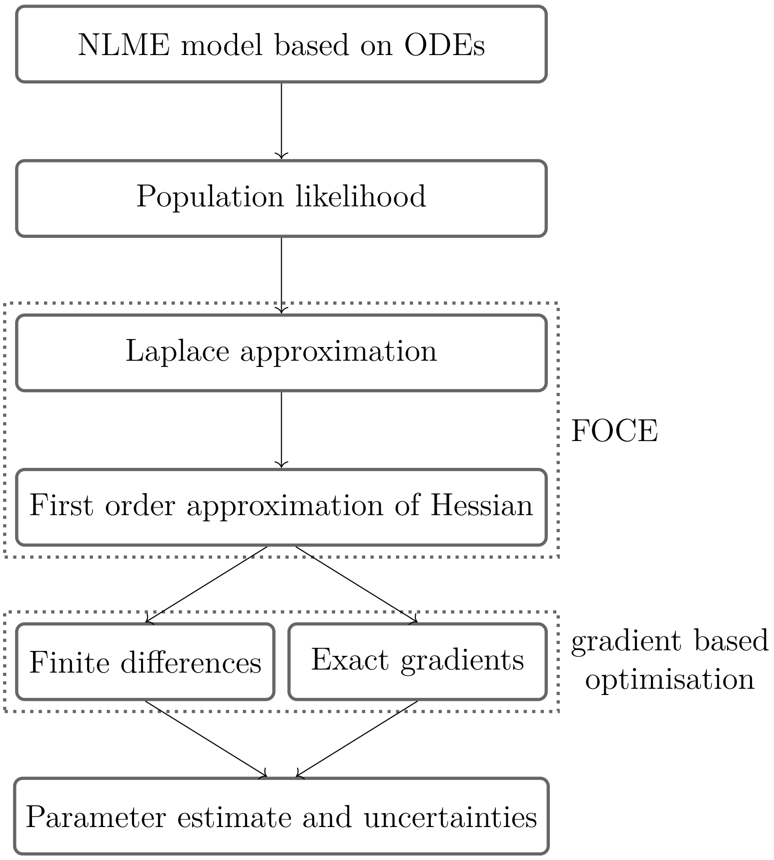


Figure 1.1: *Schematic figure of two variances of FOCE for parameter estimation of NLME models based on ODEs. The traditional finite difference version (left) will be compared to the S-FOCE version, introduced in [2] (right).*

of parameters if the optimisation algorithm does not manage to fulfil the set stopping criteria. The second failure occurs if the method does not manage to obtain an uncertainty estimate of the estimated parameters. The uncertainty estimate is described using a covariance matrix. Both failures contribute to an unsuccessful parameter estimation. This is shown in Figure 1.2.

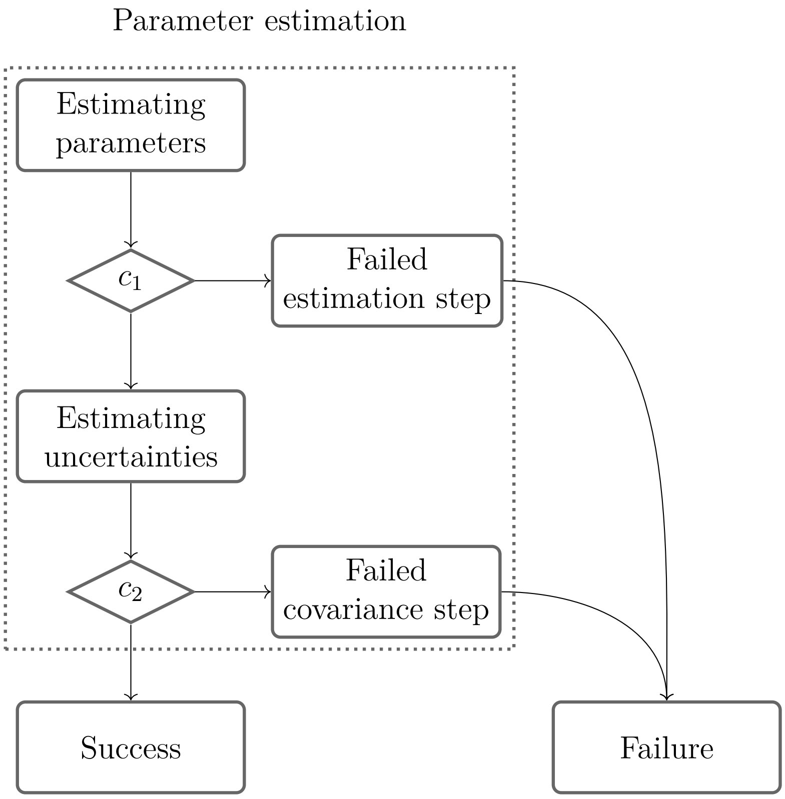


Figure 1.2: *Schematic figure of types of failure during parameter estimation. The conditions c_1 and c_2 ensure that point estimate and uncertainties of the point estimate have been found. Both need to be fulfilled for the parameter estimation to be successful.*

The model for the dynamics of the individuals is often formulated in terms of ordinary differential equations (ODEs). This makes the state variables of the model deterministic and all noise observed within an individual is modelled as measurement noise. However, a more realistic approach with dynamical systems described by stochastic differential equations (SDEs) has begun to appear within NLME modelling [12, 17]. The use of SDEs allows distinction between measurement noise and system noise. While making the models more realistic, this also helps with modelling the model structure incompleteness. However, these models add another dimension of complexity, making their analysis even more challenging.

For NLME models based on SDEs, the S-FOCE method will be extended to account for a general NLME model with stochastic dynamics when estimating parameters. In [12], the authors suggest the method of using symbolic differentiation for obtaining gradients when optimising instead of using the approximation based on finite differences. This was however done by using the symbolic algebra capability in Mathematica and the computations had to be done specifically for each model in question. In the present study, in order to verify how this method works for a general mixed effects model, the population likelihood gradient is derived analytically using sensitivity equation based gradients.

1.2 Contributions and Thesis Overview

The main contributions of this thesis are the implementation of S-FOCE and comparison with the FOCE method of NONMEM, along with the extension of the S-FOCE algorithm on general NLME models with stochastic dynamics.

The construction of the thesis is as follows. After the introduction, the parameter estimation procedure and related methods for NLME models based on ordinary differential equations are introduced. This is followed by a description of parameter estimation for NLME models based on stochastic differential equations. The discussion ends with an extension of the S-FOCE algorithm on general NLME models with stochastic dynamics, the first main contribution of the thesis. Thereafter, the main ideas behind pharmacokinetic and pharmacodynamic modelling are briefly described and the statistical tests used for robustness comparison explained. Chapter 3 introduces the software used for parameter estimation and describes how the S-FOCE method was implemented for this study and how the programs were set up for comparison. After that, Chapter 4 describes the models and simulated datasets chosen for the comparison and discusses the approach taken for comparing the different methods of parameter estimation. The results from the comparison and discussion on how the exact population likelihood gradient for a general NLME model can be derived using exact gradients are presented in Chapter 5. Finally, the main discussion and conclusions are presented in Chapter 6.

2
Theory

This chapter contains a summary of the theory necessary for this thesis. It begins with an introduction of Nonlinear Mixed Effects models based on ordinary differential equations and the parameter estimation methods studied in this project. Thereafter, an extension of the Nonlinear Mixed Effects model is introduced for which the dynamics are described using stochastic differential equations. The chapter ends with a short description of pharmacokinetic and pharmacodynamic modelling, which is necessary in order to understand the models used in this study, along with an explanation of the statistical tests used for analysing the results.

2.1 Parameter Estimation of Nonlinear Mixed Effects Models with Deterministic Dynamics

Consider a population where measurements have been collected from each individual at a number of different time points. The measurements could for example be the concentration of a drug or fatty acids in the plasma. These individuals have both inter-individual and intra-individual variability, meaning that the behaviour of each individual is slightly different, and also that the measurements are noisy. This kind of population can be modelled using a Nonlinear Mixed Effect (NLME) model [13]. For a given NLME model and dataset, the model is fitted to the data by finding the parameters that maximise the population likelihood. When the parameters have been estimated, the behaviour of the population can then be simulated and studied further.

This section describes the formulation of an NLME model based on ordinary differential equations, the population likelihood and methods to optimise the population likelihood. A way of computing the exact likelihood gradient, by solving a system of differential equations instead of using finite difference methods, is also introduced.

2.1.1 Nonlinear Mixed Effects Model

An NLME model can be defined by [13, 2]

$$\boldsymbol{y}_{ij} = \boldsymbol{h}(\boldsymbol{u}_i, \boldsymbol{x}_{ij}, t, \boldsymbol{\theta}, \boldsymbol{\eta}_i) + \boldsymbol{e}_{ij} \tag{2.1}$$

where $\boldsymbol{y}_{ij}$,$\boldsymbol{x}_{ij}$= $\boldsymbol{x}_i(t_{j_i})$ are the j-th response and predictor vector of the i-th individual, $\boldsymbol{u}_i$ is the input to the system, $\boldsymbol{e}_{ij}$ is a normally distributed noise term around zero with covariance $\boldsymbol{R}(\boldsymbol{u}_i, \boldsymbol{x}_{ij}, t, \boldsymbol{\theta}, \boldsymbol{\eta}_i)$ and $\boldsymbol{\theta}$ and $\boldsymbol{\eta}_i$ are parameter vectors. The parameter vector $\boldsymbol{\theta}$ consists of fixed population parameters and $\boldsymbol{\eta}_i$ of random effects parameters associated with individual i. The individual random parameter vector $\boldsymbol{\eta}_i$ is assumed to be normally distributed around zero with variance $\boldsymbol{\Omega}$ on the population level. It is assumed that $\boldsymbol{\Omega}$ is unknown and its elements are therefore included in the fixed elements $\boldsymbol{\theta}$. The dynamics of the predictor vector $\boldsymbol{x}_i$ are usually described with an ordinary differential equation

$$\begin{aligned} \frac{d\boldsymbol{x}_i}{dt} &= \boldsymbol{f}(\boldsymbol{u}_i, \boldsymbol{x}_i, t, , \boldsymbol{\theta}, \boldsymbol{\eta}_i) \\ \boldsymbol{x}_i(t_0) &= \boldsymbol{x}_{0i}(\boldsymbol{\theta}, \boldsymbol{\eta}_i). \end{aligned} \tag{2.2}$$

Two kinds of variability affect these kinds of NLME models. This is the inter-individual variability $\boldsymbol{\Omega}$ and the intra-individual variability $\boldsymbol{R}$.

Simple Example of a Mixed Effects Model

Below is a simple mixed effects model that illustrates the role of the fixed population parameters, individual random parameters, and measurement variability. Consider the simple model where the state variable x_i of individual i at time t is described with the differential equation

$$\begin{aligned} \frac{dx_i}{dt} &= c \\ x_i(t_0) &= \eta_i \end{aligned} \tag{2.3}$$

where c is a constant, and the measured output y_i of individual i at time t_{j_i} is

$$y_i(t_{j_i}) = x_i(t_{j_i}) + e_{ij} \tag{2.4}$$

where $\eta_i \sim N(0, \omega^2)$ and $e_{ij} \sim N(0, s^2)$. This model has the fixed population parameters $\boldsymbol{\theta} = (c, \omega, s)$ to be estimated and one random individual effect η.

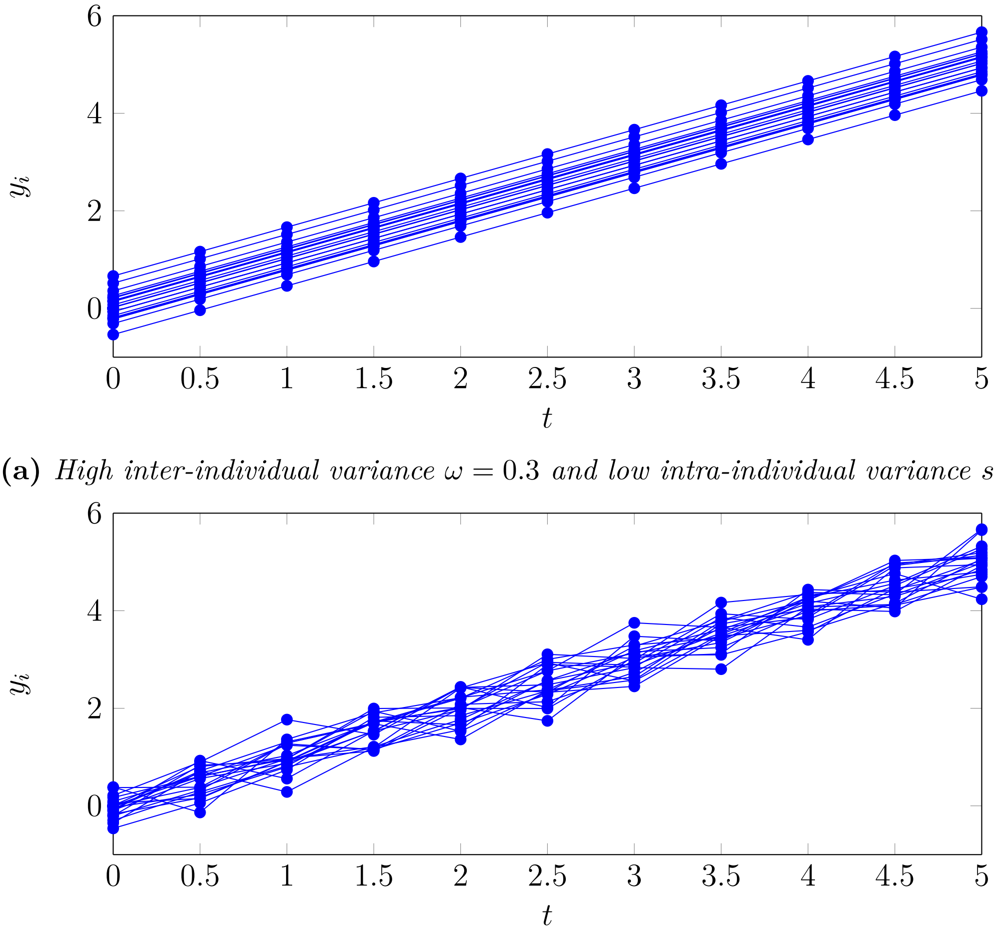
The above differential equation (Eq. 2.3) has solution

$$x_i(t) = c \cdot t + \eta_i. \tag{2.5}$$

Hence, the individual effect η_i determines the translation of the line

$$y = c \cdot t \tag{2.6}$$

in the y-direction and so does the measurement error e_{ij}. The difference between the influence of random individual effects and the measurement error can be seen in Figure 2.1, where the variability of the population is either dominated by inter-individual variability (Figure 2.1a) or intra-individual variability (Figure 2.1b).



(a) *High inter-individual variance* $\omega = 0.3$ *and low intra-individual variance* $s = 0.001$

(b) *Low inter-individual variance* $\omega = 0.001$ *and high intra-individual variance* $s = 0.3$

Figure 2.1: *Difference between inter- and intra-individual variability for computer generated time series data for a linear model with 20 individuals, one population parameter* $c = 1$*, and measurements taken at eleven equispaced time points from 0 to 5.*

2.1.2 Population Likelihood

Let $\boldsymbol{d}_{ij}$ be a set of observations for individuals $i = 1, ..., N$ at times t_{j_i}, $j_i = 1, ..., n_i$, and define the residuals $\boldsymbol{\epsilon}_{ij}$ as the difference between the measurement and expected value according to the model, i.e.

$$\boldsymbol{\epsilon}_{ij} = \boldsymbol{d}_{ij} - \hat{\boldsymbol{y}}_{ij} \tag{2.7}$$

where

$$\begin{aligned} \hat{\boldsymbol{y}}_{ij} &= E[\boldsymbol{y}_{ij}] \\ \boldsymbol{R}_{ij} &= Var[\boldsymbol{y}_{ij}]. \end{aligned} \tag{2.8}$$

The population likelihood can be written as the multiplication of the individual likelihoods yielding, as in [2]

$$\begin{aligned}
L(\boldsymbol{\theta}|\boldsymbol{d}) &= p(\boldsymbol{d}|\boldsymbol{\theta}) \\
&= \prod_{i=1}^{N} p(\boldsymbol{d}_i|\boldsymbol{\theta}) \\
&= \prod_{i=1}^{N} \int p(\boldsymbol{d}_i|\boldsymbol{\theta}, \boldsymbol{\eta}_i) p(\boldsymbol{\eta}_i|\boldsymbol{\theta}) d\boldsymbol{\eta}_i \\
&= \prod_{i=1}^{N} \int \left(\prod_{j=1}^{n_i} \frac{\exp\left(-\frac{1}{2}\boldsymbol{\epsilon}_{ij}^T \boldsymbol{R}_{ij}^{-1} \boldsymbol{\epsilon}_{ij}\right)}{\sqrt{|2\pi \boldsymbol{R}_{ij}|}} \right) \cdot \frac{\exp\left(-\frac{1}{2}\boldsymbol{\eta}_i^T \boldsymbol{\Omega}^{-1} \boldsymbol{\eta}_i\right)}{\sqrt{|2\pi\boldsymbol{\Omega}|}} d\boldsymbol{\eta}_i
\end{aligned} \tag{2.9}$$

where $\boldsymbol{d}$ denotes all observations and $\boldsymbol{d}_i$ denotes all observations of individual i. This can be further simplified by writing [2]

$$L(\boldsymbol{\theta}|\boldsymbol{d}) = \prod_{i=1}^{N} \int \exp(\boldsymbol{l}_i) d\boldsymbol{\eta}_i \tag{2.10}$$

where

$$\boldsymbol{l}_i = -\frac{1}{2} \sum_{j=1}^{n_i} \left(\boldsymbol{\epsilon}_{ij}^T \boldsymbol{R}_{ij}^{-1} \boldsymbol{\epsilon}_{ij} + \log\det(2\pi \boldsymbol{R}_{ij}) \right) - \frac{1}{2} \boldsymbol{\eta}_i \boldsymbol{\Omega}^{-1} \boldsymbol{\eta}_i - \frac{1}{2} \log\det(2\pi\boldsymbol{\Omega}). \tag{2.11}$$

Hereafter, the notation $L(\boldsymbol{\theta})$ will be used to denote the population likelihood instead of $L(\boldsymbol{\theta}|\boldsymbol{d})$.

2.1.3 Sensitivity Equation-based Gradients

This subsection describes how to compute exact gradients using sensitivity equations. Consider the simplified case of state variable x, dependent on t and $\boldsymbol{p}$, defined by a differential equation of t. Here, $\boldsymbol{p}$ is a parameter vector. Using the differential equation, the derivatives of x with regard to $\boldsymbol{p}_i$ can be found by obtaining a differential equation for the derivatives. In order to show this, the differential equation for $s_i(t, \boldsymbol{p}) := \frac{d}{d\boldsymbol{p}_i} x$ will be computed. The derivatives s_i are called (first order) sensitivities [7, 2] and hence the differential equations are called sensitivity equations.

Consider the differential equation

$$\begin{aligned}
\frac{dx(t;\boldsymbol{p})}{dt} &= f(t, x(t;\boldsymbol{p}), \boldsymbol{p}) \\
x(t_0) &= x_0(\boldsymbol{p})
\end{aligned} \tag{2.12}$$

where $\boldsymbol{p}$ is a vector of parameters of length m. The derivative of x with respect to $\boldsymbol{p}_i$ cannot be computed directly since the formulation of x as a function of $\boldsymbol{p}$ is not given. However, it holds that

$$\frac{d}{dt} s_i = \frac{d}{dt} \left(\frac{dx}{d\boldsymbol{p}_i} \right) = \frac{d}{d\boldsymbol{p}_i} \left(\frac{dx}{dt} \right) = \frac{\partial f}{\partial \boldsymbol{p}_i} + \frac{\partial f}{\partial x} \frac{dx}{d\boldsymbol{p}_i} = \frac{\partial f}{\partial \boldsymbol{p}_i} + \frac{\partial f}{\partial x} s_i \tag{2.13}$$

and thus the sensitivity equations can be obtained

$$\begin{aligned} \frac{d}{dt} s_i(t, \boldsymbol{p}) &= \frac{\partial f}{\partial \boldsymbol{p}_i} + \frac{\partial f}{\partial x} s_i \\ s_i(t_0, \boldsymbol{p}) &= \frac{dx_0(\boldsymbol{p})}{d\boldsymbol{p}_i}. \end{aligned} \tag{2.14}$$

Solving Eq. (2.14) gives us $s_i(t, \boldsymbol{p}) = \dfrac{dx}{d\boldsymbol{p}_i}$ without having to approximate any derivatives. The first order sensitivity is simply defined as the first order derivative of x with respect to p_i,

$$s_i(t, \boldsymbol{p}) = \frac{d}{d\boldsymbol{p}_i} x(t, \boldsymbol{p}), \qquad i = 1, ..., m \tag{2.15}$$

In the same way, the second order sensitivity can be defined as the second order derivative etc.

2.1.4 Parameter Estimation using First Order Conditional Estimation (FOCE)

The type of parameter estimation considered in this project depends on finding the parameters $\boldsymbol{\theta}^*$ that maximises the population likelihood $L(\boldsymbol{\theta})$ in Eq. (2.10), i.e. finding $\boldsymbol{\theta}^*$ such that

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} L(\boldsymbol{\theta}). \tag{2.16}$$

The likelihood function does not have a closed form solution [2]. One way of solving this problem is to use the First Order Conditional Estimation (FOCE) method to approximate the likelihood function. This is done by first approximating the population likelihood function (Eq. (2.10)) using Laplace's method [24]. The individual likelihood function l_i is approximated with a second-order Taylor expansion around the value $\boldsymbol{\eta}_i^*$ that maximises l_i given $\boldsymbol{\theta}$ and yields the following approximation of the population likelihood $L(\boldsymbol{\theta})$

$$L(\boldsymbol{\theta}) \approx \prod_{i=1}^{N} \left(\exp(\boldsymbol{l}_i(\boldsymbol{\eta}_i^*)) \det \left[\frac{-\nabla^2 \boldsymbol{l}_i(\boldsymbol{\eta}_i^*)}{2\pi} \right]^{-1/2} \right) \tag{2.17}$$

where $\nabla^2 \boldsymbol{l}_i(\boldsymbol{\eta}_i^*)$ denotes the Hessian of $\boldsymbol{l}_i$ at $\boldsymbol{\eta}_i^*$ [2]. After using Laplace's method, the likelihood function is further approximated by approximating the Hessian using a first order approximation. The approximation ignores all terms containing a second order derivative in the expression of the Hessian, giving

$$L(\boldsymbol{\theta}) \approx \prod_{i=1}^{N} \left(\exp(\boldsymbol{l}_i(\boldsymbol{\eta}_i^*)) \det \left[\frac{-\boldsymbol{H}_i(\boldsymbol{\eta}_i^*)}{2\pi} \right]^{-1/2} \right) \tag{2.18}$$

where $\boldsymbol{H}_i$ is the first order approximation of $\nabla^2 \boldsymbol{l}_i$ [2].

Since the log function is strictly increasing, maximising $\log L(\boldsymbol{\theta})$ is equivalent to maximising $L(\boldsymbol{\theta})$. This is done in order to avoid numerical problems with large numbers. The objective function to be maximised therefore becomes

$$\log L(\boldsymbol{\theta}) \approx \log L_F(\boldsymbol{\theta}) = \sum_{i=1}^{N} \left(\boldsymbol{l}_i(\boldsymbol{\eta}_i^*) - \frac{1}{2} \log \left| \frac{-\boldsymbol{H}_i(\boldsymbol{\eta}_i^*)}{2\pi} \right| \right) \tag{2.19}$$

called the *Approximate Population Likelihood* (APL) [2].

Using this function, the point estimate $\boldsymbol{\theta}^*$ that maximises the APL is found during the estimation step of the algorithm and using that estimate, its uncertainties are computed during the covariance step. Both steps are described below.

Estimation Step

In order to maximise the APL, two types of optimisation problems, an inner problem and an outer problem, have to be solved. The inner problem is to find $\boldsymbol{\eta}_i^*$ that maximises l_i for each individual i, given fixed parameters $\boldsymbol{\theta}$ for which the APL is evaluated. The outer problem is then to obtain the parameter estimates $\boldsymbol{\theta}^*$ such that

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \log L_F(\boldsymbol{\theta}). \tag{2.20}$$

Both problems can for example be solved using Broyden–Fletcher–Goldfarb–Shanno (BFGS) Quasi-Newton method [16]. This is a gradient based Newton method that uses an iteratively updated approximation of the Hessian.

Covariance Step

When the parameter estimates $\boldsymbol{\theta}^*$ that maximise the APL have been found, their uncertainties are obtained from the Hessian of the APL at $\boldsymbol{\theta}^*$. The covariance matrix of $\boldsymbol{\theta}^*$ can be approximated by the inverse of the negative Hessian [3, 19], i.e.

$$\mathrm{Cov}(\boldsymbol{\theta}^*) \approx -\nabla^2 \log L_F(\boldsymbol{\theta}^*)^{-1}. \tag{2.21}$$

A necessary condition to be able to compute the covariance matrix is therefore that the Hessian $\nabla^2 \log L_F(\boldsymbol{\theta}^*)$ is negative definite. Computing the Hessian also gives information whether the parameter values $\boldsymbol{\theta}^*$ result in a local minimum or a saddle point, since the estimation step ensured that the gradient of the APL in $\boldsymbol{\theta}^*$ is close to zero.

2.1.5 Failure in Parameter Estimation

Parameter estimation is not always successful and can either fail during the estimation step or the covariance step, as shown in Figure 1.2. The estimation fails when the optimisation algorithm does not converge within the desired settings, i.e. failing to achieve small enough gradient and step size to fulfil the stopping criteria of the Newton optimisation algorithm. This could for example be caused by model misspecification or insufficient data [3].

Even if the estimation step succeeds, there can still occur problems when computing the uncertainties. This happens when the Hessian at the estimate is either singular or not negative definite.

If the likelihood function is very flat, the Hessian becomes close to singular and there can be numerical problems inverting the matrix. Consider an example of a function in one variable. If the function is very flat around the estimate, the second derivative of the function at that point will be close to zero and numerical problems when inverting the Hessian might occur.

If the Hessian is negative definite, the estimate is a local maximum of the likelihood function. The other two options are that it is a local minimum or a saddle point. Finding a local minimum when maximising would suggest a poor optimisation method. It is therefore more probable that the estimate is located in a saddle point. This could for example happen when there is very small change in one dimension of the parameter space.

2.1.6 Sensitivity-based First Order Conditional Estimation (S-FOCE)

Gradient-based optimisation methods may be time consuming and numerically unstable, especially for models that require numerical integration of differential equations. In [2] a version of FOCE was introduced where the exact gradients were computed using sensitivity equations, as explained in Section 2.1.3, instead of using the standard finite difference approximation. Below is a pseudocode of the method in Algorithm 1 and examples of how the derivatives are computed using this method, taken from [2]. Further details can be found in the original article. This method will be referred in this thesis to as S-FOCE.

Using Sensitivity-based Gradients when Calculating $\nabla^2 l_i(\eta_i^*)$

In order to calculate the Hessian $\nabla^2 l_i(\eta_i^*)$, the individual likelihood l_i has to be differentiated twice, with respect to η_{ij} and η_{il}. The first order approximation of the Hessian becomes

$$H_{ikl} = -\frac{1}{2}\sum_{j=1}^{n_i}\Big(\boldsymbol{a}_l \boldsymbol{B} \boldsymbol{a}_k^T + tr(-\boldsymbol{c}_l \boldsymbol{c}_k)\Big) - \boldsymbol{\Omega}_{kl}^{-1} \tag{2.22}$$

where

$$\begin{aligned} \boldsymbol{a}_k &= \frac{d\boldsymbol{\varepsilon}_{ij}^T}{d\eta_{ik}} - \boldsymbol{\varepsilon}_{ij}^T \boldsymbol{R}_{ij}^{-1} \frac{d\boldsymbol{R}_{ij}}{d\eta_{ik}} \\ \boldsymbol{B} &= 2\boldsymbol{R}_{ij}^{-1} \\ \boldsymbol{c}_k &= \boldsymbol{R}_{ij}^{-1} \frac{d\boldsymbol{R}_{ij}}{d\eta_{ik}}. \end{aligned} \tag{2.23}$$

Algorithm 1 S-FOCE parameter estimation algorithm

```
$s := 0, \boldsymbol{\theta}_s := \boldsymbol{\theta}_{starting}$                    ▷ Initialise the algorithm
for all individuals do
    $u := 0, \boldsymbol{\eta}_s^u := 0$
end for
repeat                                                   ▷ Solve the outer problem
    for all individuals do
        $u := 0$
        repeat                                           ▷ Solve the inner problem
            Solve for $\boldsymbol{x}$ and the sensitivities $d\boldsymbol{x}/d\boldsymbol{\eta}$
            Compute $l$ and $dl/d\boldsymbol{\eta}$
            Update $\boldsymbol{\eta}_s^{u+1}$ according to BFGS
            $u := u + 1$
        until $\boldsymbol{\eta}^*$ is obtained
    end for
    for all individuals do
        Set $\boldsymbol{\eta} := \boldsymbol{\eta}_s^*$
        Solve for $\boldsymbol{x}$ and the sensitivities $d\boldsymbol{x}/d\boldsymbol{\eta}$, $d\boldsymbol{x}/d\boldsymbol{\theta}$, $d^2\boldsymbol{x}/d\boldsymbol{\eta}^2$ and $d^2\boldsymbol{x}/d\boldsymbol{\eta}d\boldsymbol{\theta}$
    end for
    Compute $\log L_F$ and $d\log L_f/d\boldsymbol{\theta}$
    Uptade $\boldsymbol{\theta}_{s+1}$ according to BFGS
    for all individuals do                   ▷ Set the starting values for inner problem
        $\boldsymbol{\eta}_{s+1}^0 = \boldsymbol{\eta}_s^* + \frac{d\boldsymbol{\eta}_s^*}{d\boldsymbol{\theta}}(\boldsymbol{\theta}_{s+1} - \boldsymbol{\theta}_s)$
    end for$s := s + 1$
until convergence of $\boldsymbol{\theta}$
```

What is left is to compute are the gradients $\frac{d\boldsymbol{\varepsilon}_{ik}}{d\eta_{ik}}$ and $\frac{d\boldsymbol{R}_{ij}}{d\eta_{ik}}$. The gradients are calculated using the chain rule, giving

$$\frac{d\boldsymbol{\varepsilon}_{ij}}{d\boldsymbol{\eta}_{ik}} = -\left(\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{\eta}_{ik}} + \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}_{ik}}\frac{d\boldsymbol{x}_{ik}}{d\boldsymbol{\eta}_{ij}}\right) \tag{2.24}$$

and

$$\frac{d\boldsymbol{R}_{ij}}{d\boldsymbol{\eta}_{ik}} = \frac{\partial \boldsymbol{R}_{ij}}{\partial \boldsymbol{\eta}_{ik}} + \frac{\partial \boldsymbol{R}_{ij}}{\partial \boldsymbol{x}_{ik}}\frac{d\boldsymbol{x}_{ij}}{d\boldsymbol{\eta}_{ik}}. \tag{2.25}$$

The derivatives

$$\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{\eta}_{ik}}, \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}_{ik}}, \frac{\partial \boldsymbol{R}_{ij}}{\partial \boldsymbol{\eta}_{ik}}, \frac{\partial \boldsymbol{R}_{ij}}{\partial \boldsymbol{x}_{ik}} \tag{2.26}$$

can be computed from the definition of $\boldsymbol{h}$ and $\boldsymbol{R}$. However, since $\boldsymbol{x}$ is not explicitly defined as a function of η, the sensitivity based approach is used to compute $\frac{d\boldsymbol{x}_{ik}}{d\eta_{ij}}$. The sensitivity equations yield

$$\begin{aligned} \frac{d}{dt}\left(\frac{d\boldsymbol{x}_i}{d\boldsymbol{\eta}_{ik}}\right) &= \frac{d}{d\boldsymbol{\eta}_{ik}}\left(\frac{d\boldsymbol{x}_i}{dt}\right) = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\eta}_{ik}} + \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}_i}\frac{d\boldsymbol{x}_i}{d\boldsymbol{\eta}_{ik}} \\ \left(\frac{d\boldsymbol{x}_i}{d\boldsymbol{\eta}_{ik}}\right)(t_0) &= \frac{d\boldsymbol{x}_{0i}}{d\boldsymbol{\eta}_{ik}}. \end{aligned} \tag{2.27}$$

The solution to Eq. (2.27) can then be used to solve Eq. (2.24) and Eq. (2.25) and ultimately to compute the approximation of the Hessian (Eq. (2.22)).

Higher Order Sensitivities

The algorithm also requires second order derivatives of the individual likelihoods, $\boldsymbol{l}_i$, which in turn requires the second order derivatives of the individual state values, $\boldsymbol{x}_i$. Those are obtained from solving sensitivity equations which are created in a similar way as shown before. An example of a sensitivity equation for a second order derivative of $\boldsymbol{x}_i$ is

$$\begin{aligned}
\frac{d}{dt}\left(\frac{d^2\boldsymbol{x}_i}{d\boldsymbol{\eta}_{ik}d\boldsymbol{\theta}_m}\right) =& \frac{d^2}{d\boldsymbol{\eta}_{ik}d\boldsymbol{\theta}_m}\left(\frac{d\boldsymbol{x}_i}{dt}\right) \\
=& \frac{\partial^2\boldsymbol{f}}{\partial\boldsymbol{\eta}_{ik}\partial\boldsymbol{\theta}_m} + \frac{\partial\boldsymbol{f}}{\partial\boldsymbol{\eta}_{ik}\partial\boldsymbol{x}_i}\frac{d\boldsymbol{x}_i}{d\boldsymbol{\theta}_m} + \left(\frac{\partial^2\boldsymbol{f}}{\partial\boldsymbol{x}_i\partial\boldsymbol{\theta}_m} + \frac{\partial^2\boldsymbol{f}}{\partial\boldsymbol{x}_i^2}\frac{d\boldsymbol{x}_i}{d\boldsymbol{\theta}_m}\right)\frac{d\boldsymbol{x}_i}{d\boldsymbol{\eta}_{ik}} \\
&+ \frac{\partial\boldsymbol{f}}{\partial\boldsymbol{x}_i}\frac{d^2\boldsymbol{x}_i}{d\boldsymbol{\eta}_{ik}\partial\boldsymbol{\theta}_m} \\
\left(\frac{d\boldsymbol{x}_i}{d\boldsymbol{\eta}_{ik}}\right)(t_0) =& \frac{d^2\boldsymbol{x}_{0i}}{d\boldsymbol{\eta}_{ik}d\boldsymbol{\theta}_m}.
\end{aligned} \tag{2.28}$$

The information in this section should suffice to understand NLME models based on ODEs and how parameter estimation can be done by maximising the population likelihood, either by using finite differences or exact gradients.

2.2 Parameter Estimation of Nonlinear Mixed Effects Models with Stochastic Dynamics

This section will gather the theory needed for extending the S-FOCE method to parameter estimation of NLME models with stochastic dynamics. The section ends with presenting the novel results from the extension, which is also the first contribution of this thesis.

2.2.1 Nonlinear Mixed Effects Model based on SDEs

NLME models can be described using SDEs to describe the dynamics of $\boldsymbol{x}$ instead of using ODEs as done in Section 2.1.1. The observations are modelled as in Eq. (2.1),

$$\boldsymbol{y}_{ij} = \boldsymbol{h}(\boldsymbol{u}_i, \boldsymbol{x}_{ij}, t, \boldsymbol{\theta}, \boldsymbol{\eta}_i) + \boldsymbol{e}_{ij} \tag{2.29}$$

but Eq. (2.2) is exchanged with [12]

$$\begin{aligned}
d\boldsymbol{x}_i &= \boldsymbol{f}(\boldsymbol{x}_i, \boldsymbol{u}_i, t, \boldsymbol{\theta}, \boldsymbol{\eta}_i)dt + \boldsymbol{\Sigma}(\boldsymbol{x}_i, \boldsymbol{u}_i, t, \boldsymbol{\theta}, \boldsymbol{\eta}_i)d\boldsymbol{W}_i \\
\boldsymbol{x}_i(0) &= \boldsymbol{x}_0(\boldsymbol{\theta}, \boldsymbol{\eta}_i).
\end{aligned} \tag{2.30}$$

The function $\boldsymbol{f}(\boldsymbol{x}_i, \boldsymbol{u}_i, t, \boldsymbol{\theta}, \boldsymbol{\eta}_i)$ is called the drift function and the new stochastic part, $\boldsymbol{\Sigma}(\boldsymbol{x}_i, \boldsymbol{u}_i, t, \boldsymbol{\theta}, \boldsymbol{\eta}_i)d\boldsymbol{W}_i$, is called the system noise. Here, $\boldsymbol{W}_i$ is defined as random walk, with $d\boldsymbol{W}_i \sim N(0, dt)$. This way of formulating the NLME model give three sources of variability in response, namely

(i) measurement noise ($\boldsymbol{e}_{ij}$)
(ii) system noise ($\boldsymbol{\Sigma}(\boldsymbol{x}_i, \boldsymbol{u}_i, t, \boldsymbol{\theta}, \boldsymbol{\eta}_i)d\boldsymbol{W}_i$)
(iii) parameter variability ($\boldsymbol{\eta}$)

Using ODEs to describe the dynamics of the predictor vector $\boldsymbol{x}_i$ makes the predictor vector deterministic which causes all unmodelled variations to be classified as noise in measurement. This is not well suited for PK/PD modelling since the distinction lacks between measurement noise and system noise. Describing the dynamics using SDEs allows this distinction and thus gives a more realistic description of the variability in observation. However, this is at the cost of a more complex model.

2.2.2 Approximate Population Likelihood for NLMEs based on SDEs

Since $\boldsymbol{x}$ is a stochastic variable, each measurement $\boldsymbol{d}_{ik}$ is dependent on the previous measurements up to time t_k. Let $Y_{i(k-1)} = \{\boldsymbol{d}_{i1}, \boldsymbol{d}_{i2}, ..., \boldsymbol{d}_{i(k-1)}\}$ denote those measurements. The residuals $\boldsymbol{\epsilon}_{ij}$ are defined similarly as in Eq. (2.7) as

$$\boldsymbol{\epsilon}_{ij} = \boldsymbol{d}_{ij} - \hat{\boldsymbol{y}}_{ij} \tag{2.31}$$

except now the expected measurement value $\hat{\boldsymbol{y}}_{ij}$ and covariance $\boldsymbol{R}_{ij}$ are conditional on Y_i and defined as and

$$\begin{aligned} \hat{\boldsymbol{y}}_{ij} &= E(\boldsymbol{d}_{ij}|Y_{i(j-1)}, \boldsymbol{\theta}) \\ \boldsymbol{R}_{ij} &= Var(\boldsymbol{d}_{ij}|Y_{i(j-1)}, \boldsymbol{\theta}). \end{aligned} \tag{2.32}$$

As shown in [12], the likelihood L_i for individual i becomes

$$\begin{aligned} L_i(\boldsymbol{\theta}|Y_{in_i}) &= p(Y_{in_i}|\boldsymbol{\theta}) \\ &= \int p(Y_{in_i}|\boldsymbol{\theta}, \boldsymbol{\eta}_i)p(\boldsymbol{\eta}_i|\boldsymbol{\theta})d\boldsymbol{\eta}_i \\ &= \int \left(p(\boldsymbol{d}_{in_i}|Y_{i(n_i-1)}, \boldsymbol{\theta}, \boldsymbol{\eta}_i)p(Y_{i(n_i-1)}|\boldsymbol{\theta}, \boldsymbol{\eta}_i)\right) p(\boldsymbol{\eta}_i|\boldsymbol{\theta})d\boldsymbol{\eta}_i \\ &= \int \left(p(\boldsymbol{d}_{i1}|\boldsymbol{\theta}, \boldsymbol{\eta}_i) \prod_{j=2}^{n_i} p(\boldsymbol{d}_{ij}|Y_{i(j-1)}, \boldsymbol{\theta}, \boldsymbol{\eta}_i)\right) p(\boldsymbol{\eta}_i|\boldsymbol{\theta})d\boldsymbol{\eta}_i \end{aligned} \tag{2.33}$$

which combined gives a population likelihood that simplifies in a similar way as Eq. (2.10) to

$$L(\boldsymbol{\theta}) = \prod_{i=1}^{N} \int \exp(\boldsymbol{l}_i)d\boldsymbol{\eta}_i \tag{2.34}$$

where

$$\boldsymbol{l}_i = -\frac{1}{2}\sum_{j=1}^{n_i} \left(\boldsymbol{\epsilon}_{ij}^T \boldsymbol{R}_{ij}^{-1} \boldsymbol{\epsilon}_{ij} + \log\det(2\pi \boldsymbol{R}_{ij})\right) - \frac{1}{2}\boldsymbol{\eta}_i \boldsymbol{\Omega}^{-1}\boldsymbol{\eta}_i - \frac{1}{2}\log\det(2\pi\boldsymbol{\Omega}). \tag{2.35}$$

The FOCE approximation of the population likelihood is derived in the same way as described in Section 2.1.4 as

$$\log L(\boldsymbol{\theta}) \approx \log L_F(\boldsymbol{\theta}) = \sum_{i=1}^{N} \left(\boldsymbol{l}_i(\boldsymbol{\eta}_i^*) - \frac{1}{2} \log \left| \frac{-\boldsymbol{H}_i(\boldsymbol{\eta}_i^*)}{2\pi} \right| \right). \tag{2.36}$$

2.2.3 Extended Kalman Filter

The SDEs introduce uncertainty to the state variables of the system. In order to estimate the state variables, the Extended Kalman filter (EKF) is used, as suggested in [12]. The EKF can be used to calculate $\hat{\boldsymbol{y}}_{ij} = E(\boldsymbol{y}_{ij}|Y_{i(j-1)}, \boldsymbol{\theta})$ and $\boldsymbol{R}_{ij} = Var(\boldsymbol{y}_{ij}|Y_{i(j-1)}, \boldsymbol{\theta})$ as noted in [17]. The continuous-discrete EKF is a state space estimator for the continuous discrete state space models of the form introduced in Eq. (2.30) [12]. It estimates the conditional expectation of the state $\hat{\boldsymbol{x}}_{i(j|j)} = E(\boldsymbol{x}_{it_j}|Y_j, \boldsymbol{\phi})$ and its covariance $\boldsymbol{P_{i(j|j)}} = E(\boldsymbol{x}_{it_j}|Y_j, \boldsymbol{\phi})$. The notation is simplified by suppressing the i and writing

$$\begin{aligned} \hat{\boldsymbol{x}}_{j|j} &= E(\boldsymbol{x}_{t_j}|Y_j, \boldsymbol{\phi}) \\ \boldsymbol{P_{j|j}} &= E(\boldsymbol{x}_{t_j}|Y_j, \boldsymbol{\phi}). \end{aligned} \tag{2.37}$$

The linearisations of the functions $\boldsymbol{f}$ and $\boldsymbol{h}$ from Eq. (2.29) and Eq. (2.30) are introduced as,

$$\begin{aligned} \boldsymbol{A}_t &= \left. \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}_t} \right|_{\boldsymbol{x}_t = \hat{\boldsymbol{x}}_{t|k}} \\ \boldsymbol{C}_k &= \left. \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}_t} \right|_{\boldsymbol{x}_t = \hat{\boldsymbol{x}}_{k|k-1}}. \end{aligned} \tag{2.38}$$

The EKF has two main steps, a prediction step and a measurement step [15]. In the prediction step, the state variables and covariance are predicted by solving the differential equations

$$\begin{aligned} \frac{d\hat{\boldsymbol{x}}_{t|k}}{dt} &= \boldsymbol{f}(\hat{\boldsymbol{x}}_{t|k}, \boldsymbol{u}_t, t, \boldsymbol{\phi}), & t \in [t_k, t_{k+1}] \\ \frac{d\boldsymbol{P}_{t|k}}{dt} &= \boldsymbol{A}_t \boldsymbol{P}_{t|k} + \boldsymbol{P}_{t|k} \boldsymbol{A}_t^T + \boldsymbol{\Sigma}\boldsymbol{\Sigma}^T, & t \in [t_k, t_{k+1}]. \end{aligned} \tag{2.39}$$

with initial conditions

$$\begin{aligned} \hat{\boldsymbol{x}}_{1|0} &= \boldsymbol{x}_0 \\ \boldsymbol{P}_{1|0} &= \boldsymbol{P}_0. \end{aligned} \tag{2.40}$$

The above prediction gives the output prediction equations

$$\begin{aligned} \hat{\boldsymbol{y}}_{k|k-1} &= \boldsymbol{h}(\hat{\boldsymbol{x}}_{k|k-1}, \boldsymbol{u}_k, t_k, \boldsymbol{\phi}) \\ \boldsymbol{R}_{k|k-1} &= \boldsymbol{C}_k \boldsymbol{P}_{k|k-1} \boldsymbol{C}_k^T + \boldsymbol{S}. \end{aligned} \tag{2.41}$$

In the measurement step the prediction is used to compute the *Kalman gain*

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1} \boldsymbol{C}_k^T \boldsymbol{R}_{k|k-1}^{-1} \tag{2.42}$$

which is used to update the state and its covariance

$$\begin{aligned}\hat{\boldsymbol{x}}_{k|k} &= \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k \boldsymbol{\epsilon}_k \\ \boldsymbol{P}_{k|k} &= \boldsymbol{P}_{k|k-1} - \boldsymbol{K}_k \boldsymbol{R}_{k|k-1} \boldsymbol{K}_k^T\end{aligned} \tag{2.43}$$

where

$$\boldsymbol{\epsilon}_k = \boldsymbol{y}_k - \hat{\boldsymbol{y}}_{k|k-1} \tag{2.44}$$

is the prediction error which is assumed to be normal distributed with zero mean and variance $\boldsymbol{R}_{k|k-1}$ [12].

2.2.4 Optimising the Approximate Population Likelihood

As for NLME models with deterministic dynamics, the fixed parameters can be estimated by maximising the APL from Eq. (2.36). This maximisation can be performed in a similar way as the maximisation for NLME models based on ODEs using sensitivity equations for computing exact gradients. The difference is however that the observations, states and their variance are no longer deterministic and have instead been estimated using the EKF. This subsection presents the equations needed for parameter estimation of a general NLME models based on SDEs using the S-FOCE which is one of the main contributions of this thesis.

In order to compute the gradient of the APL for NLME models based on SDEs, the previously introduced method of computing the gradient of the APL for NLME models based on ODEs is extended. Since the observations, state variables and their variance have been estimated using the EKF, it is necessary to obtain the gradients of the EKF equations as well. The derivatives that need to be computed differently than for the ODE case are the first order derivatives

$$\frac{d\boldsymbol{\epsilon}_k}{d\boldsymbol{\eta}_{ij}}, \quad \frac{d\boldsymbol{\epsilon}_k}{d\boldsymbol{\theta}_n}, \quad \frac{d\boldsymbol{R}_{i(k|k-1)}}{d\boldsymbol{\eta}_{ij}}, \quad \frac{d\boldsymbol{R}_{i(k|k-1)}}{d\boldsymbol{\theta}_n}, \quad \frac{d\hat{\boldsymbol{x}}_{i(k|k-1)}}{d\boldsymbol{\eta}_{ij}}, \quad \frac{d\hat{\boldsymbol{x}}_{i(k|k-1)}}{d\boldsymbol{\theta}_n} \tag{2.45}$$

and the second order derivatives

$$\frac{d^2\boldsymbol{\epsilon}_k}{d\boldsymbol{\eta}_{ij}d\boldsymbol{\theta}_n}, \quad \frac{d^2\boldsymbol{\epsilon}_k}{d\boldsymbol{\eta}_{ij}d\boldsymbol{\eta}_{il}}, \quad \frac{d^2\boldsymbol{R}_{i(k|k-1)}}{d\boldsymbol{\eta}_{ij}d\boldsymbol{\theta}_n}, \quad \frac{d^2\boldsymbol{R}_{i(k|k-1)}}{d\boldsymbol{\eta}_{ij}d\boldsymbol{\eta}_{il}}, \quad \frac{d^2\hat{\boldsymbol{x}}_{i(k|k-1)}}{d\boldsymbol{\eta}_{ij}d\boldsymbol{\theta}_n}, \quad \frac{d^2\hat{\boldsymbol{x}}_{i(k|k-1)}}{d\boldsymbol{\eta}_{ij}d\boldsymbol{\eta}_{il}}. \tag{2.46}$$

For a general model the derivations with regards to $\boldsymbol{\theta}$ and $\boldsymbol{\eta}_i$ will be on the same form, so a new parameter vector $\boldsymbol{\phi}$ is introduced to represent derivations with respect to either $\boldsymbol{\theta}$ or $\boldsymbol{\eta}$. Moreover, the individual notation i will from now on be dropped. The problem then reduces to finding the first order derivatives

$$\frac{d\boldsymbol{\epsilon}_k}{d\boldsymbol{\phi}_m}, \quad \frac{d\boldsymbol{R}_{k|k-1}}{d\boldsymbol{\phi}_m}, \quad \frac{d\hat{\boldsymbol{x}}_{k|k-1}}{d\boldsymbol{\phi}_m} \tag{2.47}$$

and the second order derivatives

$$\frac{d^2\boldsymbol{\epsilon}_k}{d\boldsymbol{\phi}_m d\boldsymbol{\phi}_n}, \quad \frac{d^2\boldsymbol{R}_{k|k-1}}{d\boldsymbol{\phi}_m d\boldsymbol{\phi}_n}, \quad \frac{d^2\hat{\boldsymbol{x}}_{k|k-1}}{d\boldsymbol{\phi}_m d\boldsymbol{\phi}_n}. \tag{2.48}$$

The derivations of the results can be found in Appendix C.

First Order Sensitivities for EKF

In order to obtain the exact gradients of the residual $\boldsymbol{\epsilon}_k$ and $\boldsymbol{R}_{k|k-1}$, the first order sensitivity equations of the predicted state and the predicted state variables

$$\begin{aligned} \frac{d}{dt}\left(\frac{d\hat{\boldsymbol{x}}_{t|k}}{d\boldsymbol{\phi}_m}\right) &= \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\phi}_m} + \frac{\partial \boldsymbol{f}}{\partial \hat{\boldsymbol{x}}_{t|k}}\frac{d\hat{\boldsymbol{x}}_{t|k}}{d\boldsymbol{\phi}_m}, \qquad t \in [t_k, t_{k+1}] \\ \frac{d\hat{\boldsymbol{x}}_{t|k}}{d\boldsymbol{\phi}_m}(t_k) &= \frac{d\hat{\boldsymbol{x}}_{k|k-1}}{d\boldsymbol{\phi}_m} + \frac{d\boldsymbol{K}_k}{d\boldsymbol{\phi}_m}\boldsymbol{\epsilon}_k + \boldsymbol{K}_k\frac{d\boldsymbol{\epsilon}_k}{d\boldsymbol{\phi}_m} \end{aligned} \tag{2.49}$$

and

$$\begin{aligned} \frac{d}{dt}\left(\frac{\partial \boldsymbol{P}_{t|k}}{\partial \boldsymbol{\phi}_m}\right) &= \frac{\partial \boldsymbol{A}_t}{\partial \boldsymbol{\phi}_m}\boldsymbol{P}_{t|k} + \boldsymbol{A}_t\frac{\partial \boldsymbol{P}_{t|k}}{\partial \boldsymbol{\phi}_m} + \boldsymbol{P}_{t|k}\frac{\partial \boldsymbol{A}_t}{\partial \boldsymbol{\phi}_m} + \frac{\partial \boldsymbol{P}_{t|k}}{\partial \boldsymbol{\phi}_m}\boldsymbol{A}_t \\ &\quad + \frac{\partial \boldsymbol{\Sigma}}{\partial \boldsymbol{\phi}_m}\boldsymbol{\Sigma}^T + \boldsymbol{\Sigma}\frac{\partial \boldsymbol{\Sigma}^T}{\partial \boldsymbol{\phi}_m}, \qquad t \in [t_k, t_{k+1}] \\ \frac{\partial \boldsymbol{P}_{t|k}}{\partial \boldsymbol{\phi}_m}(t_k) &= \frac{\partial \boldsymbol{P}_{k|k-1}}{\partial \boldsymbol{\phi}_m} \\ &\quad - \left(\frac{\partial \boldsymbol{K}_k}{\partial \boldsymbol{\phi}_m}\boldsymbol{R}_{k|k-1}\boldsymbol{K}_k^T + \boldsymbol{K}_k\frac{\partial \boldsymbol{R}_{k|k-1}}{\partial \boldsymbol{\phi}_m}\boldsymbol{K}_k^T + \boldsymbol{K}_k\boldsymbol{R}_{k|k-1}\frac{\partial \boldsymbol{K}_k^T}{\partial \boldsymbol{\phi}_m}\right) \end{aligned} \tag{2.50}$$

are required. Other derivatives can be calculated explicitly as shown in Appendix C.1

Second Order Sensitivities for EKF

In order to obtain the exact second order derivatives of the residual $\boldsymbol{\epsilon}_k$ and $\boldsymbol{R}_{k|k-1}$, the sensitivity equations for the second partial derivative of predicted expected state, the second partial derivative of predicted state variance

$$\begin{aligned} \frac{d}{dt}\left(\frac{d^2\hat{\boldsymbol{x}}_{t|k}}{d\boldsymbol{\phi}_m d\boldsymbol{\phi}_n}\right) &= \frac{\partial^2 \boldsymbol{f}}{\partial \boldsymbol{\phi}_m \partial \boldsymbol{\phi}_n} + \frac{\partial^2 \boldsymbol{f}}{\partial \eta_j \partial \hat{\boldsymbol{x}}_{t|k}}\frac{d\hat{\boldsymbol{x}}_{t|k-1}}{d\boldsymbol{\phi}_n} \\ &\quad + \left(\frac{\partial^2 \boldsymbol{f}}{\partial \hat{\boldsymbol{x}}_{t|k}\partial \boldsymbol{\phi}_n} + \frac{\partial^2 \boldsymbol{f}}{\partial \hat{\boldsymbol{x}}_{t|k}^2}\frac{d\hat{\boldsymbol{x}}_{t|k}}{d\boldsymbol{\phi}_n}\right)\frac{d\hat{\boldsymbol{x}}_{t|k}}{d\boldsymbol{\phi}_m} \\ &\quad + \frac{\partial \boldsymbol{f}}{\partial \hat{\boldsymbol{x}}_{t|k}}\frac{d^2\hat{\boldsymbol{x}}_{t|k}}{d\boldsymbol{\phi}_m \partial \boldsymbol{\phi}_n}, \qquad t \in [t_k, t_{k+1}] \\ \frac{d^2\hat{\boldsymbol{x}}_{t|k}}{d\boldsymbol{\phi}_m \partial \boldsymbol{\phi}_n}(t_k) &= \frac{d^2\hat{\boldsymbol{x}}_{k|k}}{d\boldsymbol{\phi}_m \partial \boldsymbol{\phi}_n} \end{aligned} \tag{2.51}$$

and

$$\begin{aligned}
\frac{d}{dt}\left(\frac{\partial^2 \boldsymbol{P}_{t|k}}{\partial \boldsymbol{\phi}_m \partial \boldsymbol{\phi}_n}\right) &= \frac{\partial^2 \boldsymbol{A}_t}{\partial \boldsymbol{\phi}_m \partial \boldsymbol{\phi}_n}\boldsymbol{P}_{t|k} + \frac{\partial \boldsymbol{A}_t}{\partial \boldsymbol{\phi}_m}\frac{\partial \boldsymbol{P}_{t|k}}{\partial \boldsymbol{\phi}_n} + \frac{\partial \boldsymbol{A}_t}{\partial \boldsymbol{\phi}_n}\frac{\partial \boldsymbol{P}_{t|k}}{\partial \boldsymbol{\phi}_m} + \boldsymbol{A}_t \frac{\partial^2 \boldsymbol{P}_{t|k}}{\partial \boldsymbol{\phi}_m \partial \boldsymbol{\phi}_n} \\
&\quad + \frac{\partial \boldsymbol{P}_{t|k}}{\partial \boldsymbol{\phi}_n}\frac{\partial \boldsymbol{A}_t^T}{\partial \boldsymbol{\phi}_m} + \boldsymbol{P}_{t|k}\frac{\partial^2 \boldsymbol{A}_t^T}{\partial \boldsymbol{\phi}_m \partial \boldsymbol{\phi}_n} + \frac{\partial^2 \boldsymbol{P}_{t|k}}{\partial \boldsymbol{\phi}_m \partial \boldsymbol{\phi}_n}\boldsymbol{A}_t^T + \frac{\partial \boldsymbol{P}_{t|k}}{\partial \boldsymbol{\phi}_m}\frac{\partial \boldsymbol{A}_t^T}{\partial \boldsymbol{\phi}_n} \\
&\quad + \frac{\partial^2 \boldsymbol{\Sigma}}{\partial \boldsymbol{\phi}_m \partial \boldsymbol{\phi}_n}\boldsymbol{\Sigma}^T + \frac{\partial \boldsymbol{\Sigma}}{\partial \boldsymbol{\phi}_m}\frac{\partial \boldsymbol{\Sigma}^T}{\partial \boldsymbol{\phi}_n} + \frac{\partial \boldsymbol{\Sigma}}{\partial \boldsymbol{\phi}_n}\frac{\partial \boldsymbol{\Sigma}^T}{\partial \boldsymbol{\phi}_m} + \boldsymbol{\Sigma}\frac{\partial^2 \boldsymbol{\Sigma}^T}{\partial \boldsymbol{\phi}_m \partial \boldsymbol{\phi}_n}, \qquad t \in [t_k, t_{k+1}] \\
\frac{\partial^2 \boldsymbol{P}_{t|k}}{\partial \boldsymbol{\phi}_m \partial \boldsymbol{\phi}_n}(t_k) &= \frac{\partial^2 \boldsymbol{P}_{k|k}}{\partial \boldsymbol{\phi}_m \partial \boldsymbol{\phi}_n}
\end{aligned} \tag{2.52}$$

are needed. The sensitivity equations for $\frac{\partial^2 \boldsymbol{P}_{k|k-1}}{\partial \hat{\boldsymbol{x}}_{k|k-1}\boldsymbol{\phi}_m}$ and $\frac{\partial^2 \boldsymbol{P}_{k|k-1}}{\partial \hat{\boldsymbol{x}}_{k|k-1}^2}$ are obtained in the same way. In the special case of $\frac{\partial^2 \boldsymbol{P}_{k|k-1}}{\partial \hat{\boldsymbol{x}}_{k|k-1}^2}$, the equations can be simplified to

$$\begin{aligned}
\frac{d}{dt}\left(\frac{\partial^2 \boldsymbol{P}_{t|k}}{\partial \hat{\boldsymbol{x}}_{k|k-1}^2}\right) &= \frac{\partial^2 \boldsymbol{A}_t}{\partial \hat{\boldsymbol{x}}_{k|k-1}^2}\boldsymbol{P}_{t|k} + \boldsymbol{A}_t\frac{\partial^2 \boldsymbol{P}_{t|k}}{\partial \hat{\boldsymbol{x}}_{k|k-1}^2} + \boldsymbol{P}_{t|k}\frac{\partial^2 \boldsymbol{A}_t^T}{\partial \hat{\boldsymbol{x}}_{k|k-1}^2} + \frac{\partial^2 \boldsymbol{P}_{t|k}}{\partial \hat{\boldsymbol{x}}_{k|k-1}^2}\boldsymbol{A}_t^T \\
&\quad + 2\left(\frac{\partial \boldsymbol{A}_t}{\partial \hat{\boldsymbol{x}}_{k|k-1}}\frac{\partial \boldsymbol{P}_{t|k}}{\partial \hat{\boldsymbol{x}}_{k|k-1}} + \frac{\partial \boldsymbol{P}_{t|k}}{\partial \hat{\boldsymbol{x}}_{k|k-1}}\frac{\partial \boldsymbol{A}_t^T}{\partial \hat{\boldsymbol{x}}_{k|k-1}}\right) \\
&\quad + \frac{\partial^2 \boldsymbol{\Sigma}}{\partial \hat{\boldsymbol{x}}_{k|k-1}^2}\boldsymbol{\Sigma}^T + 2\frac{\partial \boldsymbol{\Sigma}}{\partial \hat{\boldsymbol{x}}_{k|k-1}}\frac{\partial \boldsymbol{\Sigma}^T}{\partial \hat{\boldsymbol{x}}_{k|k-1}} + \boldsymbol{\Sigma}\frac{\partial^2 \boldsymbol{\Sigma}^T}{\partial \hat{\boldsymbol{x}}_{k|k-1}^2}, \qquad t \in [t_k, t_{k+1}] \\
\frac{\partial^2 \boldsymbol{P}_{t|k}}{\partial \hat{\boldsymbol{x}}_{k|k-1}^2}(t_k) &= \frac{\partial^2 \boldsymbol{P}_{k|k}}{\partial \hat{\boldsymbol{x}}_{k|k-1}^2}
\end{aligned} \tag{2.53}$$

Other derivatives can be calculated explicitly as shown in Appendix C.2. The above calculations show how the population likelihood gradient could be obtained for a general model. The resulting algorithm is Algorithm 2. The implementation and testing of this algorithm was not within the scope of this thesis.

Algorithm 2 Extended S-FOCE parameter estimation algorithm for NLME models based on SDEs

```
$\boldsymbol{s} := 0, \boldsymbol{\theta}_s := \boldsymbol{\theta}_{starting}$        ▷ Initialise the algorithm
for all individuals do
    $u := 0, \boldsymbol{\eta}_s^u := 0$
end for
repeat        ▷ Solve the outer problem
    for all individuals do
        $u := 0$
        repeat        ▷ Solve the inner problem
            Solve for $\boldsymbol{x}$ and the sensitivities $d\hat{\boldsymbol{x}}/d\boldsymbol{\eta}$,$\partial\boldsymbol{P}/\partial\boldsymbol{\eta}$,$\partial\boldsymbol{P}/\partial\hat{\boldsymbol{x}}$
            Compute $l$ and $dl/d\boldsymbol{\eta}$
            Update $\boldsymbol{\eta}_s^{u+1}$ according to BFGS
            $u := u + 1$
        until $\boldsymbol{\eta}^*$ is obtained
    end for
    for all individuals do
        Set $\boldsymbol{\eta} := \boldsymbol{\eta}_s^*$
        Solve for $\boldsymbol{x}$ and the sensitivities $d\hat{\boldsymbol{x}}/d\boldsymbol{\eta}$, $d\hat{\boldsymbol{x}}/d\boldsymbol{\theta}$, $d^2\hat{\boldsymbol{x}}/d\boldsymbol{\eta}^2$, $d^2\hat{\boldsymbol{x}}/d\boldsymbol{\eta}d\boldsymbol{\theta}$, $\partial\boldsymbol{P}/\partial\boldsymbol{\eta}$, $\partial\boldsymbol{P}/\partial\boldsymbol{\theta}$, $\partial^2\boldsymbol{P}/\partial\boldsymbol{\eta}^2$, $\partial^2\boldsymbol{P}/\partial\boldsymbol{\eta}\partial\boldsymbol{\theta}$, $\partial^2\boldsymbol{P}/\partial\hat{\boldsymbol{x}}\partial\boldsymbol{\eta}$, $\partial^2\boldsymbol{P}/\partial\hat{\boldsymbol{x}}\partial\boldsymbol{\theta}$ and $\partial^2\boldsymbol{P}/\partial\hat{\boldsymbol{x}}^2$
    end for
    Compute $\log L_F$ and $d\log L_f/d\boldsymbol{\theta}$
    Uptade $\boldsymbol{\theta}_{s+1}$ according to BFGS
    for all individuals do        ▷ Set the starting values for inner problem
        $\boldsymbol{\eta}_{s+1}^0 = \boldsymbol{\eta}_s^* + \frac{d\boldsymbol{\eta}_s^*}{d\boldsymbol{\theta}}(\boldsymbol{\theta}_{s+1} - \boldsymbol{\theta}_s)$
    end for$s := s + 1$
until convergence of $\boldsymbol{\theta}$
```

2.3 Pharmacokinetic and Pharmacodynamic Modelling

When developing drugs, it is important to understand the interaction between the drug and the body, e.g. in order to be able to determine correct doses. The term pharmacokinetics (PK) describes what the body does to a drug and pharmacodynamics (PD) describes how the drug affects the body. This chapter will describe some of the basic elements of PK and PD modelling in order to give insight into the models used for analysing and generating the simulated data which are described in Section 4.1. The models and schematic figures in this section are based on [8].

2.3.1 Pharmacokinetics

There are different ways of pharmacokinetic modelling. This section will only discuss the compartmental modelling since the models used for comparison are both based on that method. The body can be regarded as a system and the drug as an input into the body. Within the body are different organs that act and interact in a certain way. In order to model this behaviour, the different systems of the body, such as tissues, stomach and veins can be viewed as different compartments. The model then describes the changes of drug concentration in each compartment as functions of time. Usually, they can be described with a system of first order differential equations, where one equation is used for each compartment.

One-compartment Models

Consider a system with only one compartment. This could for example be the blood or plasma in the body. Imagine that a drug enters the body by injection and then the drug disappears from there over time. This can be modelled by considering the drug concentration in the compartment as a function of time that is affected by input and elimination. A simple way is to consider the elimination to be linear. This rate of change in concentration can be described by

$$\frac{dC}{dt} = -\frac{Cl}{V_c}C \tag{2.54}$$

where Cl is the clearance, defined as the volume of blood or plasma that is completely cleared of drug per unit time [8], V_c the volume of the compartment and C the unitless concentration of the compartment. This compartment is usually referred to as the central compartment. Note that in Eq. (2.54), the dynamics are only dependent on the elimination but not the input. The input could either be added as an initial condition, representing a one time dose, or as a function of time t, $I(t)$, changing the concentration dynamics into

$$\frac{dC}{dt} = I(t) - \frac{Cl}{V_c}C. \tag{2.55}$$

A schematic figure of a single compartment model is shown in Figure 2.2.

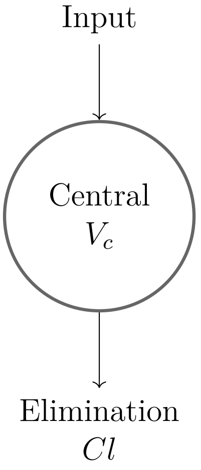


Figure 2.2: *One compartment model where the central compartment has volume V_c and clearance Cl [8].*

Two-compartment Models

Often, a single compartment is insufficient to describe the dynamics of the system. This holds for example for systems that have drug-flow between tissues in the body. One way of describing this increased complexity, is to add compartments to the model. Figure 2.3 shows a schematic figure of a two-compartment model. The model has a central compartment with similar dynamics as the single compartment model and an additional second compartment, so-called pheripheral compartment, with volume V_p. The flux between those two compartments has a clearance Cl_d in both directions. The model therefore looks as follows

$$\begin{aligned} V_c \frac{dC}{dt} &= -Cl \cdot C - Cl_d \left(C - C_p\right) \\ V_p \frac{dC_p}{dt} &= Cl_d \left(C - C_p\right). \end{aligned} \tag{2.56}$$

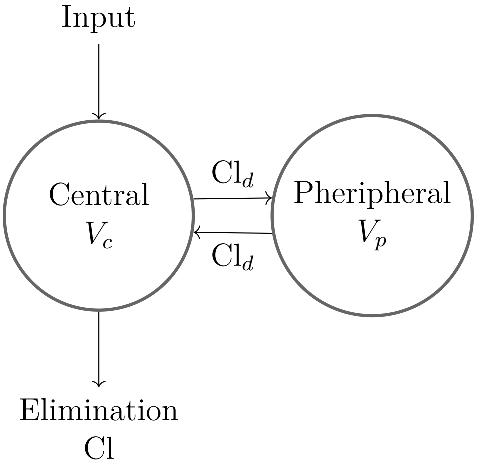


Figure 2.3: *Two compartment model where the central compartment has volume V_c and clearance Cl. In addition, there is a pheripheral compartment with volume V_p connected to the central compartment. The clearance between the compartments is the same in both directions, Cl_d [8].*

In a similar way, more complicated models can be created by adding even more compartments, either connecting them to the central compartment or to another pheripheral compartment.

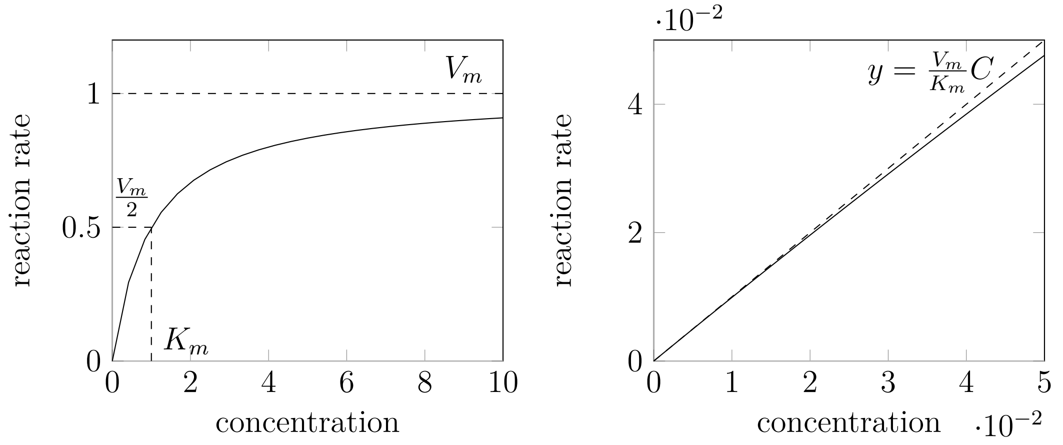
2.3.2 Michaelis-Menten Elimination

The law of mass action states that the rate of a chemical reaction is proportional to the product of the masses of the reactants [18]. However, some enzyme reactions do not follow this law and instead have a maximum reaction velocity that can be reached at high substrate concentrations. Michaelis-Menten elimination was introduced in order to model this kind of non-linear elimination [11]. The change in concentration is modelled as

$$\frac{dC}{dt} = -\frac{V_m C}{K_m + C} \tag{2.57}$$

where V_m is the maximum rate of change of concentration and K_m is the Michaelis-Menten constant.

The behaviour of the reaction rate $\dfrac{V_m C}{K_m + C}$ can be seen in Figure 2.4. For small concentration C, the reaction rate is nearly linear in the concentration as shown in Figure 2.4b. As the concentration grows, the reaction rate diverges from linearity and approaches the maximum reaction velocity V_m. This can be seen in Figure 2.4a. The Michaelis-Menten constant K_m represents the concentration that gives reaction rate equal to $V_m/2$.



(a) *Reaction rate for concentration between* 0 *and* 10.

(b) *Reaction rate for low concentration between* 0 *and* 0.05.

Figure 2.4: *Reaction rate as a function of concentration, as described by the Michaelis-Menten model with* $K_m = 1$ *and* $V_m = 1$.

2.3.3 Pharmacodynamics

Pharmacodynamics describes what the drug does to the body. The response of an individual is modelled as a function of the concentration of drug. The measured response could for example be the change of an individual's temperature, heart rate, blood pressure or level of fatty acids in the plasma. These are all examples of a continuous response. The other main type of response is quantal response. Seisures, cancer and death are examples of those responses [8]. This thesis only considers models with continuous response.

An example of a simple response model, taken from [8] is

$$E = E_0 - \frac{I_{max} C}{IC_{50} + C}. \tag{2.58}$$

Here, E is the pharmacological effect, E_0 is the baseline value, I_{max} the maximum drug-induced inhibitory effect and IC_{50} is the concentration at 50% reduction of maximal effect. This can be seen in Figure 2.5.

The two types of models, PK and PD can be combined into PK/PD modelling. These models describe what happens from drug administration to response [6].

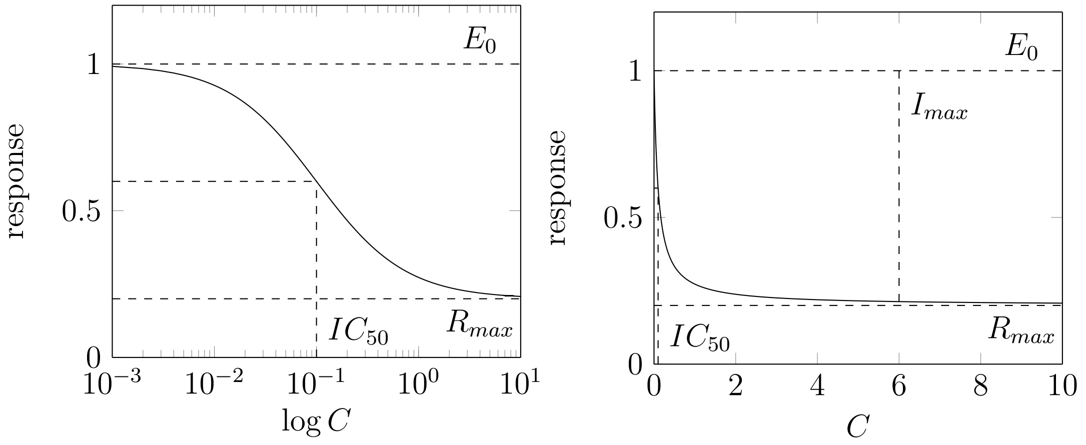


Figure 2.5: *Example of PD response with $E_0 = 1$, $I_{max} = 0.8$ and $IC_{50} = 0.1$. R_{max} is the maximal response.*

2.4 Nonparametric Statistical Tests

In statistics, hypothesis tests are used to see if data is consistent with a statistical hypothesis, often consisting of a statement about parameters of a population distribution. The given hypothesis is then either approved or rejected using these tests. Hypothesis tests where no parametric form for underlying distribution is assumed are called nonparametric [20]. Among the nonparametric test are the two-sided sign test [20], used for comparing binary data, and the Wilcoxon signed rank test [25, 21], used for comparing real-valued data. Those are described below. For further information on nonparametric statistical tests, see [20, 5].

2.4.1 The Two-sided Sign Test

Let (X_i, Y_i) be a pair of binary observations. In order to determine if the paired observations are significantly different, the following is done. First, a new set is computed with elements $Z_i = X_i - Y_i$. Thereafter all elements where $Z_i = 0$ are removed from the set. This new set is called Z. Next, let

$$n = n_p + n_n \tag{2.59}$$

where n_p is the number of elements $Z_i \in Z$ where $Z_i = 1$ and n_n is the number of elements $Z_i \in Z$ where $Z_i = -1$. The hypothesis to test is

$$H_0 : m = 0 \qquad \text{versus} \qquad H_1 : m \neq 0 \tag{2.60}$$

where m is the population median. The p-value [20] is

$$p\text{-value} = \begin{cases} 2P\{\text{Bin}(n, 0.5) \le n_n\} & \text{if } n_n < \frac{n}{2} \\ 2P\{\text{Bin}(n, 0.5) \le n_p\} & \text{if } n_n > \frac{n}{2} \\ 1 & \text{if } n_n = \frac{n}{2} \end{cases} \tag{2.61}$$

where

$$P\{\mathrm{Bin}(n, 0.5) \leq n_0\} = 0.5^n \sum_{k=0}^{n_0} \binom{n}{k}. \tag{2.62}$$

For a given percent level of significance δ, the null hypothesis is rejected if

$$p\text{-value} < \delta \tag{2.63}$$

thus strengthening the idea that the first algorithm is significantly better.

2.4.2 Wilcoxon Signed Rank Test

Let (X_i, Y_i) be a pair of real valued observations. In order to determine if the paired observations are significantly different, the following is done. From the above results, a new set of results is computed with elements $Z_i = X_i - Y_i$. Thereafter all elements where $Z_i = 0$ are removed from the set. This new set is called Z. Let n be the number of elements in Z.

Next, the $Z_i \in Z$ are ordered from the smallest to the largest absolute value and r_i is set to be the rank of Z_i. The rank ranges from 1 to n and represents where in the sorted order the value Z_i lies among all elements of Z. For elements of Z that have the same absolute value, the rank is the average of the ranks they cover.

The hypothesis to test is

$$H_0 : m = 0 \qquad \text{versus} \qquad H_1 : m \neq 0 \tag{2.64}$$

where m is the median of Z. Define T_+ as the sum of the absolute values of the positive differences and T_- as the sum of the absolute values of the negative differences, i.e.

$$\begin{aligned} T_+ &= \sum_{i=1, Z_i>0}^{n} \mathrm{sign}|Z_i| r_i \\ T_- &= \sum_{i=1, Z_i<0}^{n} \mathrm{sign}|Z_i| r_i. \end{aligned} \tag{2.65}$$

For sample sizes $n > 15$, the test statistics becomes

$$Z = \frac{T_+ - n(n+1)/4}{\sqrt{2n(n+1)(2n+1)/48}} \tag{2.66}$$

with an approximately standard normal distribution under the null hypothesis. The null hypothesis is rejected if

$$T_+ \leq C'_{\delta/2} \vee T_+ \geq C_{\delta/2} \tag{2.67}$$

where C_δ, C'_δ are defined as

$$\begin{aligned} C_\delta &= \min\left(t : \Phi\left(\frac{t - n(n+1)/4 - 0.5}{\sqrt{2n(n+1)(2n+1)/48}}\right) \leq 1 - \delta\right) \\ C'_\delta &= \max\left(t : \Phi\left(\frac{t - n(n+1)/4 + 0.5}{\sqrt{2n(n+1)(2n+1)/48}}\right) \leq \delta\right) \end{aligned} \tag{2.68}$$

and δ is the significance level [21].

3

Parameter Estimation Method Implementations

There exist numerous software for parameter estimation in NLME models. For the comparison of parameter estimation methods, a S-FOCE method is implemented in Mathematica and compared to two versions of the FOCE method that both use finite differences for computing gradients during optimisation and are implemented in a commercial software called NONMEM. This is done as a benchmark to what is known to give good results. The NONMEM methods are referred to as NM-FOCE and NM-SLOW-FOCE. This chapter describes the implementation of the S-FOCE in Mathematica, the software NONMEM and its methods, and finally defines the settings of all methods to limit the effect of other factors than the computation of gradients.

3.1 Implementation of S-FOCE in Mathematica

For this project, a program for parameter estimation was implemented in Mathematica (MMA) [26]. Mathematica was chosen due to functionality already implemented in Mathematica, such as numerical integration methods. It uses the built in numerical solver for solving the differential equations and obtaining gradients. The program is an implementation of the S-FOCE algorithm (Algorithm 1) based on an existing version developed by FCC. The modifications involved

- allowing to constrain positive parameters of the search space during the covariance step using transformations (Section 3.1.2)
- estimating uncertainties of the untransformed parameter estimates and the estimated variance of intra-individual effects, $\boldsymbol{\Omega}$ (Section 3.1.3)
- the inclusion of a new optimisation method (Section 3.1.5)
- a generalised way of handling intra-individual variance (Section 3.1.5)
- handling runs of multiple datasets (Section 3.1.5)

and general debugging of the previous version.

The implementation of the algorithm uses an unconstrained BFGS-Quasi Newton optimisation method developed at FCC. However, without constraints, some differential equations representing the system might become unrealistic for the models in question. The constraints implemented for the parameter estimation involve constraining the covariance matrix of the individual random effects to be positive semi definite and constraining parameters that should be positive. Both these constraints affect the uncertainty estimation. The implemented constraints and their effect on the uncertainty estimation is described below.

3.1.1 Estimating Covariance of Individual Random Effects

When estimating the elements of the covariance matrix $\boldsymbol{\Omega}$ of the individual random effects, one must consider that $\boldsymbol{\Omega}$ should be positive semi-definite. Note that for any matrix $\boldsymbol{\Omega}$ such that

$$\boldsymbol{\Omega} = \boldsymbol{\Lambda}\boldsymbol{\Lambda}^T \tag{3.1}$$

where $\boldsymbol{\Lambda}$ is a lower triangle matrix, it holds that

$$\boldsymbol{x}^T\boldsymbol{\Omega}\boldsymbol{x} = \boldsymbol{x}^T\boldsymbol{\Lambda}\boldsymbol{\Lambda}^T\boldsymbol{x} = (\boldsymbol{\Lambda}^T\boldsymbol{x})^T(\boldsymbol{\Lambda}^T\boldsymbol{x}) = ||\boldsymbol{\Lambda}^T\boldsymbol{x}||^2 \geq 0 \tag{3.2}$$

showing that $\boldsymbol{\Omega}$ is positive semi-definite. These are sufficient conditions for $\boldsymbol{\Omega}$ to be a covariance matrix. Moreover, $\boldsymbol{\Omega}$ is a positive definite matrix if $\boldsymbol{\Lambda}$ is not singular. Therefore, the model can be formulated so that it estimates the elements of the lower triangular matrix $\boldsymbol{\Lambda}$ instead of the elements of $\boldsymbol{\Omega}$, ensuring a positive semi-definite covariance matrix estimation $\boldsymbol{\Omega}$. There are no boundaries on the elements of $\boldsymbol{\Lambda}$.

3.1.2 Transformation of Positive Parameters

Many physiological parameters in PK/PD models have to be positive, taking for example the clearance. If the clearance becomes negative then unexplainable input is introduced to the system. In order to limit the search space, the the models can be re-parametrised by doing a log-transformation [23]. For a parameter $\boldsymbol{\theta}_i$ that should be positive, a new parameter $\tilde{\boldsymbol{\theta}}_i$ is introduced such that

$$\boldsymbol{\theta}_i = e^{\tilde{\boldsymbol{\theta}}_i}. \tag{3.3}$$

For other parameters that do not have the positive constraint, let $\tilde{\boldsymbol{\theta}}_i = \boldsymbol{\theta}_i$. The likelihood function $L(\boldsymbol{\theta})$ is written as a function $L(\tilde{\boldsymbol{\theta}})$ of $\tilde{\boldsymbol{\theta}}$ and the parameters $\tilde{\boldsymbol{\theta}}$ are estimated. From the estimated $\tilde{\boldsymbol{\theta}}$, the estimate of $\boldsymbol{\theta}$ can be computed.

3.1.3 Estimating Uncertainties

Assume that the parameter vector $\tilde{\boldsymbol{\theta}}$ has been estimated using the above transformations of the covariance matrix elements and the positive parameters. Note also that, since the elements of $\boldsymbol{\Omega}$ are included in the population parameter vector $\boldsymbol{\theta}$, the elements of $\boldsymbol{\Lambda}$ are included in the transformed parameter vector $\tilde{\boldsymbol{\theta}}$.

For an obtained parameter estimate, the uncertainties of the original parameters have to be evaluated. However, computing the Hessian of $L(\tilde{\boldsymbol{\theta}})$ only gives uncertainty approximations for the transformed elements $\tilde{\boldsymbol{\theta}}$. Therefore, the Hessian $\boldsymbol{H}$ for the untransposed model also has to be computed in order to get the correct covariance matrix of the parameter estimates.

In order to find the Hessian $\boldsymbol{H}$ for the untransposed model, note the following

$$\begin{aligned}\boldsymbol{H}_{ij} &= \frac{\partial^2 L}{\partial\boldsymbol{\theta}_i\partial\boldsymbol{\theta}_j} = \frac{\partial}{\partial\boldsymbol{\theta}_i}\left(\frac{\partial L}{\partial\boldsymbol{\theta}_j}\right) = \frac{\partial}{\partial\boldsymbol{\theta}_i}\left(\frac{\partial L}{\partial\tilde{\boldsymbol{\theta}}}\frac{\partial\tilde{\boldsymbol{\theta}}}{\partial\boldsymbol{\theta}_j}\right)\\ &= \frac{\partial\tilde{\boldsymbol{\theta}}}{\partial\boldsymbol{\theta}_i}\frac{\partial^2 L}{\partial\tilde{\boldsymbol{\theta}}^2}\frac{\partial\tilde{\boldsymbol{\theta}}}{\partial\boldsymbol{\theta}_j} + \underbrace{\frac{\partial L}{\partial\tilde{\boldsymbol{\theta}}}}_{=0}\frac{\partial^2\tilde{\boldsymbol{\theta}}}{\partial\boldsymbol{\theta}_i\partial\boldsymbol{\theta}_j} \qquad (3.4)\\ &= \frac{\partial\tilde{\boldsymbol{\theta}}^T}{\partial\boldsymbol{\theta}_i}\tilde{\boldsymbol{H}}\frac{\partial\tilde{\boldsymbol{\theta}}}{\partial\boldsymbol{\theta}_j}.\end{aligned}$$

The above calculations show that $\boldsymbol{H}$ can be computed from from $\tilde{\boldsymbol{H}}$ by using $\tilde{\boldsymbol{\theta}}$ to rescale as follows

$$\boldsymbol{H} = \boldsymbol{J}^T\tilde{\boldsymbol{H}}\boldsymbol{J}. \qquad (3.5)$$

where $\boldsymbol{J}$ is the Jacobian of $\tilde{\boldsymbol{\theta}}(\boldsymbol{\theta})$.

In the case of a one to one dependency between $\boldsymbol{\theta}_i$ and $\tilde{\boldsymbol{\theta}}_i$ for all i,

$$\frac{\partial\tilde{\boldsymbol{\theta}}_j}{\partial\boldsymbol{\theta}_i} = 0, \qquad \forall i \neq j \qquad (3.6)$$

and the results from Eq. (3.5) can be simplified to

$$\boldsymbol{H}_{ij} = \tilde{\boldsymbol{H}}_{ij}\frac{\partial\tilde{\boldsymbol{\theta}}_i}{\partial\boldsymbol{\theta}_i}\frac{\partial\tilde{\boldsymbol{\theta}}_j}{\partial\boldsymbol{\theta}_j} \qquad (3.7)$$

where

$$\begin{aligned}\frac{\partial\tilde{\boldsymbol{\theta}}_i}{\partial\boldsymbol{\theta}_i} &= \begin{cases}1 & \text{if } \tilde{\boldsymbol{\theta}}_i = \boldsymbol{\theta}_i\\ 1/\boldsymbol{\theta}_i & \text{if } \tilde{\boldsymbol{\theta}}_i = \ln\boldsymbol{\theta}_i\end{cases}\\ &= \begin{cases}1 & \text{if } \tilde{\boldsymbol{\theta}}_i = \boldsymbol{\theta}_i\\ e^{-\tilde{\boldsymbol{\theta}}_i} & \text{if } \tilde{\boldsymbol{\theta}}_i = \ln\boldsymbol{\theta}_i\end{cases}.\end{aligned} \qquad (3.8)$$

For the parametrisation responding to the elements of the covariance of individual random effects, $\tilde{\boldsymbol{\theta}}_i$ could depend on more than one element of $\boldsymbol{\theta}$. The Jacobian of $F^{-1}(\boldsymbol{\theta})$ can be found using that

$$J_{F^{-1}}(F(\tilde{\boldsymbol{\theta}})) = J_F(\tilde{\boldsymbol{\theta}})^{-1}. \qquad (3.9)$$

This way of reparametrisation is implemented in the Mathematica program for parameter estimation.

3.1.4 Hessian Computation

The Hessian in the covariance step is computed using central difference approximation in both implementations. For the S-FOCE, the exact gradients

$$\nabla\log L_F = \frac{d\log L_F}{d\boldsymbol{\theta}} \qquad (3.10)$$

of the APL are used for the Hessian computation, yielding

$$\boldsymbol{H}_j = \frac{\nabla \log L_F(\boldsymbol{\theta} + \boldsymbol{h}_j) - \nabla \log L_F(\boldsymbol{\theta} - \boldsymbol{h}_j)}{2h_j} \tag{3.11}$$

where $\boldsymbol{H}_j$ represents the j-th column of the Hessian, $\boldsymbol{h}_j = h_j\boldsymbol{e}_j$, $\boldsymbol{e}_j$ is the j-th unit vector and h_j is the step length.

3.1.5 Other Implementation

An optimisation algorithm, developed in Mathematica by FCC was appended to the S-FOCE implementation. It is an implementation of the BFGS Quasi-Newton method. The optimisation algorithm takes in two parameters, one representing the desired precision of the parameter estimates and one representing the relative digits of the objective function. The number of relative digits is used internally in the algorithm but does not affect the stopping conditions.

The covariance matrix of the measurement noise, $\boldsymbol{R}(\boldsymbol{u}_i, \boldsymbol{x}_{ij}, t, \boldsymbol{\theta}, \boldsymbol{\eta}_i)$, could depend on parameters that have to be evaluated and also on states, input and time. Therefore, the derivatives of $\boldsymbol{R}$ are also evaluated at the relevant values of $\boldsymbol{u}_i, \boldsymbol{x}_{ij}, t, \boldsymbol{\theta}, \boldsymbol{\eta}_i$.

Multiple datasets can be run on the same model by looping through the datasets one by one. In order to minimise unnecessary memory leaks in Mathematica, the variables are cleared between datasets. This can be done since the datasets are independent of each other.

3.1.6 Method Parameters in S-FOCE Implementation

For each optimisation, two parameters have to be determined, the precision p and relative digits r. These parameters are set specifically for each of the three different optimisations that have to be performed; the inner optimisation to optimise the individual likelihood l_i, the outer optimisation to determine the optimal population likelihood during the estimation step and the inner optimisation to compute the gradient of the APL during the covariance step. The constraints on the parameters are that for each optimisation the inequality

$$p \leq rd \tag{3.12}$$

should hold.

3.2 FOCE Implementation in NONMEM

The software NONMEM [4] is chosen to serve as a benchmark program for comparison. NONMEM is a well acknowledged software used for analysing NLME models. It was developed by Stuart Beal, Lewis Sheinier and Alison Boeckmann. It estimates parameters by minimising the -2 log likelihood of the model parameters. NONMEM has has multiple estimation methods, including first order conditional estimation (FOCE), first order conditional estimation with interaction (FOCEI), iterative two stage (ITS), Monte Carlo importance sampling (IMP), importance sampling assisted by mode a posteriori (IMPMAP), stochastic approximation expectation–maximization (SAEM), and Markov chain Monte Carlo Bayesian (BAYES) [9]. In NONMEM, two different versions of the FOCE algorithm, NM-FOCE and NM-SLOW-FOCE are considered. The NM-FOCE method uses finite difference approximations when computing the second order sensitivities with respect to the random individual parameters $\boldsymbol{\eta}$. Other derivatives are computed by a mixture of finite differences and another method. In NONMEM there exists a $SLOW$ version of the FOCE method where all gradients in the outer level of the optimisation are computed using the finite difference approach. This is referred to as the NM-SLOW-FOCE method.

3.2.1 Method Parameters in NONMEM

In NONMEM there are a few method parameters that have to be specified for the FOCE method. They are *NSIG*, *SIGL* and *TOL*. These parameters represent the number of significant digits that population parameters are to be evaluated at the maximum likelihood, the number of significant digits to which the objective function is evaluated and the number of relative significant digits precision to which differential equations are to be integrated.

The NONMEM user guide [4] suggests setting the parameters such that

$$\begin{aligned} \mathit{SIGL} &\leq \mathit{TOL} \\ \mathit{NSIG} &\leq \mathit{SIGL}/3 \end{aligned} \tag{3.13}$$

In the parameter estimations performed, this suggestion was followed and the relation

$$3 \cdot \mathit{NSIG} = \mathit{SIGL} \quad = \mathit{TOL} \tag{3.14}$$

used.

In the user guide it was also mentioned that using higher values of $SIGL$ and TOL during covariance step might lead to improved success in the covariance step. During the covariance step, the parameters $SIGL_{cov}$ and TOL_{cov} are therefore set as follows

$$\mathit{SIGL}_{cov} = \mathit{TOL}_{cov} = \max(6, 4 \cdot \mathit{NSIG}) \tag{3.15}$$

Using the above constraints of Eq. (3.14) and Eq. (3.15), the only method parameter to determine for parameter estimation using NONMEM is $NSIG$.

When computing the derivatives using finite difference approximation NONMEM uses both forward and central differences. The forward differences are

$$\frac{O(\theta_1(1+h)) - O(\theta_1)}{\theta_1 h} \tag{3.16}$$

Table 3.1: *Choice of h for finite differences in NONMEM.*

	Forward	Central	Forward second order	Evaluating $\boldsymbol{R}$
h	$SIGL/2$	$SIGL/3$	$SIGL/3$	$SIGL/4$

and the central differences

$$\frac{O(\theta_1(1+h)) - O(\theta_1(1-h))}{2\theta_1 h} \tag{3.17}$$

The value h is chosen as shown in Table 3.1, where $\boldsymbol{R}$ refers to the Hessian of the APL function at the obtained parameter estimate $\boldsymbol{\theta}^*$.

3.3 Estimation Method Settings

In order to make the estimation method implementation comparable, the uncertainty is estimated in the same way, the optimisation is constrained for all methods and the intra-individual variability is modelled in an equivalent way. This is done as follows.

3.3.1 Uncertainty Estimation of Estimated Parameters

As mentioned in Section 2.1.4, the standard error covariance matrix of the parameter estimates can be approximated with the inverse of the negative Hessian computed at the parameter point estimate. This is implemented in S-FOCE by computing the Hessian as a derivative of the gradient of the APL using central differences as shown in Section 3.1.4.

In NONMEM there exist three ways of computing the error covariance matrix of the parameters estimated. One way is to compute the inverse of the Hessian $\boldsymbol{R}$ evaluated at the final estimates $\boldsymbol{\theta}^*$. The second is to compute $\boldsymbol{S}$, the matrix obtained from the cross product of the gradient vector and its transpose where the gradient is a vector of first derivatives evaluated at the final estimates. The third and default way is to compute $\boldsymbol{R}^{-1}\boldsymbol{S}\boldsymbol{R}^{-1}$. In order to have comparable methods of estimating the error covariance matrix of the parameter estimates, the covariance matrix is computed using the Hessian $\boldsymbol{R}$.

3.3.2 Constraining Fixed Population Parameters

In Mathematica, the transformation presented in Section 3.1.1 is used to ensure positive covariance matrix of inter-individual random effects. The transformation presented in 3.1.2 is also used, in order to ensure positivity of the parameters not connected to the two covariance matrices. In NONMEM, the positivity of the parameters not connected to the two covariance matrices is ensured by declaring lower bound of 10^{-16}.

3.3.3 Intra-individual Variability Model

Intra-individual variability is modelled with the measurement noise. The two types of intra-individual variability used in this project are additive and combined proportional and additive error.

Additive Error

The additive error is modelled in the same way in both Mathematica and NONMEM by adding a normally distributed variable e to the output with zero mean and covariance $\mathbf{\Sigma} = [\sigma^2]$. The parameter σ is one of the parameters that have to be evaluated.

Combined Proportional and Additive Error

The combined proportional and additive error is modelled by adding a normally distributed variable e to the expected observation h with zero mean and covariance $\mathbf{\Sigma} = [\sigma_1^2 h^2 + \sigma_2^2]$. The parameters σ_1, σ_2 are two of the parameters that have to be estimated. This is done in Mathematica by defining the covariance matrix $\mathbf{\Sigma}$ and declaring the parameters σ_1, σ_2. In NONMEM, the error is modelled by setting the observation y as

$$y = h + he_1 + e_2 \tag{3.18}$$

where e_1 and e_2 are normally distributed with zero mean and covariances σ_1^2 and σ_2^2 respectively. According to the NONMEM user guide [3], the covariance of y then becomes $\mathbf{\Sigma} = [\sigma_1^2 h^2 + \sigma_2^2]$ as wanted.

4

Methods

This chapter introduces the models and the simulated datasets used for method comparison and presents the methodology behind the performed comparison.

4.1 Models and Data

Two different models are used for comparison of the estimation methods, each consisting of four parts: a dynamical system described with ordinary differential equations, a model for the observations, a model for the inter-individual variability and a computer simulated experimental design. The synthetic data is generated from each model using Mathematica from the settings described in Section 4.1.2 and Section 4.1.3

4.1.1 Generating Simulated Data

The simulated data is generated using Mathematica. In order to generate data, the model and the parameter values have to be defined. The model is described by introducing the state variables of the system and ODEs to describe them along with the initial states of the ODEs, the population parameters to be evaluated, the inter-individual variability (IIV) of population parameters and the functions describing both the output function and the intra-individual covariance matrix (the measurement noise). The IIV of a parameter $\boldsymbol{\theta}_m$ was introduced in the model by replacing $\boldsymbol{\theta}_m$ by

$$\boldsymbol{\theta}_{m_i} = \boldsymbol{\theta}_m \cdot e^{\boldsymbol{\eta}_{ij}} \tag{4.1}$$

for individual i where $\boldsymbol{\eta}_{ij}$ is the j-th individual random parameter for individual i. The parameter $\boldsymbol{\theta}_m$ becomes a fixed population parameter. Individual random effects could also be introduced by explicitly introducing them in the state equations, but that is not done for the models in this study.

The values needed for data simulation are the number of individuals in the population, the fixed population parameters and the time points at which the measurements are taken.

4.1.2 M1 (PK)

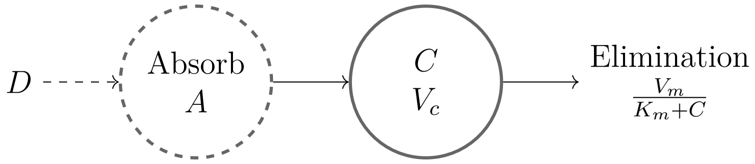


Figure 4.1: *Schematic figure of one-compartment model with absorption compartment.*

This model is a one-compartment pharmacokinetic (PK) model with an absorption compartment and non-linear Michaelis Menten elimination [14]. A schematic figure of the model can be seen in Figure 4.1. The model describes the concentration of a drug over time where the drug enters an absorption compartment at time zero and gradually moves from there into the central compartment from where it gets eliminated. The kinetics are described using the equations

$$\begin{aligned} \frac{dA(t)}{dt} &= -k_A A(t) \\ V\frac{dC(t)}{dt} &= k_A A(t) - \frac{V_\mathrm{m} C(t)}{K_\mathrm{m} + C(t)} \\ A(0) &= D \\ C(0) &= 0 \end{aligned} \tag{4.2}$$

where A is the amount of drug in the absorption compartment, C the concentration of drug in the central compartment and D an oral dose given at time zero. Note that, although there exist both an absorption compartment and a central compartment this model is considered to be one-compartmental since there are no dynamics between the compartments, only flow from A to C and hence A can be considered as an input factor.

The observations are modelled as

$$y_t = C(t) + e_t \tag{4.3}$$

with a combined error

$$\begin{aligned} e_t &\overset{i.i.d.}{\sim} N(0, \Sigma) \\ \Sigma &= \sigma_1^2 + (\sigma_2 C_2(t))^2. \end{aligned} \tag{4.4}$$

Data Simulation

When generating the data, two versions of M1 are considered:

a) two dose-groups of 10 individuals given 1 and 5 units of the drug, respectively, with inter-individual variability on V_m and V;

b) two dose-groups of 10 individuals given 1 and 5 units of the drug respectively, with inter-individual variability on K_m and V.

Samples are taken at time points 0.5, 1, 1.5, 2, 5 and 10 for both groups in model M1a, with an additional time point at 15 in model M1b. The parameters of M1 are summarised in Table A.1. An example of the computer generated data can be seen in Figure 4.2 and an example of a simulated population can be seen in Figure 4.3. Figure 4.4 shows the effect of the Michaelis Menten eliminiation where for the lower dose-groups, the elimination is linear since the concentration does not exceed K_m, but for the higher dose-groups the clearance reaches saturation, showing two phases of elimination.

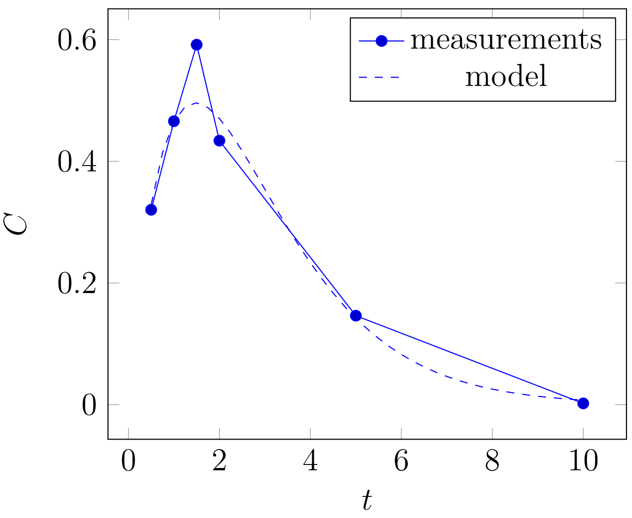
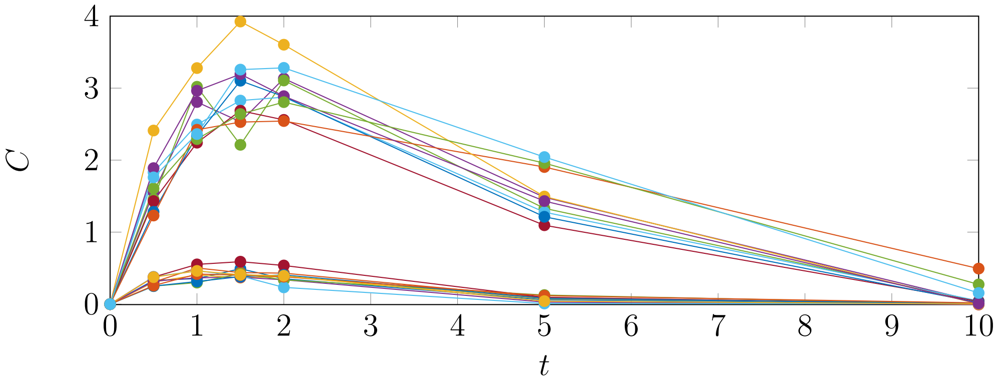
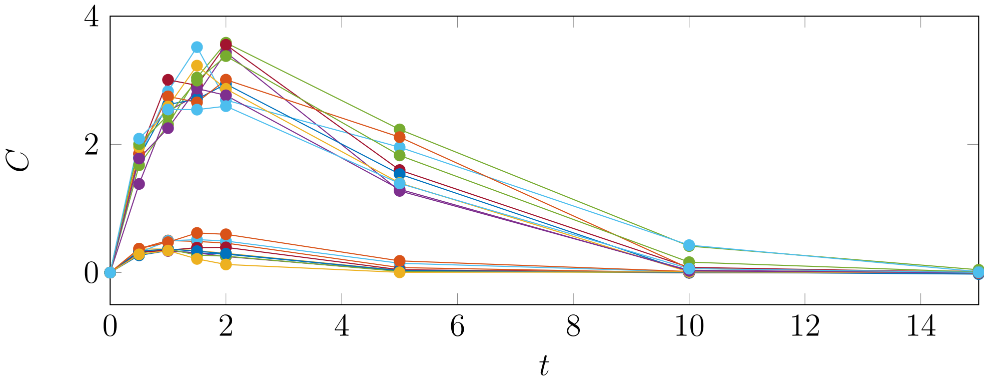


Figure 4.2: *Example of computer generated data for a single individual for model M1. The initial dose was 1. The dotted line shows the underlying model with the parameter values used for the generation of data.*

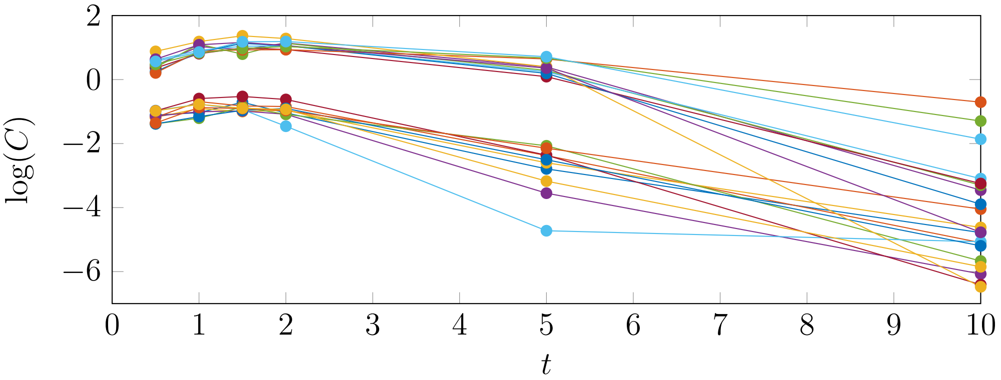


(a) *Time series showing concentration for model M1a.*

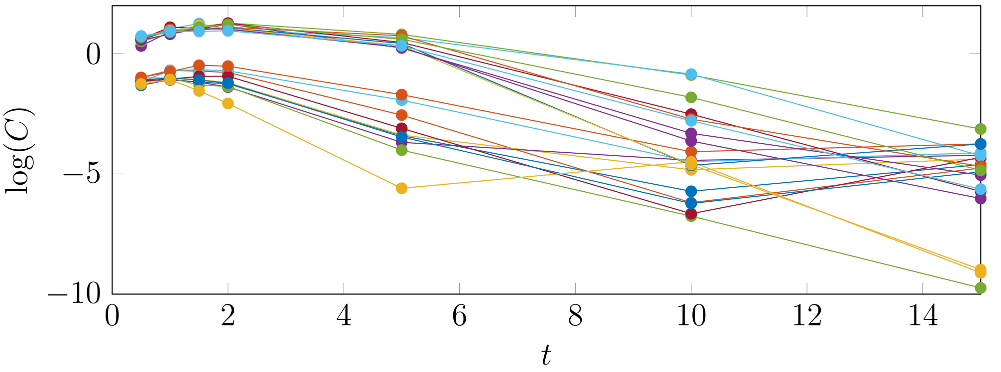


(b) *Time series showing concentration for model M1b.*

Figure 4.3: *Time series showing concentration for a computer generated dataset for model M1 with different dosing schemes.*



(a) *Time series showing log of concentration for model M1a.*



(b) *Time series showing log of concentration for model M1b.*

Figure 4.4: *Time series showing log of concentration for a computer generated dataset for model M1 with different dosing schemes. The effect of Michaelis Menten elimination can be seen for the higher dose-groups where the elimination has a different slope before and after the timepoint* $t = 5$.

4.1.3 M2 (PKPD)

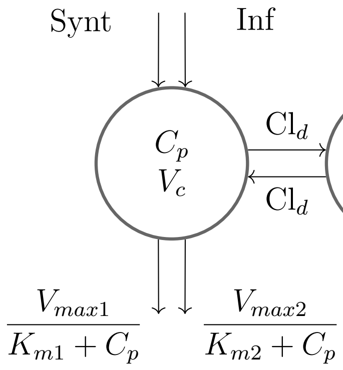


Figure 4.5: *Schematic figure of two-compartment model with two administration routes and two nonlinear elimination terms.*

This model is a pharmacokinetic-pharmacodynamic (PKPD) model of nicotinic acid intervention on fatty acid turnover in rats [22]. The measured values are the fatty acids concentration R. The PD part models the change in R. Since R depends on the unknown drug concentration, the PK part is used to model the concentration profile based on the known inputs. It consists of the drug concentrations C_p in the central compartment and C_t in the pheripheral compartment and is described with the equations

$$\begin{aligned} V_\mathrm{c}\frac{dC_\mathrm{p}(t)}{dt} &= \mathrm{Synt} - \frac{V_\mathrm{max1}}{K_\mathrm{m1}+C_\mathrm{p}(t)}C_\mathrm{p}(t) - \frac{V_\mathrm{max2}}{K_\mathrm{m2}+C_\mathrm{p}(t)}C_\mathrm{p}(t) \\ &\quad - Cl_\mathrm{d}(C_\mathrm{p}(t) - C_\mathrm{t}(t)) + \mathrm{Inf}(t) \\ V_\mathrm{t}\frac{dC_\mathrm{t}(t)}{dt} &= Cl_\mathrm{d}(C_\mathrm{p}(t) - C_\mathrm{t}(t)). \end{aligned} \tag{4.5}$$

In [22], the model also included a time dependent state variable A_g, representing the gut to simulate uptake of oral drugs in the body. To simplify the model, individuals only get drugs intravenously, allowing the removal of the state variable A_g from the model. Hence, the PK part of the model simplifies to a two compartmental model with two input rates of nicotinic acid into the central compartment, one constant (Synt) and one time dependent (Inf(t)), and two nonlinear elimination rates. A schematic figure of the pharmacokinetics part can be seen in Figure 4.5.

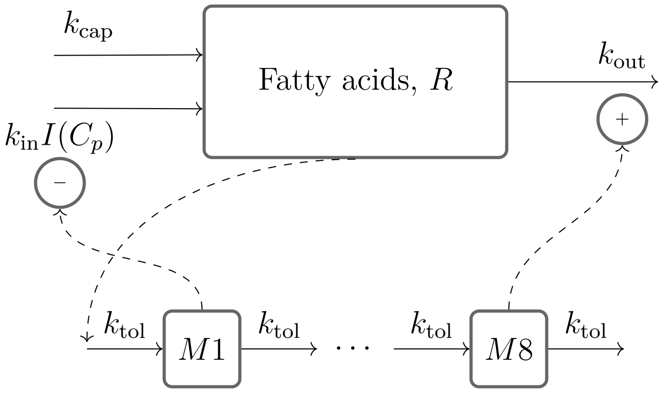


Figure 4.6: *Schematic figure of fatty acid turnover, described as a feedback model with eight moderator compartments (M1-M8). Solid lines represent fluxes and dashed lines represent control processes [22].*

The pharmacodynamic part consists of the fatty acids concentration in the measured compartment, R, and eight other so called moderator compartments, $M_1, M_2, ..., M_8$, representing a delayed feedback. A schematic figure can be seen in Figure 4.6. The PD part of the model is described with the equations

$$\begin{aligned} \frac{dR(t)}{dt} &= k_{\text{in}}\frac{1}{M_1(t)^p}I(C_{\text{p}}(t)) + k_{\text{cap}} - k_{\text{out}}R(t)M_8(t) \\ \frac{dM_1(t)}{dt} &= k_{\text{tol}}(R(t) - M_1(t)) \\ \frac{dM_i(t)}{dt} &= k_{\text{tol}}(M_{i-1}(t) - M_i(t)), \quad i = 2, 3, .., 8 \end{aligned} \tag{4.6}$$

where

$$I(C_{\text{p}}(t)) = 1 - I_{\text{max}}\frac{C_{\text{p}}(t)^\gamma}{IC_{50}^\gamma + C_{\text{p}}(t)^\gamma} \tag{4.7}$$

is the inhibitory drug function, k_{in} the turnover rate, p the amplification factor, k_{cap} the formation of fatty acids in capillaries, and k_{out} the fractional turnover rate. I_{max}, IC_{50}, and γ are the maximum drug-induced inhibitory effect, plasma concentration at 50% reduction of maximal effect (potency), and sigmoidicity factor. Finally, k_{tol} is a fractional turnover rate constant [22]. The initial conditions are

$$\begin{aligned} C_{\text{p}}(0) &= 0 \\ C_{\text{t}}(0) &= 0 \\ R(0) &= R_0 \\ M_i(0) &= R_0. \end{aligned} \tag{4.8}$$

The observations are modeled as

$$y_t = R(t) + e_t \tag{4.9}$$

with additive error

$$\begin{aligned} &e_t \overset{i.i.d.}{\sim} N(0, \Sigma) \\ &\Sigma = \sigma^2. \end{aligned} \tag{4.10}$$

Data Simulation

The synthetic data consists of 36 individuals, divided into three groups of twelve individuals, receiving infusion of 5, 10 and 51 μmol kg^{-1}, respectively, over 300 minutes. Samples are taken after 0, 20, 60, 150, 240, 300, 310, 320, 340, 360, 400, 460 and 500 minutes for every individual.

The parameters of M2 are summarised in Table A.2. An example of the computer generated data can be seen in Figure 4.7 for a single individual and in Figure 4.8 for a population of 36 individuals.

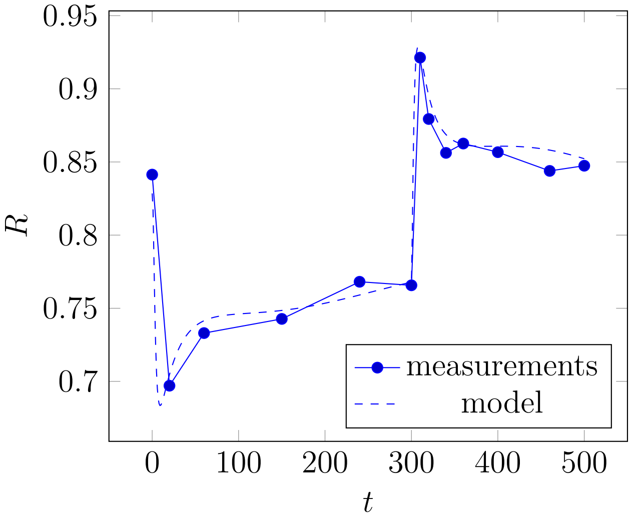


Figure 4.7: *Example of computer generated data for a single individual for model M2. The dotted line shows the underlying model with the parameter values used for the generation of data.*

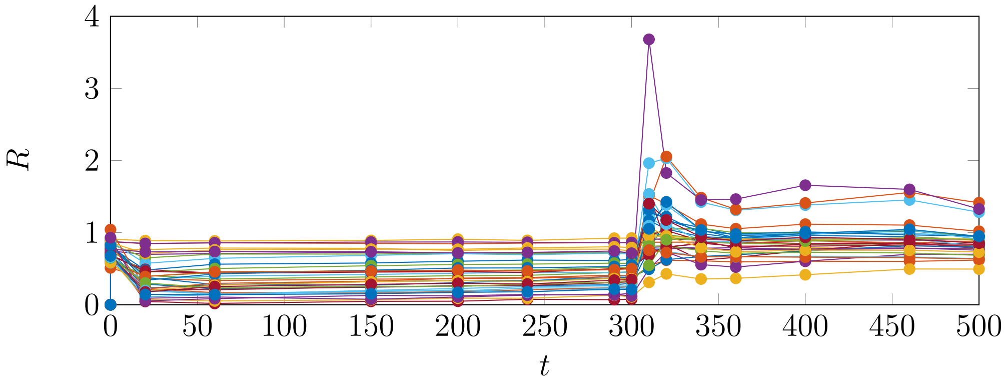


Figure 4.8: *Time series showing concentration for a computer generated dataset for model M2 with different dosing schemes.*

4.2 Robustness Comparison

The robustness of parameter estimation is investigated in two ways. One way is to run parameter estimation on various datasets of the same model to investigate general robustness. The other way is to perform parameter estimation on the same dataset multiple times using different initial values for the optimisation. The robustness of uncertainty estimates is also compared between methods. The types of analysis done for each model, along with relevant result sections are presented in Table 4.1. The comparison methods, along with the initial values and method parameters used, are described below.

Table 4.1: *Comparison types done for different models and relevant result sections.*

Model	Various datasets		Multiple	Uncertainty
	>50	<20	initial values	coverage
M1a	5.1.1			5.1.5
M1b	5.1.2		5.2	5.1.5
M2		5.1.3		

4.2.1 Initial Values for Optimisation

The optimisation during parameter estimation requires initial values of the population parameters. They are chosen by either of the two following ways. The first way is to set the initial value to the parameter values used to generate the synthetic data. These kind of initial values are referred to as *true* values. Note that although the values are called true values they do not have to be the ones that maximise the likelihood since the computer generated data is sparse and thus does not manage to describe the model with the set values exactly. The true values should, however, be close to the optimal ones. The second way is to choose initial values from a random perturbation around the true value. These kinds of initial values are referred to as *random* values. If $\boldsymbol{\theta}_m$ is the value used for generating the synthetic dataset, the initial value $\boldsymbol{\theta}_{gm}$ for each parameter $\boldsymbol{\theta}_m$ is randomised by

$$\boldsymbol{\theta}_{gm} = \boldsymbol{\theta}_m \cdot X_m \tag{4.11}$$

where $X_i \sim \ln\mathcal{N}(0, (\ln(2)/1.96)^2)$ is a log normal variable.

The perturbation of the elements of $\boldsymbol{\Omega}$ could result in a non-positive definite matrix. If that happens, the perturbation of those elements is repeated until the covariance matrix is positive definite. This approach is chosen instead of making use of the feature of the lower triangle matrix, since doing perturbation on the values of the triangular matrix results in accumulated perturbations on the elements of $\boldsymbol{\Omega}$ down the diagonal, which is considered less realistic.

4.2.2 Various Datasets

In order to investigate the general robustness of the algorithms on the models, various datasets are generated from each model by using the same parameter settings.

Parameter estimation is then performed for the different estimation methods on each dataset. Two types of initial values are chosen for each dataset to be fitted, both true and random values. The successes of individual steps are computed as well as the overall success of the estimation. For each dataset and method the results state the success of the estimation of the parameters and their uncertainties with the values 1 and 0 representing success and failure, respectively. When comparing success of two parameter estimation methods on various datasets, a pair (X_i, Y_i) of binary values is computed for every dataset i. In order to see if the success frequencies of the methods are significantly different, the two sided sign test is used, as described in Section 2.4.1. For the test, five percent level of significance is required and therefore, the value $\delta = 0.05$ is used.

4.2.3 Multiple Initial Values

The sensitivity to initial guess is investigated by performing parameter estimation on the same dataset using multiple different random initial values. Using the success results from each parameter estimation, the success frequency of each method for a dataset can be computed and reported as a real value between 0 and 1. By repeating this for multiple datasets, two parameter estimation methods are compared by creating a pair (X_i, Y_i) of the success frequency of each method for every dataset i. The significance of the difference is then estimated using the Wilcoxon signed rank test as described in Section 2.4.2, requiring five percent level of significance and using the value $\delta = 0.05$.

4.2.4 Robustness of Uncertainty Estimation

In order to compare the parameter and uncertainty estimates, confidence intervals (CIs) are computed by using the standard deviation from the covariance matrix obtained during the covariance step.

For an estimated parameter vector $\boldsymbol{\theta}^*$ with estimated covariance matrix $C = -H^{-1}$, the CI for the i-th parameter becomes the interval

$$\left[\boldsymbol{\theta}_i^* - s\sqrt{C_{ii}}, \boldsymbol{\theta}_i^* + s\sqrt{C_{ii}}\right]. \tag{4.12}$$

where $s = 1$ for 68% CI and $s = 1.96$ for 95% CI. For estimations on multiple datasets, the fraction of 68% and 95% CIs that include the true parameter value are computed for each method. This is done in order to see if the methods over- or underestimate the uncertainty of the estimated parameters.

4.2.5 Method Parameter Settings

Since the programs NONMEM and Mathematica do not have the same method-parameters, the aim is to set each program to its optimal settings. The optimal settings are found using the NONMEM guidelines and by manual testing of reasonable amount of different integer valued parameter settings (e.g. setting $NSIG = 1, 2, 3$). The method-parameters used for the different models can be seen in Table 4.2 for S-FOCE and in Table 4.3 for NM-FOCE and NM-SLOW-FOCE.

Table 4.2: *Method-parameter values used for S-FOCE for the different models*

Model	p_{out}	r_{out}	p_{inn}	r_{inn}	p_{cov}	r_{cov}
M1a	3	6	6	6	8	8
M1b	3	6	6	6	8	8
M2	3	6	6	6	6	6

Table 4.3: *Method-parameter values used for S-FOCE for the different models.*

Model	$NSIG$ NM-FOCE	$NSIG$ NM-SLOW-FOCE
M1a	2	2
M1b	2	2
M2	2	2

5

Results

5.1 Analysis of Various Datasets

For the one compartmental models M1(a-b), 100 synthetic datasets are simulated as described in Section 4.1.2. The population likelihood function is optimised using the three methods S-FOCE, NM-FOCE and NM-SLOW-FOCE. The initial values of the numerical minimisation are set in two ways; as true and random values, as described in Section 4.2.1.

5.1.1 Success Analysis of M1a of Different Datasets

Not all NLME models are hard to estimate, as can be seen in Figure 5.1 that shows the success frequency when performing parameter estimation for 100 different datasets of model M1a. This model seems to be fairly easy to estimate with the success frequency of the estimation step and the covariance step close to one for the S-FOCE methods as well as for both NONMEM methods.

Table 5.1: *Successful runs of model M1a for S-FOCE, NM-FOCE and NM-SLOW-FOCE when starting from different initial value settings.*

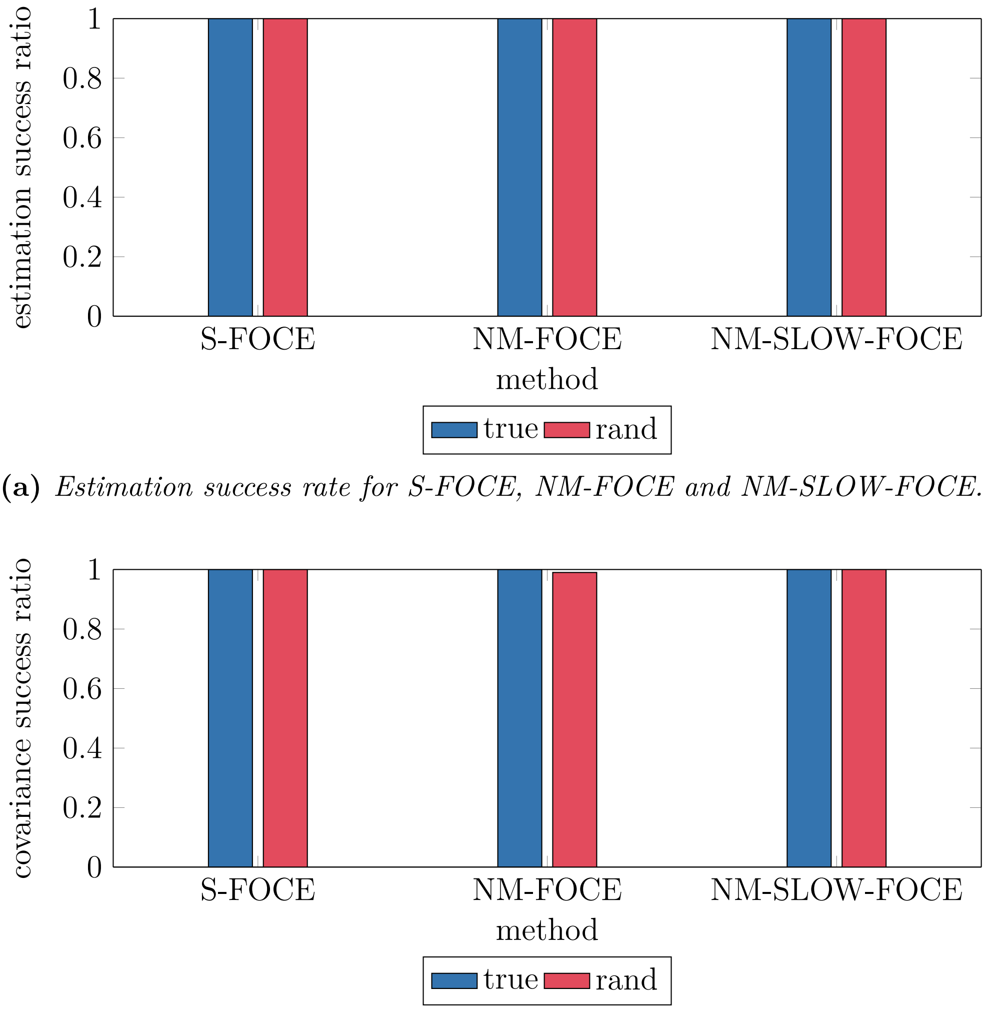
(a) *Successful runs for S-FOCE, NM-FOCE and NM-SLOW-FOCE for true starting values.*

		NM Yes		NM NO	
		NM SLOW yes	NM SLOW no	NM SLOW yes	NM SLOW no
S-FOCE	Yes	100	0	0	0
	No	0	0	0	0

(b) *Successful runs for S-FOCE, NM-FOCE and NM-SLOW-FOCE for random starting values.*

		NM Yes		NM NO	
		NM SLOW yes	NM SLOW no	NM SLOW yes	NM SLOW no
S-FOCE	Yes	98	0	1	0
	No	1	0	0	0

An estimation is considered successful if both the estimation and the covariance steps are successful. In order to compare the results of one dataset using different parameter estimation methods, Table 5.1 is generated. The table shows the division of datasets between different combination of success and failure for the three



(a) *Estimation success rate for S-FOCE, NM-FOCE and NM-SLOW-FOCE.*

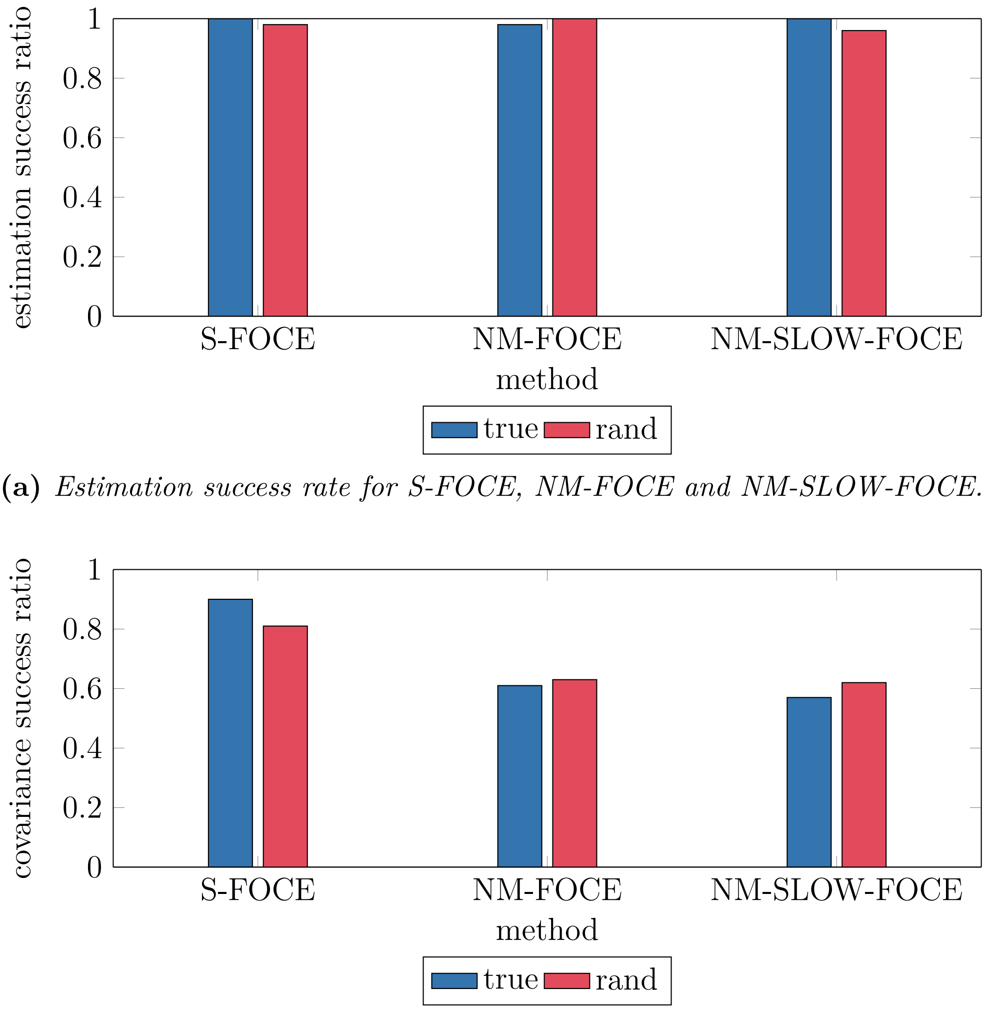
(b) *Covariance success rate for S-FOCE, NM-FOCE and NM-SLOW-FOCE.*

Figure 5.1: *Success frequency for the three methods S-FOCE, NM-FOCE and NM-SLOW-FOCE when estimating parameters for model M1a. Parameter values are estimated for 100 different computer generated datasets using both true and random initial values.*

methods. In the table, *yes* represents successful datasets while *no* represents failed datasets. The two rows of the table represent the number of datasets for which the S-FOCE algorithm was successful and unsuccessful, respectively. The success of NM-FOCE can be found by adding the values of the first two columns of the table and the success of NM-SLOW-FOCE can be found by adding the values of columns one and three. The total sum of the table is equal to the number of datasets. Table 5.1 shows that there is not a significant difference between the three methods for this model with almost perfect success frequency.

5.1.2 Success Analysis of M1b of Different Datasets

The computed success frequency of obtaining parameter estimation and in obtaining parameter uncertainties for 100 different datasets generated from the one compartmental model M1b is shown in Figure 5.2.



(a) *Estimation success rate for S-FOCE, NM-FOCE and NM-SLOW-FOCE.*

(b) *Covariance success rate for S-FOCE, NM-FOCE and NM-SLOW-FOCE.*

Figure 5.2: *Success frequency for the three methods S-FOCE, NM-FOCE and NM-SLOW-FOCE when estimating parameters for model M1b. Parameter values were estimated for 100 different computer generated datasets using both true and random initial values.*

All three methods perform similarly in obtaining the parameter values as shown in Figure 5.2a. Moreover, the choice of starting guess does not seem to affect the success. Figure 5.2b suggests that the covariance step, performs similarly for the methods NM-FOCE and NM-SLOW-FOCE but the S-FOCE is the most successful method. The starting value does not seem to have a significant effect on the success frequency.

While comparing success frequencies between methods on the same 100 datasets can give an idea on their performance, it does not show if the methods are significantly different or not. In order to evaluate the significance between two different methods, the number of runs where both, either and neither methods were successful is computed. From this information, the significance is evaluated using the sign test, described in Section 2.4.1.

The results from runs on different datasets are shown in Table 5.2. Using these results, the sign test is performed on pairs of algorithms. The p-values are shown in Table 5.3. The table shows that the null hypothesis that there is not a significant difference in success of NM-FOCE and S-FOCE should be rejected, even with one percent level of significance. The same holds for the pair NM-SLOW-FOCE and S-FOCE. However, the null hypothesis that there is no significant difference between NM-SLOW-FOCE and NM-FOCE cannot be rejected.

Table 5.2: *Successful runs of model M1b for S-FOCE, NM-FOCE and NM-SLOW-FOCE when starting fom different initial value settings.*

(a) *Successful runs for S-FOCE, NM-FOCE and NM-SLOW-FOCE for true starting values.*

		NM Yes		NM NO	
		NM SLOW yes	NM SLOW no	NM SLOW yes	NM SLOW no
S-FOCE	Yes	45	8	6	31
	No	5	2	1	2

(b) *Successful runs for S-FOCE, NM-FOCE and NM-SLOW-FOCE for random starting values.*

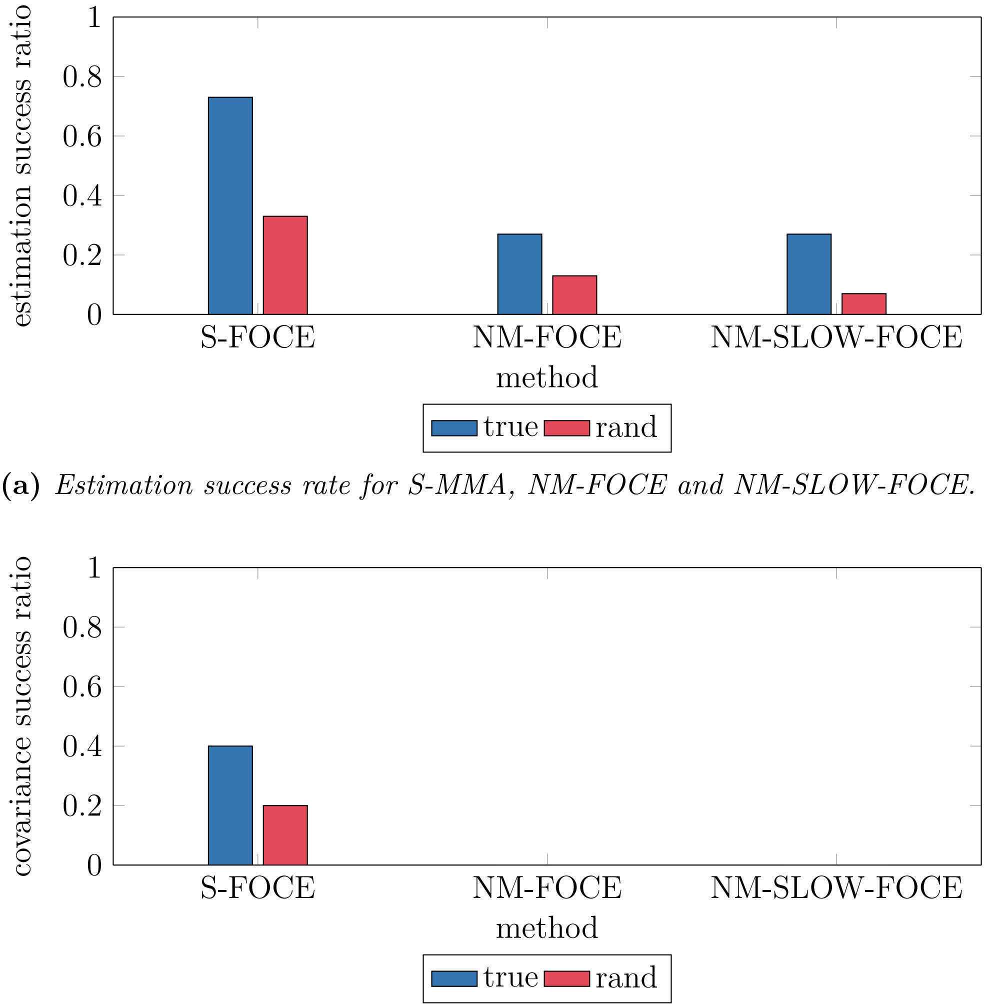
		NM Yes		NM NO	
		NM SLOW yes	NM SLOW no	NM SLOW yes	NM SLOW no
S-FOCE	Yes	41	9	9	22
	No	8	5	2	4

Table 5.3: *P-values as described with the sign test for comparison of success of different algorithms of model M1b. Two cases are considered, both when starting from true initial guess and random initial guess.*

Algorithms		p-value true	p-value rand
S-FOCE	NM-FOCE	$5.3 \cdot 10^{-6}$	$9.6 \cdot 10^{-3}$
S-FOCE	NM-SLOW-FOCE	$5.4 \cdot 10^{-7}$	$1.5 \cdot 10^{-3}$
NM-FOCE	NM-SLOW-FOCE	0.63	0.69

5.1.3 Success Analysis of M2 of Different Datasets

In order to test the methods on a more complex model, parameter estimation is performed on 15 computer generated datasets of the model M2 with true initial guesses as well as random initial guesses. Since parameter estimation for each of these datasets takes several hours it is not considered feasible to do a statistical approach using this model. The success frequency of estimate steps and covariance steps can be seen in Figure 5.3 and the results from runs of different datasets is shown in Table 5.4. For this model, only S-FOCE managed to succeed in both the estimation and the covariance step. Therefore, it is also investigated how the algorithms differ during the estimation step only. The results from estimation step for different datasets are shown in Table 5.5. There is not sufficient data to be able to reject any hypothesis about equivalent methods, except in the case of successful runs when starting from a true value. Then S-FOCE and NM-FOCE are significantly different and so are S-FOCE and NM-SLOW-FOCE. This can be seen from the p-values displayed in Table 5.6. In order to see difference for the runs starting from a random value, more runs are probably needed.



(a) *Estimation success rate for S-MMA, NM-FOCE and NM-SLOW-FOCE.*

(b) *Covariance success rate for S-MMA, NM-FOCE and NM-SLOW-FOCE.*

Figure 5.3: *Success rate for the three methods S-MMA, NM-FOCE and NM-SLOW-FOCE when estimating parameters for model M2. The estimates were found for 15 different computer generated datasets using both true and random initial values.*

When investigating when the NM methods failed during estimation, one can see that the gradient eventually increases and the algorithm terminates with large elements on the gradient and a message stating that the minimisation has been terminated due to rounding errors. Moreover, some successful estimation steps terminate with the message that the estimate should not be trusted unless the covariance step is successful, which is not the case. This is possibly due to cancellation errors when

Table 5.4: *Successful runs of model M2 for S-FOCE, NM-FOCE and NM-SLOW-FOCE when starting fom different initial value settings.*

(a) *Successful runs for S-FOCE, NM-FOCE and NM-SLOW-FOCE for true starting values.*

		NM Yes		NM NO	
		NM SLOW yes	NM SLOW no	NM SLOW yes	NM SLOW no
S-FOCE	Yes	0	0	0	6
	No	0	0	0	9

(b) *Successful runs for S-FOCE, NM-FOCE and NM-SLOW-FOCE for random starting values.*

		NM Yes		NM NO	
		NM SLOW yes	NM SLOW no	NM SLOW yes	NM SLOW no
S-FOCE	Yes	0	0	0	3
	No	0	0	0	12

Table 5.5: *Successful estimation steps of model M2 for S-FOCE, NM-FOCE and NM-SLOW-FOCE when starting fom different initial value settings.*

(a) *Successful runs for S-FOCE, NM-FOCE and NM-SLOW-FOCE for true starting values.*

		NM Yes		NM NO	
		NM SLOW yes	NM SLOW no	NM SLOW yes	NM SLOW no
S-FOCE	Yes	2	0	2	7
	No	1	2	0	1

(b) *Successful runs for S-FOCE, NM-FOCE and NM-SLOW-FOCE for random starting values.*

		NM Yes		NM NO	
		NM SLOW yes	NM SLOW no	NM SLOW yes	NM SLOW no
S-FOCE	Yes	0	0	0	5
	No	0	2	0	8

estimating gradients using finite difference approximation, suggesting even further the benefits of exact gradients.

Table 5.6: *P-values as described with the sign test for comparison of success of different algorithms of model M2. Two cases are considered, both when starting from true initial guess and random initial guess.*

(a) *Successful runs for S-FOCE, NM-FOCE and NM-SLOW-FOCE for true starting values.*

Algorithms		p-value true	p-value rand
S-FOCE	NM-FOCE	0.031	0.25
S-FOCE	NM-SLOW-FOCE	0.031	0.25
NM-FOCE	NM-SLOW-FOCE	1	1

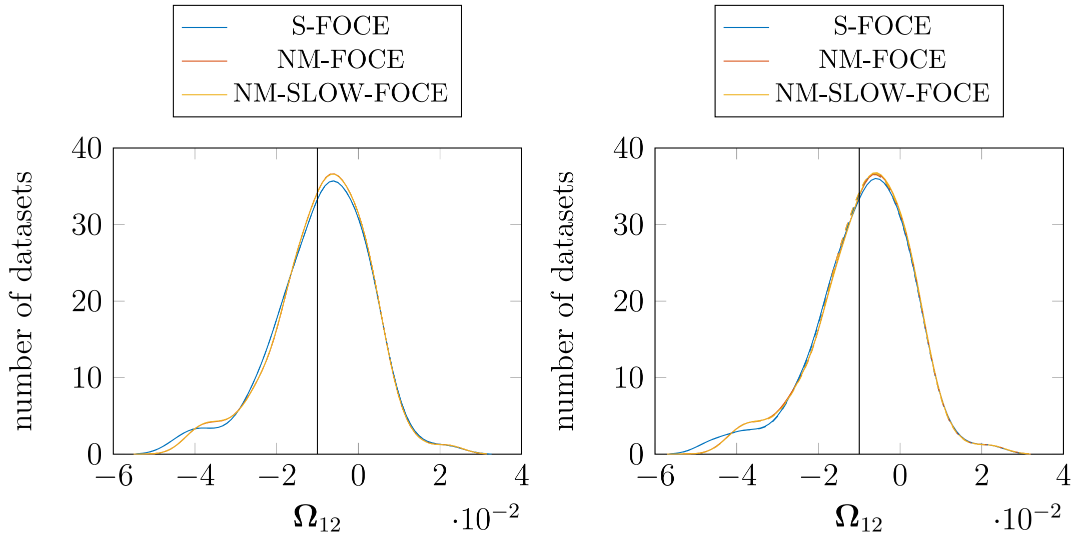
(b) *Successful estimation steps for S-FOCE, NM-FOCE and NM-SLOW-FOCE for random starting values.*

Algorithms		p-value true	p-value rand
S-FOCE	NM-FOCE	0.15	0.45
S-FOCE	NM-SLOW-FOCE	0.070	0.063
NM-FOCE	NM-SLOW-FOCE	1	0.5

5.1.4 Parameter Estimate Distribution

In order to compare how the three different methods estimate parameters for the models M1a and M1b, the distribution of the parameter estimates is plotted. Comparison is not done for M2 due to insufficient amount of found estimates. The estimated parameter values for $\boldsymbol{\Omega}_{12}$ when using the three methods S-FOCE, NM-FOCE and NM-SLOW-FOCE to estimate parameters on the 100 datasets of models M1a and M1b can be seen in Figure 5.4 and Figure 5.5. Other estimated parameter distributions can be found in Appendix B.

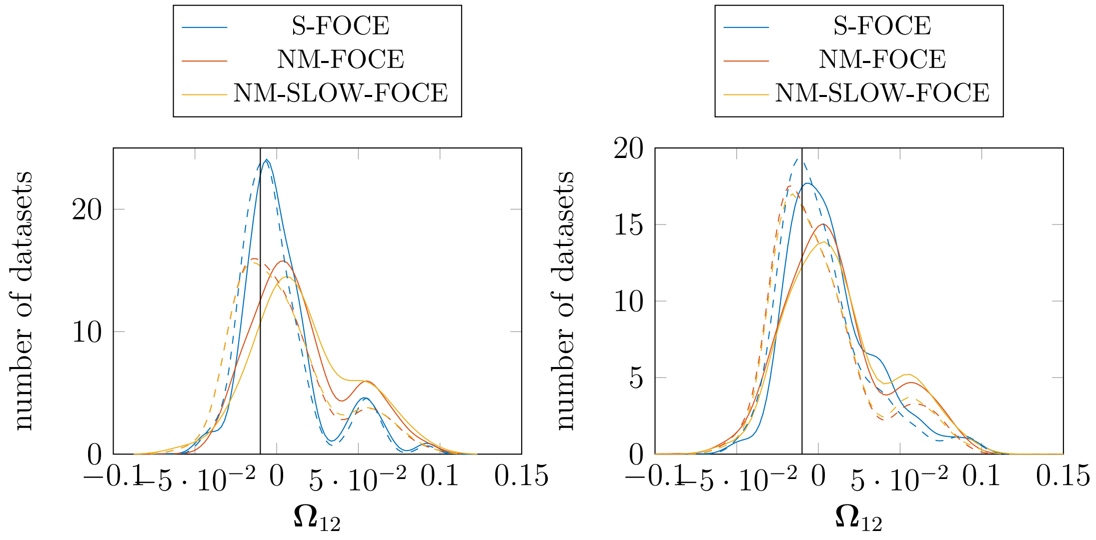
For model M1a, where all the runs are successful, the parameter estimate distributions are almost identical for the three methods. For the less successful model, M1b there is a difference in the distributions, with S-FOCE having the most focused peak around the true value. Figure 5.5a also suggests that there is more than one optimum for this parameter. For the model M1b there is one outlier successful parameter estimate when using the NM-SLOW-FOCE method and starting from random initial values. All figures show that the found estimate is close to the true value, although the might not necessarily be the optimum value as suggested with the slightly shifted distribution curve.



(a) *Estimated parameter distribution for M1a with true starting values*

(b) *Estimated parameter distribution for M1a with random starting values*

Figure 5.4: *Estimated parameter distribution for model M1a using S-FOCE, NM-FOCE and NM-SLOW-FOCE. Estimates found during a successful estimation step are dashed and the estimates that succeeded in both the estimation and the covariance step are shown with a solid line. The true value of the parameter is shown in black. There is not a distinguishable difference of estimates between the methods.*



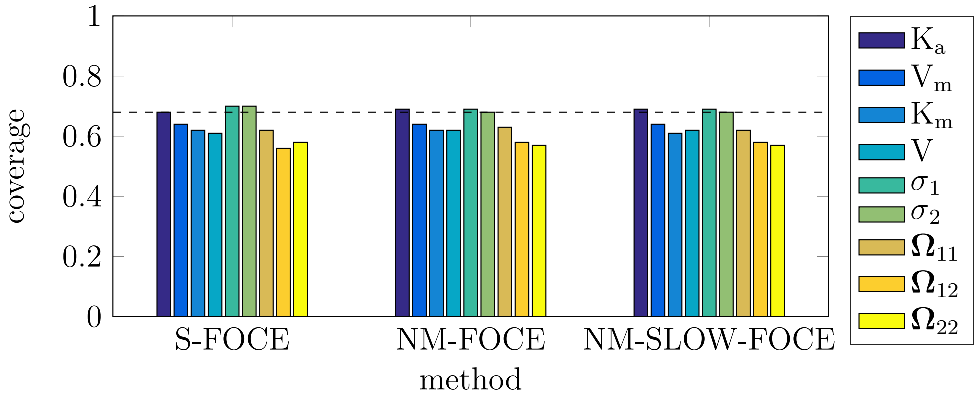
(a) *Estimated parameter distribution for M1b with true starting values*

(b) *Estimated parameter distribution for M1b with random starting values*

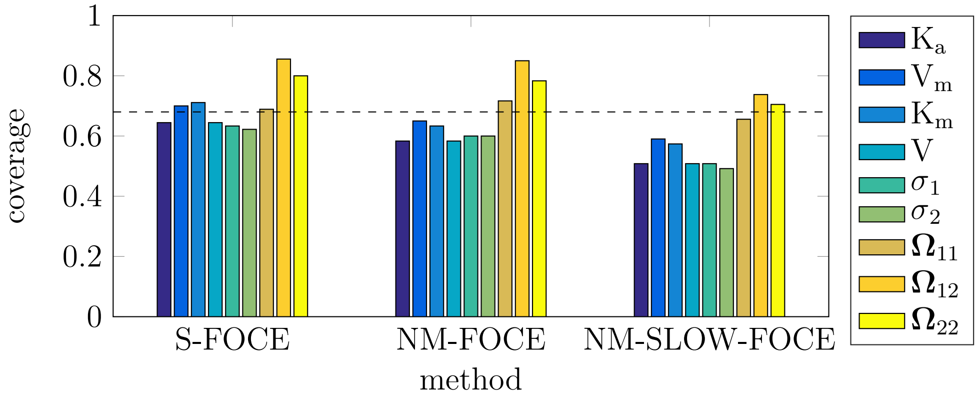
Figure 5.5: *Estimated parameter distribution for model M1a using S-FOCE, NM-FOCE and NM-SLOW-FOCE. Estimates found during a successful estimation step are dashed and the estimates that succeeded in both the estimation and the covariance step are shown with a solid line. The true value of the parameter is shown in black. Note that S-FOCE is more focused around the true value and seems to distinguish a local value in the left figure.*

5.1.5 Evaluation of Uncertainty Estimation

The coverage frequency of true values when using 68% estimated confidence interval for both M1a and M1b is shown in Figure 5.6 for true initial values and Figure 5.7 for random initial values.



(a) *Coverage of estimated 68% confidence intervals for model M1a*

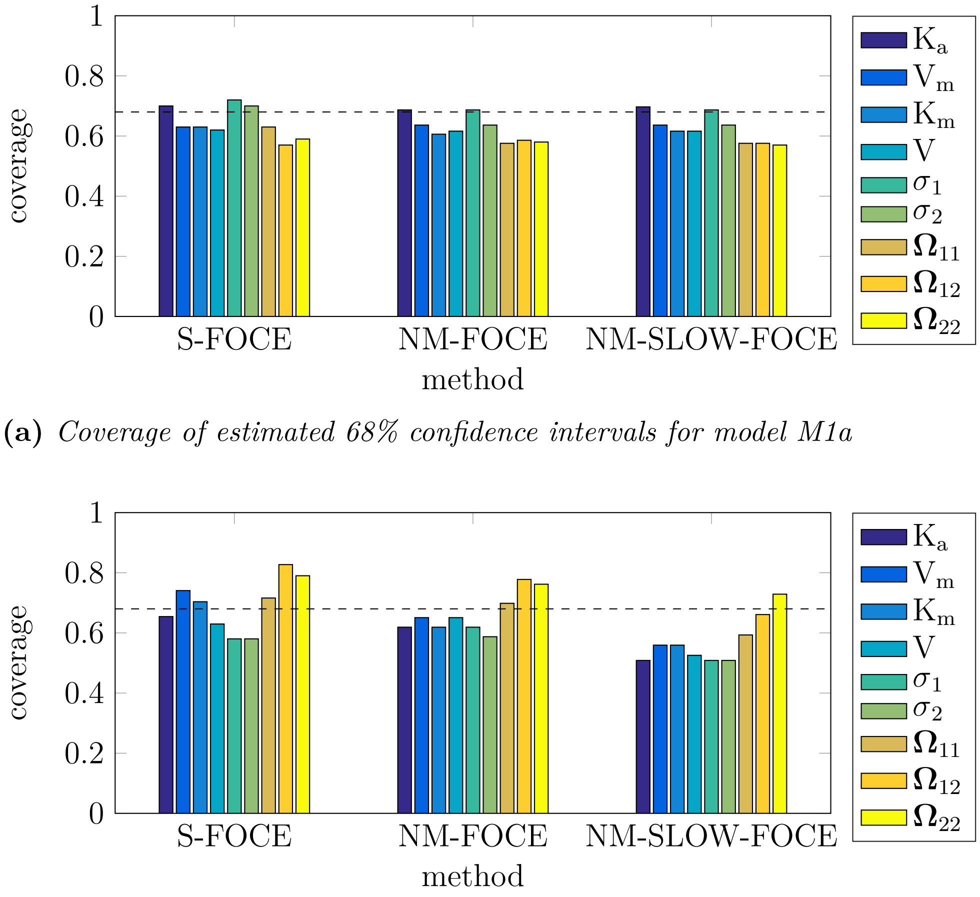


(b) *Coverage of estimated 68% confidence intervals for model M1b*

Figure 5.6: *Coverage of estimated 68% confidence intervals for models M1a and M1b using true initial values. The expected coverage is shown with a dashed line. All three methods have coverages close to the expected coverage. For M1b, the NM-SLOW-FOCE seems to be underestimating the uncertainties.*

The coverage is rather consistent for every parameter for all three methods. There cannot be seen any differences between the different kinds of initial values for model M1a. However, for model M1b, the NM-SLOW-FOCE seems to be underestimating the uncertainties. We also see that the overestimation of uncertainties for the $\mathbf{\Omega}$ values on M1b is slightly less for both S-FOCE and NM-FOCE for the random initial values.

The coverage of the 95% CIs, shown in Figure 5.8 was similar to the coverage of the 68% CIs. The conclusion is therefore that all three methods evaluate their uncertainty of parameters in a reasonable manner and that the transformation for

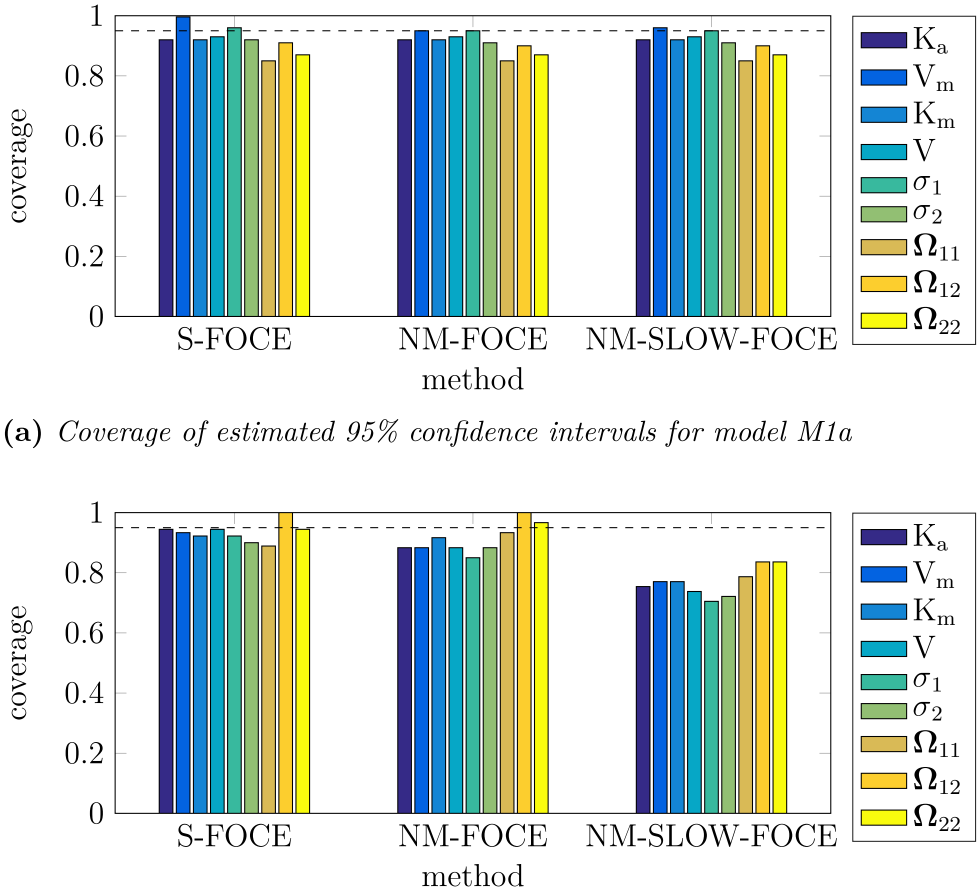


(a) *Coverage of estimated 68% confidence intervals for model M1a*

(b) *Coverage of estimated 68% confidence intervals for model M1b*

Figure 5.7: *Coverage of estimated 68% confidence intervals for models M1a and M1b using random initial values. The expected coverage is shown with a dashed line. The coverage is generally slightly lower than for true initial values.*

constraining the fixed population parameters done in the S-FOCE implementation (Section 3.3.2) does not affect the quality of the uncertainties.



(a) *Coverage of estimated 95% confidence intervals for model M1a*

(b) *Coverage of estimated 95% confidence intervals for model M1b*

Figure 5.8: *Coverage of estimated 95% confidence intervals for models M1a and M1b using true initial values. The expected coverage is shown with a dashed line. All three methods have coverages close to the expected coverage. For M1b, the NM-SLOW-FOCE seems to be underestimating the uncertainties.*

5.2 Analysis of Multiple Initial Values

For the one compartmental model M1b, 30 synthetic datasets are computed and estimated parameters from them using 30 different random initial guesses for each dataset. From this, the success frequency, f, is computed for each dataset, stating the frequency of runs that are successful for both the estimation and the covariance step. The distribution of success frequency can be seen in Figure 5.9.

In order to evaluate the difference of success frequency between the methods, the difference of success frequency for each dataset is considered. Figure 5.10 shows the difference for the three pairs of methods that are compared. The method has almost always higher success frequency than both of the methods NM-FOCE and NM-SLOW-FOCE. The difference of success frequency between NM-FOCE and NM-SLOW-FOCE is not as distinct for one method. However, NM-FOCE seems slightly better on average. The significance of difference between these three methods can

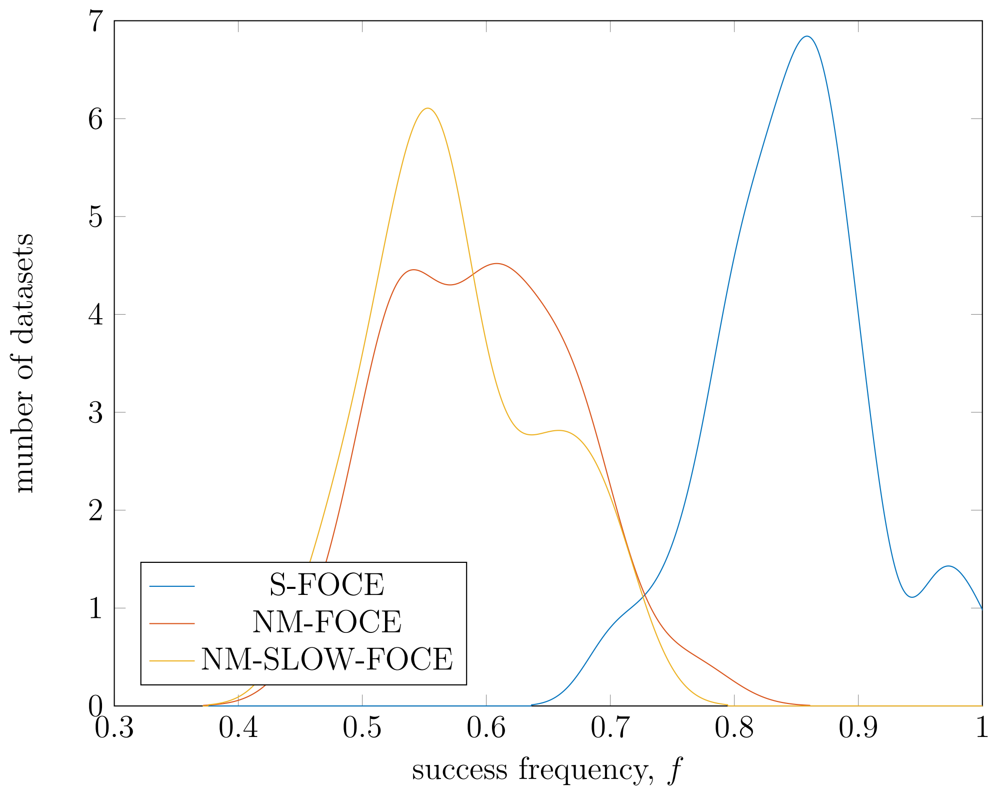


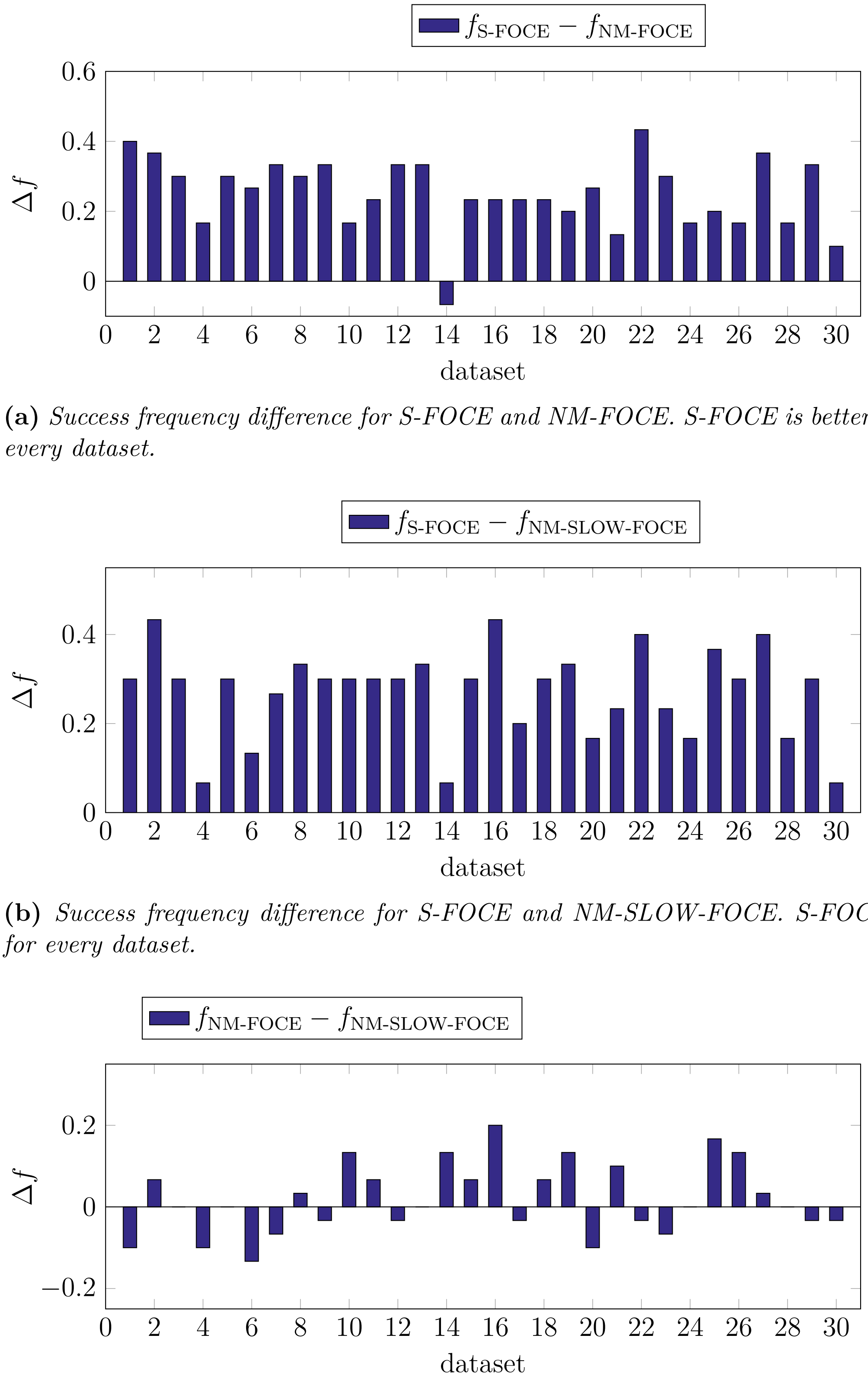
Figure 5.9: *Success frequency distribution for the three for the three methods S-FOCE, NM-FOCE and NM-SLOW-FOCE when estimating parameters for model M1b. Parameter values were found for 30 different computer generated datasets each starting from 30 random initial values.*

be evaluated using the Wilcoxon rank test, as described in Section 2.4.2. The results can be seen in Table 5.7, suggesting that the null hypothesis, that there is not a significant difference in success, should be rejected when comparing the algorithm S-FOCE to NM-FOCE and S-FOCE to NM-SLOW-FOCE. However, the null hypothesis cannot be rejected for comparison of NM-FOCE to NM-SLOW-FOCE.

Taking the mean of the success frequency yields similar results as in Section 5.1.2, as expected. These results also show that there does not seem to be any specific types of datasets of this model that some methods handle significantly better than other.

Table 5.7: *P-values as described with the Wilcoxon signed rank test for comparison of success of different algorithms of model M1b. Two cases are considered, both when starting from true initial guess and random initial guess. The p-values were found using a z-table in [20].*

Algorithms		T_+	T_-	n	p-value
S-FOCE	NM-FOCE	464	1	30	$< 10^{-3}$
S-FOCE	NM-SLOW-FOCE	465	0	30	$< 10^{-3}$
NM-FOCE	NM-SLOW-FOCE	204.5	120.5	25	0.26



(a) *Success frequency difference for S-FOCE and NM-FOCE. S-FOCE is better for almost every dataset.*

(b) *Success frequency difference for S-FOCE and NM-SLOW-FOCE. S-FOCE is better for every dataset.*

(c) *Success frequency difference for NM-FOCE and NM-SLOW-FOCE. The methods do not differ significantly, but overall, the NM-FOCE seems slightly more successful.*

Figure 5.10: *Success frequency difference for pairs of parameter estimation methods on model M1b. Parameter values were found for 30 different computer generated datasets each starting from 30 random initial values.*

5.3 Population Likelihood Gradients for NLMEs based on SDEs

For a general NLME model based on SDEs, the exact gradient of the APL can be computed in a similar way as for the a NLME model based on ODEs. These results have already been presented in Section 2.2.4. However, doing this for a specific model could yield a number of simplifications. Therefore, it could be feasible to at least partly do the computations for the specific model instead of a general model.

6
Conclusion and Future Work

This chapter gathers the main conclusion from this study and discusses potential future work.

6.1 Conclusion

When estimating parameters for NLME models, the method used has to be robust. This means that the method has to be able to find a parameter estimate and also uncertainties of the estimate with low probability of failure. The robustness was measured both in terms of success frequency for parameter estimates on multiple datasets of the same model as well as sensitivity to the initial values of the optimisation. The results suggest significantly more robust parameter estimation when using S-FOCE compared to the two finite difference approaches NM-FOCE and NM-SLOW-FOCE in NONMEM.

For this project, models with a low number of state variables and population parameters were chosen in order to have computational times that would allow multiple runs for statistical analysis of the results. This in turn led to a challenge in finding models that were possible to estimate without being too simple. However, the models that were easy in the sense that the parameter estimation was almost always successful served as a control that the methods worked in a similar way and not randomly. Comparison of the distribution of parameter estimates between methods for the slightly less successful simple model showed that the S-FOCE method manages to get closer to the optimum than NONMEM without failing in the covariance step.

When judging the quality of the uncertainty estimates by looking at the coverage frequency of true values for confidence intervals, the parameters for all methods were consistent and close to the expected coverage. However, for model M1b, the NM-SLOW-FOCE seemed to slightly underestimate the uncertainty of parameters.

The model M2 proved to be difficult without being impossible and suggested a difference between NM-FOCE and NM-SLOW-FOCE. Moreover, when estimating parameters for the M2 model using NONMEM one can observe cases where the estimation step fails with large elements in the gradient. This might imply that the exact gradient method would be able to improve the success of parameter estimation on that model.

The theoretical derivations of sensitivities needed for the computation of the exact gradient of the APL for NLME models based on SDEs showed that this can be done for a general model. The S-FOCE method could therefore be implemented in a similar way for NLME models based on SDEs as it was done for NLME models based on ODEs. However, it might be feasible in practice to simplify the calculations from the start by considering a particular model of interest.

6.2 Future Work

A thorough statistical analysis of model M2 did not fall within the scope of this project due to long execution times for parameter estimation on a single dataset. The results obtained from the model seem however promising for supporting the benefits of exact gradient computing for parameter estimation of NLME models. Future work could involve a statistical analysis of the model, which could further support the notion that some failure of parameter estimation is indeed directly connected to the way of gradient computing during optimisation. To make sure that the choice of benchmark models is representative, the analysis should also be done on at least a third model, preferably a model chosen from literature that has been proven to be difficult to solve.

For being able to do such long computations, the code running the S-FOCE method should also be optimised further, parallelising code and optimising memory use. This could either be done within Mathematica or using other faster programming languages.

For the NLME models based on stochastic differential equations, the results can be analysed further for practical purposes in terms of the order and overhead of multiplications of matrices. It is possible that the general approach is not practical and the parameter estimation would benefit from initial simplifications based on the type of model used.

Bibliography

[1] Joachim Almquist, Loubna Bendrioua, Caroline B. Adiels, Mattias Goksör, Stefan Hohmann, and Mats Jirstrand. A nonlinear mixed effects approach for modeling the cell-to-cell variability of mig1 dynamics in yeast. *PLoS ONE*, 10, 2015.

[2] Joachim Almquist, Jacob Leander, and Mats Jirstrand. Using sensitivity equations for computing gradients of the foce and focei approximations to the population likelihood. *Journal of Pharmacokinetics and Pharmacodynamics*, 42(3):191–209, 06 2015.

[3] Robert J. Bauer. *NONMEM USERS GUIDE - Part V.* ICON Development Solutions, Hanover, MD 21076, 2013.

[4] Robert J. Bauer. *NONMEM USERS GUIDE INTRODUCTION TO NONMEM 7.3.0.* ICON Development Solutions, Hanover, MD 21076, 2014.

[5] Gregory W. Corder and Dale I. Foreman. *Nonparametric statistics: a step-by-step approach.* John Wiley & Sons, Hoboken, New Jersey, second;2;2nd; edition, 2014.

[6] Hartmut Derendorf and Bernd Meibohm. Modeling of pharmacokinetic/pharmacodynamic (pk/pd) relationships: Concepts and perspectives. *Pharmaceutical research*, 16(2):176–85, 02 1999.

[7] Ernesto Aranda Enrique Castillo, Antonio J. Conejo. Sensitivity analysis in calculus of variations. some applications. *SIAM Review*, 50(2):294–312, 2008.

[8] Johan Gabrielsson and Daniel Weiner. *Pharmacokinetic & pharmacodynamic data analysis: concepts and applications.* Swedish Academy of Pharmaceutical Sciences, Stockholm, 4., rev. and expand edition, 2006.

[9] Leonid Gibiansky, Ekaterina Gibiansky, and Robert Bauer. Comparison of nonmem 7.2 estimation methods and parallel processing efficiency on a target-mediated drug disposition model. *Journal of Pharmacokinetics and Pharmacodynamics*, 39(1):17–35, 2012.

[10] Markus Karlsson, David L. Janzen, Lucia Durrieu, Alejandro Colman-Lerner, Maria C. Kjellsson, and Gunnar Cedersund. Nonlinear mixed-effects modelling for single cell estimation: when, why, and how to use it. *BMC Systems Biology*, 9, 2015.

[11] James Keener and James Sneyd. *Mathematical Physiology: I: Cellular Physiology*, volume 8/1. Springer New York, 2009.

[12] Jacob Leander, Joachim Almquist, Christine Ahlström, Johan Gabrielsson, and Mats Jirstrand. Mixed effects modeling using stochastic differential equations: Illustrated by pharmacokinetic data of nicotinic acid in obese zucker rats. *AAPS Journal*, 17:586–596, 2015.

[13] Douglas M. Bates Mary J. Lindstrom. Nonlinear mixed effects models for repeated measures data. *Biometrics*, 46(3):673–687, 1990.

[14] Leonor Michaelis, Maud L. Menten, Kenneth A. Johnson, and Roger S. Goody. The original michaelis constant: translation of the 1913 michaelis-menten paper. *Biochemistry*, 50(39):8264–8269, 2011.

[15] Kevin P. Murphy. *Adaptive Computation and Machine Learning: Machine Learning : A Probabilistic Perspective.* The MIT Press, 2012.

[16] Jorge Nocedal and Stephen J. Wright. *Numerical optimization.* Springer, New York, 2nd edition, 2006.

[17] Rune V. Overgaard, Niclas Jonsson, W. Tornøe, Christoffer, and Henrik Madsen. Non-linear mixed-effects models with stochastic differential equations: Implementation of an estimation algorithm. *Journal of Pharmacokinetics and Pharmacodynamics*, 32(1):85–107, 02 2005.

[18] William DM Paton and James Patrick Payne. *Pharmacological principles and practice.* Little, Brown, 1968.

[19] Yudi Pawitan. *In all likelihood: statistical modelling and inference using likelihood.* Clarendon, Oxford, 2001;2013;2014;.

[20] Sheldon M. Ross. *Introduction to probability and statistics for engineers and scientists.* Harcourt/Academic, San Diego, Calif, 2. edition, 2000.

[21] S. M. Taheri and G. Hesamian. A generalization of the wilcoxon signed-rank test and its applications. *Statistical Papers*, 54(2):457–470, 05 2013.

[22] Sofia Tapani, Joachim Almquist, Jacob Leander, Christine Ahlström, Lambertus A. Peletier, Mats Jirstrand, and Johan Gabrielsson. Joint feedback analysis modeling of nonesterified fatty acids in obese zucker rats and normal sprague-dawley rats after different routes of administration of nicotinic acid. *Journal of Pharmaceutical Sciences*, 103:2571–2584, 2014.

[23] RN Upton and DR Mould. Basic concepts in population modeling, simulation, and model-based drug development: Part 3-introduction to pharmacodynamic modeling methods. *CPT: Pharmacometrics & Systems Pharmacology*, 3(1):1–16, 2014.

[24] Yaning Wang. Derivation of various nonmem estimation methods. *Journal of Pharmacokinetics and Pharmacodynamics*, 34(5):575–593, 2007.

[25] Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945.

[26] Inc. Wolfram Research. *Mathematica.* Wolfram Research, Inc., Champaign, Illinois, version 10.3 edition, 2015.

A

Model Parameters

Table A.1: *M1 parameters and their interindividual variability (IIV)*

Parameter	Unit	Value	IIV[1] M1a	IIV[1] M1b
k_A	t^{-1}	1	-	-
V_m	t^{-1}	1	20	20
K_m	L^{-1}	1	-	20
V	L	1	20	-
σ_1	-	0.01	-	-
σ_2	-	0.1	-	-
$\mathrm{Corr}(V_\mathrm{m}, K_\mathrm{m})$	-	-0.25	0	0

[1] IIV measured as $\sqrt{\omega^2} \cdot 100\%$

Table A.2: *M2 parameters and their interindividual variability (IIV)*

Parameter	Unit	Value	IIV[1]
V^*_{max1}	μmol min^{-1} kg^{-1}	0.0871	92.7
K^*_{max1}	μmol L^{-1}	0.235	-
V^*_{max2}	μmol min^{-1} kg^{-1}	7.09	29.1
K^*_{max2}	μmol L^{-1}	74.5	-
V^*_{c}	L kg^{-1}	0.393	-
V^*_{t}	L kg^{-1}	0.172	-
Cl^*_{d}	L min^{-1} kg^{-1}	0.000852	-
Synt*	μmol min^{-1} kg^{-1}	0.00355	109
R_0	mmol L^{-1}	0.741	19.7
k_{out}	L mmol^{-1} min^{-1}	0.290	71.4
k_{tol}	min^{-1}	0.0245	-
k_{cap}	mmol L^{-1} min^{-1}	0.0245	-
$k^*_{\mathrm{in}} = (k_{\mathrm{out}}R_0^2 - k_{\mathrm{cap}})R_0^p$	mmol L^{-1} min^{-1}	0.0940	-
p	-	1.20	-
$\mathrm{I}^*_{\mathrm{max}}$	-	1	-
IC_{50}	μmol L^{-1}	0.0820	129
γ	-	2.16	-
σ	-	0.0110	-
$\mathrm{Corr}(R_0, k_{\mathrm{out}})$	-	0	-
$\mathrm{Corr}(R_0, \mathrm{IC}_{50})$	-	0	-
$\mathrm{Corr}(k_{\mathrm{out}}, \mathrm{IC}_{50})$	-	0	-

[1] IIV measured as $\sqrt{\omega^2} \cdot 100\%$

* not estimated

B

Parameter Estimates for M1

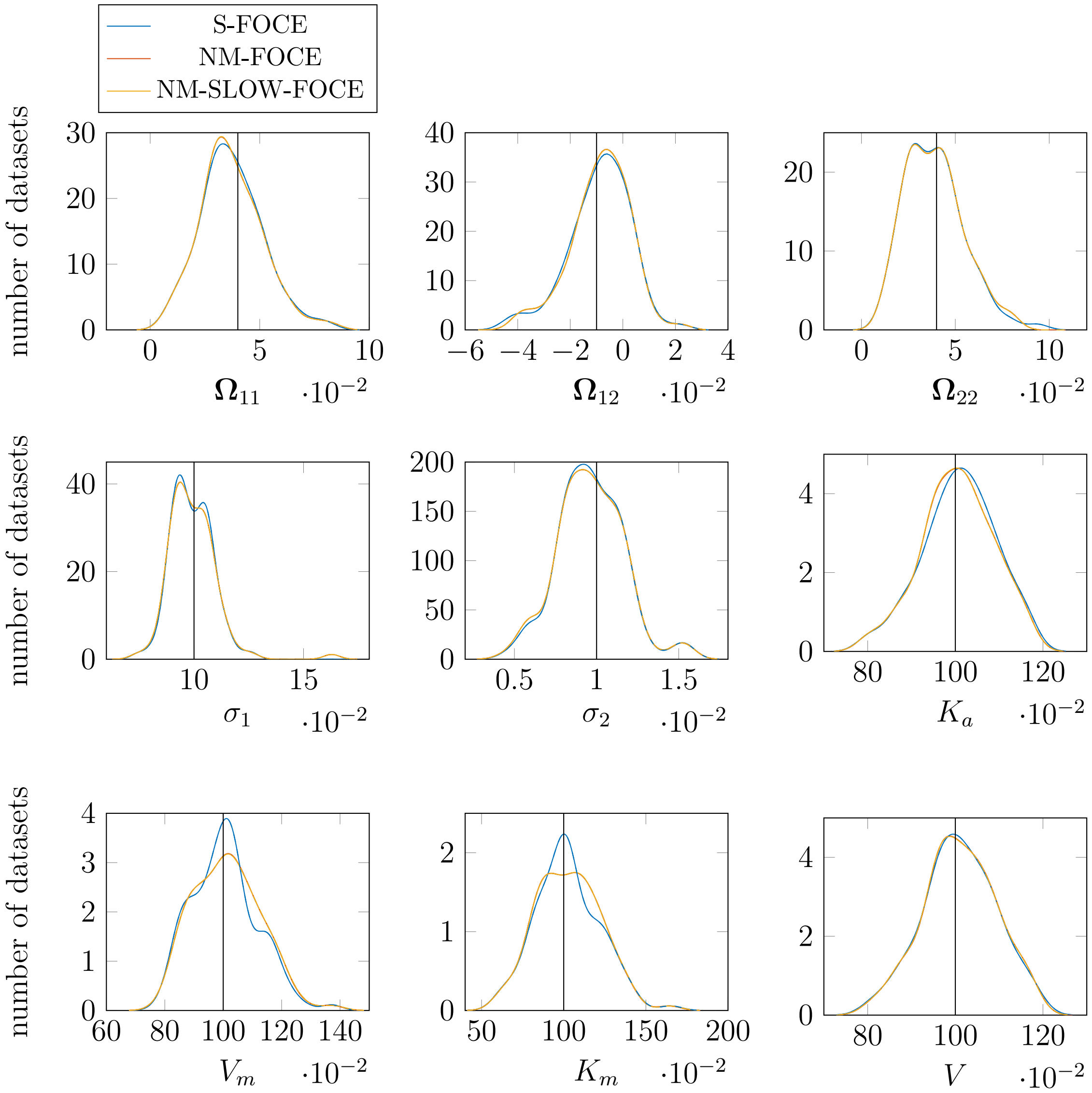


Figure B.1: *Estimated parameter distribution for M1a using S-FOCE, NM-FOCE and NM-SLOW-FOCE starting with true starting values found during a successful estimation step (dashed) and during both successful estimation and covariance steps (solid). The true value is shown in black.*

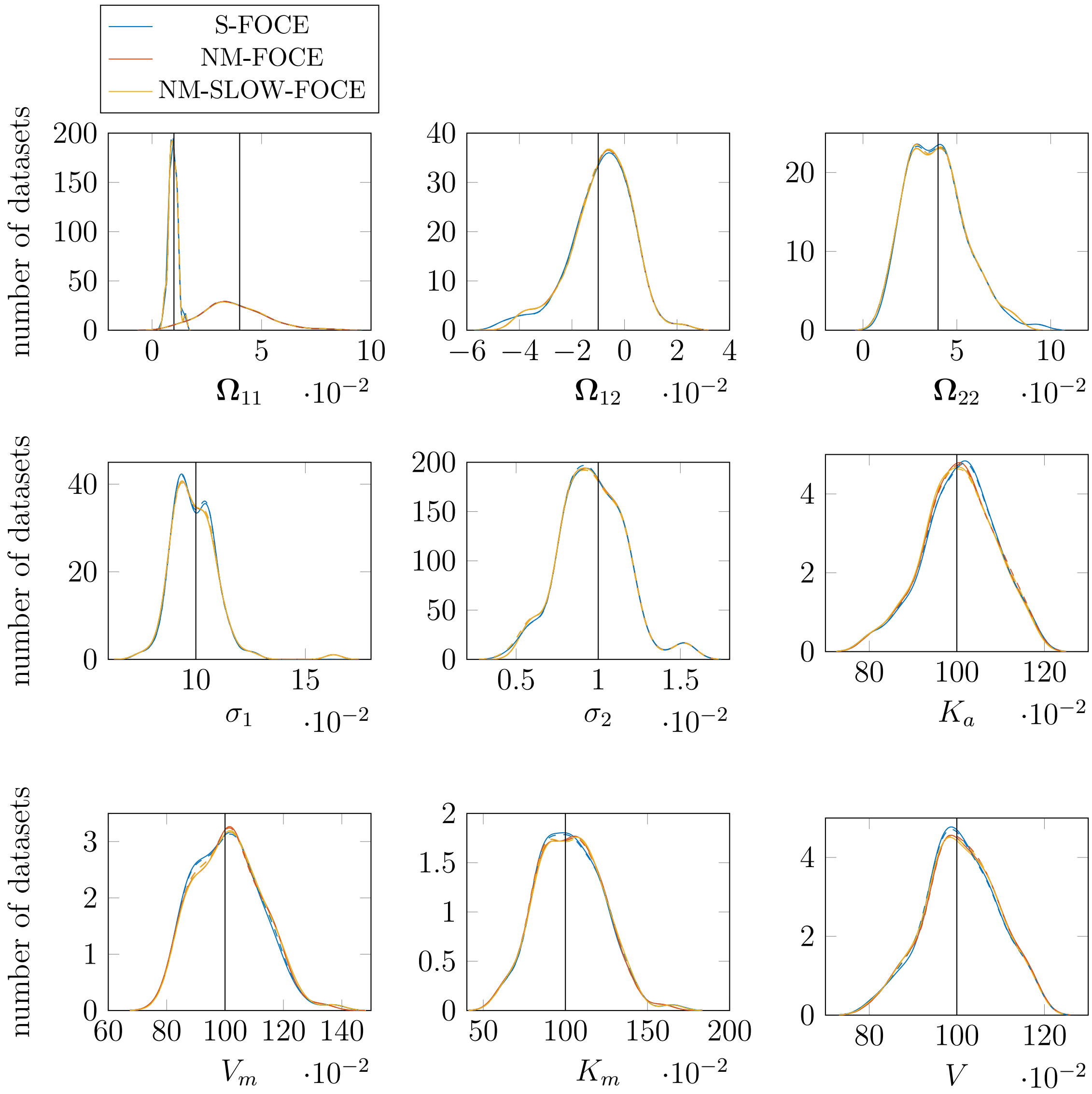


Figure B.2: *Estimated parameter distribution for M1a using S-FOCE, NM-FOCE and NM-SLOW-FOCE starting with random starting values found during a successful estimation step (dashed) and during both successful estimation and covariance steps (solid). The true value is shown in black.*

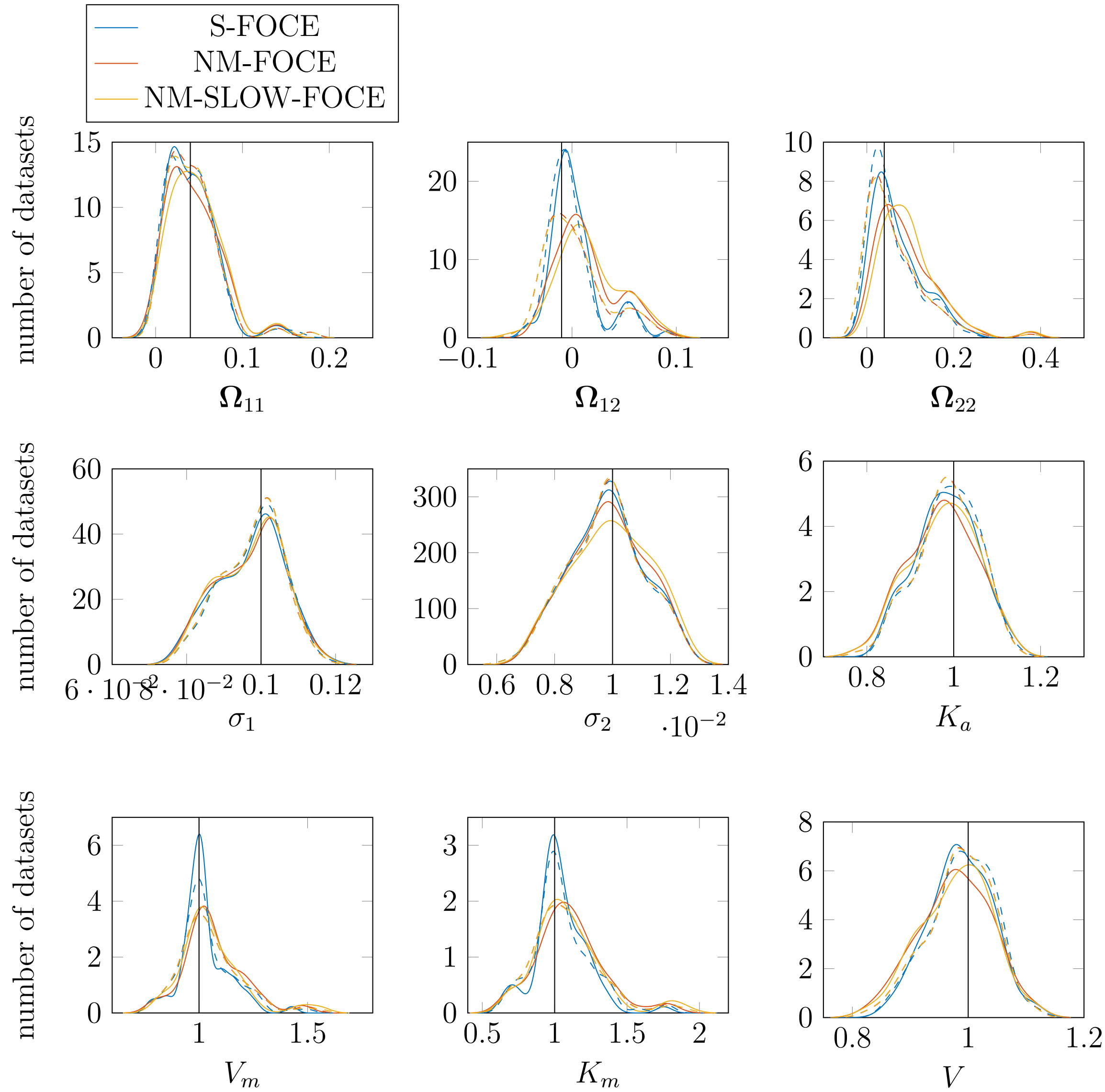


Figure B.3: *Estimated parameter distribution for M1b using S-FOCE, NM-FOCE and NM-SLOW-FOCE starting with true starting values found during a successful estimation step (dashed) and during both successful estimation and covariance steps (solid). The true value is shown in black.*

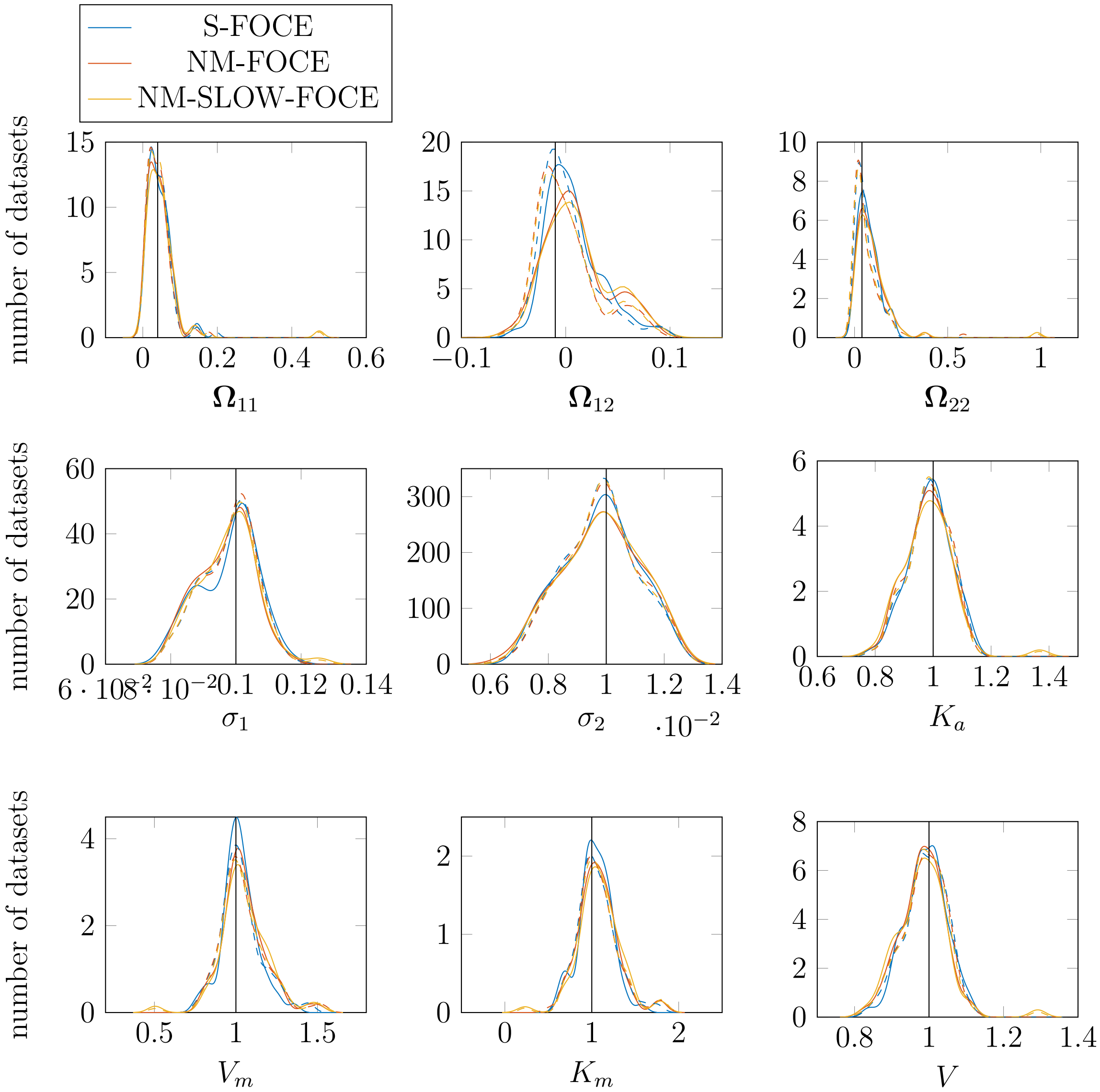


Figure B.4: *Estimated parameter distribution for M1b using S-FOCE, NM-FOCE and NM-SLOW-FOCE starting with random starting values found during a successful estimation step (dashed) and during both successful estimation and covariance steps (solid). The true value is shown in black.*

C

Derivation of EKF Sensitivities

C.1 First Order Sensitivities

We need to determine $d\boldsymbol{\epsilon}_{ik}/d\boldsymbol{\eta}_{ij}$ and $d\boldsymbol{R}_{i(k|k-1)}/d\boldsymbol{\eta}_{ij}$. Using the chain rule we get:

$$\begin{aligned}\frac{d\boldsymbol{\epsilon}_{ik}}{d\boldsymbol{\eta}_{ij}} &= \frac{d(\boldsymbol{y}_{ik} - \hat{\boldsymbol{y}}_{i(k|k-1)})}{d\boldsymbol{\eta}_{ij}} \\ &= -\frac{d\hat{\boldsymbol{y}}_{i(k|k-1)}}{d\boldsymbol{\eta}_{ij}} \\ &= -\left(\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{\eta}_{ij}} + \frac{\partial \boldsymbol{h}}{\partial \hat{\boldsymbol{x}}_{i(k|k-1)}} \frac{d\hat{\boldsymbol{x}}_{i(k|k-1)}}{d\boldsymbol{\eta}_{ij}}\right)\end{aligned} \tag{C.1}$$

and

$$\frac{d\boldsymbol{R}_{ik}}{d\boldsymbol{\eta}_{ij}} = \frac{\partial \boldsymbol{R}_{i(k|k-1)}}{\partial \boldsymbol{\eta}_{ij}} + \frac{\partial \boldsymbol{R}_{i(k|k-1)}}{\partial \hat{\boldsymbol{x}}_{i(k|k-1)}} \frac{d\hat{\boldsymbol{x}}_{i(k|k-1)}}{d\boldsymbol{\eta}_{ij}} \tag{C.2}$$

The derivatives

$$\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{\eta}_{ij}}, \text{ and } \frac{\partial \boldsymbol{h}}{\partial \hat{\boldsymbol{x}}_{i(k|k-1)}} \tag{C.3}$$

can be found directly from the definition of $\boldsymbol{h}$. However, we need to find the derivatives

$$\frac{d\hat{\boldsymbol{x}}_{i(k|k-1)}}{d\boldsymbol{\eta}_{ij}}, \frac{\partial \boldsymbol{R}_{i(k|k-1)}}{\partial \boldsymbol{\eta}_{ij}}, \frac{\partial \boldsymbol{R}_{i(k|k-1)}}{\partial \hat{\boldsymbol{x}}_{i(k|k-1)}} \tag{C.4}$$

In further calculations, the individual index i will be supressed.

C.1.1 Differentiating Predicted Expected State Variables

Let us start by considering $\frac{d\hat{\boldsymbol{x}}_{i(k|k-1)}}{d\boldsymbol{\eta}_j}$. This derivative can be obtained by solving the sensitivity equation

$$\begin{aligned}\frac{d}{dt}\left(\frac{d\hat{\boldsymbol{x}}_{t|k}}{d\boldsymbol{\eta}_j}\right) &= \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\eta}_j} + \frac{\partial \boldsymbol{f}}{\partial \hat{\boldsymbol{x}}_{t|k}} \frac{d\hat{\boldsymbol{x}}_{t|k}}{d\boldsymbol{\eta}_j}, \qquad t \in [t_k, t_{k+1}] \\ \frac{d\hat{\boldsymbol{x}}_{t|k}}{d\boldsymbol{\eta}_j}(t_k) &= \frac{d\hat{\boldsymbol{x}}_{k|k}}{d\boldsymbol{\eta}_j}\end{aligned} \tag{C.5}$$

where

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k \boldsymbol{\epsilon}_k \tag{C.6}$$

and thus

$$\frac{d\hat{\boldsymbol{x}}_{k|k}}{d\boldsymbol{\eta}_j} = \frac{d\hat{\boldsymbol{x}}_{k|k-1}}{d\boldsymbol{\eta}_j} + \frac{d\boldsymbol{K}_k}{d\boldsymbol{\eta}_j}\boldsymbol{\epsilon}_k + \boldsymbol{K}_k \frac{d\boldsymbol{\epsilon}_k}{d\boldsymbol{\eta}_j} \tag{C.7}$$

C.1.2 Deriving Predicted State Variance

Now consider the other two remaining derivatives, $\dfrac{\partial \boldsymbol{R}_{k|k-1}}{\partial \boldsymbol{\eta}_j}$ and $\dfrac{\partial \boldsymbol{R}_{k|k-1}}{\partial \hat{\boldsymbol{x}}_{k|k-1}}$.
We have

$$\boldsymbol{R}_{k|k-1} = \boldsymbol{C}_k \boldsymbol{P}_{k|k-1} \boldsymbol{C}_k^T + \boldsymbol{S} \tag{C.8}$$

so

$$\frac{\partial \boldsymbol{R}_{k|k-1}}{\partial \boldsymbol{\eta}_j} = \frac{\partial \boldsymbol{C}_k}{\partial \boldsymbol{\eta}_j}\boldsymbol{P}_{k|k-1}\boldsymbol{C}_k^T + \boldsymbol{C}_k \frac{\partial \boldsymbol{P}_{k|k-1}}{\partial \boldsymbol{\eta}_j}\boldsymbol{C}_k^T + \boldsymbol{C}_k \boldsymbol{P}_{k|k-1}\frac{\partial \boldsymbol{C}_k^T}{\partial \boldsymbol{\eta}_j} + \frac{\partial \boldsymbol{S}}{\partial \boldsymbol{\eta}_j} \tag{C.9}$$

We can get $\partial \boldsymbol{S}/\partial \boldsymbol{\eta}_j$ and $\partial \boldsymbol{C}_k/\partial \boldsymbol{\eta}_j$ from the definition of $\boldsymbol{S}$ and $\boldsymbol{h}$. Therefore we only need $\partial \boldsymbol{P}_{k|k-1}/\partial \boldsymbol{\eta}_j$.
By definition,

$$\boldsymbol{P}_{1|0} = \boldsymbol{P}_0 \tag{C.10}$$

so

$$\frac{\partial \boldsymbol{P}_{1|0}}{\partial \boldsymbol{\eta}_j} = \frac{\partial \boldsymbol{P}_0}{\partial \boldsymbol{\eta}_j} \tag{C.11}$$

can be calculated from expression. For positive integers k, we get the following sensitivity equation for $\boldsymbol{P}_{k+1|k}$

$$\begin{aligned} \frac{d}{dt}\left(\frac{\partial \boldsymbol{P}_{t|k}}{\partial \boldsymbol{\eta}_j}\right) &= \frac{\partial \boldsymbol{A}_t}{\partial \boldsymbol{\eta}_j}\boldsymbol{P}_{t|k} + \boldsymbol{A}_t \frac{\partial \boldsymbol{P}_{t|k}}{\partial \boldsymbol{\eta}_j} + \boldsymbol{P}_{t|k}\frac{\partial \boldsymbol{A}_t^T}{\partial \boldsymbol{\eta}_j} + \frac{\partial \boldsymbol{P}_{t|k}}{\partial \boldsymbol{\eta}_j}\boldsymbol{A}_t^T \\ &\quad + \frac{\partial \boldsymbol{\Sigma}}{\partial \boldsymbol{\eta}_j}\boldsymbol{\Sigma}^T + \boldsymbol{\Sigma}\frac{\partial \boldsymbol{\Sigma}^T}{\partial \boldsymbol{\eta}_j}, \qquad t \in [t_k, t_{k+1}] \\ \frac{\partial \boldsymbol{P}_{t|k}}{\partial \boldsymbol{\eta}_j}(t_k) &= \frac{\partial \boldsymbol{P}_{k|k}}{\partial \boldsymbol{\eta}_j} \end{aligned} \tag{C.12}$$

where

$$\frac{\partial \boldsymbol{P}_{k|k}}{\partial \boldsymbol{\eta}_j} = \frac{\partial \boldsymbol{P}_{k|k-1}}{\partial \boldsymbol{\eta}_j} - \left(\frac{\partial \boldsymbol{K}_k}{\partial \boldsymbol{\eta}_j}\boldsymbol{R}_{k|k-1}\boldsymbol{K}_k^T + \boldsymbol{K}_k \frac{\partial \boldsymbol{R}_{k|k-1}}{\partial \boldsymbol{\eta}_j}\boldsymbol{K}_k^T + \boldsymbol{K}_k \boldsymbol{R}_{k|k-1}\frac{\partial \boldsymbol{K}_k^T}{\partial \boldsymbol{\eta}_j}\right) \tag{C.13}$$

In the same way as for $\dfrac{\partial \boldsymbol{R}_{k|k-1}}{\partial \boldsymbol{\eta}_j}$ we get the equations for $\dfrac{\partial \boldsymbol{R}_{k|k-1}}{\partial \hat{\boldsymbol{x}}_{k|k-1}}$ by replacing $\boldsymbol{\eta}_j$ with $\hat{\boldsymbol{x}}_{k|k-1}$.

C.1.3 Deriving the Kalman Gain

In above cases, we need to compute the derivative and partial derivative of the Kalman gain $\boldsymbol{K}_k$. This can be done by first noting that

$$\frac{d\boldsymbol{K}_k}{d\boldsymbol{\eta}_j} = \frac{\partial \boldsymbol{K}_k}{\partial \boldsymbol{\eta}_j} + \frac{\partial \boldsymbol{K}_k}{\partial \hat{\boldsymbol{x}}_{k|k-1}} \frac{d\hat{\boldsymbol{x}}_{k|k-1}}{d\boldsymbol{\eta}_j} \tag{C.14}$$

and then calculating the derivative of $\boldsymbol{K}_k$ with respect to $\boldsymbol{\eta}_j$ as follows

$$\begin{aligned}
\frac{\partial \boldsymbol{K}_k}{\partial \boldsymbol{\eta}_j} &= \frac{\partial}{\partial \boldsymbol{\eta}_j}\left(\boldsymbol{P}_{k|k-1}\boldsymbol{C}_k^T \boldsymbol{R}_{k|k-1}^{-1}\right) \\
&= \frac{\partial \boldsymbol{P}_{k|k-1}}{\partial \boldsymbol{\eta}_j}\boldsymbol{C}_k^T \boldsymbol{R}_{k|k-1}^{-1} + \boldsymbol{P}_{k|k-1}\frac{\partial \boldsymbol{C}_k^T}{\partial \boldsymbol{\eta}_j}\boldsymbol{R}_{k|k-1}^{-1} + \boldsymbol{P}_{k|k-1}\boldsymbol{C}_k^T \frac{\partial \boldsymbol{R}_{k|k-1}^{-1}}{\partial \boldsymbol{\eta}_j} \\
&= \frac{\partial \boldsymbol{P}_{k|k-1}}{\partial \boldsymbol{\eta}_j}\boldsymbol{C}_k^T \boldsymbol{R}_{k|k-1}^{-1} + \boldsymbol{P}_{k|k-1}\frac{\partial \boldsymbol{C}_k^T}{\partial \boldsymbol{\eta}_j}\boldsymbol{R}_{k|k-1}^{-1} \\
&\quad - \boldsymbol{P}_{k|k-1}\boldsymbol{C}_k^T \boldsymbol{R}_{k|k-1}^{-1}\frac{\partial \boldsymbol{R}_{k|k-1}}{\partial \boldsymbol{\eta}_j}\boldsymbol{R}_{k|k-1}^{-1} \\
&= \frac{\partial \boldsymbol{P}_{k|k-1}}{\partial \boldsymbol{\eta}_j}\boldsymbol{C}_k^T \boldsymbol{R}_{k|k-1}^{-1} + \boldsymbol{P}_{k|k-1}\frac{\partial \boldsymbol{C}_k^T}{\partial \boldsymbol{\eta}_j}\boldsymbol{R}_{k|k-1}^{-1} - \boldsymbol{K}_k \frac{\partial \boldsymbol{R}_{k|k-1}}{\partial \boldsymbol{\eta}_j}\boldsymbol{R}_{k|k-1}^{-1} \\
&= \left(\frac{\partial \boldsymbol{P}_{k|k-1}}{\partial \boldsymbol{\eta}_j}\boldsymbol{C}_k^T + \boldsymbol{P}_{k|k-1}\frac{\partial \boldsymbol{C}_k^T}{\partial \boldsymbol{\eta}_j} - \boldsymbol{K}_k \frac{\partial \boldsymbol{R}_{k|k-1}}{\partial \boldsymbol{\eta}_j}\right)\boldsymbol{R}_{k|k-1}^{-1}
\end{aligned} \tag{C.15}$$

In the same way it follows that

$$\frac{\partial \boldsymbol{K}_k}{\partial \hat{\boldsymbol{x}}_{k|k-1}} = \left(\frac{\partial \boldsymbol{P}_{k|k-1}}{\partial \hat{\boldsymbol{x}}_{k|k-1}}\boldsymbol{C}_k^T + \boldsymbol{P}_{k|k-1}\frac{\partial \boldsymbol{C}_k^T}{\partial \hat{\boldsymbol{x}}_{k|k-1}} - \boldsymbol{K}_k \frac{\partial \boldsymbol{R}_{k|k-1}}{\partial \hat{\boldsymbol{x}}_{k|k-1}}\right)\boldsymbol{R}_{k|k-1}^{-1} \tag{C.16}$$

Moreover, the derivative of $\boldsymbol{K}_k$ with respect to $\boldsymbol{\theta}$ is calculated in the same way as the derivative with respect to $\boldsymbol{\eta}$.

C.2 Second Order Sensitivities

We need to compute the derivatives

$$\frac{d^2\boldsymbol{\epsilon}_k}{d\boldsymbol{\eta}_j d\boldsymbol{\theta}_n} \text{ and } \frac{d^2\boldsymbol{R}_{k|k-1}}{d\boldsymbol{\eta}_j d\boldsymbol{\theta}_n}. \tag{C.17}$$

Note that in the following calculations, $\boldsymbol{\theta}_n$ can be replaced by $\boldsymbol{\eta}_l$ in order to obtain the derivatives

$$\frac{d^2\boldsymbol{\epsilon}_k}{d\boldsymbol{\eta}_j d\boldsymbol{\eta}_l} \text{ and } \frac{d^2\boldsymbol{R}_{k|k-1}}{d\boldsymbol{\eta}_j d\boldsymbol{\eta}_l}. \tag{C.18}$$

We get

$$\begin{aligned}\frac{d^2\boldsymbol{\epsilon}_k}{d\boldsymbol{\eta}_j d\boldsymbol{\theta}_n} &= -\frac{d}{d\boldsymbol{\theta}_n}\left(\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{\eta}_j} + \frac{\partial \boldsymbol{h}}{\partial \hat{\boldsymbol{x}}_{k|k-1}}\frac{d\hat{\boldsymbol{x}}_{k|k-1}}{d\boldsymbol{\eta}_j}\right)\\ &= -\left(\frac{\partial^2 \boldsymbol{h}}{\partial \boldsymbol{\eta}_j \partial \boldsymbol{\theta}_n} + \frac{\partial^2 \boldsymbol{h}}{\partial \boldsymbol{\eta}_j \partial \hat{\boldsymbol{x}}_{k|k-1}}\frac{d\hat{\boldsymbol{x}}_{k|k-1}}{d\boldsymbol{\theta}_n}\right.\\ &\quad + \left(\frac{\partial^2 \boldsymbol{h}}{\partial \hat{\boldsymbol{x}}_{k|k-1}\partial \theta_n} + \frac{\partial^2 \boldsymbol{h}}{\partial \hat{\boldsymbol{x}}^2_{k|k-1}}\frac{d\hat{\boldsymbol{x}}_{k|k-1}}{\boldsymbol{\theta}_n}\right)\frac{d\hat{\boldsymbol{x}}_{k|k-1}}{d\boldsymbol{\eta}_j}\\ &\quad \left. + \frac{\partial \boldsymbol{h}}{\partial \hat{\boldsymbol{x}}_{k|k-1}}\frac{d^2\hat{\boldsymbol{x}}_{k|k-1}}{d\boldsymbol{\eta}_j d\boldsymbol{\theta}_n}\right)\end{aligned} \tag{C.19}$$

and

$$\begin{aligned}\frac{d^2\boldsymbol{R}_{k|k-1}}{d\boldsymbol{\eta}_j d\boldsymbol{\theta}_n} &= \frac{d}{d\boldsymbol{\theta}_n}\left(\frac{\partial \boldsymbol{R}_{k|k-1}}{\partial \boldsymbol{\eta}_j} + \frac{\partial \boldsymbol{R}_{k|k-1}}{\partial \hat{\boldsymbol{x}}_{k|k-1}}\frac{d\hat{\boldsymbol{x}}_{k|k-1}}{d\boldsymbol{\eta}_j}\right)\\ &= \frac{\partial^2 \boldsymbol{R}_{k|k-1}}{\partial \boldsymbol{\eta}_j \partial \boldsymbol{\theta}_n} + \frac{\partial^2 \boldsymbol{R}_{k|k-1}}{\partial \boldsymbol{\eta}_j \partial \hat{\boldsymbol{x}}_{k|k-1}}\frac{\partial\hat{\boldsymbol{x}}_{k|k-1}}{\partial\boldsymbol{\theta}_n}\\ &\quad + \left(\frac{\partial^2 \boldsymbol{R}_{k|k-1}}{\partial \hat{\boldsymbol{x}}_{k|k-1}\partial \boldsymbol{\theta}_n} + \frac{\partial^2 \boldsymbol{R}_{k|k-1}}{\partial \hat{\boldsymbol{x}}^2_{k|k-1}}\frac{d\hat{\boldsymbol{x}}_{k|k-1}}{d\boldsymbol{\theta}_n}\right)\frac{d\hat{\boldsymbol{x}}_{k|k-1}}{d\boldsymbol{\eta}_j}\\ &\quad + \frac{\partial \boldsymbol{R}_{k|k-1}}{\partial \hat{\boldsymbol{x}}_{k|k-1}}\frac{d^2\hat{\boldsymbol{x}}_{k|k-1}}{d\boldsymbol{\eta}_j \boldsymbol{\theta}_n}\end{aligned} \tag{C.20}$$

C.2.1 Second Derivative of Predicted Expected State

$$\begin{aligned}\frac{d}{dt}\left(\frac{d^2\hat{\boldsymbol{x}}_{t|k}}{d\boldsymbol{\eta}_j d\boldsymbol{\theta}_n}\right) &= \frac{\partial^2 \boldsymbol{f}}{\partial \boldsymbol{\eta}_j \partial \boldsymbol{\theta}_n} + \frac{\partial^2 \boldsymbol{f}}{\partial \eta_j \partial \hat{\boldsymbol{x}}_{t|k}}\frac{d\hat{\boldsymbol{x}}_{t|k-1}}{d\boldsymbol{\theta}_n}\\ &\quad + \left(\frac{\partial^2 \boldsymbol{f}}{\partial \hat{\boldsymbol{x}}_{t|k}\partial \boldsymbol{\theta}_n} + \frac{\partial^2 \boldsymbol{f}}{\partial \hat{\boldsymbol{x}}^2_{t|k}}\frac{d\hat{\boldsymbol{x}}_{t|k}}{d\boldsymbol{\theta}_n}\right)\frac{d\hat{\boldsymbol{x}}_{t|k}}{d\boldsymbol{\eta}_j}\\ &\quad + \frac{\partial \boldsymbol{f}}{\partial \hat{\boldsymbol{x}}_{t|k}}\frac{d^2\hat{\boldsymbol{x}}_{t|k}}{d\boldsymbol{\eta}_j \partial \boldsymbol{\theta}_n}, \qquad t \in [t_k, t_{k+1}]\\ \frac{d^2\hat{\boldsymbol{x}}_{t|k}}{d\boldsymbol{\eta}_j \partial \boldsymbol{\theta}_n}(t_k) &= \frac{d^2\hat{\boldsymbol{x}}_{k|k}}{d\boldsymbol{\eta}_j \partial \boldsymbol{\theta}_n}\end{aligned} \tag{C.21}$$

where

$$\frac{d^2\hat{\boldsymbol{x}}_{k|k}}{d\boldsymbol{\eta}_j \partial \boldsymbol{\theta}_n} = \frac{d^2\hat{\boldsymbol{x}}_{k|k-1}}{d\boldsymbol{\eta}_j \partial \boldsymbol{\theta}_n} + \frac{d^2\boldsymbol{K}_k}{d\boldsymbol{\eta}_j d\boldsymbol{\theta}_n}\boldsymbol{\epsilon}_k + \frac{d\boldsymbol{K}_k}{d\boldsymbol{\eta}_j}\frac{\boldsymbol{\epsilon}_k}{d\boldsymbol{\theta}_n} + \frac{d\boldsymbol{K}_k}{d\boldsymbol{\theta}_n}\frac{\boldsymbol{\epsilon}_k}{d\boldsymbol{\eta}_j} + \boldsymbol{K}_k\frac{d\boldsymbol{\epsilon}_k}{d\boldsymbol{\eta}_j d\boldsymbol{\theta}_n} \tag{C.22}$$

C.2.2 Second Derivative of Predicted State Variance

$$
\begin{aligned}
\frac{\partial^2 \boldsymbol{R}_{k|k-1}}{\partial \boldsymbol{\eta}_j \partial \boldsymbol{\theta}_n} &= \frac{\partial}{\partial \boldsymbol{\theta}_n}\left(\frac{\partial \boldsymbol{C}_k}{\partial \boldsymbol{\eta}_j}\boldsymbol{P}_{k|k-1}\boldsymbol{C}_k^T + \boldsymbol{C}_k\frac{\partial \boldsymbol{P}_{k|k-1}}{\partial \boldsymbol{\eta}_j}\boldsymbol{C}_k^T + \boldsymbol{C}_k\boldsymbol{P}_{k|k-1}\frac{\partial \boldsymbol{C}_k^T}{\partial \boldsymbol{\eta}_j} + \frac{\partial \boldsymbol{S}}{\partial \boldsymbol{\eta}_j}\right) \\
&= \frac{\partial^2 \boldsymbol{C}_k}{\partial \boldsymbol{\eta}_j \partial \boldsymbol{\theta}_n}\boldsymbol{P}_{k|k-1}\boldsymbol{C}_k^T + \frac{\partial \boldsymbol{C}_k}{\partial \boldsymbol{\eta}_j}\frac{\partial \boldsymbol{P}_{k|k-1}}{\partial \boldsymbol{\theta}_n}\boldsymbol{C}_k^T + \frac{\partial \boldsymbol{C}_k}{\partial \boldsymbol{\eta}_j}\boldsymbol{P}_{k|k-1}\frac{\boldsymbol{C}_k^T}{\partial \boldsymbol{\theta}_n} \\
&\quad + \frac{\partial \boldsymbol{C}_k}{\partial \boldsymbol{\theta}_n}\frac{\partial \boldsymbol{P}_{k|k-1}}{\partial \boldsymbol{\eta}_j}\boldsymbol{C}_k^T + \boldsymbol{C}_k\frac{\partial^2 \boldsymbol{P}_{k|k-1}}{\partial \boldsymbol{\eta}_j \partial \boldsymbol{\theta}_n}\boldsymbol{C}_k^T + \boldsymbol{C}_k\frac{\partial \boldsymbol{P}_{k|k-1}}{\partial \boldsymbol{\eta}_j}\frac{\boldsymbol{C}_k^T}{\partial \boldsymbol{\theta}_n} \\
&\quad + \frac{\partial \boldsymbol{C}_k}{\partial \boldsymbol{\theta}_n}\boldsymbol{P}_{k|k-1}\frac{\partial \boldsymbol{C}_k^T}{\partial \boldsymbol{\eta}_j} + \boldsymbol{C}_k\frac{\partial \boldsymbol{P}_{k|k-1}}{\partial \boldsymbol{\theta}_n}\frac{\partial \boldsymbol{C}_k^T}{\partial \boldsymbol{\eta}_j} + \boldsymbol{C}_k\boldsymbol{P}_{k|k-1}\frac{\partial^2 \boldsymbol{C}_k^T}{\partial \boldsymbol{\eta}_j \partial \boldsymbol{\theta}_n} \\
&\quad + \frac{\partial^2 \boldsymbol{S}}{\partial \boldsymbol{\eta}_j \partial \boldsymbol{\theta}_n}
\end{aligned}
\tag{C.23}
$$

With

$$
\begin{aligned}
\frac{d}{dt}\left(\frac{\partial^2 \boldsymbol{P}_{t|k}}{\partial \boldsymbol{\eta}_j \partial \boldsymbol{\theta}_n}\right) &= \frac{\partial^2 \boldsymbol{A}_t}{\partial \boldsymbol{\eta}_j \partial \boldsymbol{\theta}_n}\boldsymbol{P}_{t|k} + \frac{\partial \boldsymbol{A}_t}{\partial \boldsymbol{\eta}_j}\frac{\partial \boldsymbol{P}_{t|k}}{\partial \boldsymbol{\theta}_n} + \frac{\partial \boldsymbol{A}_t}{\partial \boldsymbol{\theta}_n}\frac{\partial \boldsymbol{P}_{t|k}}{\partial \boldsymbol{\eta}_j} + \boldsymbol{A}_t\frac{\partial^2 \boldsymbol{P}_{t|k}}{\partial \boldsymbol{\eta}_j \partial \boldsymbol{\theta}_n} \\
&\quad + \frac{\partial \boldsymbol{P}_{t|k}}{\partial \boldsymbol{\theta}_n}\frac{\partial \boldsymbol{A}_t^T}{\partial \boldsymbol{\eta}_j} + \boldsymbol{P}_{t|k}\frac{\partial^2 \boldsymbol{A}_t^T}{\partial \boldsymbol{\eta}_j \partial \boldsymbol{\theta}_n} + \frac{\partial^2 \boldsymbol{P}_{t|k}}{\partial \boldsymbol{\eta}_j \partial \boldsymbol{\theta}_n}\boldsymbol{A}_t^T + \frac{\partial \boldsymbol{P}_{t|k}}{\partial \boldsymbol{\eta}_j}\frac{\partial \boldsymbol{A}_t^T}{\partial \boldsymbol{\theta}_n} \\
&\quad + \frac{\partial^2 \boldsymbol{\Sigma}}{\partial \boldsymbol{\eta}_j \partial \boldsymbol{\theta}_n}\boldsymbol{\Sigma}^T + \frac{\partial \boldsymbol{\Sigma}}{\partial \boldsymbol{\eta}_j}\frac{\partial \boldsymbol{\Sigma}^T}{\partial \boldsymbol{\theta}_n} + \frac{\partial \boldsymbol{\Sigma}}{\partial \boldsymbol{\theta}_n}\frac{\partial \boldsymbol{\Sigma}^T}{\partial \boldsymbol{\eta}_j} + \boldsymbol{\Sigma}\frac{\partial^2 \boldsymbol{\Sigma}^T}{\partial \boldsymbol{\eta}_j \partial \boldsymbol{\theta}_n}, \qquad t \in [t_k, t_{k+1}] \\
\frac{\partial^2 \boldsymbol{P}_{t|k}}{\partial \boldsymbol{\eta}_j \partial \boldsymbol{\theta}_n}(t_k) &= \frac{\partial^2 \boldsymbol{P}_{k|k}}{\partial \boldsymbol{\eta}_j \partial \boldsymbol{\theta}_n}
\end{aligned}
\tag{C.24}
$$

where

$$
\begin{aligned}
\frac{\partial^2 \boldsymbol{P}_{k|k}}{\partial \boldsymbol{\eta}_j \partial \boldsymbol{\theta}_n} = &\frac{\partial^2 \boldsymbol{P}_{k|k-1}}{\partial \boldsymbol{\eta}_j \partial \boldsymbol{\theta}_n} \\
&- \left(\frac{\partial^2 \boldsymbol{K}_k}{\partial \boldsymbol{\eta}_j \partial \boldsymbol{\theta}_n}\boldsymbol{R}_{k|k-1}\boldsymbol{K}_k^T + \frac{\partial \boldsymbol{K}_k}{\partial \boldsymbol{\eta}_j}\frac{\partial \boldsymbol{R}_{k|k-1}}{\partial \boldsymbol{\theta}_n}\boldsymbol{K}_k^T + \frac{\partial \boldsymbol{K}_k}{\partial \boldsymbol{\eta}_j}\boldsymbol{R}_{k|k-1}\frac{\partial \boldsymbol{K}_k^T}{\partial \boldsymbol{\theta}_n}\right. \\
&+ \frac{\partial \boldsymbol{K}_k}{\partial \boldsymbol{\theta}_n}\frac{\partial \boldsymbol{R}_{k|k-1}}{\partial \boldsymbol{\eta}_j}\boldsymbol{K}_k^T + \boldsymbol{K}_k\frac{\partial^2 \boldsymbol{R}_{k|k-1}}{\partial \boldsymbol{\eta}_j \partial \boldsymbol{\theta}_n}\boldsymbol{K}_k^T + \boldsymbol{K}_k\frac{\partial \boldsymbol{R}_{k|k-1}}{\partial \boldsymbol{\eta}_j}\frac{\partial \boldsymbol{K}_k^T}{\partial \boldsymbol{\theta}_n} \\
&\left.+\frac{\partial \boldsymbol{K}_k}{\partial \boldsymbol{\theta}_n}\boldsymbol{R}_{k|k-1}\frac{\partial \boldsymbol{K}_k^T}{\partial \boldsymbol{\eta}_j} + \boldsymbol{K}_k\frac{\partial \boldsymbol{R}_{k|k-1}}{\partial \boldsymbol{\theta}_n}\frac{\partial \boldsymbol{K}_k^T}{\partial \boldsymbol{\eta}_j} + \boldsymbol{K}_k\boldsymbol{R}_{k|k-1}\frac{\partial^2 \boldsymbol{K}_k^T}{\partial \boldsymbol{\eta}_j \partial \boldsymbol{\theta}_n}\right)
\end{aligned}
\tag{C.25}
$$

The derivatives $\frac{\partial^2 \boldsymbol{R}_{k|k-1}}{\partial \boldsymbol{\eta}_j \partial \hat{\boldsymbol{x}}_{k|k-1}}$, $\frac{\partial^2 \boldsymbol{R}_{k|k-1}}{\partial \hat{\boldsymbol{x}}_{k|k-1} \partial \boldsymbol{\theta}_n}$ and $\frac{\partial^2 \boldsymbol{R}_{k|k-1}}{\partial \hat{\boldsymbol{x}}_{k|k-1}^2}$ are calculated in the same way. In the special case of $\frac{\partial^2 \boldsymbol{R}_{k|k-1}}{\partial \hat{\boldsymbol{x}}_{k|k-1}^2}$, the equations can be simplified to

$$\begin{aligned}\frac{\partial^2 \boldsymbol{R}_{k|k-1}}{\partial \hat{\boldsymbol{x}}^2_{k|k-1}} = {} & \frac{\partial^2 \boldsymbol{C}_k}{\partial \hat{\boldsymbol{x}}^2_{k|k-1}} \boldsymbol{P}_{k|k-1}\boldsymbol{C}_k^T + \boldsymbol{C}_k \frac{\partial^2 \boldsymbol{P}_{k|k-1}}{\partial \hat{\boldsymbol{x}}^2_{k|k-1}}\boldsymbol{C}_k^T + \boldsymbol{C}_k\boldsymbol{P}_{k|k-1}\frac{\partial^2 \boldsymbol{C}_k^T}{\partial \hat{\boldsymbol{x}}^2_{k|k-1}} \\ & + 2\left(\frac{\partial \boldsymbol{C}_k}{\partial \hat{\boldsymbol{x}}_{k|k-1}}\frac{\partial \boldsymbol{P}_{k|k-1}}{\partial \hat{\boldsymbol{x}}_{k|k-1}}\boldsymbol{C}_k^T + \frac{\partial \boldsymbol{C}_k}{\partial \hat{\boldsymbol{x}}_{k|k-1}}\boldsymbol{P}_{k|k-1}\frac{\boldsymbol{C}_k^T}{\partial \hat{\boldsymbol{x}}_{k|k-1}}\right. \\ & \left. + \boldsymbol{C}_k\frac{\partial \boldsymbol{P}_{k|k-1}}{\partial \hat{\boldsymbol{x}}_{k|k-1}}\frac{\boldsymbol{C}_k^T}{\partial \hat{\boldsymbol{x}}_{k|k-1}}\right) + \frac{\partial^2 \boldsymbol{S}}{\partial \hat{\boldsymbol{x}}^2_{k|k-1}}\end{aligned} \tag{C.26}$$

With

$$\begin{aligned}\frac{d}{dt}\left(\frac{\partial^2 \boldsymbol{P}_{t|k}}{\partial \hat{\boldsymbol{x}}^2_{k|k-1}}\right) = {} & \frac{\partial^2 \boldsymbol{A}_t}{\partial \hat{\boldsymbol{x}}^2_{k|k-1}}\boldsymbol{P}_{t|k} + \boldsymbol{A}_t\frac{\partial^2 \boldsymbol{P}_{t|k}}{\partial \hat{\boldsymbol{x}}^2_{k|k-1}} + \boldsymbol{P}_{t|k}\frac{\partial^2 \boldsymbol{A}_t^T}{\partial \hat{\boldsymbol{x}}^2_{k|k-1}} + \frac{\partial^2 \boldsymbol{P}_{t|k}}{\partial \hat{\boldsymbol{x}}^2_{k|k-1}}\boldsymbol{A}_t^T \\ & + 2\left(\frac{\partial \boldsymbol{A}_t}{\partial \hat{\boldsymbol{x}}_{k|k-1}}\frac{\partial \boldsymbol{P}_{t|k}}{\partial \hat{\boldsymbol{x}}_{k|k-1}} + \frac{\partial \boldsymbol{P}_{t|k}}{\partial \hat{\boldsymbol{x}}_{k|k-1}}\frac{\partial \boldsymbol{A}_t^T}{\partial \hat{\boldsymbol{x}}_{k|k-1}}\right) \\ & + \frac{\partial^2 \boldsymbol{\Sigma}}{\partial \hat{\boldsymbol{x}}^2_{k|k-1}}\boldsymbol{\Sigma}^T + 2\frac{\partial \boldsymbol{\Sigma}}{\partial \hat{\boldsymbol{x}}_{k|k-1}}\frac{\partial \boldsymbol{\Sigma}^T}{\partial \hat{\boldsymbol{x}}_{k|k-1}} + \boldsymbol{\Sigma}\frac{\partial^2 \boldsymbol{\Sigma}^T}{\partial \hat{\boldsymbol{x}}^2_{k|k-1}}, \qquad t \in [t_k, t_{k+1}] \\ \frac{\partial^2 \boldsymbol{P}_{t|k}}{\partial \hat{\boldsymbol{x}}^2_{k|k-1}}(t_k) = {} & \frac{\partial^2 \boldsymbol{P}_{k|k}}{\partial \hat{\boldsymbol{x}}^2_{k|k-1}}\end{aligned} \tag{C.27}$$

where

$$\begin{aligned}\frac{\partial^2 \boldsymbol{P}_{k|k}}{\partial \hat{\boldsymbol{x}}^2_{k|k-1}} = {} & \frac{\partial^2 \boldsymbol{P}_{k|k-1}}{\partial \hat{\boldsymbol{x}}^2_{k|k-1}} \\ & - \left(\frac{\partial^2 \boldsymbol{K}_k}{\partial \hat{\boldsymbol{x}}^2_{k|k-1}}\boldsymbol{R}_{k|k-1}\boldsymbol{K}_k^T + \boldsymbol{K}_k\frac{\partial^2 \boldsymbol{R}_{k|k-1}}{\partial \hat{\boldsymbol{x}}^2_{k|k-1}}\boldsymbol{K}_k^T + \boldsymbol{K}_k\boldsymbol{R}_{k|k-1}\frac{\partial^2 \boldsymbol{K}_k^T}{\partial \hat{\boldsymbol{x}}^2_{k|k-1}}\right. \\ & + 2\left(\frac{\partial \boldsymbol{K}_k}{\partial \hat{\boldsymbol{x}}_{k|k-1}}\frac{\partial \boldsymbol{R}_{k|k-1}}{\partial \hat{\boldsymbol{x}}_{k|k-1}}\boldsymbol{K}_k^T + \frac{\partial \boldsymbol{K}_k}{\partial \hat{\boldsymbol{x}}_{k|k-1}}\boldsymbol{R}_{k|k-1}\frac{\partial \boldsymbol{K}_k^T}{\partial \hat{\boldsymbol{x}}_{k|k-1}}\right. \\ & \left.\left. + \boldsymbol{K}_k\frac{\partial \boldsymbol{R}_{k|k-1}}{\partial \hat{\boldsymbol{x}}_{k|k-1}}\frac{\partial \boldsymbol{K}_k^T}{\partial \hat{\boldsymbol{x}}_{k|k-1}}\right)\right)\end{aligned} \tag{C.28}$$

C.2.3 Second Derivative of the Kalman Gain

In both above cases, we need to compute the second parial derivative of the Kalman gain $\boldsymbol{K}_k$. This can be done as follows

$$\begin{aligned}\frac{\partial^2 \boldsymbol{K}_k}{\partial \boldsymbol{\eta}_j \partial \boldsymbol{\theta}_n} &= \frac{\partial^2 \boldsymbol{P}_{k|k-1}}{\partial \boldsymbol{\eta}_j \partial \boldsymbol{\theta}_n} \boldsymbol{C}_k^T \boldsymbol{R}_{k|k-1}^{-1} + \frac{\partial \boldsymbol{P}_{k|k-1}}{\partial \boldsymbol{\eta}_j} \frac{\partial \boldsymbol{C}_k^T}{\partial \boldsymbol{\theta}_n} \boldsymbol{R}_{k|k-1}^{-1} + \frac{\partial \boldsymbol{P}_{k|k-1}}{\partial \boldsymbol{\eta}_j} \boldsymbol{C}_k^T \frac{\partial \boldsymbol{R}_{k|k-1}^{-1}}{\partial \boldsymbol{\theta}_n} \\ &\quad + \frac{\partial \boldsymbol{P}_{k|k-1}}{\partial \boldsymbol{\theta}_n} \frac{\partial \boldsymbol{C}_k^T}{\partial \boldsymbol{\eta}_j} \boldsymbol{R}_{k|k-1}^{-1} + \boldsymbol{P}_{k|k-1} \frac{\partial^2 \boldsymbol{C}_k^T}{\partial \boldsymbol{\eta}_j \partial \boldsymbol{\theta}_n} \boldsymbol{R}_{k|k-1}^{-1} + \boldsymbol{P}_{k|k-1} \frac{\partial \boldsymbol{C}_k^T}{\partial \boldsymbol{\eta}_j} \frac{\partial \boldsymbol{R}_{k|k-1}^{-1}}{\partial \boldsymbol{\theta}_n} \\ &\quad - \frac{\partial \boldsymbol{K}_k}{\partial \boldsymbol{\theta}_n} \frac{\partial \boldsymbol{R}_{k|k-1}}{\partial \boldsymbol{\eta}_j} \boldsymbol{R}_{k|k-1}^{-1} - \boldsymbol{K}_k \frac{\partial^2 \boldsymbol{R}_{k|k-1}}{\partial \boldsymbol{\eta}_j \partial \boldsymbol{\theta}_n} \boldsymbol{R}_{k|k-1}^{-1} - \boldsymbol{K}_k \frac{\partial \boldsymbol{R}_{k|k-1}}{\partial \boldsymbol{\eta}_j} \frac{\partial \boldsymbol{R}_{k|k-1}^{-1}}{\partial \boldsymbol{\theta}_n} \\ &= \left(\frac{\partial^2 \boldsymbol{P}_{k|k-1}}{\partial \boldsymbol{\eta}_j \partial \boldsymbol{\theta}_n} \boldsymbol{C}_k^T + \boldsymbol{P}_{k|k-1} \frac{\partial^2 \boldsymbol{C}_k^T}{\partial \boldsymbol{\eta}_j \partial \boldsymbol{\theta}_n} \boldsymbol{R}_{k|k-1}^{-1} - \boldsymbol{K}_k \frac{\partial^2 \boldsymbol{R}_{k|k-1}}{\partial \boldsymbol{\eta}_j \partial \boldsymbol{\theta}_n} \right. \\ &\quad \left. + \frac{\partial \boldsymbol{P}_{k|k-1}}{\partial \boldsymbol{\eta}_j} \frac{\partial \boldsymbol{C}_k^T}{\partial \boldsymbol{\theta}_n} + \frac{\partial \boldsymbol{P}_{k|k-1}}{\partial \boldsymbol{\theta}_n} \frac{\partial \boldsymbol{C}_k^T}{\partial \boldsymbol{\eta}_j} - \frac{\partial \boldsymbol{K}_k}{\partial \boldsymbol{\theta}_n} \frac{\partial \boldsymbol{R}_{k|k-1}}{\partial \boldsymbol{\eta}_j} \right) \boldsymbol{R}_{k|k-1}^{-1} \\ &\quad - \left(\frac{\partial \boldsymbol{P}_{k|k-1}}{\partial \boldsymbol{\eta}_j} \boldsymbol{C}_k^T + \boldsymbol{P}_{k|k-1} \frac{\partial \boldsymbol{C}_k^T}{\partial \boldsymbol{\eta}_j} - \boldsymbol{K}_k \frac{\partial \boldsymbol{R}_{k|k-1}}{\partial \boldsymbol{\eta}_j} \right) \boldsymbol{R}_{k|k-1}^{-1} \frac{\partial \boldsymbol{R}_{k|k-1}}{\partial \boldsymbol{\theta}_n} \boldsymbol{R}_{k|k-1}^{-1}\end{aligned} \tag{C.29}$$

The derivatives $\dfrac{\partial^2 \boldsymbol{K}_k}{\partial \boldsymbol{\eta}_j \partial \hat{\boldsymbol{x}}_{k|k-1}}$, $\dfrac{\partial^2 \boldsymbol{K}_k}{\partial \hat{\boldsymbol{x}}_{k|k-1} \partial \boldsymbol{\theta}_n}$ and $\dfrac{\partial^2 \boldsymbol{K}_k}{\partial \hat{\boldsymbol{x}}_{k|k-1}^2}$ are calculated in the same way. In the special case of $\dfrac{\partial^2 \boldsymbol{K}_k}{\partial \hat{\boldsymbol{x}}_{k|k-1}^2}$, the derivative can be simplified to

$$\begin{aligned}\frac{\partial^2 \boldsymbol{K}_k}{\partial \hat{\boldsymbol{x}}_{k|k-1}^2} &= \left(\frac{\partial^2 \boldsymbol{P}_{k|k-1}}{\hat{\boldsymbol{x}}_{k|k-1}^2} \boldsymbol{C}_k^T + \boldsymbol{P}_{k|k-1} \frac{\partial^2 \boldsymbol{C}_k^T}{\hat{\boldsymbol{x}}_{k|k-1}^2} \boldsymbol{R}_{k|k-1}^{-1} - \boldsymbol{K}_k \frac{\partial^2 \boldsymbol{R}_{k|k-1}}{\hat{\boldsymbol{x}}_{k|k-1}^2} \right. \\ &\quad \left. + \frac{\partial 2 \boldsymbol{P}_{k|k-1}}{\partial \hat{\boldsymbol{x}}_{k|k-1}} \frac{\partial \boldsymbol{C}_k^T}{\partial \hat{\boldsymbol{x}}_{k|k-1}} - \frac{\partial \boldsymbol{K}_k}{\partial \hat{\boldsymbol{x}}_{k|k-1}} \frac{\partial \boldsymbol{R}_{k|k-1}}{\partial \hat{\boldsymbol{x}}_{k|k-1}} \right) \boldsymbol{R}_{k|k-1}^{-1} \\ &\quad - \left(\frac{\partial \boldsymbol{P}_{k|k-1}}{\partial \hat{\boldsymbol{x}}_{k|k-1}} \boldsymbol{C}_k^T + \boldsymbol{P}_{k|k-1} \frac{\partial \boldsymbol{C}_k^T}{\partial \hat{\boldsymbol{x}}_{k|k-1}} - \boldsymbol{K}_k \frac{\partial \boldsymbol{R}_{k|k-1}}{\partial \hat{\boldsymbol{x}}_{k|k-1}} \right) \boldsymbol{R}_{k|k-1}^{-1} \frac{\partial \boldsymbol{R}_{k|k-1}}{\partial \hat{\boldsymbol{x}}_{k|k-1}} \boldsymbol{R}_{k|k-1}^{-1}\end{aligned} \tag{C.30}$$