



LUND UNIVERSITY

FACULTY OF ENGINEERING, LTH

CENTER FOR MATHEMATICAL SCIENCES

DIVISION OF MATHEMATICAL STATISTICS

MASTER'S THESIS IN MATHEMATICAL STATISTICS

An Adaptive Iterated Filtering Algorithm

Samuel Wqvist

supervised by
Erik Lindstöm, Adj. prof.

May 24, 2016

Abstract

Iterated Filtering (IF) is a method used to compute maximum likelihood based model parameter estimations of partially observed Markov process models in settings where the maximum likelihood function is untraceable. The objective of this thesis is to present a modified and improved version of the IF algorithm.

This thesis focuses on the IF algorithm introduced in [Ionides et al., 2011], which is denoted IF1 and introduces a modified version of the IF1 algorithm: the Adaptive Iterated Filtering algorithm (AIF). The AIF algorithm incorporates an approximation of the Hessian matrix into the IF framework and thereby utilizes second-order information to construct an optimal gain sequence.

Simulation studies show that the AIF algorithm performs better or just as good as the IF1 algorithm for the simple AR(1) model. However, the AIF algorithm is inferior when more advanced models are employed. This is probably due to the poor approximation of the Hessian matrix in the AIF algorithm.

The main conclusion is that the AIF algorithm might be a promising version of the IF algorithm if a better approximation of the Hessian matrix is obtained.

Keywords: Iterated Filtering, partially observed Markov processes, Adaptive Iterated Filtering, Hessian matrix, maximum likelihood

Acknowledgements

Firstly, I would like to thank Erik Lindstöm for proposing a master's thesis about the Iterated Filtering algorithm. During the spring of 2016 I have also had many pleasant and fruitful discussions with Erik about the progress of the thesis and how to proceed with the project. These discussions have been vital for me to be able to complete the thesis.

I would also like to thank my friends and family for the kind of support that only family and close friends can provide.

Table of Contents

Abstract	iii
Acknowledgements	v
Abbreviations and Mathematical Notation	ix
1 Introduction	1
1.1 Background	1
1.2 Problem Formulation	2
1.3 Outline	2
2 Theory for the Iterated Filtering Algorithm	5
2.1 Partially Observed Markov Processes	5
2.1.1 Mathematical Formulation	5
2.1.2 Applications	6
2.2 The Iterated Filtering Algorithm and Its Versions	7
2.2.1 The IF1 Algorithm	7
2.2.2 The IF2 Algorithm	10
2.2.3 Efficient Iterated Filtering	14
2.2.4 Other Version of IF1	14
2.2.5 Applications of the Iterated Filtering Algorithm	14
3 The Adaptive Iterated Filtering Algorithm	15
3.1 Introduction	15
3.2 Mathematical Formulation	15
3.3 Implementation	17
4 Simulations	21
4.1 Simple AR(1) model	21
4.1.1 Simulation Results for the Simple AR(1) Model	21
4.2 The Gompertz Model	23
4.2.1 Simulation Results for the Gompertz Model	25
4.3 The SIR Model	29
4.3.1 Simulation Results for the SIR Model	32
4.4 Summary of Simulation Results	34

5	Discussion	41
5.1	General Discussion on the Iterated Filtering Algorithm	41
5.2	Discussion on the Adaptive Iterated Filtering Algorithm	42
5.3	Further Research	42
6	Conclusions	45
6.1	Main Conclusion and Findings	45
6.2	Final Thoughts	46

Abbreviations and Mathematical Notation

Abbreviation	Meaning
AIF	The Adaptive Iterated Filtering Algorithm (see Chapter 3)
IF	Iterated Filtering algorithm
IF1	The Iterated Filtering algorithm introduced in [Ionides et al., 2011]
IF2	The Iterated Filtering algorithm introduced in [Ionides et al., 2015]
ML	Maximum likelihood
POMP	Partially observed Markov Process
2SG	Second-order stochastic gradient
2SPSA	Second-order simulations perturbation stochastic approximation

Notation	Meaning
f	POMP model
f_{X_0}	Simulator for the start value of the Markov process
$f_{X_n X_{n-1}}$	Simulator for the propagation of the Markov process
$f_{Y_n X_n}$	Evaluator for the observation process
$f(x \mu, \sigma)$	Distribution function for the normal distribution with mean μ and standard deviation σ
$f(x \mu, \theta)$	Distribution function for the negative binomial distribution with mean μ and shape parameter θ
$\mathcal{N}(\mu, \sigma)$	Normal random numbers with mean μ and standard deviation σ
$\mathcal{U}(a, b)$	Uniform random numbers on the interval $[a, b]$
$\ell(\theta)$	The log-likelihood function

Chapter 1

Introduction

This thesis addresses maximum likelihood based model parameter estimations of partially observed Markov process (POMP) models by examining the Iterated Filtering algorithm and introducing an adaptive version of the Iterated Filtering algorithm.

1.1 Background

POMP models are models that consist of two processes, one observation process that is known and one unknown Markov process. POMP models are also known as Hidden Markov models or state-space models and have been used in many applications [Ionides et al., 2011]. Since a POMP model consists of an unknown process the maximum likelihood function is often untraceable, which implies that estimations of model parameters of POMP models are challenging. There are however several ways to estimate the model parameters of POMP models, including sequential Monte Carlo Methods, Bayesian parameter estimations and various maximum likelihood based methods [Kantas et al., 2009].

The Iterated Filtering algorithm is a maximum likelihood based algorithm used for estimation of model parameters of POMP models such that the estimated model parameters converge to the maximum likelihood estimation of the model parameters [Ionides et al., 2011]. The Iterated Filtering algorithm also has the favorable property of being a plug-and-play method [Ionides et al., 2011]. A plug-and-play method is a method that only requires simulations of the processes and it is therefore possible to fit more general models and the restrictions on the models are also limited [He et al., 2009]. The plug-and-play method is similar to a *gradient free* algorithm in optimization theory (i.e. an algorithm that only requires the evaluation of the objective function and no information about the gradient) [He et al., 2009]. Because of its favorable properties the Iterated Filtering algorithm has been used to examine scientific questions in epidemiology [King et al., 2008], ecology [Chan and Tsay, 2011] and finance [Bretó, 2014].

The Iterated Filtering algorithm was first introduced in the paper *Inference for nonlinear dynamical systems* by Ionides in 2006. This algorithm was later developed into the Iterated Filtering algorithm (this algorithm will hereafter be denoted IF1) [Ionides et al., 2011]. Today, there exists mainly two versions of the Iterated Filtering algorithm, the IF1 algorithm and the IF2 algorithm, which was introduced in [Ionides et al., 2015]. These versions of the IF algorithm are however quite different. The IF1 algorithm replaces the true POMP model with a model that is similar, but perturbed and approximates the maximum likelihood estimation of the POMP model by estimating the moments of the perturbed process [Ionides et al., 2011]. The IF2 algorithm on the other hand is based on iteration of a Bayes map [Ionides et al., 2015]. There also exists other versions of the IF1 algorithm that provides better convergence properties than the IF1 algorithm [Ionides et al., 2012], [Lindström, 2013].

1.2 Problem Formulation

This thesis examines the Iterated Filtering algorithm and the objective is to provide an improved version of the IF1 algorithm. Hence, the objective is to introduce an improved version of the IF algorithm that estimates the model parameter of a POMP model with higher precision and faster convergence compared to the other IF algorithms.

The thesis focuses on the IF1 algorithm since the convergence properties of the IF1 algorithm and the role of the algorithm parameters are better understood compared to the IF2 algorithm.

The Iterated Filtering algorithm provides an algorithm to compute maximum likelihood based parameter estimations of complex POMP models, a problem that previously has been untraceable [Ionides et al., 2006]. Since POMP models are used in many scientific fields it is of high relevance to be able to estimate the parameters of general POMP models with high precision. It is therefore of interest and relevance to further develop the Iterated Filtering algorithm.

1.3 Outline

This thesis consists of five main chapters:

Chapter 2: A review of the partially observed Markov process model, including applications. Chapter 2 also includes a review of the Iterated Filtering algorithms (i.e. the IF1 algorithm and its versions and the IF2 algorithm) and their convergence properties. The chapter also includes pseud code for the algorithms including implementation guidelines.

Chapter 3: Introduces the Adaptive Iterated Filtering algorithm, including the motivation for the algorithm, the mathematical formulation and pseud code

for the Adaptive Iterated Filtering algorithm, including implementation guidelines

Chapter 4: Simulations studies for a simple AR(1) model, the Gompertz model and the SIR model. For each model are different versions of the Iterated Filtering algorithm compared.

Chapter 5: Both a general discussion on the Iterated filtering algorithm and a more specific discussion on the Adaptive Iterated Filtering and the simulation results. The chapter also provides a section on further research questions regarding the Iterated Filtering algorithm.

Chapter 6: The final chapter provides a synthesis of the key findings and the objectives stated in section 1.2 are also addressed. The chapter ends with a section that carries the final thoughts.

Chapter 2

Theory for the Iterated Filtering Algorithm

This chapter provides a review of the partially observed Markov process model and the different versions of the Iterated Filtering algorithm.

2.1 Partially Observed Markov Processes

In this section is the partially observed Markov process model introduced. The notation is similar to the notation in [Ionides et al., 2015].

2.1.1 Mathematical Formulation

A partially observed Markov process (POMP) model consists of two processes, one known observation process $Y_n \in \mathbb{R}^{\dim(Y_n)}$ and one unknown Markov process $X_n \in \mathbb{R}^{\dim(X_n)}$, the two processes are governed by the model parameters $\theta \in \mathbb{R}^{\dim(\theta)}$. Hence, the model consists of two sequences of random variables: $X_{0:N} = (X_0, X_1, \dots, X_N)$ and $Y_{1:N} = (Y_1, \dots, Y_N)$.

A POMP model is also known as a Hidden Markov model or stat-space model and in that framework is the process X_n the latent variable and process Y_n the observation process.

For a POMP model is the *joint density function* $f_{X_{0:N}, Y_{1:N}}(x_{0:N}, y_{0:N} \mid \theta)$ assumed. A POMP model is then specified by *initial density* $f_{X_0}(x_0 \mid \theta)$, *conditional transition density* $f_{X_n|X_{0:n-1}}(x_n \mid x_{0:n-1}, \theta)$ and *conditional transition density for the observation process* $f_{Y_n|X_n}(y_n \mid x_n, \theta)$. The propagation of the POMP model is governed by the conditional density functions in Equation 2.1 and 2.2.

$$X_n = f_{X_n|X_{0:n-1}}(x_n \mid x_{0:n-1}, \theta) \quad (2.1)$$

$$Y_n = f_{Y_n|X_n}(y_n \mid x_n, \theta) \quad (2.2)$$

The model also consists of data $y_{1:N}^* = (y_1^*, \dots, y_N^*)$, $y_n^* \in \mathbb{R}^{\dim(Y_n)}$.

By employing the Markov property of $X_{0:N}$ and the conditional independence of the observation process the *joint density* can be expressed as

$$f_{X_{0:N}, Y_{1:N}}(x_{0:N}, y_{0:N} \mid \theta) = \quad (2.3)$$

$$f_{X_0}(x_0 \mid \theta) \prod_{n=1}^N f_{X_n|X_{0:n-1}}(x_n \mid x_{0:n-1}, \theta) f_{Y_n|X_n}(y_n \mid x_n, \theta) = \quad (2.4)$$

$$f_{X_0}(x_0 \mid \theta) \prod_{n=1}^N f_{X_n, Y_n|X_{n-1}}(x_n, y_n \mid x_{n-1}, \theta) \quad (2.5)$$

The marginal likelihood function for the observations $f_{Y_{1:N}}(y_{0:N} \mid \theta)$ is obtained in Equation 2.6

$$f_{Y_{1:N}}(y_{0:N} \mid \theta) = \quad (2.6)$$

$$\int \dots \int f_{X_0}(x_0 \mid \theta) \prod_{n=1}^N f_{X_n, Y_n|X_{n-1}}(x_n, y_n \mid x_{n-1}, \theta) dx_0 \dots dx_N$$

The log-likelihood function $\ell(\theta)$ is expressed as

$$\ell(\theta) = \quad (2.7)$$

$$\log \int \dots \int f_{X_0}(x_0 \mid \theta) \prod_{n=1}^N f_{X_n, Y_n|X_{n-1}}(x_n, y_n^* \mid x_{n-1}, \theta) dx_0 \dots dx_N$$

The maximum likelihood estimation of the model parameters θ are the model parameters that maximize the probability for the data. Hence, the maximum likelihood estimation $\hat{\theta}^{ML}$ of the model parameters θ is obtained by maximizing the log-likelihood function

$$\hat{\theta}^{ML} = \arg \max_{\theta} \ell(\theta) \quad (2.8)$$

The log-likelihood function in Equation 2.7 is however not analytically tractable since the process X_n is unknown. It is therefore generally not possible to obtain the maximum likelihood estimation $\hat{\theta}^{ML}$ by Equation 2.8. However, it is possible to obtain a maximum likelihood based estimation of the parameters θ by employing the Iterated Filtering algorithm.

2.1.2 Applications

Partially observed Markov process models (or Hidden Markov models) are quite general models that have been used in many fields. Applications can for instance

be found in ecology [Chan and Tsay, 2011], epidemiology [King et al., 2008], finance [Bretó, 2014] [Johannes et al., 2009] and economics [Fernández-Villaverde and Rubio-Ramírez, 2007].

2.2 The Iterated Filtering Algorithm and Its Versions

This section provides a review of the Iterated Filtering algorithm, including the IF1 algorithm [Ionides et al., 2011], the IF2 algorithm [King et al., 2015], the Efficient Iterated Filtering algorithm (EIF) [Ionides et al., 2012] and other version of the IF1 algorithm [Lindström, 2013].

2.2.1 The IF1 Algorithm

The IF1 algorithm and the convergence properties of the IF1 algorithm were described by Ionides [Ionides et al., 2011]. This section provides a review of the IF1 algorithm and its convergence properties. The notation used in this section is similar to the notation in [Ionides et al., 2011].

Assume that the POMP model f consists of a Markov process X_n , observation process Y_n and model parameters θ that are constant and not time dependent. The main idea of the IF1 algorithm is to replace the POMP model f with the processes X_n and Y_n with a closely related but perturbed model with processes \check{X}_n , \check{Y}_n , the model parameters of the perturbed model $\check{\theta}_n$ are time dependent parameters and follow a random walk in the model parameter space. By using the perturbed model a sequence of model parameter estimations are constructed such that they converge to the maximum likelihood estimation of the model parameters of the original model.

Derivation of the IF1 Algorithm

Let f be a POMP model specified by *initial density* f_{X_0} , *conditional transition density* $f_{X_n|X_{0:n-1}}$ and *conditional transition density for the observation process* $f_{Y_n|X_n}$ with fixed model parameters θ . The model \bar{f} is an expansion of the model f with time-varying model parameters $\bar{\theta}_{0:N} = (\bar{\theta}_0, \dots, \bar{\theta}_N)$ such that $\bar{\theta}_j \in \mathbb{R}^p$ for each $0 \leq n \leq N$. The joint density of the model \bar{f} is expressed in Equation 2.9.

$$\begin{aligned} \bar{f}_{\bar{X}_{0:N}, \bar{Y}_{1:N}}(x_{0:N}, y_{0:N} \mid \bar{\theta}) = \\ f_{X_0}(x_0 \mid \theta) \prod_{n=1}^N f_{X_n, Y_n|X_{n-1}}(x_n, y_n \mid x_{n-1}, \bar{\theta}_n) \end{aligned} \quad (2.9)$$

The log likelihood function for the model \bar{f} is $\bar{\ell}(\bar{\theta}_{0:n}) = \log \bar{f}_{\bar{Y}_{1:n}}(y_{1:n}^* \mid \bar{\theta}_{0:n})$. Now also let $\theta^{[l]}$ denote l copies of the θ such that $\ell_n(\theta) = \bar{\ell}_n(\theta^{n+1})$

The following objects are now established: the time-varying model parameter process $\check{\Theta}_n$, $0 \leq n \leq N$, the perturbation parameters, τ and σ and the

density κ on \mathbb{R}^p with compact support, zero mean and covariance matrix Σ . The time-varying model parameter process $\check{\Theta}_n$ follows the processes in Equations 2.10 and 2.11 where Z_0, \dots, Z_N are independently drawn from the density κ

$$\check{\Theta}_0 = \bar{\theta}_0 + \tau Z_0, N = 0 \quad (2.10)$$

$$\check{\Theta}_j = \bar{\theta}_{n-1} + \sigma Z_n, 1 \leq n \leq N \quad (2.11)$$

Furthermore, the joint density of $\check{\Theta}_n$ is $g_{\check{\Theta}_{0:N}}(\check{\theta} \mid \bar{\theta}_{0:N}, \sigma, \tau)$. The perturbed POMP process g with Markov process $(\check{X}_n, \check{\Theta}_n, 0 \leq n \leq N)$, observation process $\check{Y}_{1:N}$ and parameters $(\bar{\theta}, \sigma, \tau)$ is also introduced. The joint density for the POMP model g is

$$g_{\check{X}_{0:N}, \check{\Theta}_{0:N}, \check{Y}_{1:N}}(x_{0:N}, \check{\theta}, y_{1:N} \mid \bar{\theta}_{0:N}, \tau, \sigma) = \quad (2.12)$$

$$g_{\check{\Theta}_{0:N}}(\check{\theta} \mid \bar{\theta}, \sigma, \tau) \bar{f}_{\check{X}_{0:N}, \check{Y}_{1:N}}(x_{0:N}, y_{1:N} \mid \check{\theta}) \quad (2.13)$$

The filtering mean and variances where $\bar{\theta}_{0:N} = \theta^{[n+1]}$ are now expressed as

$$\begin{aligned} \check{E}_{\theta^{[n+1]}, \sigma, \tau}(\check{\Theta}_n \mid \check{Y}_{1:N} = y_{1:N}^*) &= \int \check{\theta}_n g_{\check{\Theta}_n \mid \check{Y}_n}(\check{\theta}_n \mid y_{1:N}^*, \theta^{[n+1]}, \sigma, \tau) d\check{\theta}_n \quad (2.14) \\ &= \check{\theta}_n^F, \text{ for } n = 1 \cdots N, \text{ with } \check{\theta}_0^F = 0 \end{aligned}$$

$$\check{Var}_{\theta^{[n+1]}, \sigma, \tau}(\check{\Theta}_n \mid \check{Y}_{1:N} = y_{1:N}^*) = \check{V}_n^P, \text{ for } n = 1 \cdots N, \quad (2.15)$$

A key result in [Ionides et al., 2011] is that the gradient of the maximum likelihood function $\nabla l_N(\theta)$ can be approximated with the filtering mean and prediction variance in Equation 2.14 and 2.15. Hence the maximum likelihood function can be approximated with the moments of the perturbed POMP model. This result is presented in Theorem 2.2.1.

Theorem 2.2.1 (Theorem 3 in [Ionides et al., 2011]) *Let σ be a function of τ with $\lim_{\tau \rightarrow 0} \sigma(\tau)/\tau = 0$. For any compact set $K \in \mathbb{R}^p$, there exists a finite constant C such that for all τ small enough*

$$\sup_{\theta \in K} \left| \sum_{n=1}^N (\check{V}_n^P)^{-1} (\check{\theta}_n^F - \check{\theta}_{n-1}^F) - \nabla l_N(\theta) \right| \leq C(\tau + \sigma^2/\tau^2) \quad (2.16)$$

The filtering mean and prediction variance in Equations 2.14 and 2.15 are estimated using a sequential Monte Carlo method. Let $\tilde{\Theta}_{n,j}^F$ and $\tilde{\Theta}_{n,j}^P$ be estimation j in step n in the sequential Monte Carlo method of the filtered and

predicted value of θ . The estimation of the filtering mean $\tilde{\theta}_n^F$ and the estimation of the prediction variance \tilde{V}_n^P are calculated as

$$\tilde{\theta}_n^F = \frac{1}{J} \sum_{j=n}^J \tilde{\Theta}_{n,j}^F \quad (2.17)$$

$$\tilde{V}_n^P = \frac{1}{J} \sum_{j=n}^J (\tilde{\Theta}_{n,j}^P - \tilde{\theta}_{n-1}^F)^2 \quad (2.18)$$

Another important result in [Ionides et al., 2011] is Theorem 2.2.2, which states that it is possible to approximate the gradient of the maximum likelihood function using the Monte Carlo approximations of the filtering mean and predictive variance, hence a Monte Carlo version of Theorem 2.2.1.

Theorem 2.2.2 (*Theorem 4 in [Ionides et al., 2011]*) Let σ_m , τ_m and J_m be positive sequences with $\tau_m \rightarrow 0$, $\sigma_m \tau_m^{-1} \rightarrow 0$, $\tau_m J_m \rightarrow \infty$. Define $\tilde{\theta}_{n,m}^F = \tilde{\theta}_n^F(\theta, \sigma_m, \tau_m, J_m)$ and $\tilde{V}_{n,m}^P = \tilde{V}_n^P(\theta, \sigma_m, \tau_m, J_m)$ via Equation 2.17 and 2.18. Let K be any arbitrary compact subset of \mathbb{R}^p . Then,

$$\lim_{m \rightarrow \infty} \sup_{\theta \in K} |E_{MC} \left[\sum_{n=1}^N (\tilde{V}_{n,m}^P)^{-1} (\tilde{\theta}_{n,m}^F - \tilde{\theta}_{n-1,m}^F) \right] - \nabla \ell_N(\theta)| = 0 \quad (2.19)$$

$$\lim_{m \rightarrow \infty} \sup_{\theta \in K} |\tau_m^2 J_m \text{Var}_{MC} \left[\sum_{n=1}^N (\tilde{V}_{n,m}^P)^{-1} (\tilde{\theta}_{n,m}^F - \tilde{\theta}_{n-1,m}^F) \right]| < \infty \quad (2.20)$$

The key idea for the IF1 algorithm is to create a sequence of model parameter estimations $\hat{\theta}_m$ that converges to the maximum likelihood estimation $\hat{\theta}^{ML}$ of θ . Theorem 2.2.3 shows how the sequence of model parameter estimations $\hat{\theta}_m$ is formed by recursively update the model parameter estimations.

Theorem 2.2.3 (*Theorem 5 in [Ionides et al., 2011]*) Let a_m , σ_m , τ_m and J_m be positive sequences with $\tau_m \rightarrow 0$, $\sigma_m \tau_m^{-1} \rightarrow 0$, $\tau_m J_m \rightarrow \infty$, $\sum_m a_m = \infty$ and $\sum_m a_m^2 J_m^{-1} \tau_m^{-2} < \infty$. Specify a recursive sequence of parameter estimates θ_m

$$\hat{\theta}_{m+1} = \hat{\theta}_m + a_m \sum_{n=1}^N (\tilde{V}_{n,m}^P)^{-1} (\tilde{\theta}_{n,m}^F - \tilde{\theta}_{n-1,m}^F) \quad (2.21)$$

Where $\tilde{\theta}_{n,m}^F$ and $\tilde{V}_{n,m}^P$ are defined in Equations 2.17 and 2.18. Then $\lim_{m \rightarrow \infty} \hat{\theta}_m = \theta^{ML}$ with probability one.

Theorem 2.2.3 is the key contribution in [Ionides et al., 2011] since Theorem 2.2.3 states that, by choosing sequences that fulfil the conditions, the model parameter sequence $\hat{\theta}_m$ will converge to the maximum likelihood estimation θ^{ML} .

Pseudo Code

The pseudo code for the IF1 algorithm is presented in Algorithm 1. The pseudo code highlights the fact that the IF1 algorithm essentially consists of an iteration of a particle filter where the parameters are subject to a stochastic gradient update for each iteration. Hence, the IF1 algorithm consists of two nested loops. The inner loop is a particle filter in which the the filtering mean and prediction variance are estimated using a sequential Monte Carlo method. The outer loop updates the model parameters using the updating formula in Equation 2.21.

The algorithm parameters for the IF1 algorithm are: *Start value* θ_0 , *Data* y^* , *number of iterations* M , *number of time steps* N , *number of particles* J_m , *perturbation sequences* τ_m, σ_m , *cooling factor* a_m , *perturbation distribution* κ , *simulators* f_{X_0} and $f_{X_n|X_{n-1}}$, *evaluator* $f_{Y_n|X_n}$. The *Start value* is the initial guess for the IF1 algorithm and should be quite near the true parameter value. The *Data* is the data from the observation process Y . The parameters *number of iterations* M and *number of time steps* N and can be chosen quite freely. However, the parameters *number of particles* J_m , *perturbation sequences* τ_m, σ_m and *cooling factor* a_m have to fulfill the restrictions in Theorem 2.2.3.

The *perturbation distribution* κ also has to be specified to some distribution with compact support, zero mean and positive definite covariance matrix [Ionides et al., 2011]. In this thesis is the distribution κ set to a truncated Gaussian distribution.

Choosing Parameters

Theorem 2.2.3 states the conditions that the model parameters *number of particles* J_m , *perturbation sequences* τ_m, σ_m and *cooling factor* a_m have to fulfill in order for the IF1 algorithm to converge. A notable condition for the parameters is that Theorem 2.2.3 requires number of particles J_m to increase faster then τ_m^{-1} . An example of model parameters that fulfill Theorem 2.2.3 are: $J_m = m^{(\delta+1/2)}$, $\tau_m = \sqrt{m^{-1}}$, $\sigma_m = \sqrt{m^{-(1+\delta)}}$, cooling factor $a_m = m^{-1}$ where $\delta > 0$ [Ionides et al., 2011].

The cooling factor a_m determines how large the step sizes are, it is common to set the cooling factor as [Spall, 2005]

$$a_m = \frac{a}{(m + 1 + A)^\alpha} \quad (2.22)$$

Where a is chosen such that the size of the amplification of the gradient in the updating formula is suitable, $0.5 < \alpha \leq 1$ determines how fast the cooling factor decreases and A is a stabilizing part that is commonly set to 5 or 10 percent of M [Spall, 2005].

2.2.2 The IF2 Algorithm

This section provides a brief review to the theoretical framework of the IF2 algorithm, (however a more detailed outline can be fund in [Ionides et al., 2015]),

Algorithm 1 The IF1 algorithm describe in [Ionides et al., 2011]

Require: Start value θ_0 , Data y^* , number of iterations M , number of time steps N , number of particles J_m , perturbation sequences τ_m, σ_m , cooling factor a_m , parameter perturbation κ , simulators f_{X_0} and $f_{X_n|X_{n-1}}$, evaluator $f_{Y_n|X_n}$

Ensure: $\hat{\theta}^{IF1} = \hat{\theta}_M$

$\hat{\theta}_0 = \theta_0$ ▷ Set start value for $\hat{\theta}$

for $m = 1 \cdots M$ **do**

$Z_{0,j} \sim \kappa$, for $j = 1 \cdots J_m$ ▷ Draw from perturbation distribution

$\Theta_j^F = \hat{\theta}_{m-1} + \tau_m \cdot Z_{0,j}$, for $j = 1 \cdots J_m$ ▷ Initialize parameters

$X_{0,j}^F \sim f_{X_0}(\cdot | \Theta_j^F)$, for $j = 1 \cdots J_m$ ▷ Initialize states

$\bar{\theta}_0 = \frac{1}{J_m} \sum_j \Theta_j^F$ ▷ Calculate mean

for $n = 1 \cdots N$ **do**

$Z_n \sim \kappa$ for $j = 1 \cdots J_m$ ▷ Draw from perturbation distribution

$\Theta_j^P = \Theta_{n-1,j}^F + \sigma_m \cdot Z_{n,j}$ for $j = 1 \cdots J_m$ ▷ Perturb parameters

$X_j^P \sim f_{X_n|X_{n-1}}(\cdot | X_{n-1,j}^F, \Theta_j^P)$ for $j = 1 \cdots J_m$ ▷ Simulate particles

$w_j = f_{Y_n|X_n}(y_n^* | X_j^P, \Theta_j^P)$ for $j = 1 \cdots J_m$ ▷ Evaluate weights

$w_j = \frac{w_j}{\sum_{j=1}^{J_m} w_j}$ for $j = 1 \cdots J_m$ ▷ Normalize weights

$k_j = j$ with probability $\frac{w_j}{\sum_{j=1}^{J_m} w_j}$ for $j = 1 \cdots J_m$ ▷ Systematic

re-sampling of indexes

$X_{n,j}^F = X_{n,k_j}^P, \Theta_{n,j}^F = \Theta_{n,k_j}^P$ for $j = 1 \cdots J_m$ ▷ Re-sampling

$\bar{\Theta}_n = \frac{1}{J_m} \sum_j \Theta_j^F$ ▷ Calculate mean

$\mathbf{Var}(\theta_n) = \frac{1}{J_m} \sum_j (\theta_{n,j}^P - \bar{\Theta}_{n-1})^2$ ▷ Calculate variance

end for

$\hat{\theta}_{m+1} = \hat{\theta}_m + a_m \cdot \sum_n \mathbf{Var}(\theta_n)^{-1} \cdot (\bar{\theta}_n - \bar{\theta}_{n-1})$ ▷ Update $\hat{\theta}$

end for

return $\hat{\theta}^{IF1} = \hat{\theta}_M$

and a description of the pseudo code. The notation used is similar to the notation in [Ionides et al., 2015].

Mathematical Formulation

The main idea of the IF2 algorithm is to combine the data cloning methodology with model parameter perturbation. This creates a stable algorithm where the iterated Bayes map (i.e. the model particle swarm) converges to the maximum likelihood estimation of the model parameters. The IF2 algorithm is therefore an algorithm that has a similar model parameter perturbation structure as the IF1 algorithm, but where a iterated Bayes map is introduced to obtained a algorithm where the model parameter swarm converges to the maximum likelihood estimation of the model parameter. [Ionides et al., 2015]

The perturbation distribution of the IF2 algorithm, h_n , is ofter set to multivariate Gaussian distribution. In Equation 2.23 are all perturbations combined.

$$h(\theta_{0:N} \mid \varphi, \sigma) = h_0(\theta_0 \mid \varphi, \sigma) \prod_{n=1}^N h_n(\theta_n \mid \theta_{n-1}, \sigma) \quad (2.23)$$

The likelihood function is defined as

$$\begin{aligned} \check{\ell}(\theta_{0:N}) = & \int \dots \int d_{x_0} \dots d_{x_N} \left\{ f_{X_0}(x_0 \mid \theta_0) \right. \\ & \cdot \prod_{n=1}^N f_{X_n|X_{n-1}}(x_n \mid x_{n-1}, \theta_n) f_{Y_n|X_n}(y_n^* \mid x_n, \theta_n) \left. \right\} \end{aligned} \quad (2.24)$$

The IF2 algorithm then approximates the map in Equating 2.25 for each iteration of the algorithm.

$$T_\sigma f(\theta_N) = \frac{\int \check{\ell}(\theta_{0:N}) h(\theta_{0:N} \mid \varphi, \sigma) f(\varphi) d\varphi d\theta_{0:N-1}}{\int \check{\ell}(\theta_{0:N}) h(\theta_{0:N} \mid \varphi, \sigma) f(\varphi) d\varphi d\theta_{0:N}} \quad (2.25)$$

It is shown that the IF2 algorithm approximately maximizes the likelihood, with an error decreasing to zeros as $\sigma \rightarrow 0$ and $M \rightarrow 0$, for a fixed σ [Ionides et al., 2015].

Pseudo Code

The pseudo code for the IF2 algorithm is presented in Algorithm 2. The algorithm parameters for the IF2 algorithm are: *Initial particle swarm* $\Theta_{0,j}$, $j = 1 \dots J$, *Data* y^* , *number of iterations* M , *number of time steps* N , *number of particles* J , *perturbation density* h_n , *perturbation sequence* σ_m , *simulators* f_{X_0} and $f_{X_n|X_{n-1}}$, *evaluator* $f_{Y_n|X_n}$. Where *Data* y^* , *number of iterations* M , *number of time steps* N , *simulator* f_{X_0} , *simulator* $f_{X_n|X_{n-1}}$ and *evaluator* $f_{Y_n|X_n}$ are similar to the IF1 algorithm. The *initial particle swarm* $\Theta_{0,j}$

corresponds to the *start value* of the IF1 algorithm. The *perturbation sequence* determines how fast the variance of the *perturbation density* should decay, where the *perturbation density* is a density function (typically a multivariate Gaussian distribution) that governs the dispersion of the particles. Hence, most of the algorithm parameters of the IF2 algorithm correspond to algorithm parameters of the IF1 algorithm and the two algorithm parameters that are unique for the IF2 algorithm are the *perturbation sequence* σ_m and the *perturbation density* h_n , which govern the decrease in variance and the dispersion of the model parameter swarm.

Another important difference between the IF1 and IF2 algorithm is that the updating formula for the model parameters that is characteristic for IF1 algorithm is not present in the IF2 algorithm. Instead are the model parameters in the IF2 algorithm, (i.e. the particles), updated by updating the particle swarm with the N :th particle swarm in the former M-step.

Algorithm 2 The IF2 algorithm describe in [Ionides et al., 2015]

Require: Initial particle swarm $\Theta_{0,j}, j = 1 \dots J$, Data y^* , number of iterations M , number of time steps N , number of particles J , perturbation density h_n , perturbation sequence σ_m simulators f_{X_0} and f_{Y_0} , simulator $f_{X_n|X_{n-1}}$, evaluator $f_{Y_n|X_n}$

Ensure: $\hat{\theta}^{IF2} = \Theta_j^M$, for $j = 1 \dots J$

for $m = 1 \dots M$ **do**

$\Theta_{j,m}^F = h_0(\theta | \Theta_j^{m-1}, \sigma_m)$, for $j = 1 \dots J$ ▷ Initialize parameters

$X_{0,j}^F \sim f_{X_0}(\cdot | \Theta_j^F)$, for $j = 1 \dots J$ ▷ Initialize states

for $n = 1 \dots N$ **do**

$\Theta_{j,m}^P = h_0(\theta | \Theta_{n-1,j}^{F,m-1}, \sigma_m)$, for $j = 1 \dots J$ ▷ Perturb parameters

$X_j^P \sim f_{N_n|X_{n-1}}(\cdot | X_{n-1,j}^{F,m}, \Theta_j^{P,m})$, for $j = 1 \dots J$ ▷ Simulate particles

$w_j = f_{Y_n|X_n}(y_n^* | X_{n,j}^P, \Theta_{n,j}^{P,m})$, for $j = 1 \dots J$ ▷ Evaluate weights

$w_j = \frac{w_j}{\sum_{j=1}^J w_j}$, for $j = 1 \dots J$ ▷ Normalize weights

$k_j = j$ with probability $\frac{w_j}{\sum_{j=1}^J w_j}$, for $j = 1 \dots J$ ▷ Systematic

re-sampling of indexes

$X_{n,j}^F = X_{n,k_j}^P$, $\Theta_{n,j}^F = \Theta_{n,k_j}^P$ for $j = 1 \dots J$ ▷ Re-sampling

end for

$\Theta_j^m = \Theta_{N,j}^F$ for $j = 1 \dots J$ ▷ Update Θ

end for

return $\hat{\theta}^{IF2} = \Theta_j^M$, for $j = 1 \dots J$

Choosing Parameters

The IF2 algorithm is in some sense a more simple algorithm compared to the IF1 algorithm since the perturbation is governed by only two parameter the *perturbation sequence* σ_m and *perturbation density* h_n .

The *perturbation density* is typically set to a multivariate Gaussian distribution (the only type of *perturbation density* that is used in this thesis). However, it might be possible to obtain a better IF2 algorithm by varying the *perturbation density*, but that is not a question covered in this thesis. The *perturbation sequence* σ_m should decay fast enough to allow a rapid convergence, however the *perturbation sequence* should also be large enough to allow the model parameters to explore the model parameter space even for large values of M . In this thesis is the *perturbation sequence* often set to $\sigma_m = \alpha^{0:M-1}$ where $0.9 \leq \alpha < 1$.

2.2.3 Efficient Iterated Filtering

The Efficient Iterated Filtering (EIF) algorithm is another version of the IF algorithm introduced in [Ionides et al., 2012]. The updating formula in the Efficient Iterated Filtering algorithm is modified to by incorporating a recursive estimation. It is also shown that the EIF algorithm is more efficient compared to the IF algorithm.

2.2.4 Other Version of IF1

Another version of the IF algorithm is the Tuned Iterated Filtering algorithm [Lindström, 2013]. In the Tuned Iterated Filtering algorithm are the algorithms parameters tuned by estimating the Σ matrix in two different ways. Simulations studies show that the Tuned Iterated Filtering algorithm outperforms the standard IF algorithm.

2.2.5 Applications of the Iterated Filtering Algorithm

The Iterated Filtering algorithm, or rather different versions of the Iterated Filtering algorithm have mostly been applied in various ecological and epidemiological applications. As an example, the IF algorithm has been used to model cholera in [King et al., 2008] and [Ionides et al., 2015]. However, the IF algorithm has also been used in financial applications [Bretó, 2014].

Chapter 3

The Adaptive Iterated Filtering Algorithm

This chapter introduces the Adaptive Iterated Filtering algorithm (AIF), which is a version of the IF1 algorithm where second-order information is used to construct the gain sequence.

3.1 Introduction

The gain sequence a_m has a crucial impact on the performance of the IF1 algorithm. If the gain sequence is too large it is possible to obtain an algorithm that does not converge. Another common problem that occurs if the gain sequence is too large is that the step sizes for the few first iterations generates too *extrema* new model parameter guesses which causes the IF1 algorithm to fail. However, if the gain sequence is too small the step sizes will be small and the IF1 algorithm will converge slowly. The conclusion is that it is hard to choose an optimal gain sequence. The idea behind the Adaptive Iterated Filtering algorithm is to relate the gain sequence a_m to the Hessian matrix and to use second-order information in the update formula and thereby obtain an optimal gain sequence. This is done by introducing the second-order stochastic gradient algorithm (2SG) into the Iterated Filtering framework [Spall, 2000].

3.2 Mathematical Formulation

In this section is the second-order stochastic gradient algorithm (2SG) presented [Spall, 2000], the section also covers the incorporation of the 2SG algorithm into the Iterated Filtering setting. The notation will be similar to the notation used in [Spall, 2000].

The IF1 algorithm can be viewed as stochastic gradient algorithm since the update formula in Equation 2.21 is a Robbins–Monro algorithm type formula.

The asymptotically optimal gain sequence for a stochastic approximation algorithm is known to be [Spall, 2005]

$$\mathbf{a}_k = \frac{\mathbf{H}(\hat{\theta})^{-1}}{k+1}, k \geq 0 \quad (3.1)$$

Hence, the optimal gain sequence is a matrix gain sequence that is related to the Hessian matrix $\mathbf{H}(\hat{\theta})$. The result in Equation 3.1 is mostly of theoretical interest since the Hessian matrix is hard to approximate [Spall, 2005].

The second-order stochastic gradient algorithm (2SG) is an extension of the standard Robbins–Monro algorithm. In the 2SG algorithm the Hessian is approximated. Hence, the 2SG algorithm incorporates second-order information. The two main recursions in the 2SG algorithm are [Spall, 2000]

$$\hat{\theta}_{k+1} = \hat{\theta}_k - a_k \bar{\bar{H}}_k^{-1} G_k(\hat{\theta}), \bar{\bar{H}}_k = f_k(\bar{H}_k) \quad (3.2)$$

$$\bar{H}_k = \frac{k}{k+1} \bar{H}_{k-1} + \frac{1}{k+1} \hat{H}_k, k = 0, 1, 2, \dots \quad (3.3)$$

In Equation 3.2 is a_k a scalar gain sequence, G_k is the observation of the gradient and f_k a mapping to ensure positive definiteness of the approximation of the Hessian. The matrix \hat{H}_k in Equation 3.3 is the pre-iterated Hessian estimation

$$\hat{H}_k = \frac{1}{2} \left[\frac{\delta G_k^T}{2c_k \Delta_k} + \left(\frac{\delta G_k^T}{2c_k \Delta_k} \right)^T \right] \quad (3.4)$$

$$\delta G_k = G_k^{(1)}(\hat{\theta} + c_k \Delta_k) - G_k^{(1)}(\hat{\theta} - c_k \Delta_k) \quad (3.5)$$

In Equation 3.4 is c_k a positive scalar decaying to zero and Δ_k is often set to a vector of Bernoulli ± 1 random variables (for a more detailed outline see [Spall, 2000]). For the 2SG case is $G_k^{(1)}$ typically set equal to G_k .

To incorporate the 2SG algorithm into the IF1 framework and thus introduce the Adaptive Iterated Filtering (AIF) algorithm the updating formula in Equation 2.21 needs to be modified. Furthermore, the AIF algorithm also has to allow calculation of δG_k . The notation is hereafter changed from the notation in [Spall, 2000] to the notation used in Section 2.2.1.

In the AIF algorithm the stochastic gradient G_m is calculated as

$$G_m(\hat{\theta}_m) = \sum_{n=1}^N (\tilde{V}_{n,m}^P)^{-1} (\tilde{\theta}_{n,m}^F - \tilde{\theta}_{n-1,m}^F) \quad (3.6)$$

To calculate $G_m^{(1)}(\hat{\theta} + c_m \Delta_m)$ and $G_m^{(1)}(\hat{\theta} - c_m \Delta_m)$ the expressions for $\tilde{V}_{n,m}^P(\hat{\theta} + c_m \Delta_m)$, respectively $\tilde{V}_{n,m}^P(\hat{\theta} - c_m \Delta_m)$ and $\tilde{\theta}_{n,m}^F(\hat{\theta} + c_m \Delta_m)$, respectively $\tilde{\theta}_{n,m}^F(\hat{\theta} - c_m \Delta_m)$ first need to be established.

The expressions for $\tilde{\theta}_{n,m}^F(\hat{\theta} + c_m \Delta_m)$ and $\tilde{V}_{n,m}^P(\hat{\theta} + c_m \Delta_m)$ are

$$\tilde{\theta}_{n,m}^F(\hat{\theta} + c_m \Delta_m) = \frac{1}{J} \sum_{j=n}^J \tilde{\Theta}_{n,j}^F + c_m \Delta_m \quad (3.7)$$

$$\tilde{V}_{n,m}^P(\hat{\theta} + c_m \Delta_m) = \frac{1}{J_m} \sum_j \left(\theta_{n,j}^P + c_m \Delta_m - \tilde{\theta}_{n-1}^F(\hat{\theta} + c_m \Delta_m) \right)^2 \quad (3.8)$$

By employing Equation 3.7 and 3.8 it is possible to compute $G_k^{(1)}(\hat{\theta} + c_m \Delta_m)$ as

$$G_m^{(1)}(\hat{\theta} + c_m \Delta_m) = \sum_{n=1}^N \left(\tilde{V}_{n,m}^P(\hat{\theta} + c_m \Delta_m) \right)^{-1} \cdot \left(\tilde{\theta}_{n,m}^F(\hat{\theta} + c_m \Delta_m) - \tilde{\theta}_{n-1,m}^F(\hat{\theta} + c_m \Delta_m) \right) \quad (3.9)$$

It is also clear that $G_m^{(1)}(\hat{\theta} - c_m \Delta_m)$ is calculated in a similar way as in Equation 3.9.

Two different mappings of f_m are presented in Equation 3.10 and 3.11

$$\bar{\bar{H}}_m = (\bar{H}_m \bar{H}_m)^{1/2} + \delta_m I \quad (3.10)$$

$$\bar{\bar{H}}_m = \bar{H}_m + \delta_m I \quad (3.11)$$

In Equation 3.10 is $(\bar{H}_m \bar{H}_m)^{1/2}$ the positive semi-definite square root. The algorithm parameter δ_m is typically a small number in Equation 3.10 but often a quite much larger number in Equation 3.11.

The AIF algorithm is introduced by using the updating formulas in 3.2 and 3.3 and calculating the pre-iterated Hessian in Equations 3.4 and 3.5 by employing Equation 3.7, 3.8 and 3.9.

3.3 Implementation

The pseudo code for the AIF algorithm is presented in Algorithm 3. The three new algorithm parameters for the AIF algorithm are: the parameters for the pre-iterated estimate of the Hessian, c_m and Δ_m , and the parameter for the mapping δ_m . The parameter for pre-iterated estimate of the Hessian c_m governs how large the deviations from θ_m are in Equation 3.9 and the algorithm parameter c_m is often set to [Spall, 2000]

$$c_m = \frac{c}{m^\gamma} \quad (3.12)$$

In Equation 3.12 is c often a factor 2-10 greater then the one-sigma noise level of the data (i.e c is approximately 2-5 times the standard deviation of the data) and $\gamma \geq 0$.

The algorithm parameter Δ_m is set to a Bernoulli ± 1 random variable.

If the mapping in Equation 3.11 is used the algorithm parameter δ_m might have to be set to a quite large value to ensure a positive definite matrix. But if the mapping in 3.10 is used is it sufficient to let δ_m be a small number.

The other algorithm parameters for the AIF algorithm can be chosen in a similar way as there counterparts for the IF1 algorithm.

Algorithm 3 The AIF algorithm

Require: Start value θ_0 , Data y^* , number of iterations M , number of time steps N , number of particles J_m , perturbation sequences τ_m, σ_m , cooling coefficient a_m , parameter for pre-iterated estimate of the Hessian, c_m and Δ_t factor for mapping δ_m perturbation distribution κ , simulators f_{X_0} and $f_{X_n|X_{n-1}}$, evaluator $f_{Y_n|X_n}$

Ensure: $\hat{\theta}^{AIF} = \hat{\theta}_M$

$\hat{\theta}_0 = \theta_0$ ▷ Set start value for $\hat{\theta}$
for $m = 1 \cdots M$ **do**
 $Z_{0,j} \sim \kappa$, for $j = 1 \cdots J_m$ ▷ Draw from perturbation perturbation
 $\Theta_j^F = \hat{\theta}_{m-1} + \tau_m \cdot Z_{0,j}$, for $j = 1 \cdots J_m$ ▷ Initialize parameters
 $X_{0,j}^F \sim f_{X_0}(\cdot | \Theta_j^F)$, for $j = 1 \cdots J_m$ ▷ Initialize states
 $\bar{\theta}_0 = \frac{1}{J_m} \sum_j \Theta_j^F$ ▷ Calculate mean
 $\bar{\theta}_0^{add} = \frac{1}{J_m} \sum_j \Theta_j^F + c_m \Delta_m$
 $\bar{\theta}_0^{sub} = \frac{1}{J_m} \sum_j \Theta_j^F - c_m \Delta_m$
for $n = 1 \cdots N$ **do**
 $Z_n \sim \kappa$ for $j = 1 \cdots J_m$ ▷ Draw from perturbation perturbation
 $\Theta_j^P = \Theta_{n-1,j}^F + \sigma_m \cdot Z_{n,j}$ for $j = 1 \cdots J_m$ ▷ Perturb parameters
 $X_j^P \sim f_{X_n|X_{n-1}}(\cdot | X_{n-1,j}^F, \Theta_j^P)$ for $j = 1 \cdots J_m$ ▷ Simulate particles
 $w_j = f_{Y_n|X_n}(y_n^* | X_j^P, \Theta_j^P)$ for $j = 1 \cdots J_m$ ▷ Evaluate weights
 $w_j = \frac{w_j}{\sum_{j=1}^{J_m} w_j}$ for $j = 1 \cdots J_m$ ▷ Normalize weights
 $k_j = j$ with probability $\frac{w_j}{\sum_{j=1}^{J_m} w_j}$ for $j = 1 \cdots J_m$ ▷ Systematic
re-sampling of indexes
 $X_{n,j}^F = X_{n,k_j}^P$, $\Theta_{n,j}^F = \Theta_{n,k_j}^P$ for $j = 1 \cdots J_m$ ▷ Re-sampling
 $\bar{\theta}_n = \frac{1}{J_m} \sum_j \Theta_j^F$ ▷ Calculate means
 $\bar{\theta}_n^{add} = \frac{1}{J_m} \sum_j \Theta_j^F + c_m \Delta_m$
 $\bar{\theta}_n^{sub} = \frac{1}{J_m} \sum_j \Theta_j^F - c_m \Delta_m$
 $\mathbf{Var}(\theta_n) = \frac{1}{J_m} \sum_j (\theta_{n,j}^P - \bar{\theta}_{n-1})^2$ ▷ Calculate variances
 $\mathbf{Var}(\theta_n)^{add} = \frac{1}{J_m} \sum_j (\theta_{n,j}^P + c_m \Delta_m - \bar{\theta}_{n-1}^{add})^2$
 $\mathbf{Var}(\theta_n)^{sub} = \frac{1}{J_m} \sum_j (\theta_{n,j}^P - c_m \Delta_m - \bar{\theta}_{n-1}^{sub})^2$
end for
 $G_m = \sum_n \mathbf{Var}(\theta_n)^{-1} \cdot (\bar{\theta}_n - \bar{\theta}_{n-1})$ ▷ Calc gradient
 $\delta G_m = G_m(\theta_m + c_m \delta_m) - G_m(\theta_m - c_m \delta_m)$ ▷ Calc δG_m using Eq 3.9
 $\hat{H} = \frac{1}{2} \left[\frac{\delta G_k^T}{2c_k \Delta_k} + \left(\frac{\delta G_k^T}{2c_k \Delta_k} \right)^T \right]$ ▷ Calc pre-iterated estimate of the Hessian
 $\bar{H}_m = \frac{m}{m+1} \bar{H}_{m-1} + \frac{1}{m+1} \hat{H}_m$ ▷ Calc estimate of the Hessian
 $\bar{\bar{H}}_m = f(\bar{H}_m)$ ▷ Mapping in Equation 3.10 or 3.11
 $\hat{\theta}_{m+1} = \hat{\theta}_m - a_m \bar{\bar{H}}_m^{-1} G_m$ ▷ Update $\hat{\theta}$
end for
return $\hat{\theta}^{AIF} = \hat{\theta}_M$

Chapter 4

Simulations

This chapter contains simulation studies for three different POMP models: a simple AR(1) model, the Gompertz model and the SIR model. For each model are the IF1, IF2 and AIF algorithms compared.

4.1 Simple AR(1) model

The simple AR(1) model follows Equation 4.1.

$$\begin{cases} X_n &= \theta X_{n-1} + \epsilon_n, \text{ where } \epsilon_n \in \mathcal{N}(0, 1) \\ Y_n &= X_n + \eta_n, \text{ where } \eta_n \in \mathcal{N}(0, 1) \end{cases} \quad (4.1)$$

The index n in Equation 4.1 is the time index, which is defined on $n = 1, \dots, N$.

The Simple AR(1) model is one of the least complex POMP models that are possible and is therefore a suitable model to first consider when to test different and new versions of the IF algorithm. Mainly since the implementation of the model is easy and since the model itself is non-complex.

4.1.1 Simulation Results for the Simple AR(1) Model

To estimate the model parameters with the IF algorithm the *simulators* f_{X_0} , $f_{X_n|X_{n-1}}$ and the *evaluator* $f_{Y_n|X_n}$ have to be defined. For the simple AR(1) model these are set to

$$\begin{cases} f_{X_0} &= \mathcal{N}(0, 1) \\ f_{X_n|X_{n-1}} &= \mathcal{N}(\Theta^P \cdot X^F, 1) \\ f_{Y_n|X_n} &= f(y_n^* | X^P, 1)^1 \end{cases} \quad (4.2)$$

¹ $f(y_n^* | X^P, 1)$ is the density function $f(x | \mu, \sigma)$ for the normal distribution.

The *start value*, θ_0 , is drawn from a Uniform distribution on $[-1, 1]$

The algorithm parameter settings used for the IF1, IF2 and AIF model are presented in Table 4.1. The algorithm parameters are set to the values that have shown to provide best convergence properties, but also algorithm parameter values such that the algorithm parameters of the different versions of the IF algorithm are as similar as possible. Hence, many of the algorithm parameters are set to the same value for the different versions of the IF algorithm.

For the simple AR(1) mode is the mapping in Equation 3.10 used to ensure a semi-positive Hessian matrix.

Table 4.1: Presents the algorithm parameter settings for the IF1, IF2 and AIF algorithm for the simple AR(1) model

IF1		AIF		IF2	
Parameter	Value	Parameter	Value	Parameter	Value
M	1000	M	1000	M	1000
N	500	N	500	N	500
δ	2	δ	2	J	1000
J_m	$50 \cdot m$	τ_m	$\sqrt{m^{-1}}$	σ_m	$0.2 \cdot 0.99^m$
τ_m	$\sqrt{m^{-1}}$	σ_m	$\sqrt{m^{-0.5-\delta}}$		
σ_m	$\sqrt{m^{-0.5-\delta}}$	a_m	$\frac{1}{(m+0.1 \cdot M)^{0.7}}$		
α	1	c_m	$\frac{5 \cdot std(y^*)}{(m+1)^{0.3}}$		
a	0.1	Δ_m	Bernoulli ± 1		
A	$0.05 \cdot M$	δ_m	0.1		
a_m	$\frac{a}{(1+m+A)^m}$	Σ	0.2		
Σ	0.2				

The main result is presented in Figure 4.1 where the *root mean square error* (RMSE) is presented. The *RMSE* is calculated as

$$RMSE = \sqrt{\frac{\sum_{i=k}^K (\hat{\theta}^{ML} - \hat{\theta}_{m,k}^{IF})^2}{K}} \quad (4.3)$$

In Equation 4.3 is *RMSE* calculated against the true maximum likelihood estimation $\hat{\theta}^{ML}$, which was calculated with the MATLAB function *estimate*. This is the case since the IF algorithm ensures convergence to the maximum likelihood estimation. The term $\hat{\theta}_{m,k}^{IF}$ in Equation 4.3 is the m :th approximation of the model parameter θ , (i.e the last approximation), in the k :th run of the IF algorithm.

The main conclusion of Figure 4.1 is that the AIF algorithm converges faster than the IF1 algorithm, the convergence rate of the AIF algorithm also seems

to follow the fast convergence rate of $\mathcal{O}(\frac{1}{m^2})$. The convergence rate of the AIF algorithm can however vary with the data used, but the AIF algorithm has the same or better convergence rate compared to the IF1 algorithm independently of the data.

In Figure 4.1 is it also possible to conclude that the estimate of the model parameter θ is constant for the first iterations of the AIF algorithm. This is due to the poor approximation of the Hessian matrix in the AIF algorithm for the first iterations. Hence, during the first iterations are new model parameter estimations generated that are far away from the previous model parameter estimation, which is due to the poorly approximated Hessian matrix. These new model parameter estimations are however not accepted since they are too far way from the last estimation of the model parameter.

From Figure 4.1 is it also possible to conclude that the IF1 algorithm converges quite slow after the first 20 iterations and that the converges rate seems to approximately follow the rate of $\mathcal{O}(\frac{1}{\sqrt{m}})$.

The convergence to the true model parameter θ is presented in Figure 4.2 and 4.3. By inspecting the plot in Figure 4.2 one concludes that the IF1 algorithm converges faster for the first iterations while the AIF algorithm converges slower, this pattern is however also present in Figure 4.4. In Figure 4.3, where the transient part of the convergence to the true model parameter is shown, is it possible to note that the AIF algorithm seems to continue to converge to the maximum likelihood value of the model parameter θ , while the IF1 algorithm fluctuates around the maximum likelihood value. This observation of the convergence behavior of the AIF algorithm is the main explanation of why the AIF algorithm converges faster after the first 20 iterations compared to the IF1 algorithm.

The approximate log-likelihood values are also presented in Figure 4.4 and 4.5. The log likelihood values increases with the number of iteration, hence the the estimation of model parameter θ improves as the number of iterations increases. In Figure 4.5 where the transient phase is shown is it possible to conclude that the IF1 and AIF algorithm converges to the same log likelihood value $loglik \sim -920$. This indicates that the estimation of the model parameter θ is equivalently good for the IF1 and AIF algorithm.

Paragraph: approximation of the Hessian matrix!

4.2 The Gompertz Model

The Gompertz model is a discrete time model for population growth [Ionides et al., 2015]. The Markov process X_t in the Gompertz model is the population density and the observation process Y_t is the population density, which is observed on an logarithmic scale and with an error. The model is presented in Equation 4.4

$$\begin{cases} X_{t+\Delta t} &= K^{1-e^{-r\Delta t}} X_t^{e^{-r\Delta t}} \epsilon_t \\ \log Y_t &\sim \mathcal{N}(\log X_t, \tau) \end{cases} \quad (4.4)$$

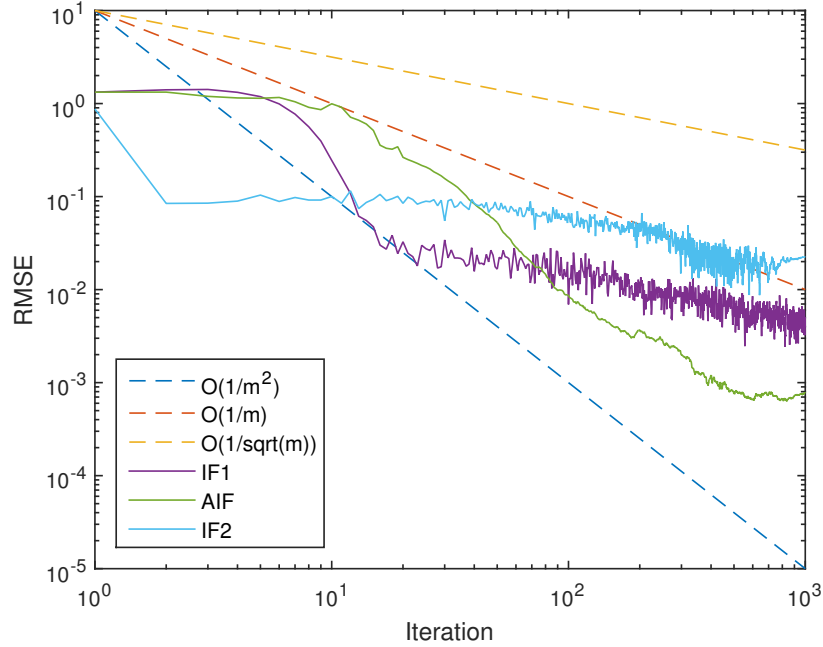


Figure 4.1: Shows the RMSE (i.e $|\hat{\theta}^{ML} - \hat{\theta}^{IF}|$ calculated over 10 runs) for different versions of the IF algorithm. The algorithm parameter settings for the IF1, AIF and IF2 algorithm are presented in Table 4.1

In Equation 4.4 are the ϵ_t 's independently identically distributed random variable such that $\log \epsilon_t \sim \mathcal{N}(0, \sigma)$. The length of the time step Δt is assumed to be 1. If the Markov process X_t is also transformed into the logarithmic scale the model then follows

$$\begin{cases} \log X_{t+\Delta t} & \sim \mathcal{N}((1 - e^{-r\Delta t})K + e^{-r\Delta t}X_t, \sigma) \\ \log Y_t & \sim \mathcal{N}(\log X_t, \tau) \end{cases} \quad (4.5)$$

The transformed version of the Gompertz model in Equation 4.5 is convenient since it allows exact maximum likelihood estimation of the model parameters by employing a Kalman filter [Ionides et al., 2015]. However, in this thesis is the Gompertz model only considered in the non-transformed version in Equation 4.4.

The model parameters r , σ and τ are estimated using the IF1, IF2 and AIF algorithm, but the parameter K is assumed to be known. However, since the model parameters have to be strictly large then zero the model parameters are transformed with the logarithm function to facilitate the estimation of the model parameters and transfer the allowed range of the parameter from $[0 \infty]$

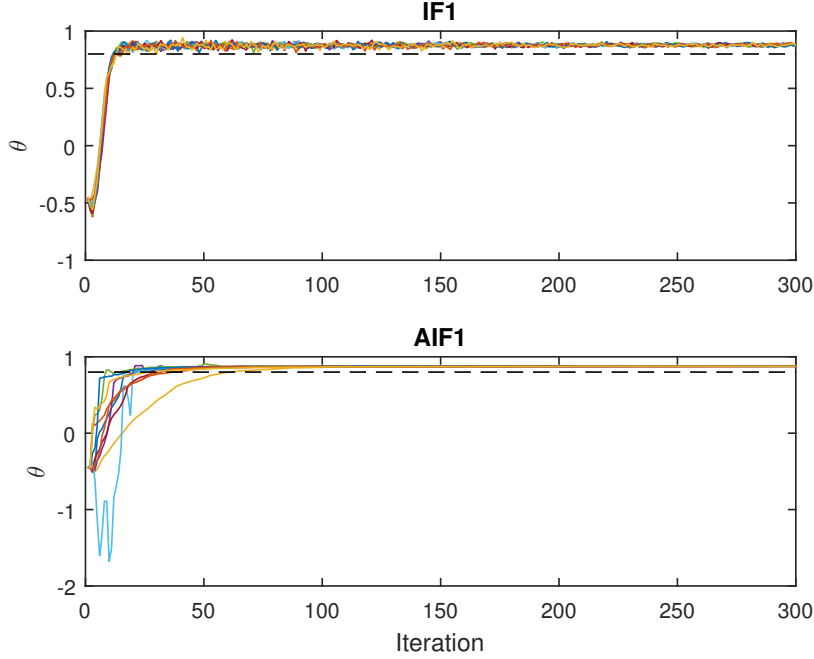


Figure 4.2: Shows the convergence to the true value of the model parameter θ for different version of the IF algorithm in the converging phase. The parameter settings for the IF1, IF2 and AIF algorithm are presented in Table 4.1

to the whole real axis.

For the Gompertz mode is the mapping in Equation 3.11 used to ensure a semi-positive Hessian matrix. The mapping in Equation 3.11 is used to avoid problems with obtaining a complex matrix after calculating the square root in Equation 3.10

4.2.1 Simulation Results for the Gompertz Model

For the Gompertz model is the *simulators* f_{X_0} and $f_{X_n|X_{n-1}}$, and the *evaluator* $f_{Y_n|X_n}$ set to

$$\begin{cases} f_{X_0} &= \mathcal{U}(0, 1) \\ f_{X_n|X_{n-1}} &= K^{1-e^{-r\Delta t}} X_t^{e^{-r\Delta t}} \epsilon_t, \text{ where } \log \epsilon_t \sim \mathcal{N}(0, \sigma) \\ f_{Y_n|X_n} &= f(y_n^* | \log X_t, \tau)^2 \end{cases} \quad (4.6)$$

The algorithm parameters for the IF1, IF2 and AIF algorithm are presented in Table 4.2. The *start values* are drawn from a uniform distribution

² $f(y_n^* | \log X_t, \tau)$ is the normal probability density function $f(x | \mu, \sigma)$

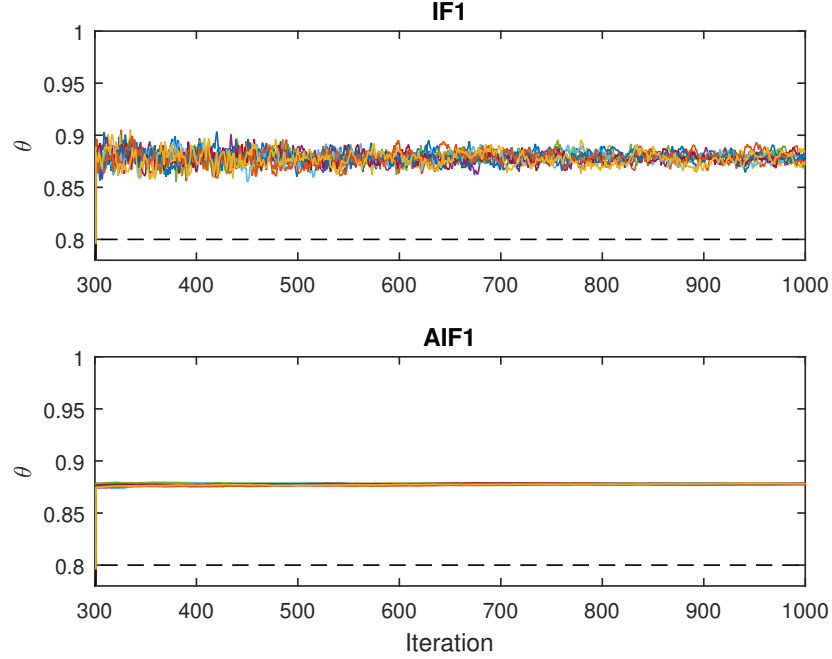


Figure 4.3: Shows the convergence to the true value of the model parameter θ for different version of the IF algorithm in the transient phase. The algorithm parameter settings for the IF1, IF2 and AIF algorithm are presented in Table 4.1

on $[0, 1]$. The model parameter estimations are computed on the same data as in [King et al., 2015]. This is done to be able to compare the estimated model parameters with the exact maximum likelihood estimation of the model parameters that are provided in [King et al., 2015].

Estimating three parameters

The model parameters r , σ and τ are first estimated using the IF1, IF2 and AIF algorithm. The obtained model parameter estimations, as well as the maximum likelihood estimation of the parameters are presented in Table 4.3. From Table 4.3 it is possible to conclude that it is only the IF1 and IF2 algorithm that generates reasonable model parameter estimations when three model parameters are estimated. The estimation of the parameter r also seems to be better for the IF1 and IF2 model compared to the maximum likelihood estimation since the true value for all model parameters is 0.1.

The convergence of the IF1, IF2 and AIF algorithm is presented in Figure 4.7, where the model parameter estimation for each iteration is shown. It is clear that only the IF1 and IF2 algorithms converge. The model parameter

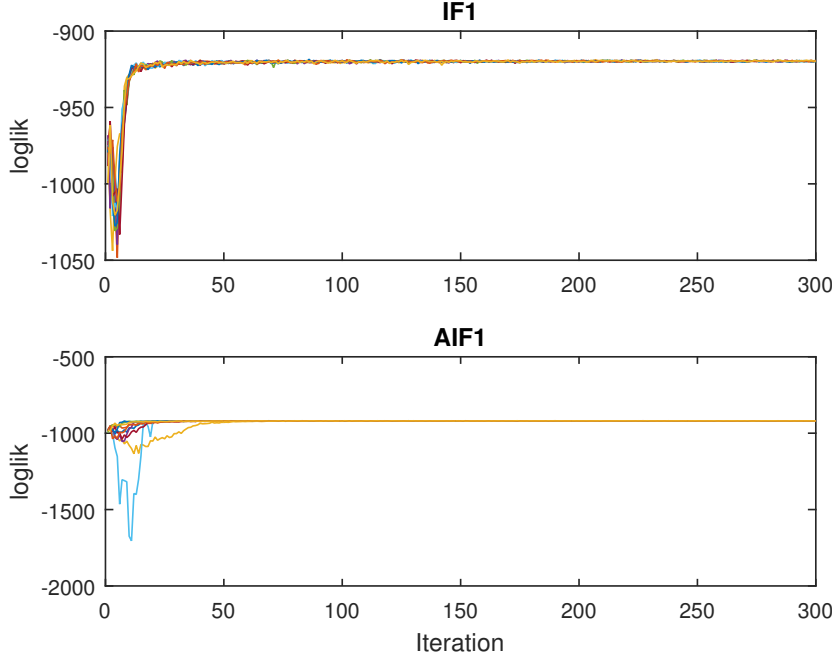


Figure 4.4: Shows the log-likelihood values for different versions of the IF algorithm in the converging phase. The algorithm parameter settings for the IF1, AIF and IF2 algorithm are presented in Table 4.1

estimation of the AIF algorithm does converge, but it does not converge to the correct value. It is also notable the the model parameter estimation of the AIF algorithm only moves very little away from the start value.

Estimating two parameters

The estimation results obtained when estimating three model parameters were not satisfying. The next step is therefore to only estimate the parameters σ and τ and treat the model parameter r as known. The obtained model parameter estimations, as well as the exact maximum likelihood estimation of the parameters are presented in Table 4.4. From Table 4.4 is it possible to conclude that only the IF1 and IF2 algorithms generate reasonable model parameter estimations when two model parameters are estimated.

The convergence of the IF1, IF2 and AIF algorithm is presented in Figure 4.8, where the model parameter estimation for each iteration is shown. The IF1 algorithm seems to converge quite nicely to the true model parameter. The IF2 algorithm also seems to converge quite well. However, both the IF1 and IF2 algorithm seem to better estimate the model parameter τ , then the model parameter σ . Hence, the IF1 and IF2 algorithms only generate good

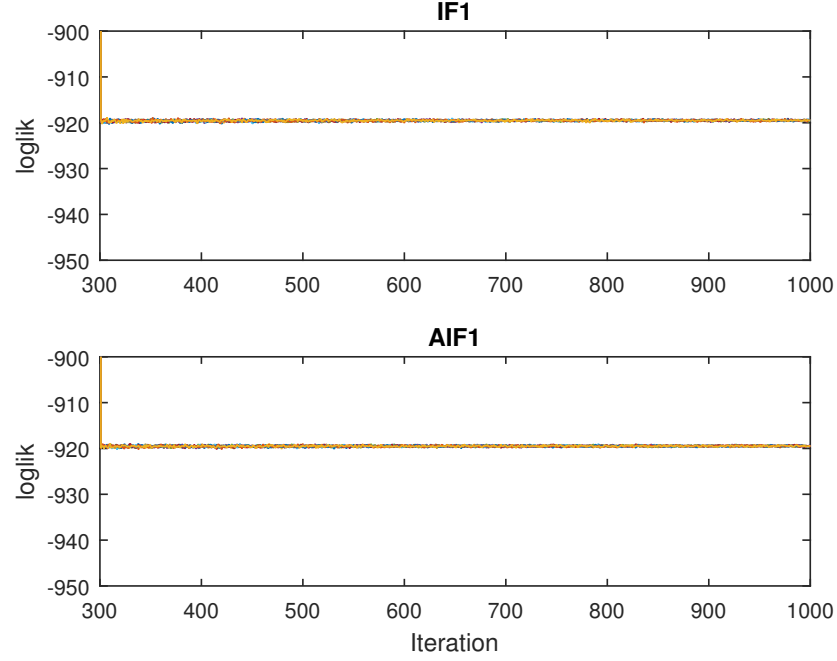


Figure 4.5: Shows the log-likelihood values for different versions of the IF algorithm in the transient phase. The algorithm parameter settings for the IF1, IF2 and AIF algorithm are presented in Table 4.1

model parameter estimations for one of the two model parameters. The AIF algorithm on the other hand does not converge to true model parameters. The AIF algorithm instead seems to converge to some other model parameter value. Hence, the performance of the algorithms are only slightly improved when only two parameters are estimated and AIF algorithms still does not generate reliable model parameter estimations.

Conclusion regarding the Gompertz model

The conclusion is that both the IF1 and IF2 algorithms generate reasonable model parameter estimations for the Gompertz model. However, the AIF algorithm does not generate reliable model parameter estimations since the AIF algorithm does not converge to the true model parameters, this is likely due to the poor estimation of the Hessian matrix in the AIF algorithm.

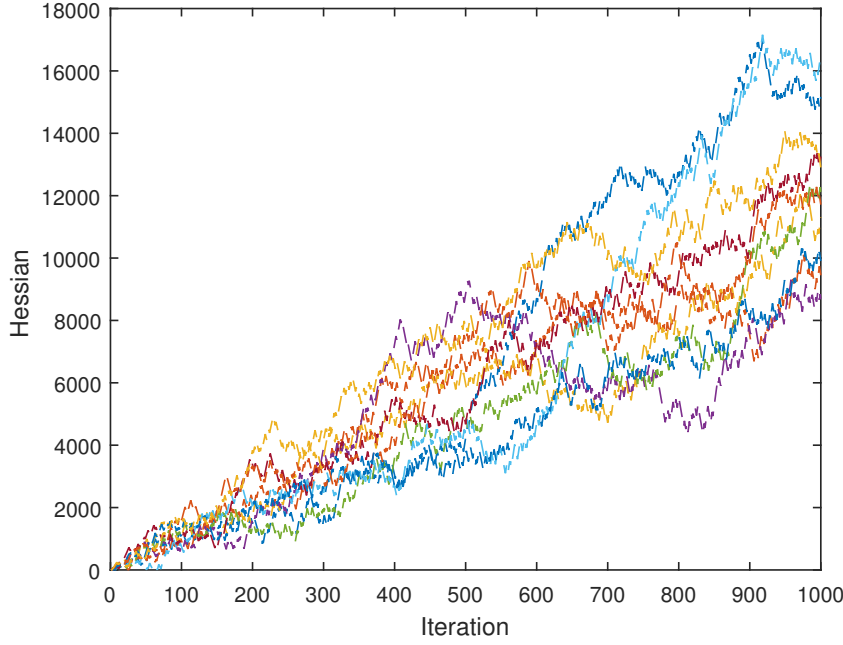


Figure 4.6: Shows the approximation of the Hessian for each iterations for 10 runs of the AIF algorithm. The algorithm parameter settings for the AIF algorithm are presented in Table 4.1

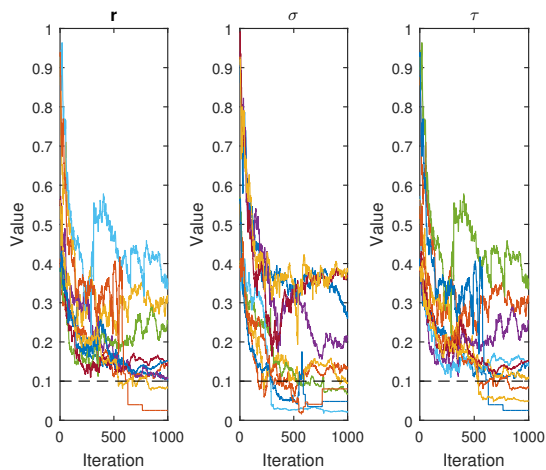
4.3 The SIR Model

The SIR model is an epidemiological model used to model how a disease spreads in a population of hosts. The model consists of four classes, *number susceptible* (S), *number infectious* (I) and *number recovered* (immune) (R). A diagram for the SIR model is presented at Figure 4.9. [King et al., 2015]

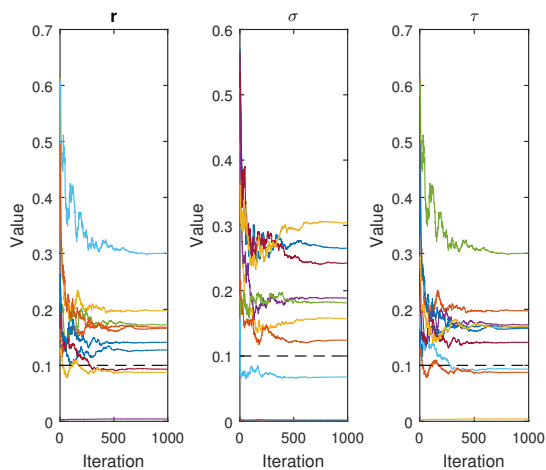
The SIR model can be viewed as a POMP model where the Markov process X_t consists of the processes S , I and R , while the observation process Y_t is the process C . The dynamic for the Markov process X_t (i.e for the processes S , I and R) is

$$\begin{cases} \frac{dS}{dt} = \mu(N - S) - \beta \frac{I}{N} S \\ \frac{dI}{dt} = \beta \frac{I}{N} S - \gamma I - \mu I \\ \frac{dR}{dt} = \gamma I - \mu R \end{cases} \quad (4.7)$$

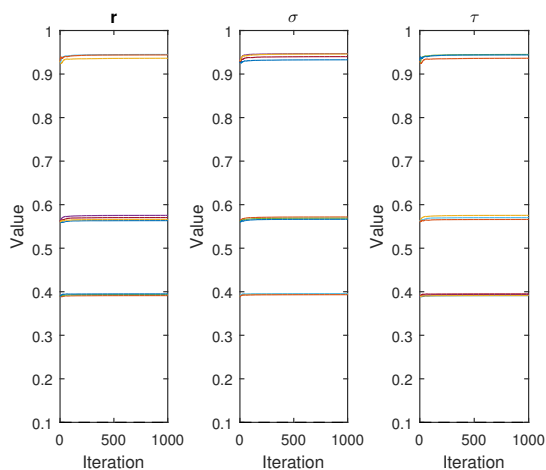
The number of cases on a time interval $[t_1, t_2]$ is assumed to be negative binomial distributed



(a) IF1

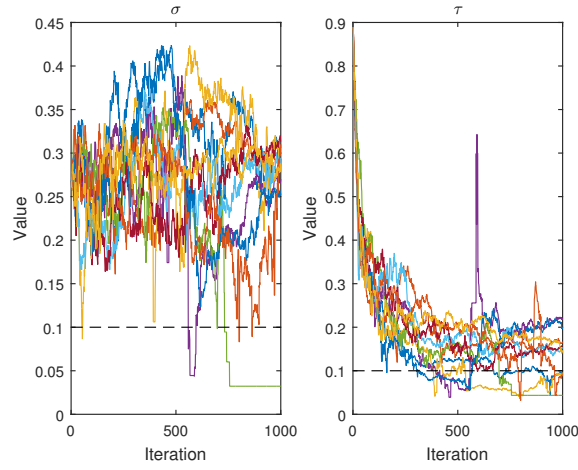


(b) IF2

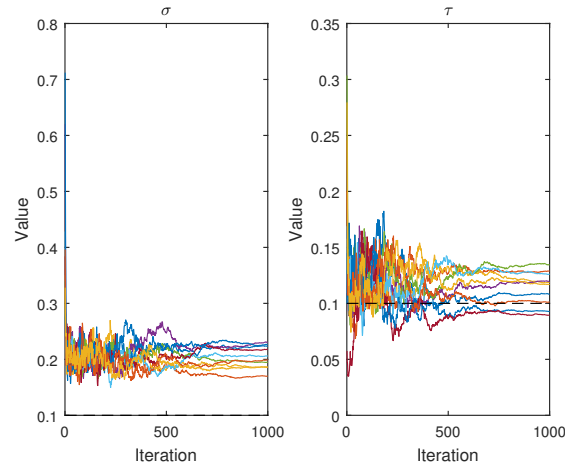


(c) AIF

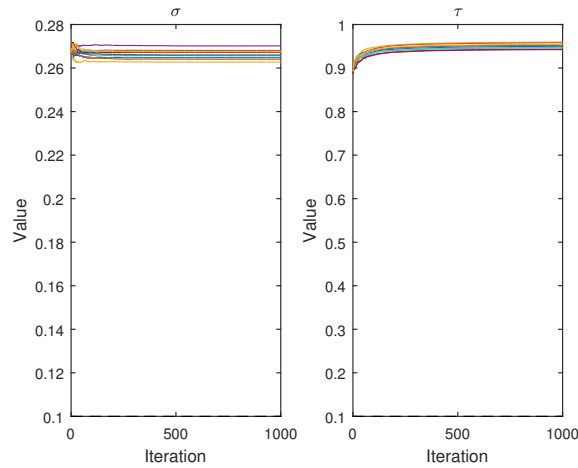
Figure 4.7: Convergence for the IF1, IF2 and AIF algorithm where three model parameters in the Gompertz model are estimated. The algorithm parameter settings are presented in Table 4.2



(a) IF1



(b) IF2



(c) AIF

Figure 4.8: Convergence for the IF1, IF2 and AIF algorithm where two model parameters in the Gompertz model are estimated. The algorithm parameter settings are presented in Table 4.2

Table 4.2: Presents the model parameter settings for the IF1, AIF and IF2 algorithm for the Gompertz model

IF1		AIF		IF2	
Parameter	Value	Parameter	Value	Parameter	Value
M	1000	M	$200 + 5$	M	1000
N	100	N	100	N	100
δ	1	δ	2	J	1000
J_m	$50 \cdot m$	m	$50 \cdot m$	σ_m	$0.001 \cdot 0.99^{m-1}$
τ_m	$\sqrt{m^{-1}}$	τ_m	$\sqrt{m^{-1}}$		
σ_m	$\sqrt{m^{-0.5-\delta}}$	σ_m	$\sqrt{m^{-0.5-\delta}}$		
α	1	a_m	$\frac{50}{(1+m+0.05 \cdot M)^{0.9}}$		
a	0.02	c_m	$\frac{std(y^*) \cdot 2}{(m+1)^{0.2}}$		
A	$0.05 \cdot M$	Δ_m	Bernoulli ± 1		
a_m	$\frac{a}{(1+m+A)^{0.9}}$	δ_m	$10000 \cdot m$		
Σ	$\begin{pmatrix} 0.02 & 0 & 0 \\ 0 & 0.02 & 0 \\ 0 & 0 & 0.02 \end{pmatrix}$	Σ	$\begin{pmatrix} 0.02 & 0 & 0 \\ 0 & 0.02 & 0 \\ 0 & 0 & 0.02 \end{pmatrix}$		

$$C(t_1, t_2) \sim negbin(\rho \Delta_{I \rightarrow R}(t_1, t_2), \theta) \quad (4.8)$$

In Equation 4.8 is $\rho \Delta_{I \rightarrow R}(t_1, t_2)$ the mean for the negative binomial distribution and θ is the shape parameter.

The SIR model is fully described by the model parameters, μ , β , γ , ρ and θ . However, μ governs the birth/death rate and is therefore not linked to the specifications of the disease. Furthermore, the model parameter θ is the shape parameter for the negative binomial distribution and therefore does not have a direct affect on the dynamic of the disease. The parameters β , γ and ρ are however directly linked to the specifications of the disease. The objective in this thesis is therefore to estimate the parameters β , γ and ρ . The model parameter ρ can only take values on the interval $[0, 1]$ and the model parameter ρ is therefore transformed by the cumulative distribution function of the standard normal distribution to facilitate the estimation of the ρ parameter, which transfers the allowed range of the parameter ρ from $[0, 1]$ to the whole real axis.

4.3.1 Simulation Results for the SIR Model

The *simulator* f_{X_0} , and *evaluator* $f_{Y_n|X_n}$ for the SIR model are

$$\begin{cases} f_{X_0} &= \mathcal{N}\left(\begin{pmatrix} 30000 & 0 & 0 & 0 \\ 0 & 800 & 0 & 0 \\ 0 & 0 & 470000 & 0 \\ 0 & 0 & 0 & 400 \end{pmatrix}, \begin{pmatrix} 50 & 0 & 0 & 0 \\ 0 & 30 & 0 & 0 \\ 0 & 0 & 500 & 0 \\ 0 & 0 & 0 & 10 \end{pmatrix}\right) \\ f_{Y_n|X_n} &= f(y_n^* | \rho H, \theta)^3 \text{ where } H = \Delta_{I \rightarrow R} \end{cases} \quad (4.9)$$

Table 4.3: Presents the obtained model parameter estimation for the IF1, IF2 and AIF algorithm where three model parameters in the Gompertz model are estimated. The model parameter estimations are calculated as the mean of the parameter estimations over 10 runs of the algorithms. Also presented the exact maximum likelihood estimation of the model parameters for the Gompertz model [King et al., 2015]

	r	σ	τ
IF1	0.1663	0.1675	0.1607
IF2	0.1458	0.1534	0.1500
AIF	0.6280	0.6656	0.6108
Exact ml [King et al., 2015]	0.0322	0.0694	0.1170

Table 4.4: Presents the obtained model parameter estimation for the IF1, IF2 and AIF algorithm where two model parameters in the Gompertz model are estimated. The model parameter estimations are calculated as the mean of the parameter estimations over 10 runs of the algorithms. Also presented the exact maximum likelihood estimation of the model parameters for the Gompertz model [King et al., 2015]

	σ	τ
IF1	0.2634	0.1359
IF2	0.2039	0.1136
AIF	0.2663	0.9507
Exact ml [King et al., 2015]	0.0694	0.1170

The *simulator* $f_{X_n|X_{n-1}}$ for the SIR model follows a Euler-multinational distribution. The process X_n is advanced one time step by employing the Euler-method and by simulating the system in Equation 4.7 over one time step. A more detailed outline of the Euler-multinational distribution and the Euler-method can be found in [King et al., 2015].

Before estimating the model parameters is the data y^* generated by simulating the system in described in Equation 4.7 and Figure 4.9. The simulated X_t process is presented in Figure 4.10 and the data y^* (i.e. the process for the number of cases) is presented in Figure 4.11

The algorithm parameter settings used for the IF1, IF2 and AIF algorithm are presented in Table 4.5.

For the SIR mode is the mapping in Equation 3.11 used to ensure a semi-positive Hessian matrix. The mapping in Equation 3.11 is used to avoid prob-

³ $f(y_n^* | \rho H, \theta)$ is the negative binomial distribution function $f(x | \mu, \theta)$

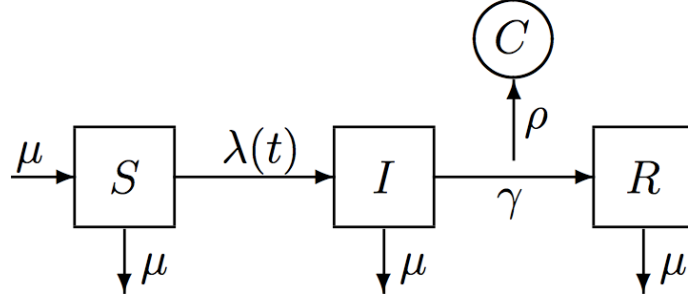


Figure 4.9: The diagram show the SIR model. Where S is *number susceptible*, I *number infectious* and R *number recovered* (immune) and total population $N = S + I + R$. The parameter μ is the birth/death rate, $\lambda(t) = \beta \frac{I(t)}{P}$ is the force of infection, γ is the recovery rate. The parameter ρ is the probability of which new infections are recorded. Source [King et al., 2015]

lems with obtaining a complex matrix after calculating the square root in Equation 3.10

The simulation results are presented in Figure 4.12, 4.13 and 4.14. In Figure 4.12 and 4.13 it is notable that both the IF1 and the IF2 algorithm can estimate the ρ parameter quite well. For the IF1 algorithm is the estimation of the γ parameter also quite good. However non of the models manage to estimate the β parameter with any high precision. While the IF1 and IF2 algorithms seem to converge the parameter estimations of the AIF algorithm generally do not seem to converge. Only the estimation of the γ parameter for the AIF model seem to converge while the estimation of the β and ρ parameter do not converge.

4.4 Summary of Simulation Results

The three different models used for simulations are quite different. The simple AR(1) model is a model driven by the noise terms and the SIR model is a dynamical model, while the Gompertz model is model with both dynamical and noise driven parts. The complexity of the models also varies where the Simple AR(1) model is the least complex model and the SIR model the most complex mode. The three models were chosen because of their different characteristics.

The conclusions from the simulation studies are.

(1) *AIF is only efficient for the Simple AR(1) model* From the simulation results is it clear that the AIF algorithm is only comparable with the IF1 algorithm for the simple AR(1) model. The performance of the AIF algorithm is also considerable worse compared to the IF1 algorithm for the other models.

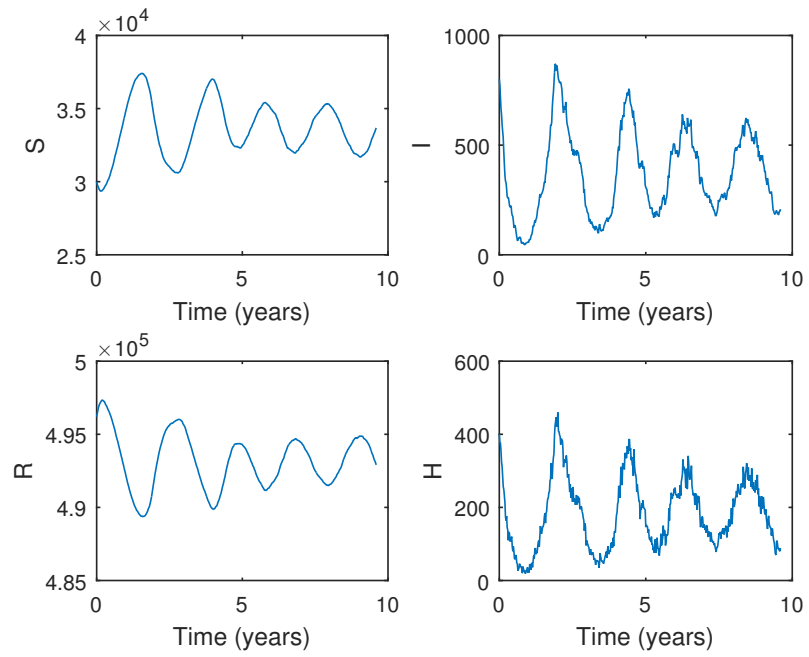
(2) *The Hessian approximation is poor* The poor performance of the AIF

Table 4.5: Presents the algorithm parameter settings for the IF1, AIF and IF2 algorithm for the SIR model

IF1		AIF		IF2	
Parameter	Value	Parameter	Value	Parameter	Value
M	50	M	$200 + 1$	M	50
N	500	N	100	N	500
δ	2	δ	2	J	1000
m	$50 \cdot m$	m	$50 \cdot m$	σ_m	$1 \cdot 0.99^{m-1}$
τ_m	$\sqrt{m^{-1}}$	τ_m	$\sqrt{m^{-1}}$		
σ_m	$\sqrt{m^{-0.5-\delta}}$	σ_m	$\sqrt{m^{-0.5-\delta}}$		
α	1	a_m	$\begin{pmatrix} 10 & 5 & 0.5 \end{pmatrix}$		
a	$\begin{pmatrix} 20 & 1 & 0.01 \end{pmatrix}$	c_m	$\frac{(m+0.05 \cdot M)^{0.9}}{\text{std}(y^*) \cdot 2}$		
A	$0.05 \cdot M$	Δ_m	Bernoulli ± 1		
A_m	$\frac{a}{(1+m+A)^\alpha}$	δ	$10000 \cdot m$		
Σ	$\begin{pmatrix} 20 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.2 \end{pmatrix}$	Σ	$\begin{pmatrix} 20 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.2 \end{pmatrix}$		

model is probably due to the fact that the approximation of the Hessian matrix is quite bad. This observation is possible to draw since [hur den approximativet hessianen relaterar till den verkliga hessianen]. A problematic part with approximation of the Hessian matrix is the mapping in Equation 3.11 that is used for the Gompertz and SIR model. To ensure a semi-definite matrix with using this mapping the δ_m has to be very large to counteract the non-positive elements of the matrix \bar{H}_m . Hence, the matrix \bar{H}_m undergoes a quite large transformation to ensure a semi-definite matrix and it is therefore possible that important information in the matrix \bar{H}_m is partly lost.

(3) *The IF1 algorithm seems to generate the best model parameters estimations* An interesting finding is that the IF1 algorithm seems to be the algorithm that generates the best model parameter estimations. However, the objective of this thesis is not to compare the IF1 and IF2 algorithm so this finding is not further analysed but rather noticed.

Figure 4.10: Simulated X_t process for the SIR model

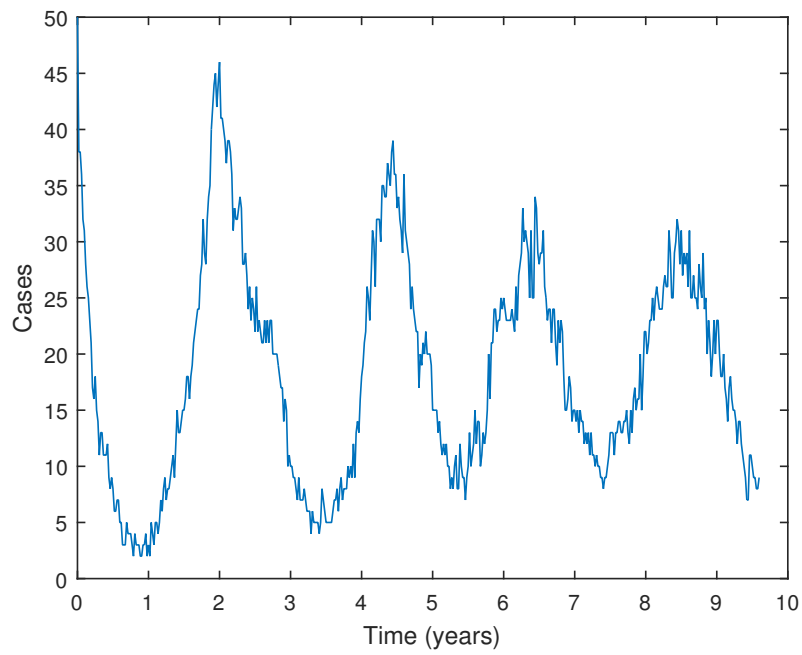


Figure 4.11: Simulated data y^* (i.e. number of cases C) for the SIR model

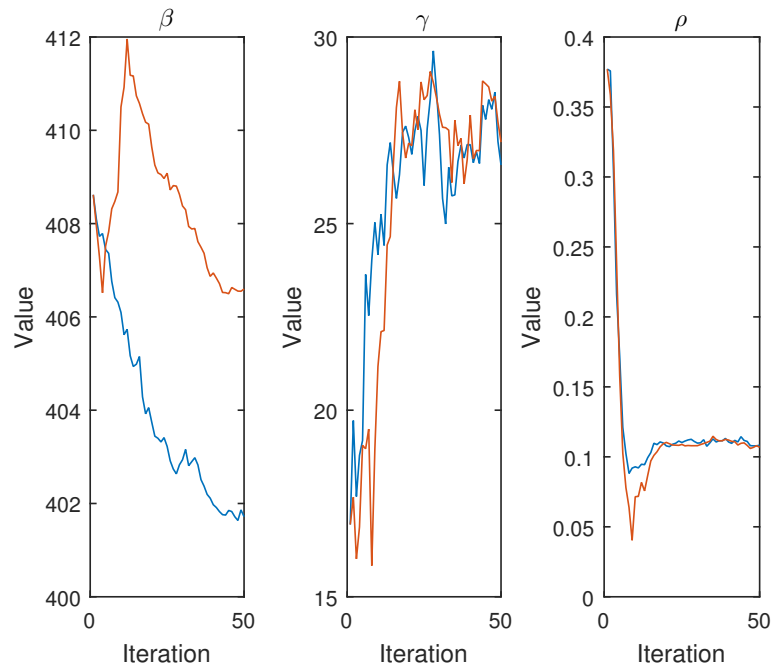


Figure 4.12: Shows the convergence of the model parameters β , γ and ρ for the SIR model using the IF1 algorithm computed over 10 runs of the algorithm. The algorithm parameter settings for the IF1 algorithm are presented in Table 4.5

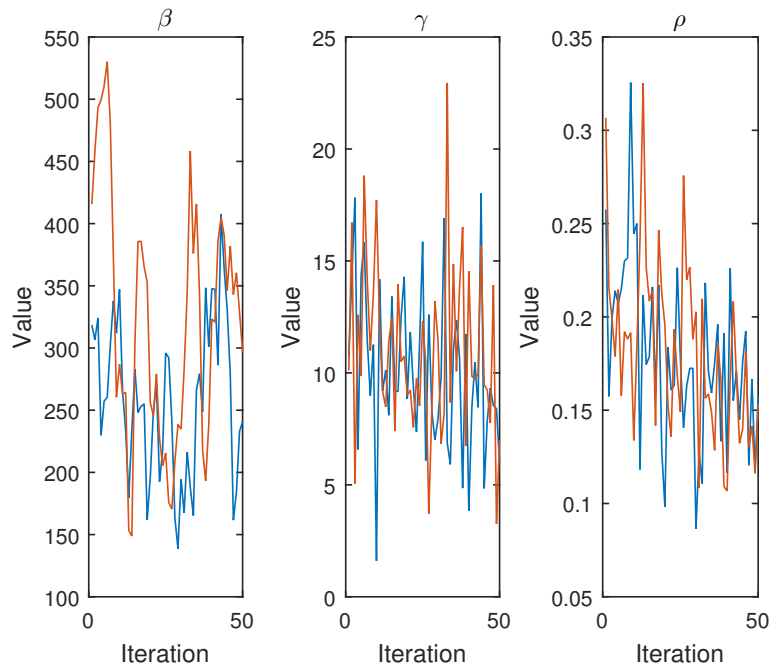


Figure 4.13: Shows the convergence of the model parameters β , γ and ρ for the SIR model using the IF2 algorithm computed over 10 runs of the algorithm. The algorithm parameter settings for the IF2 algorithm are presented in Table 4.1

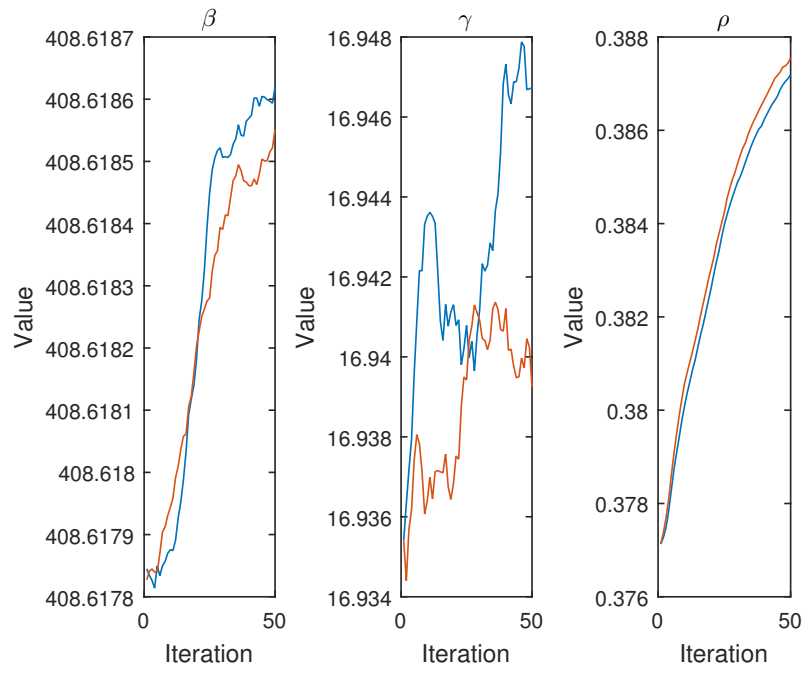


Figure 4.14: Shows the convergence of the model parameters β , γ and ρ for the SIR model using the AIF algorithm computed over 10 runs of the algorithm. The algorithm parameter settings for the AIF algorithm are presented in Table 4.1

Chapter 5

Discussion

This chapter contains a general discussion on the Iterated Filtering algorithm and also a specific discussion on the Adaptive Iterated Filtering algorithm. The chapter also discusses further research questions regarding the IF algorithm.

5.1 General Discussion on the Iterated Filtering Algorithm

The Iterated Filtering algorithm is a suitable algorithm to compute maximum likelihood based parameter estimations of POMP models. However, the IF algorithm (both the IF1 and IF2 version of the Iterated Filtering algorithm), has two main drawbacks.

(1) *Sensitive of choice of algorithm parameters* The IF algorithm is highly sensitive to algorithm parameter changes. Especially a well stated gain sequence is crucial to obtain an algorithm that is stable and that converges properly. Furthermore, even though guidelines are available is it not quite clear how the algorithm parameters for the IF1 and IF2 algorithms should be chosen to obtain a stable and optimal IF algorithm. This problem especially applies to the IF2 algorithm.

(2) *Tuning* Since the IF algorithm is sensitive to algorithm parameter changes the algorithm requires quite a bit of tuning before a stable algorithm that converges properly is obtained.

Hence, the IF algorithm is not an algorithm that can be applied to a problem straight away, but requires tuning and proper choices of algorithm parameters.

The algorithm parameters for the IF1 algorithm are all easily interpreted and it is clear how they effect the algorithm. However, the connection between the algorithm parameters of the IF1 algorithm and the IF2 algorithm is not clear. Specifically the connection between the two perturbation sequences of the IF1 algorithm σ_m and τ_m and the perturbation sequence σ_m and perturbation density h_m is unclear. Due to this reason and the fact that the IF1 and IF2

algorithm are two quite different versions of the Iterated Filtering algorithm are comparisons between the two algorithms hard to do and interpret.

The main advantage of the IF algorithm is that the algorithm can be applied to a large amount of models in many different fields. It is also simple to implement the IF algorithm to a new model if the IF algorithm has been implemented once since the main different for the IF algorithm of different models are the specification of the simulators f_{X_0} and $f_{X_{n+1}|X_n}$, and the evaluator $f_{Y_n|X_n}$. There also exists a R package that allows the user to apply the IF algorithm to a model of choice with only specifying the model and not actually implementing the IF algorithm [King et al., 2015].

5.2 Discussion on the Adaptive Iterated Filtering Algorithm

The main idea of the Adaptive Iterated Filtering algorithm is to construct an IF algorithm with an optimal gain sequence. This is done by incorporating second-order information by approximating the Hessian matrix.

The simulation results in Chapter 4 however indicates that the proposed AIF algorithm might only work satisfying if it is possible to approximate the Hessian well. The results for the simple AR(1) model propose that this is possible for the simple AR(a) model, however for the two more complex models, i.e. the Gompertz model and the SIR model it is clear that the Hessian approximation is not satisfying.

Furthermore, the AIF algorithm is also, as the other version the IF algorithm, sensitive to parameter settings and some tuning of algorithm parameters, particular the gain sequence, is necessary to obtain a stable algorithm that does not diverge or in other ways behave inadequate. The importance of tuning is also increased since the AIF algorithm introduces three new algorithm parameters that need to be specified compared to the IF1 algorithm, however poorly specified values for the algorithm parameters c_m and Δ_m do not seem to have a severe impact on the behavior of the AIF algorithm compared to a poorly specified gain sequence. However, a well specified δ_m parameter is crucial to obtain a semi-definite approximation of the Hessian matrix.

The problem with the bad estimation of the Hessian is a notorious problem since it is hard to estimate the Hessian matrix [Spall, 2005]. Hence, it is not unreasonable that the approximation of the Hessian matrix is bad since the Hessian matrix is quite hard to approximate.

5.3 Further Research

Further research on the IF algorithm could focus on conditions on algorithm parameter of the IF1 and IF2 algorithm and methods to find the optimal algorithm parameter for the IF1 and IF2 algorithm. This is relevant since the IF algorithm is highly sensitive to algorithm parameter settings. It would therefore

be useful to be able to use optimal algorithm parameters without first having to tune the parameters for the specific problem at hand.

Another aspect to consider is the rate of convergence. Theorem 2.2.3 only states that the IF1 algorithm will converge but does not specify the convergence rate. The rate of convergence can in fact vary quite a lot depending on the algorithm parameters and specifically depending on the gain sequence. For this question could theoretical analysis be employed to investigate the rate of convergence.

Further research regarding the AIF algorithm should focus on how the Hessian approximation can be improved. An idea for this research could be to base the Hessian approximation on the gradient approximation that is calculated in the IF1 algorithm instead of employing the 2SG algorithm. It could also be interesting to develop a on-line version of the AIF algorithm. An on-line version of the AIF algorithm might outperform the AIF algorithm, (assuming that the approximation of the Hessian matrix is good), since the on-line version would allow algorithm parameters that carried the most information about the problem at all times. This is of course a very non-formal argument for an on-line version of the AIF algorithm but nevertheless captures a key advantage for an on-line version of the AIF algorithm.

Chapter 6

Conclusions

This final chapter provides a summary of the main findings and conclusions in this thesis.

6.1 Main Conclusion and Findings

The research questions for this thesis were stated in Section 1.2 and the main objective was to introduce an improved version of the IF1 algorithm. This objective is partially fulfilled since the AIF algorithm is introduced in this thesis. However, the AIF algorithm is not superior to the IF1 algorithm in a convergence property sense.

The key idea of the AIF algorithm is to utilize second-order information and thereby construct an optimal gain sequence, this is done by relating the gain sequence to the Hessian matrix. The Hessian matrix is approximated by incorporating the framework in [Spall, 2005] into the IF setting. The Hessian matrix is however hard to approximate and the main problem regarding the AIF algorithm is the poor approximation of the Hessian matrix, which causes the AIF algorithm to perform badly, particular for more advanced models where a good approximation of the Hessian matrix is harder to obtain.

The key finding from the simulation studies in Chapter 4 are that the IF1 algorithm is the algorithm that performs best. The IF1 algorithm outperforms both the AIF algorithm and the IF2 algorithm for the two more complex models and the IF1 and AIF algorithms perform similar for the simple AR(1) model. An expected result would however be that the IF2 would outperform the other algorithm since the IF2 algorithm has been found to be superior to the IF1 algorithm [Ionides et al., 2015]. These findings might occur because of bad algorithm parameter settings for the IF2, however the algorithm parameters of the IF2 algorithm are hard to interpret and this thesis has been focused on the IF1 algorithm, so the performance of the IF2 algorithm might be explained by non-optimal choices of algorithm parameters.

The IF algorithm is an effective algorithm to estimate model parameters in

POMP models, and both the IF1 and IF2 algorithm can carry out this task well. However, the IF algorithm can require some tuning of the algorithm parameters before the IF algorithm is stable and converges properly. A not insignificant amount of time in this project has been spent on tuning the algorithms to obtain algorithms that have pleasant properties (i.e. stable and converge properly). Hence, the IF algorithm might not be the easiest algorithm to work with, but if the algorithm parameters are well specified for the problem at hand is it possible to estimate the model parameters in even complex and non-linear POMDP models.

6.2 Final Thoughts

The AIF algorithm introduced in this thesis is a IF algorithm built upon the IF1 algorithm. The main advantage of the AIF algorithm is the utilization of second-order information, which allows the AIF algorithm to employ an optimal gain sequence. However, the AIF algorithm still has some of the drawbacks as the other IF algorithms since the AIF algorithm requires tuning of the algorithm parameters. Furthermore, the Hessian approximation in the AIF algorithm is not satisfying since the Hessian approximation is very crude. However, an AIF algorithm where a better approximation of the Hessian matrix is incorporated has good prospects to perform as well as or better than the IF1 and IF2 algorithm since the AIF algorithm can employ a theoretical near optimal gain sequence.

Bibliography

- [Bretó, 2014] Bretó, C. (2014). On idiosyncratic stochasticity of financial leverage effects. *Statistics & Probability Letters*, 91:20–26.
- [Chan and Tsay, 2011] Chan, K.-S. and Tsay, R. S. (2011). Discussion of” feature matching in time series modeling” by y. xia and h. tong. *Statistical science*, 26(1):53–56.
- [Fernández-Villaverde and Rubio-Ramírez, 2007] Fernández-Villaverde, J. and Rubio-Ramírez, J. F. (2007). Estimating macroeconomic models: A likelihood approach. *The Review of Economic Studies*, 74(4):1059–1087.
- [He et al., 2009] He, D., Ionides, E. L., and King, A. A. (2009). Plug-and-play inference for disease dynamics: measles in large and small populations as a case study. *Journal of the Royal Society Interface*.
- [Ionides et al., 2006] Ionides, E., Bretó, C., and King, A. (2006). Inference for nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 103(49):18438–18443.
- [Ionides et al., 2012] Ionides, E., Frydendall, J., Madsen, H., et al. (2012). Efficient iterated filtering. In *System Identification*, volume 16, pages 1785–1790.
- [Ionides et al., 2011] Ionides, E. L., Bhadra, A., Atchadé, Y., King, A., et al. (2011). Iterated filtering. *The Annals of Statistics*, 39(3):1776–1802.
- [Ionides et al., 2015] Ionides, E. L., Nguyen, D., Atchadé, Y., Stoev, S., and King, A. A. (2015). Inference for dynamic and latent variable models via iterated, perturbed bayes maps. *Proceedings of the National Academy of Sciences*, 112(3):719–724.
- [Johannes et al., 2009] Johannes, M. S., Polson, N. G., and Stroud, J. R. (2009). Optimal filtering of jump diffusions: Extracting latent states from asset prices. *Review of Financial Studies*, page hhn110.
- [Kantas et al., 2009] Kantas, N., Doucet, A., Singh, S. S., and Maciejowski, J. M. (2009). An overview of sequential monte carlo methods for parameter estimation in general state-space models. In *15th IFAC Symposium on System Identification (SYSID), Saint-Malo, France.(invited paper)*, volume 102, page 117.

- [King et al., 2008] King, A. A., Ionides, E. L., Pascual, M., and Bouma, M. J. (2008). Inapparent infections and cholera dynamics. *Nature*, 454(7206):877–880.
- [King et al., 2015] King, A. A., Nguyen, D., and Ionides, E. L. (2015). Statistical inference for partially observed markov processes via the r package pomp. *arXiv preprint arXiv:1509.00503*.
- [Lindström, 2013] Lindström, E. (2013). Tuned iterated filtering. *Statistics & Probability Letters*, 83(9):2077–2080.
- [Spall, 2000] Spall, J. C. (2000). Adaptive stochastic approximation by the simultaneous perturbation method. *Automatic Control, IEEE Transactions on*, 45(10):1839–1853.
- [Spall, 2005] Spall, J. C. (2005). *Introduction to stochastic search and optimization: estimation, simulation, and control*, volume 65. John Wiley & Sons.