Model misspecification diagnosis in mixed effects models through SDEs

Ruben Taieb

August 19, 2023

Introduction

In many settings, namely in biology and medicine, we are lead to take several measurements of a certain time-varying process (e.g drug concentration, tumour size, neutrophil levels...) in a set of different individuals, who form a population.

In order to study this data and formulate an adequate model for the studied process, we naturally turn towards a mixed effects model setting, which allows for intra-individual variability (like measurement errors) and inter-individual variability, meaning that each person has a different parameter defining his underlying process.
Usually, the studied process $(x_t)_{t\geq 0}$ is defined by an ODE, which depends on a function f, called the *structural model* , and on a vector ϕ_i , the *individual parameter*: $\mathrm{d}x_t = f(t, x_t, \phi_i)\,\mathrm{d}t$.
The structural model usually stems from physiological, chemical or physical equations. However, this presupposed structural model could be wrong, for a myriad of reasons. In a pharmaceutical and pharmacological setting, this could lead to catastrophic results. As such, diagnosing whether the structural model is significantly incorrect is crucial.

One possibility for testing for model misspecification is to add another parameter accounting for error: that is, adding a diffusion term to the ODE, effectively modelling the studied process with an SDE : $\mathrm{d}x_t = f(t, x_t, \phi_i)\,\mathrm{d}t + \gamma\,\mathrm{d}W_t$. Thus, if we find a diffusion term that is too large, then the discrepancy between model and data will have to have come from a structural error, and not just from parameter variability.

During our internship, we have expanded this idea from a single individual study, to a hierarchical setting (composed of a whole population of individuals), and so needing the manipulation of latent variables. We will see how studying a population increases greatly the power of our test, at the cost of needing a much more complex and robust algorithm for parameter estimation.

In this report, we will first present rigorously the model, and the test of structural misspecification. Then, we will explain the algorithm we coded for estimating population parameters in an SDE non-linear mixed effects model, which will use an SAEM-Metropolis-Hastings-within-Gibbs algorithm, coupled with

an Euler scheme method or a Kalman filter. We will also study the implementation and coding of this algorithm, that we have done in R. Finally, we will check the efficacy of the test on a simple pharmacological model, with simulated data.

This internship was conducted at Potsdam University, under the supervision of Dr Niklas Hartung. It is the continuation of a previous Master thesis, where we try to extend from an individual setting to a population model. This research is a topic of interest of CRC1294 Data Assimilation, a DFG funded Collaborative Research Center.

Contents

Chapter 1

Misspecfication of mixed-effects structural models

1 ODE based mixed-effects model

Suppose that you are conducting clinical trials of a new drug: you have a set of I individuals, to whom you give a certain dose. Then, you measure the drug concentration $y_{i,j}$ for the i^{th} patient at times $t_{i,j} > 0$.
You now have a set of I independent trajectories: here's how they can be modelled to account for intra-individual and inter-individual variability: for $i \in \{1, ..., I\}$ and $j \in \{1, ..., J_i\}$

$$\mathrm{d}X_t^{(i)} = f(t, X_t, \phi_i)\,\mathrm{d}t \tag{1.1}$$

$$y_{i,j} = X_{t_{i,j}}^{(i)} + \epsilon_{i,j} \tag{1.2}$$

$$\epsilon_{i,j} \sim \mathcal{N}(0, \sigma^2 I_d) \text{ iid} \tag{1.3}$$

$$\phi_i \sim \Pi(\ \cdot\ ; \theta) \text{ iid} \tag{1.4}$$

- $X \in \mathcal{F}(\mathbb{R}_+, \mathbb{R}^d)$ is the underlying process that we wish to study: it could represent drug concentration, or tumour size, or, in finances, the value of a stock, or the position of a particle in statistical physics... In this report, we will mainly focus on clinical applications, however.
- $f : \mathbb{R}_+ \times \mathbb{R}^d \times \mathbb{R}^p \to \mathbb{R}^d$ is called the structural model: it is what will define the evolution of X, and is usually chosen according to some more or less physiological or phenomenological reasoning.
- The $\epsilon_{i,j}$ are the errors which occur when measuring the process X. They are the source of intra-individual variability in the model.

- $\phi \in \mathbb{R}^p$ is the individual parameter: it is a *latent* random variable (meaning, that we cannot measure it directly), which will change the drift function f , and allow for inter-individual variability.
- The (ϕ_i) are supposed to be sampled from a distribution defined by a vector $\theta \in \Theta$, which is called the population parameter; it defines the general trends of the population, and correlations between individual parameters. When studying mixed effects models, the objective often is to estimate θ, and possibly later estimate the parameters ϕ_i by a Maximum a Posteriori method (a Bayesian estimation, basically).
- Often, we take the distribution Π to be normal or log-normal: in that case, we may write θ as $(\phi_{pop}, \Omega, \sigma^2)$ and $\phi \sim \mathcal{N}(\phi_{pop}, \Omega)$ or $\log(\phi) \sim \mathcal{N}(\phi_{pop}, \Omega)$. We will suppose normality of the individual parameters in the rest of this report; the methods here presented are easily generalizable to more general distributions (from exponential families, as we will see in part II-5).
- We can also consider the case where we only have a partially observed process: meaning that we have $y_{i,j} = h(X_{t_{i,j}}) + \epsilon_{i,j}$, where h is a function. We often choose h to be a logarithm (so we get log-normal residuals), or a projection into a lower dimensional space. In what follows, we will not consider X to be partially observed.

2 SDE based mixed-effects models

Sometimes, the underlying process we wish to study may be too complex to be modelled as an ODE solution, or may be intrinsically stochastic: the position of a physical molecule, or a biological/physiological characteristic may come to mind.

In this case, a generalization of the classical non-linear mixed-effects model is in order: the main difference being the addition of a diffusion term to the definition of the underlying process, which becomes stochastic:

$$\mathrm{d}X_t = f(t, X_t, \phi_i)\,\mathrm{d}t + \gamma(t, X_t)\,\mathrm{d}W_t \tag{1.5}$$

where W is a brownian process.

In what follows, we will consider a constant diffusion term, with $\gamma(t, X_t) = \gamma$. The methods that we will present in this report are easily generalizable to any diffusion term (for $\gamma(t, X_t) = \gamma X_t$, it suffices to consider $log(X_t)$), but we will make this constance assumption to simplify notation and implementation.

This diffusion term γ adds a new layer of error to the model: while the measurement errors $\epsilon_{i,j}$ only allow for independent residuals, using an SDE-based model allows us to account for correlation between such residuals. This is why

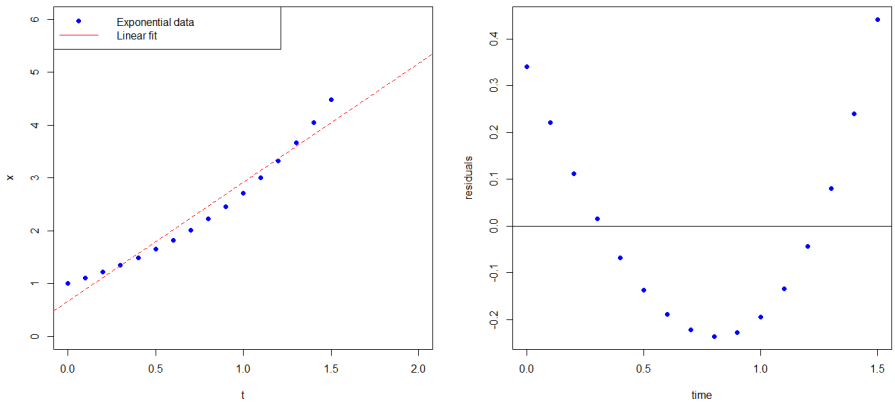

Figure 1.1: On the left, we have the best linear fit for an exponentially generated data ($x_i = e^{t_i}$), and on the right, the plot of the residuals with relation to the time. This inter-residual correlation cannot be explained by a classical linear fit, and would thus indicate model misspecification

we will attempt, in this paper, to diagnose model misspecification through SDE mixed effects modelling.

Indeed, inter-residual correlation often indicates model misspecification (in the classical ODE model): suppose, for instance, that we try to fit an exponential curve with a linear model (as shown in figure 1.1). Then, our estimation overfits at some periods in time, and underfits in others. We see a clear correlation between residuals (and some Markov properties of them) , or, at the very least, some time-dependence of the residuals.

More precisely, consider some data $(y_{i,j})_{i,j}$ observed from a process X, with $\mathrm{d}X_t = f(t, X_t, \phi)\,\mathrm{d}t$.
Now, we try to model it through a wrong process $\mathrm{d}\tilde{X} = \tilde{f}(t, \tilde{X}_t, \tilde{\phi})\,\mathrm{d}t$. We have:

$$\begin{aligned} y_{i,j} &= X_{t_{i,j}} + \epsilon_{i,j} \\ &= \int_0^{t_{i,j}} f(t, X_t, \phi)\,\mathrm{d}t + \epsilon_{i,j} \\ &= \tilde{X}_{t_{i,j}} + \int_0^{t_{i,j}} \left(f(t, X_t, \phi) - \tilde{f}(t, \tilde{X}_t, \tilde{\phi})\right)\mathrm{d}t + \epsilon_{i,j} \\ &= \tilde{X}_{t_{i,j}} + g(t_{i,j}) + \epsilon_{i,j} \end{aligned}$$

And so, we get residuals of the form $\tilde{\epsilon}_{i,j} = g(t_{i,j}) + \epsilon_{i,j}$.
Thus, we cannot guarantee iid residuals: we then turn to SDE-based error models to account for it.

Of course, we could also use a time-dependent or state-dependent $\epsilon_{i,j}$: however, we would need to know in advance the profile of the function g. Moreover, seeing as the measurements are usually taken with the same tools throughout an experiment, it is not very realistic to model the measurement errors by a time-varying function (although we could model it through an error proportional to X_t ; but in that case, considering the logarithm the process makes such a term vanish).

3 SDE based test for misspecfication diagnosis in ODE models

We have seen that fits with correlated residuals are often indicative of structural misfit, and that an SDE model could model such a correlation. And so, we present here a test for diagnosing whether a model is misspecified:

$$\mathcal{H}_0 : \gamma = 0 \qquad \text{vs} \qquad \mathcal{H}_1 : \gamma > 0$$

Let us explain this test a bit more:
Suppose you have some data $\mathbf{y}$ generated by an ODE mixed-effects model $\mathcal{M}_1$. We model it through an SDE-based model $\mathcal{M}_2$ with the same variability and structural model as $\mathcal{M}_1$: if the structural function f is correctly specified, then the estimate of γ should be close to 0, since
$\mathcal{M}_1 = \mathcal{M}_2 \cap \{\gamma = 0\}$.

However, if f is misspecified, then, to account for residual correlation, we would find a high estimate of γ.

Thus, in our diagnostic test, $\mathcal{H}_0$ is "the model is correctly specified" , and $\mathcal{H}_1$ is "the model is misspecified".

In order to be able to use this test, we need the distribution of the estimated parameter $\hat{\gamma}$ under the null hypothesis: this distribution, of course, depends on what estimator is used. While we try to estimate it through likelihood maximization, we will see in chapter 2 that there is no straightforward way of computing it.
We have different algorithms and methods for approximating the MLE $\hat{\gamma}_{MLE}$, which thus have different distributions. Since we do not know beforehand which distribution this estimate follows, we use a Monte-Carlo method to estimate it. Here is a sketch of how the test is conducted on some data $\mathbf{y}$:

1. We estimate the population parameters $\hat{\theta}_{ODE}$ of $\mathbf{y}$ under the assumption $\mathcal{H}_0 : \gamma = 0$; that is, we estimate the population parameters using the ODE mixed-effects model.

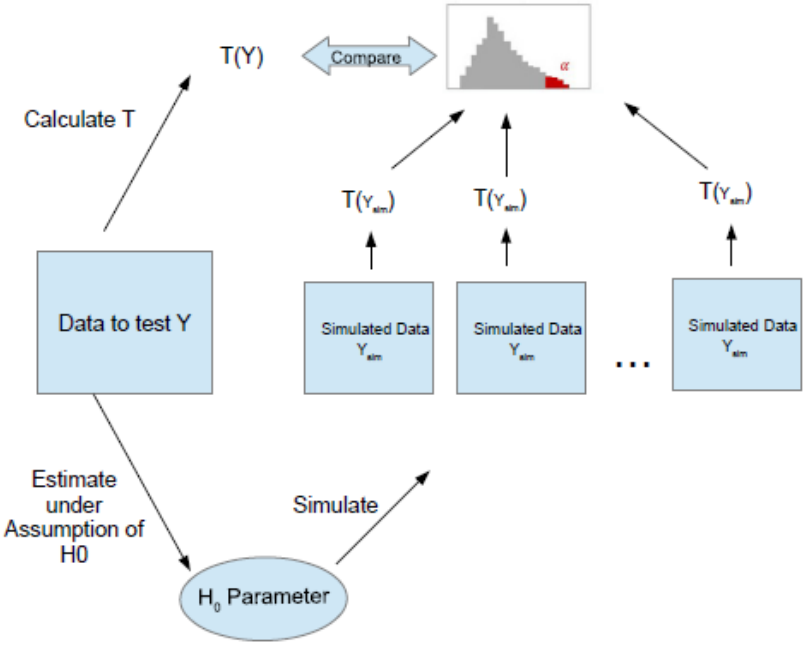

Figure 1.2: A figure representing the MC method for testing fitness; figure taken from a previous master's internship at Potsdam University

2. We simulate $\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, ..., \mathbf{y}^{(M)}$ different datasets using the ODE mixed-effects model, and $\hat{\theta}_{ODE}$.

3. Using the SDE-based model, we estimate the diffusion terms $\hat{\gamma}^{(1)}, \hat{\gamma}^{(2)}, ..., \hat{\gamma}^{(M)}$ of the generated datasets.

4. We estimate $\hat{\gamma}$ the diffusion term of $\mathbf{y}$ using the SDE-based model, adn we calculate the empirical quantile of it in the set $\{\hat{\gamma}^{(1)}, \hat{\gamma}^{(2)}, ..., \hat{\gamma}^{(M)}\}$

5. If the quantile is above a certain threshold (which will be 0.95 in this report) , then the model is deemed misspecified; else, we don't reject the null hypothesis.

Basically, we take our data, and calculate the "total residual error" contained in σ^2 , using an ODE-based estimation. Then, using an SDE-based model, we can distribute this error into two different categories: measurement error, and stochastic error, each having distinct correlation and time-dependency properties.
We apply the same procedure to generated data, which have the same total error, and we compare the percentages of error attributed to stochasticity, as

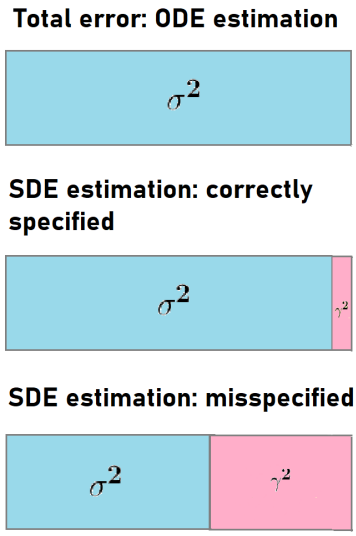

Figure 1.3: Illustration of the effect of adding a new error accountability parameter ; a misspecified model will shift much of the "total error" estimated through an ODE model to the diffusion term γ^2, as in the bottom ; while almost all of it will be accounted by the measurement error σ^2 for a correctly specified model.

opposed to measurement error.
And thus, this test has the advantage of not needing comparison between the data's "total error" with the one from other experiments, or with expected values from the measurement tools used. The model fitness can be tested in a vacuum, without the use of prior data and textbook reference values (for the measurement error σ, for instance).

Chapter 2

Parameter estimation in SDE mixed-effects models

In order to apply the test to our data, we must first be able to estimate the population parameters in an SDE/ODE mixed-effects model. This chapter will focus on methods and algorithms to efficiently estimate them.

4 Challenges

Usually, when we wish to esimate a parameter in any statistical model, the Maximum Likelihood estimator is favoured: it has several useful properties, such as consistency, asymptotic normality, and it reaches asymptotically the Cramer-Rao bound.

However, sometimes, we do not have direct access to the likelihood of our model: it is the case in our model.
An SDE-based mixed effects model likelihood cannot be written in closed form, as it has two different intractabilities:

Latent Variables

This intractability is common to all types of mixed-effects models: the presence of the individual parameters $(\phi_i)_{i\in[\![1;I]\!]}$ do not allow for direct calculation of the likelihood. Indeed, since we can not measure the variables ϕ_i (they are latent), the likelihood can only be written as:

$$p_\theta(\mathbf{y}) = \prod_{i=1}^{I} p_\theta(\mathbf{y}_i) \tag{2.1}$$

$$= \prod_{i=1}^{I} \int_{\mathbb{R}^p} p_\theta(\mathbf{y}_i \mid \phi_i)\Pi(\phi_i;\theta)\,\mathrm{d}\phi_i \tag{2.2}$$

The likelihood is thus not in closed form, as the integrals involved are very usually not analytically calculable.
To circumvent this problem, two main algorithms can be used: EM, and FOCE.

The EM (expectation-maximization) algorithm is the most widespread, and has several variations, like the SAEM algorithm, which is the one we will use in our report. We will present them in section 5 in more detail.
The FOCE (first-order conditional estimation) algorithm, also called Laplace's method, relies on a second-order expansion of the complete log-likelihood function $l_i : \phi \to \log(p_\theta(\mathbf{y}_i, \phi))$ around its maximum $\hat{\phi}_i$, yielding:

$$\int_{\mathbb{R}^p} p_\theta(\mathbf{y}_i \mid \phi_i)\Pi(\phi_i;\theta)\,\mathrm{d}\phi_i = \int_{\mathbb{R}^p} e^{l_i(\phi_i)}\,\mathrm{d}\phi_i \tag{2.3}$$

$$\approx \int_{\mathbb{R}^p} e^{l_i(\hat{\phi}_i)+\frac{1}{2}\Delta l_i(\hat{\phi}_i)\cdot(\phi_i-\hat{\phi}_i)^2} \tag{2.4}$$

$$= e^{l_i(\hat{\phi}_i)}\sqrt{\frac{2\pi}{\Delta l_i(\hat{\phi}_i)}} \tag{2.5}$$

This method is further explained by Christoffer W. Tornøe et. al [1], but it will not be our focus.

These two algorithms are very different, conceptually, and so have different pros and cons:
As we will see in section 5, SAEM is an iterative and stochastic algorithm, whose convergence will greatly depend on different choices of parameters (such as number of iterations, convergence criteria, proposal distributions...), while FOCE is less complicated to implement. Moreover, results stemming from FOCE can be better replicated.
However, SAEM is very good at avoiding local maxima, because of its stochastic nature, and works better when the initial conditions given are far from the MLE (since FOCE is based on a Taylor expansion around it). Since it is very hard to visually have an idea of the value of $\hat{\gamma}_{MLE}$, SAEM could work better for SDE-based models.
We have made the choice in this paper to use the SAEM algorithm.

Intractable trajectories

Almost all SDEs do not possess closed-form trajectories, and the randomness of them adds another layer of intractability to the likelihood function: we cannot write $p_\theta(\mathbf{y}_i \mid \phi_i)$ in closed form.

We can, however, use the Markov property of a classic SDE solution to write:

$$\begin{aligned} p_\theta(\mathbf{y}_i \mid \phi_i) &= p_\theta(\mathbf{y}_i \mid \phi_i, X_{t_1}, ..., X_{t_{J_i}}) p_\theta(X_{t_{i,1}}, ..., X_{t_{i,J_i}} \mid \phi_i,) \\ &= \prod_{j=1}^{J_i} p_\sigma(\mathbf{y}_{i,j} \mid X_{t_{i,j}}) \cdot \prod_{j=1}^{J_i - 1} p_\gamma(X_{t_{i,j+1}} \mid \phi_i, X_{t_{i,j}}) \cdot p_\theta(X_{t_{i,1}} \mid \phi_i) \end{aligned}$$

Usually, the value of $X_{t,1}$ is either considered to be known in advance, or is a parameter of the statistical problem.

We know how to calculate the first product term explicitly (it is just a normal distribution, or a log-normal distribution). The key to correctly approximate the complete log-likelihood for an SDE relies then on correctly approximating the term $p_\gamma(X_{t_{i,j+1}} \mid \phi_i, X_{t_{i,j}})$, which does not have closed form.

We have found two different methods to approximate this conditional likelihood:

- The Kalman filter method is very widespread in many different fields, which relies on the assumption that $X_{t_{i,j+1}}$ follows a Gaussian distribution conditionally to ϕ and $X_{t_{i,j}}$. It was used when coupled with the SAEM algorithm by Marc Lavielle and Maud Delattre [2] ; we will explain it in further detail in section 7.
- Another method was to consider a slightly different model, which approximates the SDE trajectories with an Euler-Maruyama scheme, presented by Donnet, S. and Samson, A [3]: we will also explain this method in further detail in section 6.

We will see that (in theory), these methods also have different pros and cons:
The Euler-Maruyama scheme is computationally expensive, and the use of a huge amount of latent variables make it very hard to implement; however, it should also work better in sparse data, when compared to a Kalman filter, seeing as it relies on data "enrichment", and an Extended Kalman filter uses linearization between data points.

5 SAEM algorithm

EM algorithm

The EM algorithm is a very widespread and well known method of dealing with latent variables (i.e variables that cannot be directly measured, but only inferred from some other variable) in statistical problems.

It is an iterative algorithm to find the maximum of the (incomplete) likelihood of a model containing hidden variables. EM stands for expectation-maximization, which are the two main steps of this algorithm.

Suppose the value of $\theta^{(k)}$ at step k is known. Then, we follow the following procedure:

- **Expectation step (E)** : We calculate

$$\mathbf{Q}(\theta \mid \theta^{(k)}) = \mathbb{E}_{\phi \sim \Pi(\cdot;\theta^{(k)})}[\log p_\theta(\mathbf{y}, \phi)] \tag{2.6}$$

  the expectation of the complete log-likelihood, conditionally on the distribution defined by the previous step's population parameter.

- **Maximization step (M)** : We update θ as

$$\theta^{(k+1)} = \arg\max_\theta \mathbf{Q}(\theta \mid \theta^{(k)})$$

It has been shown [4], [5] in many different settings, and under very relaxed assumptions, that the EM algorithm converges to a local maximum of the likelihood.

However, the implementation of the EM algorithm requires for us to be able to calculate the function $\mathbf{Q}(\theta \mid \theta')$, which is intractable in most cases. This is why we must use an alternate version of the EM algorithm, called SAEM.

SAEM algorithm

The SAEM (stochastic approximation expectation-maximization) algorithm uses a Monte-Carlo method of estimating the needed expectation.

At step k, we generate M(k) realizations $\phi^1, ..., \phi^{M(k)}$ of the individual parameter under the complete distribution $p_\theta(\mathbf{y}, \phi)$. Then, we define recursively

$$\hat{\mathbf{Q}}_k(\theta) = \hat{\mathbf{Q}}_{k-1}(\theta) + \alpha_k(\frac{1}{M(k)} \sum_{j=1}^{M(k)} \log \mathcal{L}(y, \phi^j; \theta) - \hat{\mathbf{Q}}_{k-1}(\theta))$$

with

- $\hat{\mathbf{Q}}_0(\theta) = \frac{1}{m(0)} \sum_{j=1}^{M(0)} \log p_\theta(y, \phi^j)$
- $(\alpha_k)_{k\geq 1}$ is a non-increasing positive sequence such that $\alpha_1 = 1$, $\sum \alpha_k = \infty$ and $\sum \alpha_k^2 < \infty$.

First, notice that in the case $\alpha_k = 1$, we estimate the expectation using a simple Monte-Carlo method (which is called MCEM, Monte Carlo EM).

More generally, we have that $\hat{\mathbf{Q}}_k(\theta)$ is a convex combination of a "memory term" $\hat{\mathbf{Q}}_{k-1}(\theta)$ and a stochastic term $\frac{1}{m(k)} \sum_{j=1}^{m(k)} \log p_\theta(y, \phi^{(m)})$, which is a moments estimation of the expectation.
The advantage of SAEM is that we are able to balance the stochastic and memory terms; the memory term helps with the convergence of the sequence, while the stochastic term may allow us to avoid local minima/maxima (or saddle points). This is why, usually, at the beginning of the algorithm (ie for small k), we set α_k close to 1, and for k large, we get closer and closer to 0 (the usual choice is $\alpha_k \sim \frac{1}{k}$).
Indeed, at the start of the algortihm, we allow ourselves a very large window to move; the algorithm relies more on randomness/"stochasticness", thus, θ^k will fluctuate more. This reduces the risk of getting stuck in a non-global minimum or a saddle point. Then, to have convergence, we add the memory term, which will give some Cauchy properties to the sequence.

Convergence of the SAEM algorithm relies on the Robbins-Monro theorem for convergence of stochastic procedures, studied by Kushner and Clark (1978) [6]. Under some quite broad assumptions, Bernard Delyon, Marc Lavielle and Eric Moulines [5] proved the convergence of $\theta^{(k)}$ to a local maximum of the likelihood.

It is very hard to directly implement the SAEM algorithm directly, however: optimizing a function which calls previous ones recursively has an extreme computational cost, especially if we have to go through tens or hundreds of steps.
A solution to bypass this problem is to use minimal sufficient statistics of the complete model: if there are ψ, ν functions of θ and S a function of $\mathbf{y}$ and ϕ , such that

$$\log\big(p_\theta(y, \phi)\big) = \psi(\theta) + S(y, \phi) \cdot \nu(\theta) \tag{2.7}$$

(i.e $p_\theta(y, \phi)$ belongs to the exponential family, and verifies the conditions of the Fisher–Neyman factorization theorem), then updating the sufficient statistic S will give us access to all the needed information about the estimation through each step, without having to compute the function recursively.
More specifically, if we set $s_0 = 0$, and update the sufficient statistic as such:

$$s_{k+1} = s_k + \alpha_k(S(y, \phi) - s_k) \tag{2.8}$$

then we can easily prove recursively that

$$Q_k(\theta) = \psi(\theta) + s_k \cdot \nu(\theta) \tag{2.9}$$

and we only need to know the sufficient parametric statistic s_k to optimize Q_k.

Metropolis-Hastings (within-Gibbs) sampler

A crucial step of the SAEM algorithm is the sampling of the individual parameters ϕ under the complete probability distribution. Seeing as it is hard to sample directly from this intractable and multidimensional probability distribution, we

must use a Metropolis-Hastings type algorithm.

A Metropolis-Hastings (MH) algorithm creates an ergodic discrete-time Markov chain whose stationary distribution is π , the one from which we want to sample: thus, iteratively updating values according to this chain's transition matrix will yield an approximation of a sample under π.
More specifically, supposing that we have x_r at step r:

1. Propose a candidate value x_c. It is a sample from a simpler distribution which will depend on x_r, called $g(\cdot \mid x_r)$ the proposal distribution.

2. Calculate the acceptance ratio

$$A(x_c, x_r) = \min\left(1, \frac{\pi(x_c)g(x_r \mid x_c)}{\pi(x_r)g(x_c \mid x_r)}\right) \tag{2.10}$$

3. Generate $u \sim \mathcal{U}([0,1])$. If $u < A$, we accept the proposed state, and $x_{r+1} = x_c$. Else, we reject the candidate and keep $x_{r+1} = x_r$.

Thus, we have built a Markov chain, with transition kernel

$$K(x' \mid x) = g(x' \mid x) \cdot A(x', x) \tag{2.11}$$

Now, by noticing that $A(x, x') = A(x', x)^{-1}$, then either $A(x, x') = 1$, or $A(x', x) = 1$. And so, supposing that $A(x, x') = 1$, we find that the detailed balance property is verified for the distribution π, since

$$K(x' \mid x)\pi(x) = g(x \mid x')\pi(x') = K(x \mid x')\pi(x') \tag{2.12}$$

Thus, if the chain is ergodic , it has a unique stationary distribution which will be π.
Namely, if the proposal distribution g has the same support as the target π , then the chain is clearly irreducible (since we can go from any state to any other in just one step with positive probability).
If we have a probability > 0 of getting a rejection (some values will yield an acceptance ratio < 1), the chain stays in the same state with positive probability; thus, it is also aperiodic.
Under these two assumptions, the chain is irreducible and aperiodic, and so is ergodic. These two theoretical conditions ensure the convergence of the distribution of the chain generated by M-H towards the stationary distribution π.

An MH algorithm has some parameters that are very important for the correct convergence of the chain that have to be chosen. Namely, we have an extremely vast choice of possible proposal distributions, and choosing one may be quite a challenge, especially when the sample space is very high-dimensional (see section 6, in the Euler-Maruyama case).

In order to apply the MH algorithm in our case, we need to be able to estimate efficiently (up to a normalizing factor) the complete likelihood $p_\theta(\mathbf{y}, \phi)$, so we may compute the acceptance ratio A. Being able to estimate this intractable function with a Kalman filter or an Euler scheme is thus key to the use of SAEM.

6 Euler-Maruyama method

Theoretical overview

To estimate $p_\theta(\mathbf{y}, \phi)$, Donnet S. and Samson A. [3] consider a slightly different enriched model, by introducing a new set of latent variables w.
Basically, w_i will be the value of the stochastic process $X^{(i)}$ at intermediate time steps $t_1 = \tau_1 < ... < \tau_{N_i} = t_{J_i}$, verifying that for all j, there is an integer n_j such that $t_j = \tau_{n_j}$.
Then, we will approximate the process X with a classic Euler-Maruyama scheme between each τ_n, yielding:

$$h_n = \tau_n - \tau_{n-1} \tag{2.13}$$

$$w_{i,n} = w_{i,n-1} + h_n f(\tau_{n-1}, w_{n-1}, \phi_i) + \gamma\sqrt{h_n}\xi_{i,n} \tag{2.14}$$

$$\xi_n \sim \mathcal{N}(0, 1) \tag{2.15}$$

$$y_{i,j} = w_{i,n_j} + \epsilon_{i,j} \tag{2.16}$$

$$\epsilon_{i,j} \sim \mathcal{N}(0, \sigma^2) \tag{2.17}$$

This new model will depend on the Euler time-steps $h = (h_1, ..., h_n)$. We will consider that all h_i's are equal to $h > 0$ for notation simplicity, and denote this new approximate model $\mathcal{M}_h$.

Now, by adding w as a latent variable, we are able to write the complete likelihood in closed form:

$$\begin{aligned} p_\theta(\mathbf{y}, \phi, w) &= \prod_{i=1}^{I} p_\theta(y_i, \phi_i, w_i) \\ &= \prod_{i=1}^{I} p_\theta(y_i \mid \phi_i, w_i) p_\theta(w_i \mid \phi_i) p_\theta(\phi_i) \\ &= \prod_{i=1}^{I} \left(p_{\sigma^2}(y_i \mid w_i) \prod_{n=1}^{N} p_{\gamma^2}(w_{i,n} \mid w_{i,n-1}, \phi_i) \Pi(\phi_i; \theta) \right) \end{aligned}$$

and

$$p_{\sigma^2}(y_i \mid w_i) = \prod_{j=1}^{J_i} p_{\sigma^2}(y_{i,j} \mid w_{i,n_j})$$

where we used the independence between subjects, and the Markov property for a SDE solution.

All terms in the product may be written in closed form:

- $y_{i,j} \mid w_{i,n_j} \sim \mathcal{N}(w_{i,n_j}, \sigma^2)$
- $w_{i,n} \mid \big(w_{i,n-1}, \phi_i\big) \sim \mathcal{N}(w_{i,n-1} + hf(\tau_{n-1}, w_{n-1}, \phi_i), h\gamma^2)$
- $\phi_i \sim \mathcal{N}(\phi_{pop}, \Omega)$ (or possibly some other distribution)

Moreover, we may see that the complete model belongs to an exponential family (supposing the parameter ϕ to be in $\mathbb{R}$) , since:

$$\begin{aligned}
-\log\big(p_\theta(\mathbf{y}, \phi, w)\big) = &\sum_{i=1}^{I}\sum_{j=1}^{J_i} \frac{(y_{i,j} - w_{i,n_j})^2}{2\sigma^2} \\
&+ \sum_{i=1}^{I}\sum_{n=1}^{N} \frac{(w_{i,n} - w_{i,n-1} - hf(\tau_{n-1}, w_{i,n-1}, \phi_i))^2}{2\gamma^2} \\
&+ \frac{1}{2}\sum_{i=1}^{I} (\phi_i - \phi_{pop})^T \Omega^{-1} (\phi_i - \phi_{pop}) \\
&+ IJ\log(\sqrt{2\pi\sigma^2}) + IN\log(\sqrt{2\pi\gamma^2}) + I\log(\sqrt{2\pi\det\Omega})
\end{aligned}$$

And so, setting $\psi(\theta) - \frac{1}{2}\sum_{i=1}^{I} \phi_{pop}^T \Omega^{-1} \phi_{pop}$ as the last line of the equation , with a sufficient statistics

$$S_h(\mathbf{y}, \phi, w) = \begin{bmatrix} \frac{1}{M(k)} \sum_{m=1}^{M(k)} \sum_{i=1}^{I} \sum_{j=1}^{J_i} (y_{i,j} - w_{i,n_j})^2 \\ \frac{1}{M(k)} \sum_{m=1}^{M(k)} \sum_{i=1}^{I} \sum_{n=1}^{N} (w_{i,n} - w_{i,n-1} - hf(\tau_{n-1}, w_{i,n-1}, \phi_i))^2 \\ \frac{1}{M(k)} \sum_{m=1}^{M(k)} \sum_{i=1}^{I} \phi_i \phi_i^T \\ \frac{1}{M(k)} \sum_{m=1}^{M(k)} \sum_{i=1}^{I} \phi_i \end{bmatrix}$$

and

$$\nu(\theta) = \begin{bmatrix} 1/(2\sigma^2) \\ 1/(2\gamma^2) \\ \Omega^{-1}/2 \\ \Omega^{-1}\phi_{pop} \end{bmatrix}$$

we find the required exponential property.

Moreover, the optimization of $Q_k(\theta)$ in the M-step of SAEM defined in (2.9) actually has closed form, which greatly reduces computation time. Simply by differentiating Q_k along each coordinate of θ, we get that

$$\sigma^{(k)} = \frac{(s_k)_1}{IJ} \tag{2.18}$$

$$\gamma^{(k)} = \frac{(s_k)_2}{IN} \tag{2.19}$$

$$\phi_{pop}^{(k)} = \frac{(s_k)_4}{I} \tag{2.20}$$

$$\Omega^{(k)} = \frac{(s_k)_3}{I} - \frac{(s_k)_4^2}{I^2} \tag{2.21}$$

It is important to notice, however, that this version of SAEM converges to a local maximum of the likelihood of the **approximate** model $\mathcal{M}_h$, which we name $\hat{\theta}_h$.
Thankfully, it has been proven [3] that there exists a constant $C > 0$ independent of θ such that

$$|| \hat{\theta}_{MLE} - \hat{\theta}_h ||_\infty \leq Ch \tag{2.22}$$

And so, while the estimator $\hat{\theta}_h$ is not actually consistent, we know that it can be arbitrarily close to a consistent estimator.

Proposal distribution

The main problem with this Euler scheme method is the introduction of a huge dimensional latent space (the dimension of w can be in the order of thousands) : it is victim of the curse of dimensionality.
As such, the MH algorithm encounters some difficulties: namely, it cannot explore the latent space in a realistic amount of runs. While theoretically the created Markov chain is ergodic, in practice, the amount of steps needed to go from some state to another can be extremely high.

And thus, the proposal distribution actually has a great impact on the convergence of the algorithm, even if it theoretically verifies the required assumptions: it needs to explore efficiently the latent space, and more specifically, the w space. The candidates w_c must be likely trajectories of the stochastic process.

For this, we use a Metropolis-Hastings-within-Gibbs sampler:

1. we propose a candidate of the individual parameter ϕ^c, using either $\phi^c \sim \mathcal{N}(\phi_{pop}, \Omega)$ or $\phi^c \sim \mathcal{N}(\phi, \delta^2)$; the second option, which consists of a random walk MH, is the one we opt for; δ is a parameter that we have to tune.

2. we then propose w^c conditionally on the candidate ϕ^c , following $g(w^c \mid w, \phi, \phi^c)$

There are quite a few possibilities for the distribution $g(w^c \mid w, \phi, \phi^c)$; let us consider, for instance, a data-independent Euler scheme: :

$$\begin{aligned}
&w^c_{i,0} = y_{i,0} + \epsilon_{i,0} \\
&w^c_{i,n} = w^c_{i,n-1} + h_n f(\tau_{n-1}, w^c_{i,n-1}, \phi^c_i) + \gamma\sqrt{h_n}\xi_{i,n} \\
&\epsilon_{i,0} \sim \mathcal{N}(0, \sigma^2) \\
&\xi_{i,n} \sim \mathcal{N}(0, 1)
\end{aligned}$$

In the case $\alpha_k = 1$ in (2.8) , we get that

$$\hat{\gamma}^{(k+1)} = \frac{1}{INM(k)} \sum_{m=1}^{M(k)} \sum_{i=1}^{I} \sum_{n=1}^{N} (w_{i,n} - w_{i,n-1} - hf(\tau_{n-1}, w_{i,n-1}, \phi_i))^2$$

which is proportional to a variable of law $\frac{1}{INM(k)}\chi^2(INM(k))$; thus

$$\mathbb{V}[\hat{\gamma}^{(k+1)}] \propto \frac{1}{INM(k)} \tag{2.23}$$

Seeing as N is huge, of order 10^4 or more, this indicates that the γ parameter will vary very slowly between iterations. This can be seen in the plot (2.1) , where we ran the algorithm with this proposal distribution on data generated by the model presented in 3.1 , and 10 individuals : while all other parameters are close to convergence in less than 50 iterations, the γ parameter takes around 600 iterations to be ready for the phase $\alpha_k = 1/k$

In order to circumvent this problem, we thought of another proposal distribution:

- $w^c_{i,n_j} = y_{i,j} + \epsilon_{i,j}$
- $w^c_{i,n} = w^c_{i,n-1} + h_n f(\tau_{n-1}, w^c_{i,n-1}, \phi^c_i) + \gamma\sqrt{h_n}\xi_{i,n}$ if $n \notin \{n_j\}_j$

This way, the error of the sampled trajectory would be transferred from σ to γ ; the variability of γ would be much higher from iteration to iteration. We now have, however, $\mathbb{V}[\hat{\sigma}^{(k+1)}] \propto \frac{1}{IJM(k)}$, but since $J << N$, the variability of σ is not too compromised.

A big issue with this proposal, however, is that such a trajectory is highly unlikely (as the stochastic will often be concentrated in a few very specific points), and γ will be grossly overestimated as a result.
An option would be then to consider a convex combination of both proposals:

$$w^c_{i,n} = \beta w^{(1)}_{i,n} + (1-\beta) w^{(2)}_{i,n} \tag{2.24}$$

Choosing β, however, is not simple, and we will see that varying the value of β will change the estimates of σ and γ that we get (but not the estimates of ϕ_{pop}

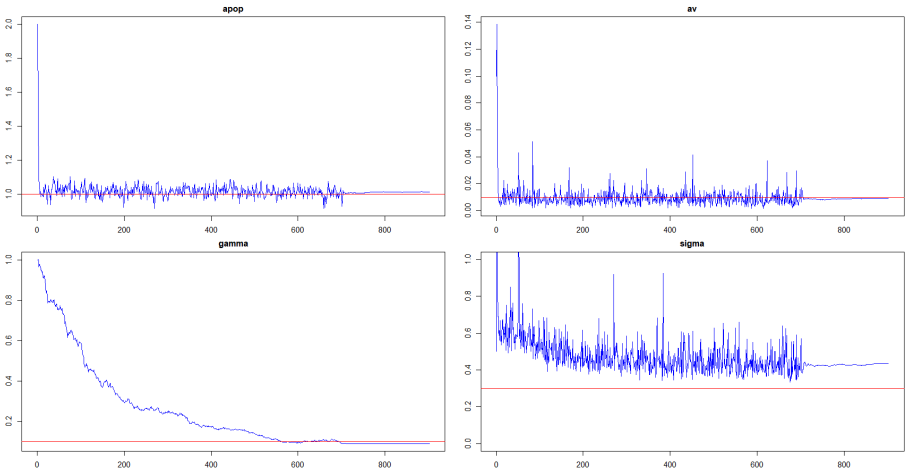

Figure 2.1: Plot of the variation of the estimates wrt the iteration number of SAEM. The red horizontal lines represent the true value of the parameters. We first distinguish a first phase, where $\alpha_k = 1$ and the estimates fluctuate heavily (as the expectation in the E step is approximated by a purely stochastic, Monte-Carlo approach). Then, we have a second phase, where we set $\alpha_k = 1/k$, and the estimates properly stabilize and converge.

and Ω), which, unfortunately, is precisely the values that we need to estimate correctly to run the structural misspecification test.

It is worth noting, however, that these proposal distributions verify the required theoretical properties required for convergence of Metropolis-Hastings presented in section 5: since the support of a normal distribution is $\mathbb{R}$, then w can take any value in $\mathbb{R}^{IN}$. Moreover, if we write down the acceptance ratio $A(w, w_c)$, we see that it converges to 0 when $|| \ w_c \ ||_\infty \to \infty$; meaning that there will be a region of $\mathbb{R}^{NI}$ attainable with positive probability by w, in which $A(w, w_c) < 1/2$; we conclude that we stay in the same step after an iteration with strictly positive probability. The chain is thus ergodic, and satisfies the required properties for MH convergence.

7 The Kalman filter method

Kalman filtering is a very widespread method for estimating parameters in SDEs ; the R package, ctsmr [7] , uses this algorithm for singular-individual (no latent variables) cases. We included this package in the coding of our SAEM-Kalman algorithm.

The coupling of SAEM with a Kalman filter has been studied by M Delattre and M Lavielle [2] ; in that paper, however, it was one of the individual

parameters which was an SDE solution, instead of the process per se.

We have seen that, using the Markov property of SDE solutions, we may write

$$p_\theta(y_i, \phi_i) = p_\theta(y_{i,0}) \prod_{j=1}^{J} p_\theta(y_{i,j} \mid y_{i,j-1}, \phi_i) \Pi(\phi_i; \theta)$$

The Kalman filter, then, supposes normality of this likelihood to estimate this value: $y_{i,j} \mid y_{i,j-1}, \phi_i \sim \mathcal{N}(\hat{y}_{i,j|j-1}, R_{i,j|j-1})$.
With this approximation, we find that

$$p_\theta(y_i, \phi_i) \propto p_\theta(y_{i,0}) \prod_{j=1}^{J} \frac{1}{\sqrt{\det(R_{i,j|j-1})}} e^{-\Delta_{i,j} R^{-1}_{i,j|j-1} \Delta^T_{i,j}/2} \Pi(\phi_i; \theta)$$

where $\Delta_{i,j} = y_{i,j} - \hat{y}_{i,j|j-1}$.
Thus, to estimate the complete likelihood, it suffices to have adequate values for the sequences $(R_{i,j|j-1})_j$ and $(\hat{y}_{i,j|j-1})_j$. The idea will be to estimate this sequence recursively.

First, let us define the following:

$$\begin{aligned}
\hat{x}_{t|j} &= \mathbb{E}[X_t \mid y_0, \dots, y_j] \\
\hat{x}_{l|j} &= \mathbb{E}[X_{t_l} \mid y_0, \dots, y_j] \\
P_{t|j} &= \mathbb{V}[X_t \mid y_0, \dots, y_j] \\
P_{l|j} &= \mathbb{V}[X_{t_l} \mid y_0, \dots, y_j] \\
R_{l|j} &= \mathbb{V}[y_l \mid y_0, \dots, y_j]
\end{aligned}$$

We have the following relationships:

$$\hat{y}_{j|j-1} = \hat{x}_{j|j-1} \tag{2.25}$$

$$R_{j|j-1} = P_{j|j-1} + \sigma^2 \tag{2.26}$$

$$\hat{x}_{j|j} = \hat{x}_{j|j-1} + P_{j|j-1} R^{-1}_{j|j-1} \Delta_j \tag{2.27}$$

$$P_{j|j} = P_{j|j-1} - P^T_{j|j-1} R^{(-1)}_{j|j-1} P_{j|j-1} \tag{2.28}$$

Equations (2.25) and (2.26) come directly from $y_{i,j} = x_{t_{i,j}} + \epsilon_{i,j}$. For (2.27) and (2.28), we write

$$\begin{aligned}
\hat{x}_{j|j} &= \mathbb{E}[X_{t_j} \mid y_0, \dots, y_j] \\
&= \mathbb{E}\Big[\mathbb{E}[X_{t_j} \mid y_j] \mid y_0, \dots, y_{j-1}\Big]
\end{aligned}$$

Conditionally to $y_0, \dots, y_{j-1}$, we have that $X_{t_j} \sim \mathcal{N}(\hat{x}_{j|j-1}, P_{j|j-1})$ and $y_j \sim \mathcal{N}(\hat{y}_{j|j-1}, R_{j|j-1})$.

Moreover, $\mathbf{cov}_{y_0,\dots,y_{j-1}}(X_{t_j}, y_j) = \mathbb{V}[X_{t_j} \mid y_0, \dots, y_{j-1}] = P_{j|j-1}$, since the $\epsilon_{i,j}$ are independent of the process. Thus,

$$X_{t_j} \mid y_j \sim \mathcal{N}(\hat{x}_{j|j-1} + P_{j|j-1}R_{j|j-1}^{-1}\Delta_j, P_{j|j-1} - P_{j|j-1}^T R_{j|j-1}^{-1} P_{j|j-1})$$

which directly yields the relations (2.27) and (2.28).

We know how to calculate $R_{j|j}$ and $\hat{y}_{j|j}$ from $R_{j|j-1}$ and $\hat{y}_{j|j-1}$; if we manage to give an expression of $R_{j+1|j}$ and $\hat{y}_{j+1|j}$ from these, than we will be able to calculate the target sequence recursively.

Linear case First, suppose the SDE equation to be linear:

$$\mathrm{d}X_t = (A_\phi X_t + b_\phi(t))\,\mathrm{d}t + \gamma\,\mathrm{d}W_t \tag{2.29}$$

Then, we have, by switching expectation and differentiation:

$$\begin{aligned}
\mathrm{d}\hat{x}_{t|j} &= \mathbb{E}\Big[\mathrm{d}X_t \mid y_0, \dots, y_j\Big] \\
&= \mathbb{E}\Big[(A_\phi X_t + b_\phi(t))\,\mathrm{d}t + \gamma\,\mathrm{d}W_t \mid y_0, \dots, y_j\Big] \\
&= (A_\phi \hat{x}_{t|j} + b_\phi(t))\,\mathrm{d}t
\end{aligned}$$

Similarly, we find that, by using Itô's lemma,

()

By solving these simple ODEs, we may then calculate $P_{j+1|j}$ and $\hat{x}_{j+1|j}$, which will allow us to compute the needed likelihood.

Non-linear case In the more general case, we linearize the drift function f between the time points t_j; if we set $\tilde{f}(t) = f(t, \hat{x}_{t|j}, \phi_i)$, then

$$\mathrm{d}X_t \approx \tilde{f}(t) + \partial_x f(t, \hat{x}_{t|j}, \phi_i)\Big(X_t - \hat{x}_{t|j}\Big)\,\mathrm{d}t + \gamma\,\mathrm{d}W_t$$

And so, using the same method as previously, we get

$$\begin{aligned}
\mathrm{d}\hat{x}_{t|j} &= \tilde{f}(t)\,\mathrm{d}t \\
\mathrm{d}P_{t|j} &= 2\partial_x f(t, \hat{x}_{t|j}, \phi_i) P_{t|j} + \gamma^2
\end{aligned}$$

These are both classical 1D ODEs, and can be solved by several methods, including implicit/explicit Euler schemes, or a Runge-Kutta algorithm.

Chapter 3

Running the algorithms

8 Implementation

In order to realize the misspecification diagnosis test, we first searched for packages or softwares in the literature that could realize parameter estimation in SDE based mixed effects models.
We found an R package, called PSM [10], but it was not functional: the package was removed from the CRAN repository, and the source doe ran into a core error that we could not repair (as it was coded in FORTRAN). So, we had to code the estimator from scratch, in R, with the help of the software Monolix and the package ctsmr [7].

This proved to be very challenging and time consuming: the SAEM algorithm has many different parameters to be tuned to ensure good convergence. For instance:

- We must set the condition upon which we change from the fluctuating regime ($\alpha_k = 1$) to the converging regime ($\alpha_k = 1/k$). We chose to use the following condition: if the empirical likelihood varies less than a certain ϵ in N_{var} iterations, then we change regimes. (we still have then 2 parameters to tune: ϵ and N_{var})
- We must choose the number of iterations of SAEM in the converging regime.
- We must choose the sequence $M(k)$
- In the Euler case, we must choose h the step of the latent variable scheme.
- In the MH sampler, we always sample $\phi^c \sim \mathcal{N}(\phi, \delta^2)$ as a proposal distribution. δ is a parameter to be tuned: if it is too small, then we will not explore correctly the latent space, and converge will be very slow. If it is too high, however, the acceptance rate (i.e the amount of proposals that are accepted) will be too low, and the algorithm will converge poorly.

Moreover, consecutive runs of the algorithms can be very computationally expensive, as one SAEM run can take from 20 to 40 minutes, depending on the number of iterations and the method used. This means that one misspecification test could take a day to run, if we estimate the test distribution with 40-50 samples.
Not only does this mean that the tests take very long to run, but it also made testing and repairing errors in the code quite a long process.

9 Algorithms convergence

Euler-SAEM

Using the non-saturable model described in section 3.8, we ran the Euler-SAEM algorithm on data generated using the set of parameters $\phi^*_{pop} = 1$, $\Omega^* = 0.01$, ${\gamma^*}^2 = 0.01$, ${\sigma^*}^2 = 0.05$, $I = 10$, and 11 measurements in the first cycle.
The initial values given were ${\phi_{pop}}_0 = 1.5$, $\Omega_0 = 0.05$, $\gamma_0^2 = 10^{-5}$ and $\sigma_0^2 = 0.01$

The figure (2.2) shows that, while the structural parameters ϕ_{pop} and Ω converge consistently, and quite fast (~ 30 iterations) , the error parameters do not converge to the same values depending on the value of β ; if β increases, then so does $\hat{\gamma}$, while $\hat{\sigma}$ decreases, as is shown in figure (2.3).
Here, β indicates the convexity factor in equation (2.24), for the proposal distribution of the MH algorithm.

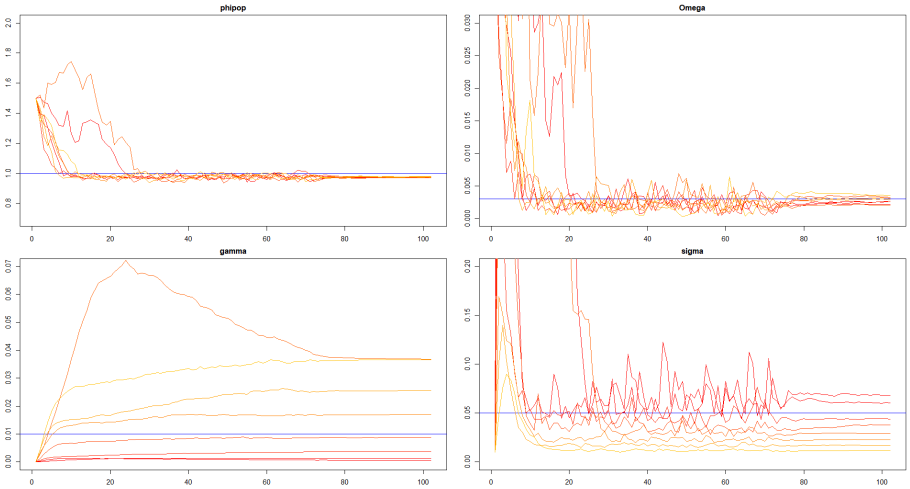

Figure 3.1: Evolution of the different parameters at each SAEM interation: redder lines represent lower βs , while yellower lines represent a higher β value, which are in $\{0, 1/10, \ldots, 9/10, 1\}$.
The blue line represents the true value of the parameter.

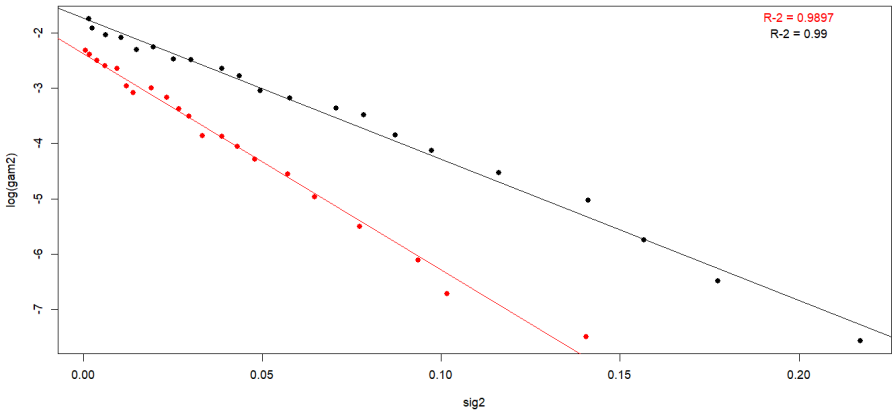

Figure 3.2: Plot of $\log(\hat{\gamma})$ wrt $\hat{\sigma}$; each point represents the estimations stemming from SAEM-Euler with a specific $\beta \in \{0, 1/20, 2/20, \ldots, 19/20, 1\}$. The value of β used is decreasing from left to right.
We have used 2 different sets of parameters for the two different colored dots (the red represents ${\gamma^*}^2 = 0.01$ and ${\sigma^*}^2 = 0.05$, and the black ${\gamma^*}^2 = 0.01$ and ${\sigma^*}^2 = 0.1$)

Since different β values yield the same (and correct) values of ϕ_{pop} and Ω, but different error parameter estimates, we may believe that the fit is similar along a certain direction of the plane (σ^2, γ^2) . To verify this, we plot in figure (2.3) a log-likelihood profile, that is, a 3D plot of the likelihood, when ϕ^*_{pop} and Ω^* are fixed to be the parameters with which the data was generated.

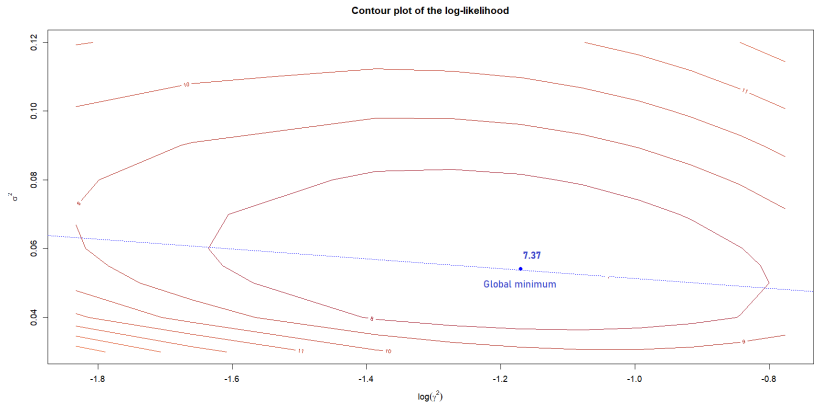

Figure 3.4: Contour plot of the likelihood

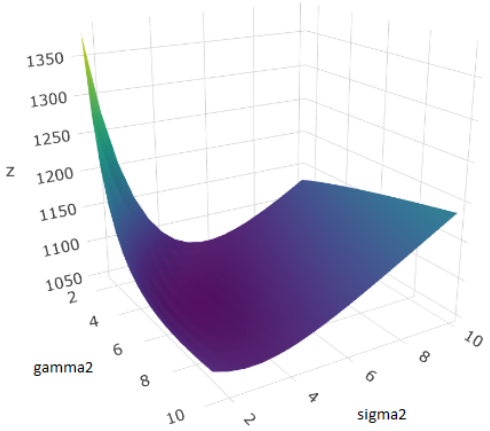

Figure 3.3: 3D log-likelihood profile, for ϕ_{pop} and Ω fixed, and varying γ and σ

We see that there is a line on which the likelihood is very flat (it is not constant, however), and we verified that the values on figure (2.2) appear on this flat 1D surface.
A hypothesis for the variability of the parameter error estimates would thus be that they get "stuck" in this flat surface, and that the proposal distributions do not allow for a proper exploration of the latent space, and thus, of this flat line. And so, there is an α that would yield a correct estimate, but it is impossible to know (or even estimate) its value in a real-life scenario.
Maybe running Euler-SAEM many times, with different values of β , and then choosing the estimate with higher empirical likelihood could work, but such a method would be very computationally expensive.

Kalman-SAEM

In order to quantify the convergence of the SAEM-Kalman algorithm, we simulate data using the model in chapter 4, and a population parameter θ^* ; then, we run the algorithm on this data to estimate $\hat{\theta}_k$. After doing this M times, we calculate the values

$$RMSE = \sqrt{\frac{1}{M}\sum_{k=1}^{M}\frac{|| \hat{\theta}_k - \theta^* ||^2}{|| \theta^* ||^2}}$$

$$\text{bias} = \frac{1}{M}\sum_{k=1}^{M}\frac{\hat{\theta}_k - \theta^*}{\theta^*}$$

A low relative mean square error (RMSE) and bias indicate good convergence properties of the algorithm.
Here, we will run $M = 40$ different simulations and estimations.
We will use 2 sets of parameters for this: the design 1 is $\phi^*_{pop} = 1$, $\Omega^* = 0.01$, $\gamma^* = 0.3$, $\sigma^2 = 0.1$; design 2 is design 1, but with $\gamma^* = 0$ (to see how the algorithm performs in the context of our Monte-Carlo test).

Parameter	ϕ_{pop}	Ω	γ	σ
RMSE	3.2 %	27.5 %	22.2 %	16 %
bias	0.06 %	- 6.1 %	5.3 %	- 3.9 %

Table 3.1: Kalman Filter Algorithm Convergence for design 1

We may compare these results with the ones obtained in [3] ; we find slightly smaller RMSE's and biases for all parameters, which could also be indicative of the simpler model that we chose, with only one individual parameter.
We see that we have an extremely good convergence of ϕ_{pop} (RMSE $< 5\%$), and satisfactory convergence of the other population parameter (RMSE $< 30\%$).
However, our algorithm underestimates σ to overestimate γ, as seen in the sign of the bias for these parameters. This bias is quite small ($< 10\%$) , so we still consider the convergence to be satisfactory, but we see here again a challenge in separating the error stemming from iid measurement errors σ, from inherent process stochasticity γ (refer to the flat surface seen in the Euler-SAEM convergence diagnosis).

We see, all in all, that for the simple model we will use, the convergence is satisfactory, and will allow us to run the structural misfit test in section 11.

Chapter 4

Simulation study

10 Compartmental Models in pharmacology

The methods studied in this report all had pharmacology in mind as an application; pharmacology is the study of the action of drugs and medication in organisms, namely the human body.

The individuals in our population are patients, to whom we give a certain dose of a drug; then, after a fixed amount of time, we re-administer the same dose to each patient.
The measured process, C_t , will be the drug concentration in the patient's blood. Knowing, and being able to predict the variation of a drug's concentration in someone's body allows for precision and individualised dosing, which allows for optimization of drug effect, in tandem with harm reduction from secondary effects.
In certain specific cases, such as with chemotherapy, it is crucial not to give a patient too high of a dose, at the risk of having him too immunosuppressed. This drug dose optimization, as explained in Maier C et al [8] , is key to give a patient the correct therapy.
It is clear to say that a structural misspecification in such a model could lead to disastrous consequences.

A very widespread model for drug concentration in pharmacology is the compartment model.
The human body is modeled as a certain amount of compartments, each representing an organ, or a part of the body: the blood, the liver, the lymphatic system, the kidneys, the plasma, and so on. Each of these compartments represent an important function in the metabolization of the studied drug.

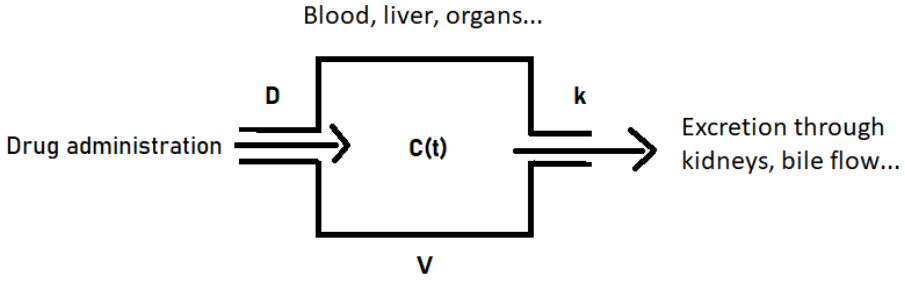

Figure 4.1: Sketch of a one-compartment pharmacological model

10.1 The studied model

In our case, we will consider a model with only one compartment. We choose this very simplified model for clarity, and for a faster implementation of the SAEM algorithm that we coded.

Figure (3.1) shows how the model works: first, we administer a certain amount D of the drug to the patient at times $\tau_1, \ldots, \tau_{n_{cyc}} \in \Delta$, creating then n_cyc sequential cycles; this might be a bolus/IV infusion (which is almost instantaneous) , or an oral administration (which will make C increase gradually, as opposed to the almost Dirac-like peak gotten from IV infusion).
Then, the drug is excreted, with a clearance parameter ϕ. The more of it you have in your system, the more the body will excrete it; and so, we consider a proportional clearance of it. In total, we get the model equation

$$\mathrm{d}C_t = -\phi C_t \,\mathrm{d}t + \frac{D}{V}\delta_\Delta \,\mathrm{d}t + \gamma \,\mathrm{d}W_t \tag{4.1}$$

where k and V are individual parameters of the model, and K is a constant.
In what follows, we will however consider V to be a known constant; we have checked that the test power varies very little when we fix V , and we will use the simplified model for simplicity of implementation.

Another clearance model is possible, however. Usually, the body has a limit as to how much of the drug can be flushed out of its system; this would suggest a saturable clearance, and we would change equation (3.1) to

$$\mathrm{d}C_t^{sat} = -\phi \frac{C_t^{sat}}{K + C_t^{sat}} \,\mathrm{d}t + \frac{D}{V}\delta_\Delta \,\mathrm{d}t + \gamma \,\mathrm{d}W_t \tag{4.2}$$

Notice that we have

$$\mathrm{d}C_t^{sat} \approx \mathrm{d}C_t \text{ if } C_t^{sat} << K$$
$$\mathrm{d}C_t^{sat} \approx -\phi \,\mathrm{d}t + \frac{D}{V}\delta_\Delta \,\mathrm{d}t + \gamma \,\mathrm{d}W_t \text{ if } C_t^{sat} >> K$$

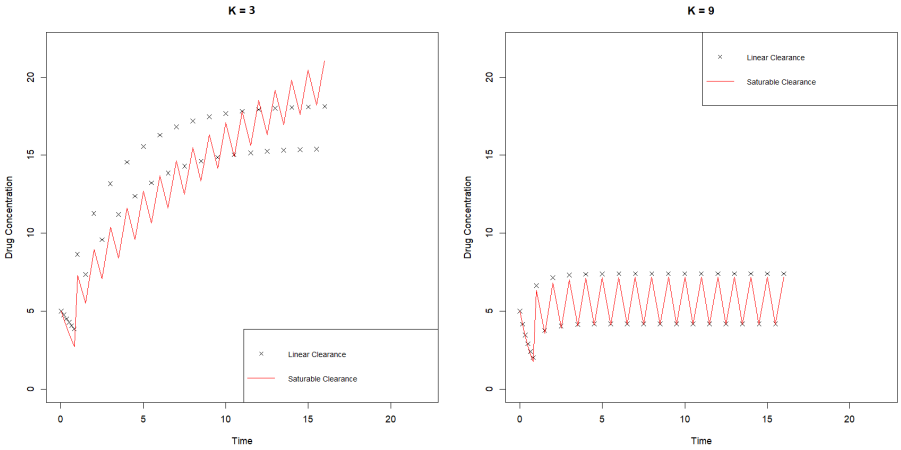

Figure 4.2: Two plots, for different K values, of C_t^{sat} and its (ODE) linear clearance fit

The non-saturable model is used in practice, with real data, as, for instance, to model absorption and clearance of the drug warfarin [11], a drug which prevents blood clots, and the saturable model has been used to model the absorption of phenytoin [12], an anti-epileptic medication.

A saturable model can be very similar to a non-saturable one, and as shown in figure (3.2) , they can be very hard to distinguish, graphically and numerically. Being able to distinguish between the two can as such be quite challenging, even if done heuristically.

We choose $C_0 = 5$, $\frac{D}{V} = 5$, $\Delta = \{1, 2, \ldots, 16\}$, $n_{cyc} = 16$, and K will vary according to the experiments that we do.
The measurements are taken n>2 times, uniformly, in the first cycle, and then we take measurements right before and right after each new dose.

10.2 Model distinction

We may wonder how different the two structural models are, in a more quantitative manner. For this, we will compare the trajectory means of the SDE solutions, that is, the solutions of the ODEs generated by the different drift functions.

Let us first consider the case where $n_{cyc} = 1$ (that is, with only one drug administration).
We write $X_{\phi,k}(t) = \mathbb{E}[C_t^{sat}]$ the trajectory average of a saturable clearance process, and $Y_\psi(t)$ the same for a non-saturable clearance.

The following theorem will give inequalities involving the distance between $X_{\phi,K}$ and its best non-saturable fit.

Theorem 1. *For all ϕ , T , there are constants $C_1 > 0$, $C_2 > 0$, $C_3 > 0$, such that for all $t \in [0,T]$, all $K > 0$*

$$\min_{\psi} \mid X_{\phi,K}(t) - Y_{\psi}(t) \mid \leq \frac{C_1 t}{K^2} \tag{4.3}$$

$$\min_{\psi} \mid X_{\phi,K}(t) - Y_{\psi}(t) \mid \geq C_2 - C_3 t K \tag{4.4}$$

Proof. For inequality (4.3), we consider the process $Y_{\phi/K}$, with initial value X_0; then, we have

$$\begin{aligned}
\frac{\mathrm{d}(X_{\phi,K} - Y_{\phi/K})}{\mathrm{d}t}(t) &= -\frac{\phi}{K}\left(X_{\phi,K}(t) - \frac{Y_{\phi/K}(t)}{1 + X_{\phi,K}(t)/K}\right) \\
&= -\frac{\phi}{K}\big(X_{\phi,K}(t) - Y_{\phi/K}(t)\big) + \frac{\phi}{K^2}\frac{X_{\phi,K}(t)^2}{1 + X_{\phi,K}(t)/K} \\
&\leq -\frac{\phi}{K}\big(Y_{\phi/K}(t) - X_{\phi,K}(t)\big) - \frac{\phi}{K^2}X_0^2
\end{aligned}$$

Where we used the positivity and monotonicity of X. Moreover,

$$\frac{\mathrm{d}(Y_{\phi/K} - X_{\phi,K})}{\mathrm{d}t}(t) \leq -\frac{\phi}{K}(Y_{\phi/K}(t) - X_{\phi,K}(t))$$

Thus, using Gromwall's lemma, we have

$$0 \leq X_{\phi,K}(t) - Y_{\phi/K}(t) \leq \frac{X_0^2}{K}(1 - e^{-\frac{\phi}{K}t}) \leq \frac{\phi X_0^2}{K^2}t \tag{4.5}$$

which yields inequality (4.3).
For (4.4), if we consider $Z(t) = X_0 - \phi t$, then

$$0 \leq (X'_{\phi,K} - Z')(t) = \frac{\phi K}{X_{\phi,K}(t) + K} \leq \frac{\phi}{X_{\phi,K}(T)}K$$

Then, supposing that $X_0 - \phi T > 0$,

$$\mid X_{\phi,K}(t) - Z(t) \mid \leq \frac{\phi}{X_0 - \phi T}Kt$$

Now, since Z is not in the closure of $\mathcal{E}_- = \{t \to Ce^{-at}, a > 0, C > 0\}$, then $d(Z, \mathcal{E}_-)_\infty = C_2 > 0$, which yields (4.4) through a triangle inequality. □

Thanks to this theorem, we may consider that we have made a good model choice for our test: now we know that, by increasing K, we may test the misfit for a model arbitrarily close to the real one. And so, we can "test the limits" of our test.
We may also check that decreasing K will indeed increase the test power, to see if the test efficiency is consistent with how different two models are, de facto.

11 Results and Discussion

Now that we have a model, and the necessary algorithm to estimate the target parameters, we can run the structural misfit test on simulated data, in order to estimate empirically the test power.

We generate data through a classic ODE Euler scheme, either using a saturable or a non-saturable model, as per the models in 10.1. Then, we test for structural misfit.

In all generated data, we kept $\frac{\phi_{pop}}{K}$ as a constant, equal to 1.75, and $\Omega = 0.1$.
We also considered V first to be a known constant with no inter-individual variability, equal for all individuals: another implementation and battery of tests including V as an individual parameter showed us that the inclusion of volume variability did not visibly influence the power of the test.

For our tests, we varied 3 different design parameters: K, n, σ and I:

- varying K, as shown 10.2, changes how similar the two opposing models are, and allows us to see how distinct 2 models have to be to be separated by our test.
- varying σ allows us to see to what extent noise can muddle the test results
- n is the number of measurements in the first cycle; varying it allows us to see how well the test fares when the data is sparse.
- I will show us how increasing the population size (and thus, the amount of data) will affect the test power

We ran the tests using both the Euler and the Kalman filter method, coupled with SAEM. For the case $I = 1$, which was studied during a previous master's internship in Potsdam university, only the Kalman method was used, without the need for an SAEM-type algorithm (since a population approach for 1 individuals does not make sense, and yields a non-identifiable problem).

For the Euler method, we had to choose a value of β for the proposal distribution, as per the problem that we raised in chapter 3.
We decided to fix a "reference value" σ_{ref} of σ (that we took to be σ^*), and chose the β value for which we get σ_{ref} as an estimate (basically, we ran SAEM

	Test power	I = 1	I = 5	I = 10	I = 20
Design 1	K = 7, $\sigma^2 = 0.05$, n = 11	48 %	100 %	100 %	100 %
Design 2	K = 7, $\sigma^2 = 0.05$, n = 3	37 %	100 %	100 %	100 %
Design 3	K = 9, $\sigma^2 = 0.1$, n = 11	12 %	55 %	73 %	83 %
Design 4	K = 9, $\sigma^2 = 0.1$, n = 3	10 %	35 %	42 %	58 %
Design 5	K = 7, $\sigma^2 = 0.1$, n = 11	29 %	95 %	100 %	100 %
Design 6	K = 7, $\sigma^2 = 0.1$, n = 3	22 %	90 %	98 %	100 %

Table 4.1: Kalman Filter Method test powers

	Test power	I = 1	I = 10	I = 20
Design 1	K = 7, $\sigma^2 = 0.05$, n = 11	55 %	53 %	77 %
Design 2	K = 7, $\sigma^2 = 0.05$, n = 3	37 %	43 %	60 %
Design 3	K = 9, $\sigma^2 = 0.1$, n = 11	12 %	21 %	20 %
Design 4	K = 9, $\sigma^2 = 0.1$, n = 3	10 %	19 %	21 %

Table 4.2: Euler-Maruyama Method test powers

with ~ 20 different β values, and realized a regression to estimate a β value to be used).

The results are summarized in tables 4.1 and 4.2. The type I error ranges between 3% and 8%.

The first thing that we notice, is that a Kalman Filter -based method outperforms an Euler-Maruyama implementation in every considered design choice; the estimated test power is much higher when using a Kalman filter.
We could attribute this to the fact that SAEM convergence with an Euler method is shaky, as per chapter 3. However, we should also consider the fact that the model design also favour a Kalman filter, theoretically: the drift function is linear between doses (i.e when the data is sparse); and when we find non-linearities, the time points are very close (as the measurements are taken just before and just after dosing). And so, the main downside of a Kalman filter, which is the linearization of the drift between data points, is not very present in our case study.
Though, seeing as this model design, dosing regimen and measurement times are very commonplace in pharmacology, our choice of model is quite relevant, and the use of a Kalman filter is to be considered. And so, our discussion will mainly focus on the results yielded by a Kalman-SAEM coupling.

- As expected with section 10.2, increasing K will decrease the test power. Its efficacy is thus consistent with model similarity, and models more distant (in the "L^∞ norm for their respective processes sense) are easier to distinguish using our Monte-Carlo test. The choice of K is very defining of the test power: for 5 individuals, changing K from 7 to 9 in design 5 makes the test go from almost flawless to not very good (95 % $\rightarrow$ 55 %)

- Increasing the size of the dataset, either longitudinally (with n), or trough population size (with I) also increases test power. This is a quite obvious and expected result. What really is interesting, however, is the drastic increase in test power when going from a single individual study (I = 1) to a population setting: in design 3, we see an increase of almost 5 times the power (12% $\rightarrow$ 55%) , and in design 2, an increase almost threefold (37% $\rightarrow$ 100%) with only 5 individuals; in design 6, we go from a poor test power (22 %) to a very good test efficiency (90 % power). The jump from a single individual study to a population approach is very effective.

- In the master's internship preceding this one [13], it was also found that the test for a single individual was very numerically unstable; in our population approach, we did not find any divergence or numerical instability when running our algorithm.

- We also find that increasing the noise level σ decreases test power, more so when K is large then when it is small: the noise σ has to be of the order of the distance between models to have a significant impact on the test power. When $K = 7$, the power decrease of increasing σ^2 from 0.05 to 0.1 is barely noticeable, with variations in power of $\leq 5\%$ in the population approach; when $K = 9$, however, the drop in power is much larger: $\geq 20\%$. Numerically, using R simulations and non-linear individual fits, we saw that the L^∞ distance between a saturable process with $K = 9$ and its best non-saturable fit is of order 0.1.

All in all, we see that the test has very positive results, and that the misfit diagnosis (and thus our method) works extremely well with our model. When looking at figure 4.2, we see that distinguishing visually and graphically a saturable model from a non-saturable one is nearly impossible when K becomes large ($\sim$ 9 or 7). However, we find extremely good results for $K = 7$, as with just 5 individuals we get a perfect test power of 100 %; for $K = 9$, that is, with extremely similar models, and only 20 individuals, we find very good results with frequent measurements (84 %), and a decent test power with sparse data (58 %).
The test seems to scale very well with the amount of patients considered, and can confidently automate a diagnosis and selection of structural model that would have otherwise been done manually and graphically.
However, the test (or at least our implementation of it) was very computationally expensive, with a run time of sometimes more than one or two days, when using only 20 individuals. When treating very rich and extensive data (with 100

or 1000 individuals), our algorithm would simply take too long; a more efficient implementation, or another method would be in order.

Moreover, we should attempt to run this test on more complex models, either with more parameters and inter-parameter correlations, with more important non-linearities, or with covariates. A good performance of our misfit test on such a complex model could show definitively the usefulness and efficiency of it, as we showed in our considered simple model.

Chapter 5

Outlook

A mixed population model

In what precedes, we have always considered that either the whole population is correctly specified, or everyone else is misspecified.

We could, however, consider the possibility of having a mixed population: some individuals fit the model correctly, and some don't.
Some extreme conditions can account for such a discrepancy: obesity, interference with other treatments, very old/young age, and others. And so, being able to diagnose that an individual is misspecified, and identify him in a population can be crucial for treating him appropriately.
This task can be modelled in the following manner:

$$\mathrm{d}X_t^{(i)} = f(t, X_t, \phi_i)\,\mathrm{d}t + \gamma_i\,\mathrm{d}W_t \tag{5.1}$$

$$y_{i,j} = X_{t_{i,j}}^{(i)} + \epsilon_{i,j} \tag{5.2}$$

$$\epsilon_{i,j} \sim \mathcal{N}(0, \sigma^2 I_d) \text{ iid} \tag{5.3}$$

$$\phi_i \sim \Pi(\ \cdot\ ; \theta) \text{ iid} \tag{5.4}$$

$$\gamma_i \in \mathbf{1}_{c_i=0}\mathcal{N}(0, \delta_1^2) + \mathbf{1}_{c_i=1}\mathcal{N}(\gamma, \delta_2^2) \tag{5.5}$$

$$c_i \sim \mathcal{B}e(\pi_s) \tag{5.6}$$

This model is very similar to the one considered previously, with a main difference being the transformation of the diffusion term from a population parameter to an individual parameter.
Each individual has a different diffusion term, and so, the test could be extended to account for individuals separately within a population.
We have opted for a Gaussian mixture as a prior distribution for the diffusion term, which is dependent on a newly added latent, c_i : simply put, c_i denotes to which "subpopulation" the individual i belongs: the correctly specified subpopulation, or the misspecified one. If $c_i = 1$, then $\gamma_i \sim \mathcal{N}(\gamma, \delta_2^2)$ indicates a

structural misspecification, as per the test we considered in this report.

In this model, the population parameters to be estimated by SAEM are thus $\theta, \sigma^2, \delta_1, \delta_2, \gamma, \pi_s$.
The model is thus considerably more complex, and may also be non-identifiable: in the plane $\{\pi_s = 0\}$, the likelihood is constant when varying γ and δ_2. A population where all individuals are correctly specified is not identifiable. Moreover, we would have to fix γ above a certain threshold, since in the plane $\{\gamma = 0\}$, the parameter π_s is non-identifiable.

Population estimation in such a model is also quite challenging; while trying to code such an estimator in R, we found a problem with the classic SAEM algorithm: the Markov chain created by Metropolis-Hastings is no longer ergodic, because of the existence of absorbing states. More specifically, in M-H-within-Gibbs, at step k:

- We simulate $c_i \sim \mathcal{B}e(\pi_s^{(k)})$
- However, if $\pi_s^{(k)} = 0$ or $\pi_s^{(k)} = 1$, then all c_i will have the same value as π_s.
- In the optimization step, we set $\pi_s^{(k+1)}$ as the empirical average of the c_i; if $\pi_s^{(k)} = 0$, then $\pi_s^{(k+1)} = 0$.
- We have the presence of 2 absorbing states, which can stop prematurely SAEM.

M. Lavielle [9] proposed a new algorithm, an extension of SAEM called MSAEM to circumvent this issue; however, it is only applied to a classic ODE model; with no diffusion term. We have not yet seen a Kalman-MSAEM coupling in the literature, and did not have time to try to formulate and implement it.

A proposed method for identifying misspecified individuals within a population

Here is a proposal for such a method:

1. We estimate the population parameters $\hat{\theta}$ with our data, using the mixture model previously presented, through MSAEM.
2. $\hat{\pi}_s$ indicates the estimated fraction of misspecified individuals within our population
3. We realize a MAP (maximum a posteriori) to estimate the individual parameters γ_i: this is done by maximizing

$$p_{\hat{\theta}}(\phi_i, \gamma_i, c_i \mid y_i) \propto p_{\hat{\theta}}(y_i \mid \phi_i, \gamma_i, c_i) p_{\hat{\theta}}(\phi_i, \gamma_i, c_i)$$

4. Then, the individuals with $c_i = 1$ are considered misspecified; we may also consider the $I(1-\hat{\pi}_s)$ individuals with highest γ_i values to be misspecified.

This method, however, may not work well in situations where the whole population is either misspecified or correctly specified, in part because of the identifiability issue. It also requires an implementation of the MSAEM algorithm for SDEs, which may prove challenging to tune correctly, and we have not found an implementation, or theoretical article in the literature.

Bibliography

[1] Stochastic Differential Equations in NONMEM Implementation, Application, and Comparison with Ordinary Differential Equations,Christoffer W. Tornøe, Rune V. Overgaard,2 Henrik Agersø, Henrik A. Nielsen, Henrik Madsen, and E. Niclas Jonsson

[2] Coupling the SAEM algorithm and the extended Kalman filter for maximum likelihood estimation in mixed-effects diffusion models, Maud Delattre and Marc Lavielle, Statistics and Its Interface Volume 6 (2013) 519–532

[3] Donnet, S. and Samson, A. (2008). Parametric inference for mixed models defined by stochastic differential equations. ESAIM: Probability and Statistics 12 196–218. MR2374638

[4] M Lavielle, *Some EM-type algorithms for incomplete data model building* 2021. hal-03512130

[5] Bernard Delyon, Marc Lavielle and Eric Moulines , *Convergence of a stochastic approximation version of the EM algorithm* , IRISA/INRIA Université Paris V and Université Paris-Sud and Ecole National Supérieure des Télécommunications

[6] Kushner, H. and Clark, D. (1978). *Stochastic Approximation for Constrained and Unconstrained Systems* Springer, New York.

[7] Rune Juhl, Jan Kloppenborg Møller, and Henrik Madsen. Continuous time stochastic modeling in r-user's guide and reference manual. Available at http://ctsm.info/ctsmr-reference.pdf (2021/10/14).

[8] Maier C, de Wiljes J, Hartung N, Kloft C, Huisinga W. A continued learning approach for model-informed precision dosing: Updating models in clinical practice. CPT Pharmacometrics Syst Pharmacol. 2022;11:185-198. doi:10.1002/psp4.12745 21638306, 2022, 2,

[9] An improved SAEM algorithm for maximum likelihood estimation in mixtures of non linear mixed effects models

September 2014Statistics and Computing 24(5)

DOI:10.1007/s11222-013-9396-2

[10] Population stochastic modelling (PSM) - An R package for mixed-effects models based on stochastic differential equations, Klim S, Mortensen SB, Kristensen NR, Overgaard RV and Madsen H (2009), Computer Methods and Programs in Biomedicine, 94(3), pp. 279-289.

[11] Xue L, Holford N, Ding XL, Shen ZY, Huang CR, Zhang H, Zhang JJ, Guo ZN, Xie C, Zhou L, Chen ZY, Liu LS, Miao LY. Theory-based pharmacokinetics and pharmacodynamics of S- and R-warfarin and effects on international normalized ratio: influence of body size, composition and genotype in cardiac surgery patients. Br J Clin Pharmacol. 2017 Apr;83(4):823-835. doi: 10.1111/bcp.13157. Epub 2016 Nov 25. Erratum in: Br J Clin Pharmacol. 2017 Jul;83(7):1602. PMID: 27763679; PMCID: PMC5346875.

[12] Tanaka J, Kasai H, Shimizu K, Shimasaki S, Kumagai Y. Population pharmacokinetics of phenytoin after intravenous administration of fosphenytoin sodium in pediatric patients, adult patients, and healthy volunteers. Eur J Clin Pharmacol. 2013 Mar;69(3):489-97. doi: 10.1007/s00228-012-1373-8. Epub 2012 Aug 24. PMID: 22918614; PMCID: PMC3572369.

[13] Tom Rodenhagen, Stochastische Differentialgleichungen zur Behandlung von Missspezifkationen in der Pharmakokinetik, Master thesis, Uni Potsdam, Oct 2021 https://www.math.uni-potsdam.de/professuren/mathematische-modellierung-und-systembiologie/abschlussarbeiten