INAUGURAL - DISSERTATION

zur

Erlangung der Doktorwürde

der

Naturwissenschaftlich-Mathematischen Gesamtfakultät

der

Ruprecht-Karls-Universität

Heidelberg

vorgelegt von

Diplom-Mathematiker mit Ausrichtung Wissenschaftliches Rechnen

und Diplom-Volkswirt

Jan Christoph Neddermeyer

aus Darmstadt

Tag der mündlichen Prüfung: 15. Juli 2010

Importance Sampling-Based Monte Carlo Methods with Applications to Quantitative Finance

Gutachter:	Prof. Dr. Rainer Dahlhaus
	Prof. Dr. Dieter W. Heermann

Abstract

In the present work advanced Monte Carlo methods for discrete-time stochastic processes are developed and investigated. A particular focus is on sequential Monte Carlo methods (particle filters and particle smoothers) which allow the estimation of nonlinear, non-Gaussian state-space models. The key technique which underlies the proposed algorithms is importance sampling. Computationally efficient nonparametric variants of importance sampling which are generally applicable are developed. Asymptotic properties of these methods are analyzed theoretically and it is shown empirically that they improve over existing methods for relevant applications. Particularly, it is shown that they can be applied for financial derivative pricing which constitutes a high-dimensional integration problem and that they can be used to improve sequential Monte Carlo methods.

Original models in general state-space form for two important applications are proposed and new sequential Monte Carlo algorithms for their estimation are developed. The first application concerns the on-line estimation of the spot cross-volatility for ultra high-frequency financial data. This is a challenging problem because of the presence of microstructure noise and non-synchronous trading. For the first time state-space models with non-synchronously evolving states and observations are discussed and a particle filter which can cope with these models is designed. In addition, a new sequential variant of the EM algorithm for parameter estimation is proposed. The second application is a non-linear model for time series with an oscillatory pattern and a phase process in the background. This model can be applied, for instance, to noisy quasi-periodic oscillators occurring in physics and other fields. The estimation of the model is based on an advanced particle smoother and a new nonparametric EM algorithm.

The dissertation is accompanied by object-oriented C++ implementations of all proposed algorithms which were developed with a focus on reusability and extendability.

Zusammenfassung

In der vorliegenden Arbeit werden fortgeschrittene Monte-Carlo-Verfahren für zeitdiskrete stochastische Prozesse entwickelt und untersucht. Ein Schwerpunkt wird dabei auf sequentielle Monte-Carlo-Verfahren (Partikel-Filter und Partikel-Smoother) gesetzt; diese werden zur Schätzung von nichtlinearen, nicht-Gauß'schen State-Space-Modellen verwendet. Importance Sampling ist die Schlüsselmethode, auf der die entwickelten Algorithmen basieren. Es werden nichtparametrische Varianten des Importance Samplings entwickelt, die recheneffizient und allgemein anwendbar sind. Asymptotische Eigenschaften dieser Methoden werden theoretisch untersucht und es wird anhand relevanter Anwendungen gezeigt, dass sie bessere Ergebnisse liefern als existierende Verfahren. Insbesondere wird gezeigt, dass sie für die Bewertung von Finanzderivaten, ein hochdimensionales Integrationsproblem, verwendet werden können und dass sie benutzt werden können um sequentielle Monte-Carlo-Verfahren zu verbessern.

Für zwei wichtige Anwendungen werden neue Modelle in State-Space-Form entwickelt und sequentielle Monte-Carlo-Algorithmen beschrieben, die für deren Schätzung genutzt werden können. Die erste Anwendung betrifft die Online-Schätzung der Spot Kreuz-Volatilität für ultrahochfrequente Finanzdaten. Aufgrund des Mikrostruktur-Rauschens und der nicht-synchronen Handelszeitpunkte stellt dies ein schwieriges Problem dar. Im Zuge dieser Anwendung werden erstmals State-Space Modelle mit nicht-synchronen Zuständen und Beobachtungen betrachtet und ein Partikel-Filter konstruiert, der für solche Modelle geeignet ist. Außerdem wird ein neuartiger sequentieller EM-Algorithmus für die Parameter-Schätzung entwickelt. Als zweite Anwendung wird ein nichtlineares Modell für Zeitreihen vorgeschlagen, die ein periodisches Muster und einen latenten Phasen-Prozess aufweisen. Dieses Modell kann u.a. verwendet werden um verrauschte quasi-periodische Oszillatoren zu beschreiben, die in verschiedenen Disziplinen (z.B. der Physik) vorkommen. Die Schätzung dieses Modells basiert auf einem erweiterten Partikel-Smoother und einem neuen nichtparametrischen EM-Algorithmus.

Teil dieser Dissertation sind außerdem objektorientierte C++-Implementierungen aller vorgeschlagenen Algorithmen, die besonders auf die Wiederverwendbarkeit und Erweiterbarkeit Wert legen.

Acknowledgements

I extend my thanks to my supervisor Prof. Dr. Rainer Dahlhaus and my co-supervisor Prof. Dr. Dieter W. Heermann for giving me the opportunity to work on this exciting topic. Particularly, I like to thank Prof. Dr. Rainer Dahlhaus for his invaluable support in terms of scientific advice and hardware/software equipment. I would also like to express my gratitude to Dr. Cornelia Wichelhaus, Prof. Dr. Jan Johannes, Konstantinos Paraschakis, Dr. Markus Fischer, and Julian Kunkel for very useful comments and inspiring discussions at different stages of this project. Similarly, my gratitude goes Dr. Ulrich Brandt-Pollmann who made possible my three month stay in London. Finally, I want to thank my wife Paula and the rest of my family for their continuing support.

This work was supported by the Deutsche Forschungsgemeinschaft under DA 187/15-1 and by the University of Heidelberg under Frontier D.801000/08.023. In addition, I am grateful to Heidelberg Graduate School of Mathematical and Computational Methods for the Sciences for funding my travels to various conferences and workshops.

Contents

Appendix

Abbreviations

APF	Auxiliary particle filter
BSPS	Backward simulation particle smoother
CDIS	Change-of-drift importance sampling
CV	Coefficient of variation
ED	Effective dimension
GSSM	General state-space model
IS	Importance sampling
LBFP	Linear blend frequency polygon
LSIS	Least-squares importance sampling
MC	Monte Carlo
NPIS	Nonparametric partial importance sampling
NIS	Nonparametric importance sampling
NPF	Nonparametric particle filter
NPS	Nonparametric particle smoother
NSIS	Nonparametric self-normalized importance sampling
PCA	Principal component analysis
QMC	Quasi-Monte Carlo
RBPS	Rao-Blackwellized particle smoother
RCE	Relative computational efficiency
RE	Relative efficiency
RMSE	Root mean square error
SIRMH	Bootstrap particle filter with Metropolis-Hastings moves
SCVE	Spot cross-volatility estimation
SPS	Simple particle smoother
SVE	Spot volatility estimation
VR	Variance reduction

Chapter 1

Introduction

1.1 Overview of the Problems, Methods, and Applications

The simulation of stochastic processes and the approximation of complex, high-dimensional integrals which depend on stochastic processes are frequent problems in many fields. Numerical integration schemes are often infeasible as a result of the curse of dimensionality and computational limitations. Monte Carlo simulation is frequently the only tractable method because its convergence rate is independent of the problem dimension. However, if the problem dimension is large or the integrand very irregular crude Monte Carlo is inefficient. This establishes a need for advanced Monte Carlo algorithms. In the last decades, advanced Monte Carlo methods became more and more relevant for practical applications as a result of increasing computing power. In particular, in Bayesian inference Monte Carlo methods such as Markov Chain Monte Carlo and sequential Monte Carlo methods experienced a distinct surge in popularity.

In this dissertation, the main focus is on sequential, discrete-time models and methods. Discrete-time models occur, for instance, as discretizations of continuous-time stochastic processes. The on-line or sequential estimation which is frequently required when dealing with stochastic processes is of key focus in this work. However, off-line settings are also considered. New Monte Carlo methods are proposed and analyzed concerning both theoretical and computational aspects. Efficient software implementations of the algorithms are provided. The usefulness of the proposed methods is verified through relevant applications. Several complex applications in the field of quantitative finance are considered in detail.

The technique which constitutes the fundamental concept of the methods developed in this dissertation is importance sampling. This is a very flexible sampling method which can be used to generate random samples from intractable distributions or to reduce the Monte Carlo variance. It is frequently applied to rare event simulation. A typical application is the computation of rare event probabilities. Already in the 1950s, importance sampling were used in rare event applications in physics (Kahn 1950; Kahn and Marshall 1953). However, importance sampling is much more powerful and by far not limited to the simulation of rare events. Generally speaking, almost any Monte Carlo approximation can be improved significantly through the use of importance sampling. The basic idea of importance sampling is to generate samples from an auxiliary distribution which is known as proposal instead of from the target distribution. Subsequently, the samples are weighted such that they approximate the target distribution. The main difficulty of applying importance sampling in practice is the choice of a suitable proposal. This issue is tackled in this work.

Most existing importance sampling methods are based on a parametric choice of the proposal, that is the proposal is chosen from a parametrized family of distributions. In addition, nonparametric importance sampling methods have been developed. They are based on nonparametric approximations of the (optimal) proposal. Until now, nonparametric importance sampling was merely a nice theoretical alternative to parametric importance sampling with no practical applications. The reason for this in founded in the computational inefficiency of existing nonparametric importance sampling techniques. In this dissertation, computationally efficient nonparametric importance sampling algorithms which are suitable for practical application are proposed and investigated.

A relevant application where (nonparametric) importance sampling can be effectively applied is the pricing of path-dependent financial derivatives. There, Monte Carlo approximations of complex high-dimensional integrals are required. In addition, the computational efficiency of the method used is important because the evaluation of financial derivative prices is often time-critical. Although parametric importance sampling methods already belong to the standard toolbox in financial engineering, nonparametric importance sampling techniques have not been applied until now. The evolution of a financial asset can be described through a stochastic differential equation with a Brownian motion as driving process. Based on this model the price of a European option can be approximated through a high-dimensional integral which depends on a discretization of the stochastic differential equation. It is shown that the proposed nonparametric importance sampling algorithms lead to massive efficiency gains for such kind of integration problems.

In this work, particle filters and particle smoothers which belong to the class of sequential Monte Carlo methods are of particular interest. Sequential Monte Carlo methods are Bayesian simulation techniques that allow the approximation of the filtering and smoothing distributions of general state-space models. Numerous applications which comply with the class of general state-space models are readily available, for instance object tracking problems in engineering and stochastic volatility estimation in finance. General state-space models often occur naturally as discretizations of stochastic differential equations with hidden components. In contrast to the traditional linear state-space models, general state-space models allow for nonlinear functions and non-Gaussian noise distributions. Consequently, standard methods for filtering and smoothing in (linear) state-space models such as the Kalman filter and the Kalman smoother can usually not be applied.

An essential ingredient of the sequential Monte Carlo methods considered here is importance sampling. It is required because direct sampling from the target distribution is impossible. A goal is to develop more efficient particle filters and smoothers by employing nonparametric importance sampling schemes and quasi-Monte Carlo techniques.

In this dissertation, two new applications of general state-space models and sequential Monte

Carlo methods, which are of great importance, are considered in detail. Both are hot topics in their areas of research. The methods and models proposed in this work are original contributions and they have not been used for these applications before. The first application is the on-line estimation of spot cross-volatility for ultra high-frequency financial data. The spot cross-volatility is the key quantity in risk management, portfolio optimization, and trading. The main problems are the presence of so-called market microstructure noise and the non-synchronous trading times of different securities. Particularly for the non-synchronous trading times there is a lack of appropriate models. Our approach is different from existing approaches and it includes several new modeling and estimation aspects. It is shown, in particular, that ultra high-frequency financial data can be effectively treated in a nonlinear state-space framework.

The second application concerns the estimation of a general time series model which is also newly proposed. It is a model for stationary time series with a specific oscillatory component which can be written in state-space form. This model includes quasi-periodic oscillators with a latent phase process in the background. Our approach is very general and allows the modeling and estimation of nonlinear phase transitions, time-varying amplitudes, baseline shifts, and general oscillatory patterns. In particular, the estimation of the phase is of interest because it is, for instance, required for the analysis of phase synchronization of coupled oscillators. Many applications complying with our model exist in different fields such as physics, engineering, and neuroscience. An interesting example which is briefly considered in this dissertation are electrocardiogram (ECG) recordings measuring the electrical activity of the heart over time. For ECG data existing methods for phase estimation such as the Hilbert transform are inappropriate because of baseline shifts and a non-trigonometric oscillatory pattern which are present in the data. Our method not only allows inference on the phase but also the nonparametric estimation of the characteristic oscillatory pattern.

1.2 Outline of the Results

The results of this dissertation are presented in several research papers which are already published or available as preprints. The following description summarizes the major contributions of this work and indicates the corresponding research papers. Chapter 2 gives a literature review which provides the foundation of the present work. Chapter 8 overviews the software packages which were developed.

Chapter 3 (Neddermeyer 2009)

The variance reduction established by importance sampling strongly depends on the choice of the importance sampling distribution. A good choice is often hard to achieve especially for high-dimensional integration problems. It is shown that nonparametric estimation of the optimal importance sampling distribution (known as nonparametric importance sampling) is a reasonable alternative to parametric approaches. New nonparametric variants of both the self-normalized and the unnormalized importance sampling estimator are proposed and investigated. A common critique on nonparametric importance sampling is the increased computational burden compared with parametric methods. This problem is solved to a large degree by utilizing the linear blend frequency polygon estimator instead of a kernel estimator. Mean square error convergence properties are investigated theoretically leading to recommendations for the efficient application of nonparametric importance sampling. Particularly, it is shown that nonparametric importance sampling asymptotically attains optimal importance sampling variance. As an application, the estimation of the distribution of the queue length of a spam filter queueing system based on real data is considered.

Chapter 4 (Neddermeyer 2010a)
It is shown how nonparametric importance sampling can be effectively used for financial derivative pricing. Standard nonparametric importance sampling is inefficient for this task because the approximation of high-dimensional integrals are required. This issue is solved by applying the procedure to a low-dimensional subspace, which is identified through principal component analysis and the concept of the effective dimension. This leads to the method of nonparametric partial importance sampling. The mean square error properties of the algorithm are investigated and its asymptotic optimality is shown. Quasi-Monte Carlo is used for further improvement of the method. It is demonstrated through path-dependent and multi-asset option pricing problems that the algorithm leads to significant efficiency gains compared with existing methods.

Chapter 5 (Neddermeyer 2010b)
An original particle filter and an original particle smoother which employ nonparametric importance sampling are developed. It is shown that these algorithms provide a better approximation of the filtering and smoothing distributions than standard methods. The methods' advantage is most distinct in severely nonlinear situations. In contrast to most existing methods, they allow the use of quasi-Monte Carlo sampling. In addition, they do not suffer from weight degeneration rendering unnecessary a resampling step. For the estimation of model parameters an efficient on-line maximum likelihood estimation technique is proposed which is also based on nonparametric approximations. All suggested algorithms have almost linear complexity for low-dimensional state-spaces. This is an advantage over standard smoothing and maximum likelihood procedures. Particularly, all existing sequential Monte Carlo methods that incorporate quasi-Monte Carlo sampling have quadratic complexity. As an application, stochastic volatility estimation for high-frequency financial data is considered, which is of great importance in practice.

Chapter 6 (Dahlhaus and Neddermeyer 2010a, b)
We develop a new technique for the on-line estimation of both time-constant and time-varying spot covariance matrices (spot cross-volatilities) in the presence of market microstructure noise. The algorithm works directly on the non-synchronous transaction data and updates the covariance estimate immediately after the occurrence of a new transaction. The transaction prices are considered as noisy observations of latent efficient log-price processes. A new transaction time model for the efficient log-prices is proposed which models

the evolution of different securities in individual transaction times. In addition, a new nonlinear market microstructure noise model is developed which reproduces the major stylized facts of high-frequency data such as the price discreteness and the negative first-order autocorrelation of the returns. Our model takes the form of a nonlinear state-space model with non-synchronous states and observations. Based on this representation a new particle filter is designed that allows the approximation of the filtering distributions of the efficient log-prices. It is shown that the spot covariance matrix of the latent log-price processes can be estimated as a parameter of the state-space model. For this purpose we propose a sequential variant of the EM algorithm that uses the output of the particle filter. For the univariate case we also propose an on-line bias correction and a method for adaptive step size selection. The practical usefulness of our technique is verified through Monte Carlo simulations and through an application to real transaction data.

Chapter 7 (Dahlhaus and Neddermeyer 2009)

We introduce a new model for stationary time series with a quasi-periodic component. The aim is to model time series with a specific fluctuation pattern and an unobserved phase process in the background. The model also includes a time-varying amplitude and baseline. This allows the modeling of data occurring in physics, biology, life science, and many other fields. The goals are to estimate the unobserved phase, amplitude, and baseline processes as well as the fluctuation pattern. The model can be written as a nonlinear, non-Gaussian state-space model treating the phase, the amplitude, and the baseline as latent Markov processes. For the estimation, we suggest a Rao-Blackwellized particle smoother that combines the Kalman smoother and an efficient sequential Monte Carlo smoother. For the estimation of the fluctuation pattern, an original nonparametric EM algorithm is developed. The proposed algorithms can be applied on-line, they are easy to implement and computationally efficient. The method's potential for practical applications is demonstrated through simulations and an application to human electrocardiogram recordings.

Chapter 2

Monte Carlo Methods and General State-Space Models

This chapter introduces notation and provides a brief overview of major concepts of Monte Carlo simulation which are relevant for the methods developed in the following chapters.

2.1 Monte Carlo Approximation

Monte Carlo simulation can be used to approximate a probability density function p on $\mathbb{R}^d$ or the expectation

$$\mathbf{E}_p[\varphi] = I_\varphi = \int \varphi(\mathbf{x})p(\mathbf{x})d\mathbf{x}$$

of some function $\varphi : \mathbb{R}^d \to \mathbb{R}$ with respect to p. Let's assume N i.i.d. samples $\{\mathbf{x}^i\}_{i=1}^N$ from p are available. Then, an empirical estimate of density p is given by

$$p(\mathbf{x}) \approx \frac{1}{N}\sum_{i=1}^{N} \delta_{\mathbf{x}^i}(\mathbf{x})$$

with δ being the Dirac delta function. The integral I_φ can be estimated through

$$\hat{I}_\varphi^{\mathrm{MC}} = \frac{1}{N}\sum_{i=1}^{N} \varphi(\mathbf{x}^i). \tag{2.1}$$

Clearly, $\hat{I}_\varphi^{\mathrm{MC}}$ is a consistent, unbiased estimator of I_φ. If the variance $\sigma^2 = \mathrm{Var}[\varphi]$ is finite then the standard central limit theorem gives

$$\sqrt{N}(\hat{I}_\varphi^{\mathrm{MC}} - I_\varphi) \Rightarrow \mathcal{N}(0, \sigma^2).$$

The central limit theorem implies that the convergence rate of (standard) Monte Carlo integration is independent of the dimension of the integrand. This constitutes a major advantage of Monte Carlo simulation-based methods compared with numerical integration techniques.

In many applications, it is impossible or at least very difficult to generate i.i.d samples from p. This is often the case when p is high-dimensional, not given in closed form, or as usual in Bayesian settings when p is only known up to a constant. In Section 2.3, the concept of importance sampling is described which allows the generation of weighted samples that approximate arbitrary probability density functions.

To reduce the (Monte Carlo) variance of the estimator (2.1) one can apply variance reduction techniques which are briefly considered in the following section.

2.2 Computational Efficiency and Variance Reduction Techniques

To make Monte Carlo algorithms comparable, it is useful to quantify the efficiency of a Monte Carlo estimator. Let's assume X is a random variable defined on a probability space $(\Omega, \mathcal{B}, \mathbf{P})$ and is used to estimate some quantity μ. The computational efficiency of the estimator X can be defined through

$$\mathrm{CE}[X] = (\mathrm{MSE}[X]C[X])^{-1},$$

where $\mathrm{MSE}[X]$ denotes the mean square error of X and $C[X]$ the average costs of computing one realization of X (L'Ecuyer 1994). From this definition, one observes that efficiency improvements can be achieved either by reducing the mean square error or the computational costs. The computational costs can be reduced through more efficient algorithms or a faster implementation. In particular, the random number generator used is a critical component because of its frequent use in Monte Carlo simulation (see Section 2.6).

Methods which are applied to reduce the mean square error $\mathrm{MSE}[X]$ are known as variance reduction techniques. Relevant methods are importance sampling, antithetic sampling, moment matching, and control variates (Jäckel 2002; Glasserman 2004; Robert and Casella 2004). Most of these techniques aim at improving a given set of samples which is used for Monte Carlo integration. More precisely, the i.i.d. samples $\{\mathbf{x}^i\}_{i=1}^N$ are transformed into non-i.i.d. samples in a way such that the mean square error of the estimator (for instance $\hat{I}_\varphi^{\mathrm{MC}}$ in (2.1)) is reduced. In contrast, importance sampling is based on a change of the distribution from which the samples are drawn. The Monte Carlo methods proposed in this dissertation are based on importance sampling. However, it is mentioned that most of the proposed methods could be (further) improved by the additional use of other variance reduction techniques.

2.3 Importance Sampling

Importance sampling is the fundamental concept underlying the Monte Carlo methods developed in this dissertation. It is a general sampling technique which can be used to approximate an integral I_φ. It is often applied if direct sampling from the distribution p is computationally too demanding or intractable. But it is not limited to this purpose. Unless φ is constant, importance sampling can often yield a massive reduction of the estimator's variance, if applied carefully. Formally, importance sampling is a change of measure. The expectation $\mathbf{E}_p[\varphi]$ is rewritten as

$$\mathbf{E}_q[\varphi w] = \int \varphi(\mathbf{x})w(\mathbf{x})q(\mathbf{x})d\mathbf{x},$$

where q is the probability density function of an importance sampling distribution (also known as proposal) and $w(\mathbf{x}) = p(\mathbf{x})/q(\mathbf{x})$ is the Radon-Nikodym derivative of p with respect to q. The proposal needs to be chosen so that its support includes the support of $|\varphi|p$ or p, which imposes a first constraint on q. Using importance sampling the integral I_φ can be estimated by

$$\hat{I}_\varphi^{\mathrm{IS}} = \frac{1}{N}\sum_{i=1}^{N} \varphi(\mathbf{x}^i)w(\mathbf{x}^i), \tag{2.2}$$

where the samples $\{\mathbf{x}^i\}_{i=1}^N$ are drawn from proposal q. Note, $\hat{I}_\varphi^{\mathrm{IS}}$ is an unbiased estimator of I_φ.

In Bayesian inference, it is often the case that either p or the proposal q (or both) are only known up to some constant. In this case an alternative is the self-normalized importance sampling estimator given by

$$\hat{I}_\varphi^{\mathrm{SIS}} = \frac{\sum_{i=1}^N \varphi(\mathbf{x}^i)w(\mathbf{x}^i)}{\sum_{i=1}^N w(\mathbf{x}^i)}. \tag{2.3}$$

In contrast to $\hat{I}_\varphi^{\mathrm{IS}}$, $\hat{I}_\varphi^{\mathrm{SIS}}$ is biased. The strong law of large numbers implies that both $\hat{I}_\varphi^{\mathrm{IS}}$ and $\hat{I}_\varphi^{\mathrm{SIS}}$ converge almost surely to the expectation I_φ if it is finite. However, this result is neither of help for assessing the precision of the estimators for a finite set of samples nor for the rate of convergence. In order to construct error bounds, it is desirable to have a central limit theorem at hand. Under the assumptions that I_φ and $\mathrm{Var}_q[\varphi w]$ are finite, a central limit theorem guaranties

$$\sqrt{N}(\hat{I}_\varphi^{\mathrm{IS}} - I_\varphi) \Rightarrow \mathcal{N}(0, \sigma_{\mathrm{IS}}^2),$$

where $\sigma_{\mathrm{IS}}^2 = \mathbf{E}_q[\varphi w - I_\varphi]^2$ (Rubinstein 1981). The proposal which minimizes the variance σ_{IS}^2 is given by

$$q_\varphi^{\mathrm{IS}}(\mathbf{x}) = \frac{|\varphi(\mathbf{x})|p(\mathbf{x})}{\int |\varphi(\mathbf{x})|p(\mathbf{x})d\mathbf{x}}. \tag{2.4}$$

q_φ^{IS} is called the optimal proposal. Remarkably, the importance sampling estimator based on the optimal proposal has zero variance for functions φ with a definite sign. However, the optimal proposal is unavailable in practice because of its unknown denominator. A central limit theorem for the self-normalised importance sampling estimator $\hat{I}_\varphi^{\mathrm{SIS}}$ can be established

$$\sqrt{N}(\hat{I}_\varphi^{\mathrm{SIS}} - I_\varphi) \Rightarrow \mathcal{N}(0, \sigma_{\mathrm{SIS}}^2) \tag{2.5}$$

with limiting variance $\sigma_{\mathrm{SIS}}^2 = \mathbf{E}_q[(\varphi - I_\varphi)w]^2$ under the additional assumption that $\mathrm{Var}_q[w] < \infty$ (Geweke 1989). The variance σ_{SIS}^2 is minimized by the proposal

$$q_\varphi^{\mathrm{SIS}}(\mathbf{x}) = \frac{|\varphi(\mathbf{x}) - I_\varphi|p(\mathbf{x})}{\int |\varphi(\mathbf{x}) - I_\varphi|p(\mathbf{x})d\mathbf{x}}, \tag{2.6}$$

provided that the median of φ with respect to p exists. The optimal proposals (2.4) and (2.6) are merely of conceptual help, because the computation of their denominators is typically at least as difficult as the original integration problem. In practice, the objective is to find an easy-to-sample density that approximates the optimal proposals. Traditionally, a proposal is chosen from some parametric family of densities $\{q_{\varphi,\theta}; \theta \in \Theta\}$ that satisfy the assumptions of the central limit theorems or some related conditions. Typically, it is demanded that the support of $q_{\varphi,\theta}$ includes the support of $|\varphi|p$ or $|\varphi - I_\varphi|p$, respectively, and that the tails of q do not decay faster than those of $|\varphi|p$. Many different density classes have been investigated in the literature including multivariate Student t, mixture, and exponential family distributions (see for instance Geweke 1989; Stadler and Roy 1993; Oh and Berger 1993). The parametrized choice of the proposal can be adaptively revised during the importance sampling which is known as adaptive importance sampling (Oh and Berger 1992; Kollman et al. 1999). Often expectation I_φ needs to be computed for many different functions φ leading to different optimal proposals. Consequently, it is necessary to investigate the structure of any new φ in order to find a suitable parametric family.

2.4 General State-Space Models and Sequential Monte Carlo

A general state-space model describes the joint evolution of a hidden state sequence and an observation sequence. The states $\mathbf{X}_t$, $t = 0, 1, \ldots$, taking values in $\mathbb{R}^d$ constitute an unobserved Markov process. The observations $\mathbf{Y}_t$, $t = 1, 2, \ldots$, which take values in $\mathbb{R}^s$, are conditionally independent given the states. A general state-space model is fully specified by the transition distributions and the observation distributions

$$\begin{aligned} \mathbf{X}_t|\mathbf{X}_{t-1} &\sim p(\mathbf{x}_t|\mathbf{x}_{t-1}), \quad (2.7)\\ \mathbf{Y}_t|\mathbf{X}_t &\sim p(\mathbf{y}_t|\mathbf{x}_t). \quad (2.8)\end{aligned}$$

The initial state $\mathbf{X}_0$ is distributed according to some prior density $p(\mathbf{x}_0)$. Alternatively, a general state-space model can be specified through a state equation and an observation equation

$$\begin{aligned} \mathbf{X}_t &= f_t(\mathbf{X}_{t-1}, \eta_t), \quad (2.9)\\ \mathbf{Y}_t &= g_t(\mathbf{X}_t, \epsilon_t), \quad (2.10)\end{aligned}$$

where f_t, g_t are (nonlinear) functions and η_t, ϵ_t are mutually and serially independent noise variables.

Many stochastic processes with latent components occurring in engineering, physics, finance, and other fields have a natural (general) state-space representation (see Doucet, de Freitas, and Gordon 2001; Lin et al. 2005 and the references there). For instance, discretizations of stochastic differential equations often lead directly to state-space models. A state-space model has a nice representation in terms of a graphical model (see Figure 2.1) and it exhibits certain conditional independence properties which can be easily verified, for instance, $p(\mathbf{y}_t|\mathbf{x}_{0:t}, \mathbf{y}_{1:t-1}) = p(\mathbf{y}_t|\mathbf{x}_t)$ and $p(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t-1}) = p(\mathbf{x}_t|\mathbf{x}_{t-1})$. Note, the notation $\mathbf{y}_{1:t} = \{\mathbf{y}_1, \ldots, \mathbf{y}_t\}$ is used throughout this dissertation.

The objective is to compute the filtering distributions $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ or smoothing distributions $p(\mathbf{x}_t|\mathbf{y}_{1:T})$ (where $T > t$) of the hidden state variable $\mathbf{X}_t$. In addition, the posterior distributions

$$\begin{tikzcd}
 & \mathbf{Y}_1 & \cdots & \mathbf{Y}_{t-1} & \mathbf{Y}_t & \mathbf{Y}_{t+1} & \cdots \\
\mathbf{X}_0 \arrow[r] & \mathbf{X}_1 \arrow[u] \arrow[r] & \cdots \arrow[r] & \mathbf{X}_{t-1} \arrow[u] \arrow[r] & \mathbf{X}_t \arrow[u] \arrow[r] & \mathbf{X}_{t+1} \arrow[u] \arrow[r] & \cdots
\end{tikzcd}$$

Figure 2.1: Representation of a state-space model with observations $\{\mathbf{Y}_t\}_{t\geq 1}$ and states $\{\mathbf{X}_t\}_{t\geq 0}$ as a directed acyclic graph.

$p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$ are often of interest. In the case when the functions and noise variables in the state equation (2.9) and observation equation (2.10) are linear and Gaussian, respectively, the well-known Kalman filter and Kalman smoother represent the optimal procedures for computing these distributions. For the non-Gaussian case, various methods have been suggested: The extended Kalman filter, the unscented Kalman filter (Julier and Uhlmann 1997), grid-based filters, particle filters (Gordon, Salmond, and Smith 1993; Kitagawa 1996), and particle smoothers (Godsill, Doucet, and West 2004) are among these. See Arulampalam et al. (2002) or Neddermeyer (2007) for an overview.

Now, particle filters and particle smoothers are considered which belong to the class of sequential Monte Carlo methods (Doucet, de Freitas, and Gordon 2001). They are based on the idea to approximate the distributions of interest sequentially by sets of weighted samples $\{\mathbf{x}_t^i, \omega_t^i\}_{i=1}^N$, $t \geq 0$, termed as particles. Typically, it is desired to compute the expectation I_{φ_t} of some function $\varphi_t(\mathbf{x}_t)$ with respect to the filtering or smoothing distributions. Given particles which approximate the filtering or smoothing distribution at time t, I_{φ_t} can be estimated through

$$\hat{I}_{\varphi_t} = \sum_{i=1}^{N} \omega_t^i \varphi_t(\mathbf{x}_t^i).$$

The estimator $\hat{I}_{\varphi_t}$ converges almost surely to I_{φ_t} and achieves mean square error rate $\mathcal{O}(N^{-1})$ under appropriate conditions on the general state-space model and the particle methods used (Crisan 2001; Crisan and Doucet 2002; Godsill, Doucet, and West 2004). In addition, central limit theorems have been proven (Del Moral and Guionnet 1999; Chopin 2004). Note that the particles $\{\mathbf{x}_t^i, \omega_t^i\}_{i=1}^N$ are not i.i.d. and, therefore, standard convergence results do not apply.

In the filtering setting, the particles are obtained sequentially in time by making use of the relation

$$\begin{aligned}
p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) &= \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})} \qquad (2.11)\\
&\propto p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})
\end{aligned}$$

which follows from the conditional independence properties of state-space models. Note, the constant $p(\mathbf{y}_t|\mathbf{y}_{1:t-1})$ is unknown but does not depend on $\mathbf{X}_{0:t}$. The distributions $p(\mathbf{y}_t|\mathbf{x}_t)$ and $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ are termed likelihood and transition prior, respectively. The iteration of the basic particle filter is based on sequential importance sampling with proposal $q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_t)$

and can be stated as follows (Gordon, Salmond, and Smith 1993): Assume weighted particles $\{\mathbf{x}^i_{0:t-1}, \omega^i_{t-1}\}^N_{i=1}$ approximating $p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})$ are given.

- For $i = 1, \ldots, N$:
  - Sample $\mathbf{x}^i_t \sim q(\mathbf{x}_t|\mathbf{x}^i_{t-1}, \mathbf{y}_t)$.
  - Compute importance weights $\breve{\omega}^i_t \propto \omega^i_{t-1} p(\mathbf{y}_t|\mathbf{x}^i_t) p(\mathbf{x}^i_t|\mathbf{x}^i_{t-1})/q(\mathbf{x}^i_t|\mathbf{x}^i_{t-1}, \mathbf{y}_t)$.
- For $i = 1, \ldots, N$:
  - Normalize importance weights $\omega^i_t = \breve{\omega}^i_t/(\sum^N_{j=1} \breve{\omega}^j_t)$.

The particles $\{\mathbf{x}^i_{0:t}, \omega^i_t\}^N_{i=1}$ are obtained, which approximate the target distribution

$$p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) \approx \sum_{i=1}^{N} \omega^i_t \delta_{\mathbf{x}^i_{0:t}}(\mathbf{x}_{0:t}).$$

Through marginalization one obtains approximations of the filtering distribution

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \approx \sum_{i=1}^{N} \omega^i_t \delta_{\mathbf{x}^i_t}(\mathbf{x}_t)$$

and the smoothing distribution

$$p(\mathbf{x}_s|\mathbf{y}_{1:t}) \approx \sum_{i=1}^{N} \omega^i_t \delta_{\mathbf{x}^i_s}(\mathbf{x}_s) \tag{2.12}$$

with $s < t$. However, the approximation of the smoothing distribution is poor if s is not close to t.

To understand the basic particle filter in more detail, let's consider the unknown constant

$$p(\mathbf{y}_t|\mathbf{y}_{1:t-1}) = \int p(\mathbf{y}_t|\mathbf{x}_t) p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) d\mathbf{x}_t.$$

It is easy to verify that the empirical mean of the unnormalized importance weights $\frac{1}{N} \sum^N_{j=1} \breve{\omega}^j_t$, which is computed in the particle filter, precisely estimates this constant. Therefore, the basic particle filter can be interpreted as a Monte Carlo method to compute the unknown integrals $p(\mathbf{y}_t|\mathbf{y}_{1:t-1})$.

The choice of the proposal is crucial for the efficiency of particle filters. It is often hard to find a suitable proposal which incorporates the observation $\mathbf{y}_t$. The trivial choice is $q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_t) = p(\mathbf{x}_t|\mathbf{x}_{t-1})$. Particle filters which choose non-trivial proposals in an automatic fashion have been suggested, e. g. the auxiliary particle filter (Pitt and Shephard 1999) and the unscented particle filter (van der Merwe et al. 2000).

The basic particle filter and other particle filters suffer from weight degeneracy which means that after a few time steps only a small number of particles have significant weight. Weight degeneration is worsened when no good proposal is available. This problem is usually tackled

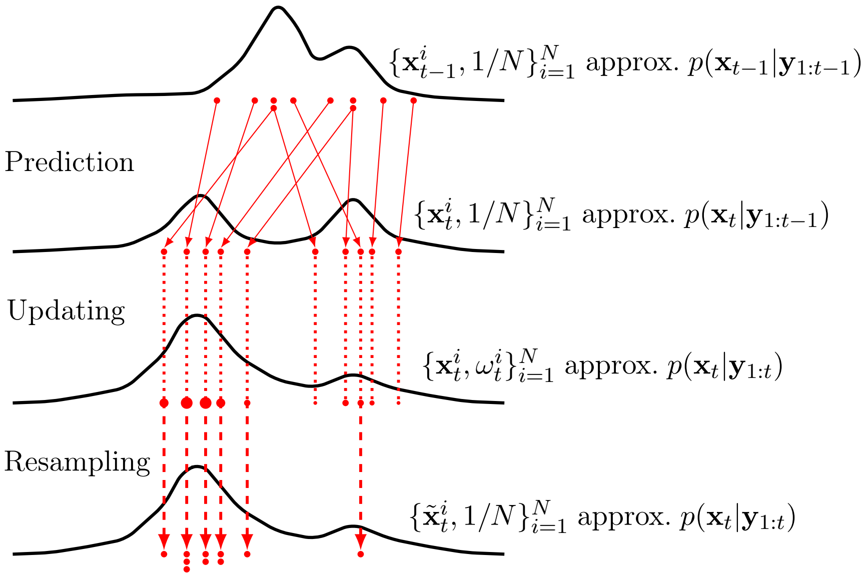
$\{\mathbf{x}_{t-1}^i, 1/N\}_{i=1}^N$ approx. $p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})$
Prediction
$\{\mathbf{x}_t^i, 1/N\}_{i=1}^N$ approx. $p(\mathbf{x}_t|\mathbf{y}_{1:t-1})$
Updating
$\{\mathbf{x}_t^i, \omega_t^i\}_{i=1}^N$ approx. $p(\mathbf{x}_t|\mathbf{y}_{1:t})$
Resampling
$\{\tilde{\mathbf{x}}_t^i, 1/N\}_{i=1}^N$ approx. $p(\mathbf{x}_t|\mathbf{y}_{1:t})$

Figure 2.2: Outline of an iteration of the basic particle filter with proposal $p(\mathbf{x}_t|\mathbf{x}_{t-1})$. The particles and target distributions are displayed as red circles and black lines, respectively.

by introducing a resampling step that maps the particle system $\{\mathbf{x}_{0:t}^i, \omega_t^i\}_{i=1}^N$ onto an equally weighted particle system $\{\tilde{\mathbf{x}}_{0:t}^i, 1/N\}_{i=1}^N$. The basic idea to duplicate the particles which have large weights and to discard those with small weights. Resampling is carried out whenever the effective sample size (Kong, Liu, and Wong 1994) defined through

$$\mathrm{ESS}(\{\omega_t^i\}_{i=1}^N) = \frac{1}{\sum_{i=1}^N (\omega_t^i)^2},$$

is below some threshold. Different resampling schemes are discussed by Douc, Cappé, and Moulines (2005). The iteration of the basic particle filter with proposal $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ and endowed with resampling is visualized in Figure 2.2. Note that this choice of the proposal implies that the particles $\{\mathbf{x}_t^i, \omega_{t-1}^i\}_{i=1}^N$ approximate the prediction distribution $p(\mathbf{x}_t|\mathbf{y}_{1:t-1})$.

Alternatively to (2.12), smoothing particles which approximate the smoothing (posterior) distribution $p(\mathbf{x}_{1:T}|\mathbf{y}_{1:T})$ and its marginals $p(\mathbf{x}_t|\mathbf{y}_{1:T})$ can be obtained by particle smoothing algorithms. Different methods have been developed by Kitagawa (1996), Hürzeler and Künsch (1998), Doucet, Godsill, and Andrieu (2000), Godsill, Doucet, and West (2004), Briers, Doucet, and Maskell (2010), and others. The major drawback of most existing particle smoothers is their quadratic complexity which make them computationally very expensive.

Although there is vast literature on sequential Monte Carlo methods, there are still issues which have not been tackled sufficiently. Some of these issues which are treated and (at least partly) solved in this dissertation include: there is still a lack of methods for the automatic construction of good proposals (see Section 5.2); methods that allow the use of quasi-Monte Carlo (see Section 5.5); particle smoothers with less than quadratic complexity (see Section 5.3); and techniques for on-line estimation of parameters (see sections 2.5, 5.4, 6.3.3, 7.3.3, and 7.4).

2.5 Expectation-Maximization Algorithm

In practice, the transition and observation distributions of state-space models usually depend on an unknown parameter vector θ which needs to be estimated. A very useful approach is to apply the Expectation-Maximization (EM) algorithm (Dempster, Laird, and Rubin 1977) which is now briefly discussed. Let's assume the observations $\mathbf{y}_t$, $t = 1, \ldots, T$, are given. The EM algorithm computes the maximum likelihood estimator of θ by maximizing the likelihood $p_\theta(\mathbf{y}_{1:T})$ through an iterative application of an E-step and an M-step. In the E-step, the conditional expectation

$$\mathcal{Q}(\theta|\theta^{(m)}) = \mathbf{E}_{\theta^{(m)}}[\log p_\theta(\mathbf{X}_{0:T}, \mathbf{y}_{1:T})|\mathbf{y}_{1:T}]$$

is computed, where $\theta^{(m)}$ is the current parameter estimate. In the M-step, a new parameter estimate $\theta^{(m+1)}$ is obtained by maximizing $\mathcal{Q}(\theta|\theta^{(m)})$. From

$$\begin{aligned}\log \frac{p_{\theta^{(m+1)}}(\mathbf{y}_{1:T})}{p_{\theta^{(m)}}(\mathbf{y}_{1:T})} &= \log \mathbf{E}_{\theta^{(m)}}\left[\frac{p_{\theta^{(m+1)}}(\mathbf{X}_{0:T}, \mathbf{y}_{1:T})}{p_{\theta^{(m)}}(\mathbf{X}_{0:T}, \mathbf{y}_{1:T})}\middle|\mathbf{y}_{1:T}\right] \\ &\geq \mathbf{E}_{\theta^{(m)}}[\log p_{\theta^{(m+1)}}(\mathbf{X}_{0:T}, \mathbf{y}_{1:T})|\mathbf{y}_{1:T}] - \mathbf{E}_{\theta^{(m)}}[\log p_{\theta^{(m)}}(\mathbf{X}_{0:T}, \mathbf{y}_{1:T})|\mathbf{y}_{1:T}] \\ &\geq 0\end{aligned}$$

it follows that the likelihood is never decreased by an iteration of the EM algorithm. Results on the convergence properties of the EM algorithm are discussed by Wu (1983).

As a result of the conditional independence properties of general state-space models $\mathcal{Q}(\theta|\theta^{(m)})$ can be written as

$$\begin{aligned}\mathcal{Q}(\theta|\theta^{(m)}) &= \mathbf{E}_{\theta^{(m)}}[\log p_\theta(\mathbf{X}_0)|\mathbf{y}_{1:T}] + \sum_{t=1}^{T} \mathbf{E}_{\theta^{(m)}}[\log p_\theta(\mathbf{y}_t|\mathbf{X}_t)|\mathbf{y}_{1:T}] \\ &\qquad + \sum_{t=2}^{T} \mathbf{E}_{\theta^{(m)}}[\log p_\theta(\mathbf{X}_t|\mathbf{X}_{t-1})|\mathbf{y}_{1:T}]. \end{aligned} \tag{2.13}$$

In practice, the E-step can usually not be performed analytically because the conditional expectations in (2.13) depend on the unknown smoothing distributions. To obtain an approximation of the conditional expectations particles which approximate the smoothing distributions can be used. The clue is that it is often possible to carry out the M-step analytically. If this is the case, a closed-from expression for the estimator $\theta^{(m+1)}$ is obtained which solely depends on the smoothing particles which are generated with respect to the old parameter estimate $\theta^{(m)}$. An advanced example is given in Section 7.3.3. Note that the EM algorithm is essentially an off-line method because the conditional expectations in (2.13) depend on all observations up to time T. Variants of the standard EM algorithm that can be applied on-line are proposed (see sections 6.3.3, 7.3.3, and 7.4).

2.6 Pseudo- and Quasi-Random Number Generation

The generation of random numbers is of core importance for Monte Carlo simulations. Typically, uniformly distributed random numbers are generated which are subsequently transformed such that they follow the desired distributions. If normal random numbers are required the inversion method can be used. It works as follows. Let's assume u is uniformly distributed on $[0, 1)$. Then, $\Phi^{-1}(u)$ is distributed according to $\mathcal{N}(0, 1)$, where Φ is the cumulative distribution function of $\mathcal{N}(0, 1)$. A problem is that even in the Gaussian case the inverse of the cumulative distribution function is not available in closed form. However, reliable approximations exist such as the Beasley-Springer-Moro approximation (Moro 1995). An alternative method for the transformation of uniform random numbers into normal random numbers is the Box-Muller algorithm (Box and Muller 1958) which does not require the (inverse) cumulative distribution function. An extensive overview of the methods for the generation of random numbers is given in Devroye (1986). A brief overview with the focus on Gaussian variates is provided in Glasserman (2004, Chapter 2). For the simulations presented in this dissertation, the inversion method is used because it is a monotonic transformation. Therefore, it can be straightforwardly combined with quasi-Monte Carlo (see below).

The generation of (uniformly distributed) random numbers is usually done with pseudo-random number generators. A famous pseudo-random number generator is the Mersenne Twister 19937 (Matsumoto and Nishimura 1998). The prefix pseudo indicates that pseudo-random numbers are not truly random but constructed by deterministic algorithms. However, they are random enough in the sense that they pass statistical randomness and distribution tests.

Now the concept of quasi-Monte Carlo is briefly reviewed. Quasi-Monte Carlo is often used to improve Monte Carlo estimators. In contrast to Monte Carlo, quasi-Monte Carlo integration uses so-called low-discrepancy sequences instead of (pseudo-) random numbers. Low-discrepancy numbers are constructed to fill the space more evenly. For a detailed description of the construction and properties of low-discrepancy sequences the reader is referred to Niederreiter (1992) and the references given there. A nice overview is given in Glasserman (2004, Chapter 5). Pseudo-random numbers from the Mersenne Twister 19937 and quasi-random numbers from the two-dimensional Sobol sequence (Sobol 1967) are compared visually in Figure 2.3. It can be observed that, in contrast to the quasi-random numbers, the pseudo-random numbers exhibit cluster-like features.

The incentive to work with quasi-Monte Carlo is justified by its deterministic error bound of order $\mathcal{O}(N^{-1} \log^d N)$, which follows from the well-known Koksma-Hlawka inequality (see Niederreiter (1992)). This bound is merely of theoretical benefit because the computation of the involved constants (including the Hardy-Krause variation of the integrand) is infeasible or at least very difficult. However, it suggests that quasi-Monte Carlo should massively outperform Monte Carlo in low-dimensional integration problems. The advantage of quasi-Monte Carlo diminishes with increasing dimension. Nevertheless, it is well-known in the financial engineering literature, that quasi-Monte Carlo may be effectively applied to high-dimensional problems (Paskov and Traub 1995; Ninomiya and Tezuka 1996; Traub and Werschulz 1998). This stems from the fact that many integration tasks in finance have rather low effective dimension compared with the nominal dimension. In Chapter 4, this is discussed in more detail.

A drawback of quasi-Monte Carlo is the lack of randomness, which impedes the computation of the mean square error for assessing the accuracy of the estimator. This issue can be resolved

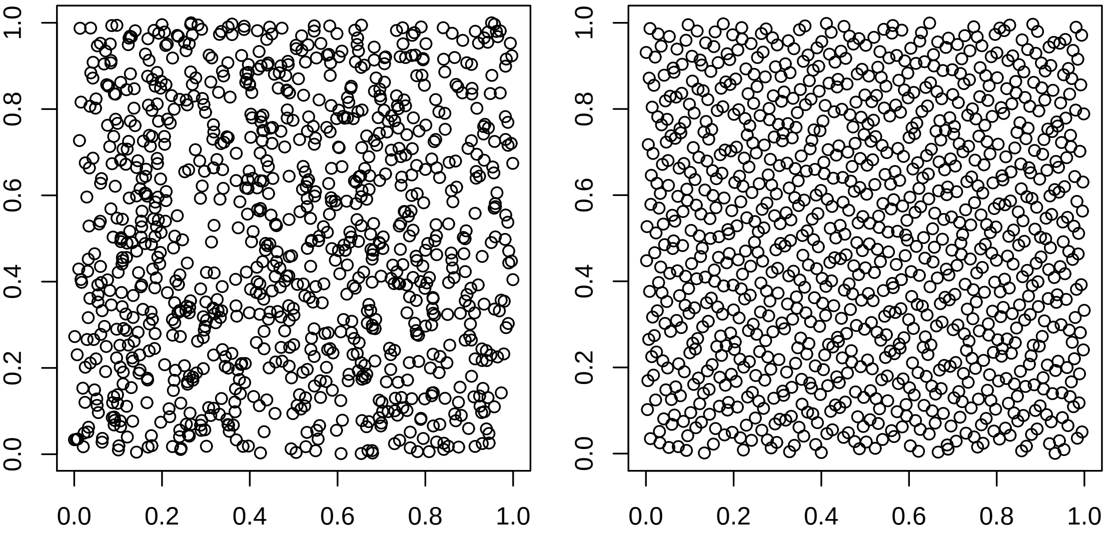

Figure 2.3: Comparison of pseudo-random numbers (left plot) with quasi-random numbers (right plot). 1023 two-dimensional variates from the Mersenne Twister 19937 and from the Sobol sequence are shown.

by randomizing the deterministic low-discrepancy sequence to achieve independent realizations of the quasi-Monte Carlo estimator. Different approaches for randomizing low-discrepancy sequences are available including Owen's scrambling (Owen 1995), random digit scrambling (Matoušek 1998), or random shifts (see Ökten and Eastman (2004) for a survey). Priority is often given to the random shift technique because of its straightforward implementation. It is based on the idea to shift the entire sequence by a random vector $\mathbf{v}$ modulo one. $\mathbf{v}$ is drawn from the uniform distribution on $[0, 1)^d$. That is, a randomized sequence is obtained by substituting the quasi-random vectors $\mathbf{y}^i$ of the original low-discrepancy sequence by $(\mathbf{y}^i + \mathbf{v}) \bmod 1$.

Chapter 3

Nonparametric Importance Sampling

3.1 Introduction

In Section 2.3, the importance sampling method was described along with a discussion on the use of parametric proposals. An alternative to the classical parametric importance sampling approaches is nonparametric importance sampling. It is based on the idea to approximate the optimal proposal (or another suitable proposal) nonparametrically. The advantage of nonparametric methods is that, at least in low dimensions, one can expect to achieve a better approximation of the optimal proposal compared with parametric techniques. In addition, nonparametric importance sampling can be applied in an automatic fashion because it does not require the prior investigation of the structure of the integrand to set up a suitable parametric family of proposals.

Nonparametric approximations based on kernel estimators for the construction of proposals have been used before (West 1992, 1993; Givens and Raftery 1996; Kim, Roh, and Lee 2000). Under restrictive conditions it has been shown that nonparametric (unnormalized) importance sampling can not only reduce the variance of the estimator but may also improve its rate of convergence of the mean square error to $\mathcal{O}(N^{-(d+8)/(d+4)})$ (Zhang 1996). Except for special cases, parametric importance sampling strategies achieve the standard Monte Carlo rate of $\mathcal{O}(N^{-1})$, because the optimal proposal is typically not included in the employed distribution family. There is still a lack of theoretical results for nonparametric importance sampling, particularly for the self-normalized importance sampler. Furthermore, computationally aspects, that critically effect the performance of nonparametric importance sampling, have only been insufficiently treated in the literature (Zlochin and Baram 2002).

The competitiveness of nonparametric importance sampling compared with parametric importance sampling heavily relies on the computational efficiency of the employed nonparametric estimator. In fact, until now nonparametric importance sampling is only of theoretical interest because of the computational shortcomings of the kernel estimator. In this chapter, we propose nonparametric importance sampling algorithms which are based on a multivariate frequency polygon estimator. This nonparametric estimator is shown to be computationally superior to kernel estimators. In addition, it allows the combination of nonparametric importance sampling with other variance reduction techniques (such as stratified sampling) which is another advantage over kernel estimators. We investigate nonparametric importance sampling not only for unnormalized importance sampling but also for self-normalized importance sampling, which has not been done before. Under loose conditions on the integrand, the mean square error convergence properties of the proposed algorithms are explored (sections 3.2 and 3.3). The theoretical findings result in distinct suggestions for efficient application of nonparametric importance sampling. The large potential of nonparametric importance sampling to reduce Monte Carlo variance is verified empirically by means of different integration problems (sections 3.5 and 3.6). Overall, we provide strong evidence that our nonparametric importance sampling algorithms solve well-known problems of existing nonparametric importance sampling techniques. This suggests that nonparametric importance sampling is a promising alternative to parametric importance sampling in practical applications.

3.2 A New Nonparametric Importance Sampling Algorithm

A nonparametric importance sampling algorithm based on a kernel density estimator, that approximates the analytically unavailable optimal proposal $q_{\varphi}^{\mathrm{IS}}$, is considered in Zhang (1996). Theoretical evidence of the usefulness of this approach has been established. In particular, it was proved that nonparametric importance sampling may yield mean square error convergence of order $\mathcal{O}(N^{-(d+8)/(d+4)})$ essentially under the very restrictive assumption that φp has compact support on which φ is strictly positive. The theoretical results derived in this chapter require much weaker assumptions. From a practical point of view a kernel density estimator is computationally too demanding. For the purpose of nonparametric importance sampling it does not suffice that the employed nonparametric estimator provides a fast and accurate approximation of the distribution of interest. It is also required to allow efficient sampling as well as fast evaluation at arbitrary points. As a computationally more efficient alternative to the kernel estimator, it is suggested that one uses a histogram estimator (Zhang 1996). The drawback of a histogram is its slow convergence rate of $\mathcal{O}(N^{-2/(2+d)})$ compared with kernel estimators, which typically achieve $\mathcal{O}(N^{-4/(4+d)})$. Here we propose the usage of a multivariate frequency polygon which is known as linear blend frequency polygon (LBFP) (Terrell 1983). It is constructed by interpolation of histogram bin midpoints. Though computationally only slightly more expensive than ordinary histograms, it achieves the same convergence rate as standard kernel estimators. Consider a multivariate histogram estimator with bin height $\hat{f}^{\mathrm{H}}_{k_1,\ldots,k_d}$ for bin $B_{k_1,\ldots,k_d} = \prod_{i=1}^{d}[t_{k_i} - h/2, t_{k_i} + h/2)$ where h is the bin width and $(t_{k_1},\ldots,t_{k_d})$ the bin mid-point. For $\mathbf{x} \in \prod_{i=1}^{d}[t_{k_i}, t_{k_i}+h)$ the LBFP estimator is defined as

$$\hat{f}(\mathbf{x}) = \sum_{j_1,\ldots,j_d \in \{0,1\}} \left[\prod_{i=1}^{d} \left(\frac{x_i - t_{k_i}}{h} \right)^{j_i} \left(1 - \frac{x_i - t_{k_i}}{h} \right)^{1-j_i} \right] \hat{f}^{\mathrm{H}}_{k_1+j_1,\ldots,k_d+j_d}. \tag{3.1}$$

It can be shown that $\hat{f}$ integrates to one. A one-dimensional (linear blend) frequency polygon and the underlying histogram are shown in Figure 3.1.

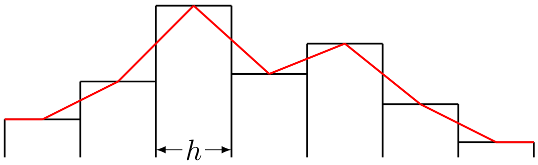

Figure 3.1: A frequency polygon and the underlying histogram with bin width h.

Our nonparametric importance sampling algorithm consists of two steps. In the first step the optimal proposal q_φ^{IS} given in (2.4) is estimated nonparametrically using samples drawn from a trial distribution q_0 and weighted according to the importance ratio $q_\varphi^{\mathrm{IS}}/q_0$. In the second step an ordinary importance sampling is carried out, subject to the proposal estimated in the first step. Before we can state the algorithm we need to introduce the following quantities. Let A_M be an increasing sequence of compact sets defined by $A_M = \{\mathbf{x} \in \mathbb{R}^d : q_\varphi^{\mathrm{IS}}(\mathbf{x}) \geq c_M\}$, where $c_M > 0$ and $c_M \to 0$ as M goes to infinity. For any function g we denote the restriction of g on A_M by g_M and we abbreviate $q_M^{\mathrm{IS}} = q_{\varphi_M}^{\mathrm{IS}}$. Furthermore, the volume of A_M is denoted by V_M. Note that, by definition, A_M converges to the support of q_φ^{IS}. The theorems in this section consider the following algorithm (NIS).

Algorithm 1: Nonparametric Importance Sampling (NIS)

Step 1: Proposal estimation

- **For** $j = 1, \ldots, M$: Sample $\tilde{\mathbf{x}}^j \sim q_0$.
- Obtain estimate $\hat{q}_M^{\mathrm{IS}}(\mathbf{x}) = \frac{\hat{f}_M(\mathbf{x})+\delta_M}{\overline{\omega}_M+V_M\delta_M}\mathbf{1}_{A_M}(\mathbf{x})$,
  where $\overline{\omega}_M = 1/M\sum_{j=1}^M \omega_M^j$, $\omega_M^j = |\varphi_M(\tilde{\mathbf{x}}^j)|p(\tilde{\mathbf{x}}^j)q_0(\tilde{\mathbf{x}}^j)^{-1}$, and

$$\begin{aligned}\hat{f}_M(\mathbf{x}) &= \frac{1}{Mh^d}\sum_{j_1,\ldots,j_d\in\{0,1\}}\left[\prod_{i=1}^d\left(\frac{x_i-t_{k_i}}{h}\right)^{j_i}\left(1-\frac{x_i-t_{k_i}}{h}\right)^{1-j_i}\right]\\ &\qquad\times\sum_{j=1}^M \omega_M^j \mathbf{1}_{\prod_{i=1}^d[t_{k_i+j_i}-h/2,t_{k_i+j_i}+h/2)}(\tilde{\mathbf{x}}^j)\end{aligned}$$

  for $\mathbf{x} \in \prod_{i=1}^d [t_{k_i}, t_{k_i}+h)$.

Step 2: Importance Sampling

- **For** $i = 1, \ldots, N-M$: Generate sample $\mathbf{x}^i$ from proposal $\hat{q}_M^{\mathrm{IS}}$.
- Evaluate $\hat{I}_{\varphi_M}^{\mathrm{NIS}} = (N-M)^{-1}\sum_{i=1}^{N-M}\varphi_M(\mathbf{x}^i)p(\mathbf{x}^i)\hat{q}_M^{\mathrm{IS}}(\mathbf{x}^i)^{-1}$.

The quantities A_M, V_M, and δ_M are required in the proofs of the following theorems, but they can be omitted in practice.

Assumption 1 Both φ and p have three continuous and square integrable derivatives on $\mathrm{supp}(|\varphi|p)$, and $|\varphi|p$ is bounded. Furthermore, it is assumed that $\int(\nabla^2|\varphi|p)^4(|\varphi|p)^{-3} < \infty$ where $\nabla^2|\varphi|p = \partial^2|\varphi|p/\partial x_1^2 + \ldots + \partial^2|\varphi|p/\partial x_d^2$.

Assumption 2 $\mathbf{E}[|\varphi|pq_0^{-1}]^4$ is finite on $\mathrm{supp}(|\varphi|p)$.

Assumption 3 As total sample size $N \to \infty$, bin width h satisfies $h \to 0$ and $Mh^d \to \infty$. Additionally, we have $\delta_M > 0$, $V_M\delta_M = o(h^2)$ and $M^3(V_M\delta_M)^4 \to \infty$.

Assumption 4a c_M satisfies $\frac{h^8+(Mh^d)^{-2}}{\delta_M c_M^3} = o(\frac{h^4+(Mh^d)^{-1}}{c_M})$ and $\frac{h^4+(Mh^d)^{-1}}{c_M} \to 0$.

Assumption 5a c_M satisfies $(\int q_\varphi^{\mathrm{IS}}\mathbf{1}_{\{q_\varphi^{\mathrm{IS}}<c_M\}})^2 = o(M^{-1}h^4 + (M^2h^d)^{-1})$.

For fixed sample size M and conditional on the samples $\{\tilde{\mathbf{x}}^i\}_{i=1}^M$, it is not hard to show that $\hat{I}_{\varphi_M}^{\mathrm{NIS}}$ is an unbiased estimator with variance

$$\mathrm{Var}[\hat{I}_{\varphi_M}^{\mathrm{NIS}}] = \frac{1}{N-M}\int\left(\frac{\varphi_M(\mathbf{x})p(\mathbf{x})}{\hat{q}_M^{\mathrm{IS}}(\mathbf{x})} - I_{\varphi_M}\right)^2 \hat{q}_M^{\mathrm{IS}}(\mathbf{x})d\mathbf{x}. \tag{3.2}$$

For the special case $\varphi \geq 0$ we have $q_M^{\mathrm{IS}} = \varphi_M p I_{\varphi_M}^{-1}$, and (3.2) can be rewritten as

$$\frac{I_{\varphi_M}^2}{N-M}\int\frac{(\hat{q}_M^{\mathrm{IS}}(\mathbf{x}) - q_M^{\mathrm{IS}}(\mathbf{x}))^2}{\hat{q}_M^{\mathrm{IS}}(\mathbf{x})}d\mathbf{x}. \tag{3.3}$$

Under the aforementioned assumptions, we now prove that the variance (3.3) attains convergence rate $\mathcal{O}(N^{-(d+8)/(d+4)})$, if bin width h is chosen optimally.

Theorem 3.1. *Suppose that the assumptions 1 through 3, 4a, 5a hold, $\varphi \geq 0$, and $q = q_\varphi^{IS}$. We obtain*

$$\mathbf{E}[\hat{I}_{\varphi_M}^{NIS} - I_\varphi]^2 \quad = \quad \frac{I_\varphi^2}{N-M}\left\{h^4H_1 + \frac{2^d}{3^dMh^d}H_2\right\} \times (o(1)+1)$$

and the optimal bin width

$$h^* = \left(\frac{dH_2 2^d}{4H_1 3^d}\right)^{\frac{1}{d+4}} M^{-\frac{1}{d+4}},$$

where

$$H_1 = \frac{49}{2880}\sum_{i=1}^d\int\frac{(\partial_i^2 q)^2}{q} + \frac{1}{64}\sum_{i\neq j}\int\frac{\partial_i^2 q\partial_j^2 q}{q},\ H_2 = \int\frac{q}{q_0}.$$

Proof. See Appendix A.1.

A direct implication of Theorem 3.1 is the following corollary.

Corollary 3.2. *Under the assumptions of Theorem 3.1 and the further assumption that $M/N \to \lambda$ $(0 < \lambda < 1)$, and $h = h^*$ we yield*

$$\lim_{N\to\infty} N^{\frac{d+8}{d+4}}\mathbf{E}\left[\hat{I}_{\varphi_M}^{NIS} - I_\varphi\right]^2 = \lambda^{-\frac{4}{d+4}}(1-\lambda)^{-1} \times I_\varphi^2 D$$

and optimal proportion $\lambda^* = 4/(d+8)$ where

$$D = \left\{(d/4)^{4/(d+4)} + (d/4)^{-d/(d+4)}\right\}\left[H_1^d(2^d3^{-d}H_2)^4\right]^{1/(d+4)}.$$

We remark that under much stronger assumptions, corresponding results for nonparametric importance sampling based on kernel estimators were obtained in Zhang (1996).

We now move to a more general case. Assume $\varphi \geq 0$ (and $\varphi \leq 0$) does not hold. For this case we show that the NIS algorithm achieves the minimum importance sampling variance asymptotically. By substituting the optimal importance sampling distribution q_φ^{IS} into variance σ_{IS}^2 and writing shorthand $\overline{I}_\varphi = \int |\varphi(\mathbf{x})| p(\mathbf{x}) d\mathbf{x}$, we see the optimal variance of the importance sampling estimator to be $\overline{I}_\varphi^2 - I_\varphi^2$.

Assumption 4b c_M guaranties $\frac{h^8+(Mh^d)^{-2}}{\delta_M c_M^5} = o(\frac{h^4+(Mh^d)^{-1}}{c_M^3})$ and $\frac{h^4+(Mh^d)^{-1}}{c_M^3} \to 0$.

Assumption 5b c_M guaranties $(\int q_\varphi^{\mathrm{IS}} \mathbf{1}_{\{q_\varphi^{\mathrm{IS}} < c_M\}})^2 = o(M^{-1}h^2 + (M^2h^d)^{-1})$.

Theorem 3.3. *Suppose that the assumptions 1 through 3, 4b, 5b hold, φ does not have a definite sign, and $q = q_\varphi^{IS}$. Then we obtain*

$$\mathbf{E}[\hat{I}_{\varphi_M}^{NIS} - I_\varphi]^2 \quad = \quad \frac{1}{N-M}\left[(\overline{I}_\varphi^2 - I_\varphi^2) \ + \ I_\varphi^2 \left\{ h^2 \overline{H}_1 + \frac{2^d}{3^d M h^d} \overline{H}_2 \right\} \times (1 + o(1))\right]$$

and the optimal bin width

$$h^{**} = \left(\frac{d\overline{H}_2 2^{d-1}}{\overline{H}_1 3^d} \right)^{\frac{1}{d+2}} M^{-\frac{1}{d+2}},$$

where $\overline{H}_1 = -\left(\int f_\varphi^2 \frac{\nabla^2 q}{8q^2} + \int f_\varphi \frac{\nabla^2 q}{4q} \right)$, $\overline{H}_2 = \left(\int \frac{q}{q_0} - 2\int \frac{f_\varphi}{q_0} - \int \frac{f_\varphi^2}{q_0 q} \right)$, *and* $f_\varphi = \left(\frac{\varphi p}{I_\varphi} - \frac{|\varphi| p}{\overline{I}_\varphi} \right)$.

Proof. See Appendix A.4.

As a consequence of Theorem 3.3, the NIS algorithm does not lead to a mean square error rate improvement for functions φ, which take positive and negative values. But if the optimal bin width h^{**} is used, we have

$$\mathbf{E}[\hat{I}_{\varphi_M}^{\mathrm{NIS}} - I_\varphi]^2 = \frac{\overline{I}_\varphi^2 - I_\varphi^2}{N-M} + o(N^{-1}).$$

That is, the optimal importance sampling variance is achieved asymptotically. Unlike Theorem 3.1, the optimal proportion λ cannot be computed analytically as a result of its dependency on N. But theoretically, it can be computed as $\lambda^{**} = \mathrm{argmin}_\lambda G(N, h^{**}, \lambda)$ where $G = \mathbf{E}[\hat{I}_{\varphi_M}^{\mathrm{NIS}} - I_\varphi]^2$. Clearly, λ^{**} decreases in N. Note, that for the optimal asymptotic variance to be achieved, it suffices that $0 < \lambda < 1$.

Corollary 3.2 and Theorem 3.3 suggest that importance sampling-based Monte Carlo integration can be much more efficient for functions $\varphi \geq 0$ (and $\varphi \leq 0$) than for arbitrary functions. This stems from the fact that for non-negative (non-positive) functions, the usage of the optimal proposal leads to a zero variance estimator. By approximating the optimal proposal with a consistent estimator it is therefore not surprising that the standard Monte Carlo rate can be surmounted. Consequently, it should be reasonable to decompose φ into positive and negative part, $\varphi = \varphi^+ - \varphi^-$, and to apply Algorithm 1 to φ^+ and φ^- separately. Since then, we can expect to achieve the superior rate $\mathcal{O}(N^{-(d+8)/(d+4)})$. Note that the partitioning of φ needs not to be done analytically. It may be carried out implicitly in Step 1 of the algorithm. This approach, denoted by NIS+/-, is investigated in a simulation study in Section 3.5.

3.3 A New Nonparametric Self-Normalized Importance Sampling Algorithm

Many problems in Bayesian inference can be written as the expectation of some function of interest, φ, with respect to the posterior distribution p, which is only known up to some constant. This leads to the evaluation of integrals

$$\mathbf{E}_p[\varphi] = \frac{\int \varphi(\mathbf{x})\tilde{p}(\mathbf{x})d\mathbf{x}}{\int \tilde{p}(\mathbf{x})d\mathbf{x}},$$

where $\tilde{p} = \alpha p$ with unknown constant α. Self-normalized importance sampling is a standard approach for solving such problems. It is often suggested to choose the proposal close to the posterior. But from the central limit theorem we know that one can do better by choosing it close to the optimal proposal, which is proportional to $|\varphi - I_\varphi|p$. Next, we introduce a nonparametric self-normalized importance sampling algorithm (NSIS).

Analogous to the definition of A_M we define $\widetilde{A}_M = \{\mathbf{x} \in \mathbb{R}^d : q_\varphi^{\text{SIS}}(\mathbf{x}) \geq \tilde{c}_M\}$, where $\tilde{c}_M > 0$ and $\tilde{c}_M \to 0$ as M goes to infinity. Its volume is denoted by $\widetilde{V}_M$. The optimal proposal q_φ^{SIS} is defined in (2.6).

Algorithm 2: Nonparametric Self-Normalized Importance Sampling (NSIS)

Step 1: Proposal estimation

- **For** $j = 1, \dots, M$: Sample $\tilde{\mathbf{x}}^j \sim q_0$.
- Obtain estimate $\hat{q}_M^{\text{SIS}}(\mathbf{x}) = \frac{\hat{f}_M(\mathbf{x})+\delta_M}{\overline{\omega}_M+\widetilde{V}_M\delta_M}\mathbf{1}_{\widetilde{A}_M}(\mathbf{x})$, where $\overline{\omega}_M = 1/M\sum_{j=1}^M \widetilde{\omega}_M^j$, $\widetilde{\omega}_M^j = |\varphi_M(\tilde{\mathbf{x}}^j) - \breve{I}_{\varphi_M}|\tilde{p}(\tilde{\mathbf{x}}^j)q_0(\tilde{\mathbf{x}}^j)^{-1}$, $\hat{f}_M(\mathbf{x})$ analogous to Algorithm 1, and
$$\breve{I}_{\varphi_M} = \frac{\sum_{j=1}^M \varphi_M(\tilde{\mathbf{x}}^j)\tilde{p}(\tilde{\mathbf{x}}^j)q_0(\tilde{\mathbf{x}}^j)^{-1}}{\sum_{j=1}^M \tilde{p}(\tilde{\mathbf{x}}^j)q_0(\tilde{\mathbf{x}}^j)^{-1}}.$$

Step 2: Self-Normalized Importance Sampling

- **For** $i = 1, \dots, N-M$: Generate sample $\mathbf{x}^i$ from proposal $\hat{q}_M^{\text{SIS}}$.
- Evaluate
$$\hat{I}_{\varphi_M}^{\text{NSIS}} = \frac{\sum_{i=1}^{N-M} \varphi_M(\mathbf{x}^i)\widetilde{w}_M(\mathbf{x}^i)}{\sum_{i=1}^{N-M} \widetilde{w}_M(\mathbf{x}^i)},$$
where $\widetilde{w}_M(\mathbf{x}^i) = \tilde{p}(\mathbf{x}^i)\hat{q}_M^{\text{SIS}}(\mathbf{x}^i)^{-1}$.

Both the self-normalized importance sampling estimator (2.3) and the NSIS algorithm produce biased estimates. However, the estimates are asymptotically unbiased. Under the assumptions 1 through 3 (with p, $|\varphi|$, c_M, V_M replaced by $\tilde{p}$, $|\varphi - I_\varphi|$, $\tilde{c}_M$, $\widetilde{V}_M$) it is easy to verify that, conditional on the samples $\{\tilde{\mathbf{x}}^i\}_{i=1}^M$, the central limit theorem of Geweke (1989) holds for $\hat{I}_{\varphi_M}^{\text{NSIS}}$ (2.5). The asymptotic variance of the central limit theorem can be written as

$$\sigma_{\text{SIS}}^2 = \tilde{I}_{\varphi_M}^2 \left[1 + \int \frac{(q_M^{\text{SIS}}(\mathbf{x}) - \hat{q}_M^{\text{SIS}}(\mathbf{x}))^2}{\hat{q}_M^{\text{SIS}}(\mathbf{x})} d\mathbf{x}\right] \tag{3.4}$$

with $\tilde{I}_{\varphi_M} = \int |\varphi_M(\mathbf{x}) - I_{\varphi_M}| p(\mathbf{x}) d\mathbf{x}$ being the median of φ. Consequently, $\tilde{I}_{\varphi_M}^2$ is the (asymptotically) optimal variance that can be achieved by self-normalized importance sampling. Unless φ is constant, it is impossible to build up a zero variance estimator based on self-normalized importance sampling. This renders it unnecessary to investigate separately the mean square error convergence of NSIS for non-negative and arbitrary functions.

The structure of σ_{SIS}^2 is very similar to the structure of the variance in (3.3) but the weights $\widetilde{\omega}_M^j$ introduce inter-sample dependencies which make the reasoning in the proofs of Theorem 3.1 and Theorem 3.3 not directly applicable. However, similarly to Theorem 3.3, we can show that NSIS attains optimal variance asymptotically for certain bin width h and proportion $0 < \lambda < 1$.

Theorem 3.4. *Suppose that the assumptions 1 through 3, 4a, 5a (with p, $|\varphi|$, c_M, V_M replaced by $\tilde{p}$, $|\varphi - I_\varphi|$, $\tilde{c}_M$, $\widetilde{V}_M$) hold, and $q = q_\varphi^{SIS}$. Then we obtain*

$$\mathbf{E}[\hat{I}_{\varphi_M}^{NSIS} - I_\varphi]^2 \quad = \quad \frac{\tilde{I}_\varphi^2}{N - M}\left[1 \ + \ h^4 H_1 + \frac{2^d}{3^d M h^d} H_2\right] \times (1 + o(1))$$

and the optimal bin width

$$\tilde{h}^* = \left(\frac{d H_2 2^d}{4 H_1 3^d}\right)^{\frac{1}{d+4}} M^{-\frac{1}{d+4}},$$

where H_1 and H_2 are defined as in Theorem 3.1 (with q_φ^{IS} replaced by q_φ^{SIS}).

Proof. See Appendix A.5.

First, note that analogous to Theorem 3.3, there is no analytic solution for the optimal λ. Second, the theorem implies that with NSIS, the mean square error rate cannot be improved. Therefore, NSIS is (at least asymptotically) less efficient than NIS+/-. There is, consequently, no reason to apply NSIS in cases where NIS+/- is applicable. However, this does not impair the usefulness of NSIS in cases when normalization is required as a result of unknown constants.

3.4 Applying Nonparametric Importance Sampling

In this section we discuss what is required for implementing NIS/NSIS. First, one needs to take care of the selection of q_0, h, and λ. Second, an implementation of the LBFP estimator, which allows the generation of samples, is required. Given these ingredients, the implementation of Algorithm 1 and 2 is straightforward.

3.4.1 Parameter Selection

q_0 From a practical point of view, trial distribution q_0 should be chosen such that its support is close to the support of $|\varphi|p$ or $|\varphi - I_\varphi|p$, respectively, and such that it has heavier tails than the corresponding optimal proposal. However, it is not required that q_0 emulates any structure of the optimal proposal. Obviously, the choice should also comply with Assumption 2. Note that the expectations in the assumptions may not exist if q_0 is too close to the optimal proposals. In addition, it is important to choose an easy-to-sample density. For low-dimensional problems, even a uniform distribution may suffice.

h As the optimal bin width incorporates unknown quantities dependent on the optimal proposal, it typically cannot be computed analytically. The unknown quantities can be estimated using the plug-in method based on the samples of Step 1 of the algorithms, as suggested in Zhang (1996). If the second derivative of the optimal proposal is unknown, the plug-in method cannot be applied. In this case, a Gaussian reference rule is an alternative.

λ Except for the case investigated in Theorem 3.1 and Corollary 3.2, where the optimal proportion λ^* is given by a beautifully easy expression only depending on the problem dimension, it is not clear how to choose λ. However, from the mean square error expressions in the theorems, we know that λ^* (from Corollary 3.2) serves as an upper bound. Empirical evidence suggests that λ should never exceed .25.

A_M, δ_M In practical applications, the restriction of the estimator on a compact set A_M can be omitted because the induced bias can be made arbitrarily small and particularly smaller than the desired precision of the integral value. Hence, the sequence c_M does not need to be defined. Sequence δ_M can also be skipped in practice as mentioned earlier.

3.4.2 Implementing the LBFP Estimator

The implementation of the LBFP estimator $\hat{f}$ should take into account efficient sampling and evaluation. Given the multivariate histogram defined through bin heights $\hat{f}^{\mathrm{H}}_{k_1,\ldots,k_d}$, the implementation of the evaluation of $\hat{f}$ is simple (see (3.1)). We emphasize that for storing $\hat{f}$ on a computer, it suffices to store the underlying histogram. Sampling from a LBFP is more involved than evaluation, and to the author's knowledge this has not been discussed in the literature until now. We propose to apply the inversion method. The crucial fact is that a LBFP can be written as a product of (conditional) univariate frequency polygons

$$\hat{f}(\mathbf{x}) = \hat{f}^{\mathrm{FP}}(x_1) \prod_{i=2}^{d} \hat{f}^{\mathrm{FP}}(x_i | x_{1:i-1})$$

with $\{x_{1:i-1}\} = \{x_1, \ldots, x_{i-1}\}$. This representation suggests to produce draws from $\hat{f}$ by sampling iteratively from the univariate frequency polygons $\hat{f}^{\mathrm{FP}}$ using their inverse cumulative distribution functions. A frequency polygon is a convenient object because it is just a linear interpolated univariate histogram (see Figure 3.1). Furthermore, we have

$$\hat{f}^{\text{FP}}(x_i|x_{1:i-1}) = \frac{\hat{f}(x_{1:i})}{\hat{f}(x_{1:i-1})} \tag{3.5}$$

where $\hat{f}(x_{1:i})$ is a (marginalized) LBFP, $i = 1, \ldots, d$. We will see later that the $\hat{f}^{\text{FP}}(x_i|x_{1:i-1})$ are not required directly but the cumulative distribution functions $\hat{F}(x_i|x_{1:i-1})$. Because frequency polygons are piecewise linear functions and as a result of relation (3.5), the latter are obtained without difficulty provided that LBFPs $\hat{f}(x_{1:i})$ can be evaluated. Hence, it is required to calculate the marginalized histograms underlying the LBFPs $\hat{f}(x_{1:i})$. These are specified through bins $B_{k_1,\ldots,k_i}$ and bin heights $\hat{f}^{\text{H}}_{k_1,\ldots,k_i}$.

Let $\mathbf{y} = \{y_1, \ldots, y_d\} \in [0,1)^d$ and $y_i \in [\hat{F}(t_{k_i}|x_{1:i-1}), \hat{F}(t_{k_i+1}|x_{1:i-1}))$. We now describe how the inverse cumulative distribution functions $\hat{F}^{-1}(\cdot|x_{1:i-1})$ of $\hat{f}^{\text{FP}}(x_i|x_{1:i-1})$ can be evaluated at y_i by making use of $\hat{F}(x_i|x_{1:i-1})$. It is easy to see that, for $x_i \in [t_{k_i}, t_{k_i+1})$, $\hat{f}^{\text{FP}}(x_i|x_{1:i-1})$ is a linear function with intercept α and slope β, where

$$\alpha = \frac{\hat{f}(x_{1:i-1}, t_{k_i})}{\hat{f}(x_{1:i-1})} \qquad \text{and} \qquad \beta = \frac{1}{h}\left[\frac{\hat{f}(x_{1:i-1}, t_{k_i+1})}{\hat{f}(x_{1:i-1})} - \alpha\right].$$

Hence, $\hat{F}^{-1}(y_i|x_{1:i-1})$ is the solution of the quadratic equation

$$y_i - \hat{F}(t_{k_i}|x_{1:i-1}) = \int_{t_{k_i}}^{z} \hat{f}^{\text{FP}}(x_i|x_{1:i-1})dx_i = \alpha z + \frac{\beta}{2}z^2,$$

which is given by

$$\hat{F}^{-1}(y_i|x_{1:i-1}) = \begin{cases} -\frac{\alpha}{\beta} + \operatorname{sgn}(\beta)\sqrt{\frac{\alpha^2}{\beta^2} - 2\frac{\gamma_1 - y_i}{\beta}} & \text{for} \quad \beta \neq 0, \\ \left[(\gamma_2 - y_i)t_{k_i} + (y_i - \gamma_1)t_{k_i+1}\right]/(\gamma_2 - \gamma_1) & \text{for} \quad \beta = 0, \end{cases} \tag{3.6}$$

where $\gamma_1 = \hat{F}(t_{k_i}|x_{1:i-1})$ and $\gamma_2 = \hat{F}(t_{k_i+1}|x_{1:i-1})$.

Summarizing, a sample $\mathbf{x}^j$ from the LBFP $\hat{f}$ is obtained through the following iteration. Let $\mathbf{y}^j$ be a sample from the uniform distribution on $[0,1)^d$. Then, for $i = 1, \ldots, d$:

1. Compute the marginalized histogram associated with LBFP $\hat{f}(x_{1:i})$.

2. Calculate the cumulative distribution function $\hat{F}(x_i|x^j_{1:i-1})$ (or $\hat{F}(x_1)$ for $i = 1$) at the (marginal) bin mid points t_{k_i} using (3.5).

3. Evaluate $x^j_i = \hat{F}^{-1}(y^j_i|x^j_{1:i-1})$ (or $x^j_1 = \hat{F}^{-1}(y^j_1)$ for $i = 1$) using (3.6).

We remark that for generating N samples, Step 1 needs only to be carried out once because it is independent of the particular sample $\mathbf{x}^j$. A C++ implementation of the LBFP estimator and an R-package are available (see Section 8.2).

3.4.3 Computational Remarks

Now the computational complexity of the LBFP is discussed. For $h = h^*$, it can be shown that the complexity for generating N samples from a LBFP is of order $\mathcal{O}(2^d d^2 N^{(d+5)/(d+4)})$ (see Appendix A.6 for details). The complexity of evaluation is of lower order. Compared with crude Monte Carlo, which has $\mathcal{O}(dN)$, sampling from a LBFP is only slightly more expensive for small d. For kernel estimators, sampling and evaluation is of order $\mathcal{O}(dN^2)$ (Zlochin and Baram 2002), proving that the LBFP is computationally more efficient for all relevant d and N. Note, more efficient sampling from kernel estimates is possible using regularization with whitening (see for instance Musso, Oudjane, and Le Gland 2001). However, this can induce severe bias if the target distribution is non-Gaussian.

Besides asymptotic complexity properties there are other computational aspects which are of relevance in practice. With computer systems, the evaluation of functions such as `exp` and `pow` is known to be much more expensive than standard arithmetic operations. Contrary to most parametric importance sampling approaches, nonparametric importance sampling methods do not require calls to those functions.

3.5 Simulations

We consider three toy examples to test our nonparametric procedures against (parametric) alternatives. The first two examples are designed to evaluate certain properties of the NIS algorithm and to demonstrate the degraded performance of the NSIS algorithm. The third example is a two-dimensional benchmark problem for self-normalized importance sampling.

A reasonable measure for the effect of a variance reduction technique is the relative efficiency. It is defined as the ratio of the crude Monte Carlo mean square error to the mean square error of the method of interest. In the case when both estimators are unbiased, the relative efficiency is also known as variance reduction factor. The performance of the different algorithms will be measured by relative efficiency and computation time. In all examples, the simulation is done for sample sizes N = 1,000, N = 5,000, and N = 10,000. All computation were carried out on a Dell Precision PWS390, Intel CPU 2.66GHz, and the algorithms are coded in C++. For the details of the software see Chapter 8. For pseudo-random number generation we used the Mersenne Twister 19937 (Matsumoto and Nishimura 1998). All computation times are reported in milliseconds.

Example 1.

As our first example, consider a simple integrand that is to be integrated with respect to the standard normal distribution of dimension d. The integrand is defined by $\varphi(\mathbf{x}) = x_1 \mathbf{1}_{[-1,1]^d}(\mathbf{x})$. It takes positive and negative values on the d-dimensional unit cube. This allows the evaluation of the strategy to apply Algorithm 1 separately to the positive and negative part of the integrand (NIS+/-). In our simulation, d varies from 1 to 8. The trial distribution q_0 is set to the uniform distribution on $[-1, 1]^d$, and the bin width h is chosen with the plug-in method. λ is set to

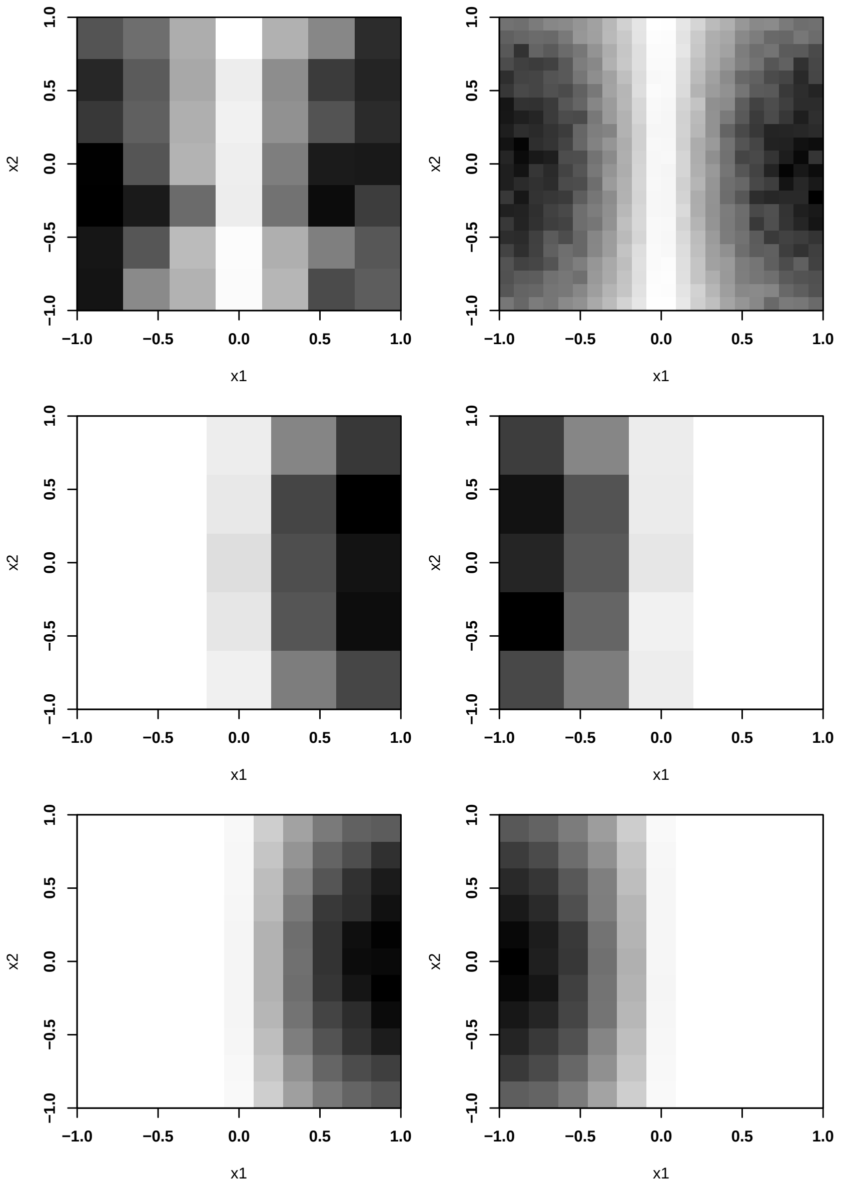

Figure 3.2: Histograms underlying the LBFP estimates of the proposal densities for $d = 2$ (Example 1). The plots correspond to NIS with $N = 10{,}000$ and $N = 1$ Mio. (upper plots), NIS+/- with $N = 10{,}000$ (middle plots), and NIS+/- with $N = 1$ Mio. (lower plots).

0.15 and to the optimal value $4/(d + 8)$ for NIS and NIS+/-, respectively. In order to obtain comparable results, for NIS+/- the total sample size is equally spread to the integration of the positive and negative parts. In Figure 3.2, the estimated proposal densities for $d = 2$ are plotted.

Table 3.1 shows the relative efficiency (RE) and computation times for crude Monte Carlo (MC), NIS, NIS+/-, and ordinary importance sampling (IS) (subject to the uniform distribution on $[-1, 1]^d$). The relative efficiency values for NIS+/- report large variance reduction, which is

		N = 1,000		N = 5,000		N = 10,000	
Method	d	RE	Time (millisec.)	RE	Time (millisec.)	RE	Time (millisec.)
MC	1	1.0	1	1.0	9	1.0	16
IS	1	1.5	3	1.8	16	1.6	31
NIS	1	1.6	13	1.8	28	1.7	50
NIS+/-	1	25.0	13	57.3	24	51.3	40
MC	4	1.0	4	1.0	20	1.0	45
IS	4	5.0	7	5.2	38	4.2	80
NIS	4	3.1	112	4.0	234	3.8	408
NIS+/-	4	9.1	105	26.0	195	22.0	326
MC	8	1.0	13	1.0	60	1.0	121
IS	8	18.6	20	23.0	104	26.3	209
NIS	8	7.8	600	17.4	1,460	5.4	4,020
NIS+/-	8	7.5	572	30.2	1,290	37.4	2,170

Table 3.1: Simulation results for Example 1. All figures are computed/averaged over 100 independent runs.

present at least up to dimension $d = 8$. Even if we take computation time into account, we find significant efficiency improvement: For instance, for $d = 4$ and $N = 10{,}000$ we obtain a relative efficiency value of 22 whereas the computation time surplus factor is about 7. Also, note that importance sampling becomes more favorable as d increases. To investigate the computationally efficiency, we plotted mean square error × computation time (Figure 3.3). Contrary to relative efficiency, smaller values are favourable. We can observe that the critical dimension, up to which NIS+/- is computationally more efficient than the other methods, strongly depends on the magnitude of N. Although for $N = 1{,}000$ one would prefer NIS+/- to importance sampling only for $d = 1$, for $N = 10{,}000$ one would do so up to $d = 4$. Finally, the convergence of the NIS variance towards the optimal importance sampling variance is examined. The minimum importance sampling variance $\overline{I}_{\varphi}^{2} - I_{\varphi}^{2}$ is approximately 0.098 and 0.0099 for $d = 1, 4$, respectively. In Figure 3.4, the estimated variances of NIS$\times(1-\lambda)N$ are plotted for $100 \leq N \leq 2{,}500$. The plots indicate rapid convergence to the optimal values. For comparison, the variance of crude Monte Carlo $\times N$ is roughly 0.198 (for $d = 1$) and 0.063 (for $d = 4$).

Example 2.

This example is concerned with the pricing of a call option within the Black-Scholes model. Given interest rate r and volatility σ, the evolution of a stock is described by the stochastic differential equation

$$dS(t)/S(t) = rdt + \sigma dW(t)$$

with standard Brownian motion W. The solution of the stochastic differential equation is given by

$$S(T) = S(0)\exp[(r - 0.5\sigma^2)T + \sigma\sqrt{T}Z],$$

where Z is a standard normal random variable. At time T, the call option pays the amount $(S(T) - K)^+$, depending on the strike level K. The price of the option at time 0 is given by the

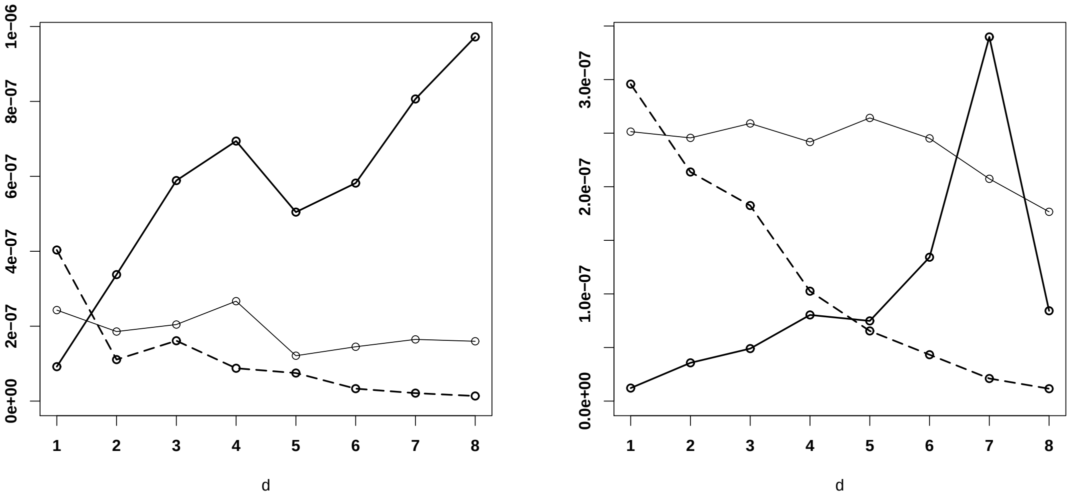

Figure 3.3: Computational efficiency (measured by mean square error × computation time) of crude Monte Carlo (thin solid line), importance sampling (dashed line), and NIS+/- (thick solid line) for N = 1,000 (left), and N = 10,000 (right) for Example 1. All figures are computed/averaged over 100 independent runs.

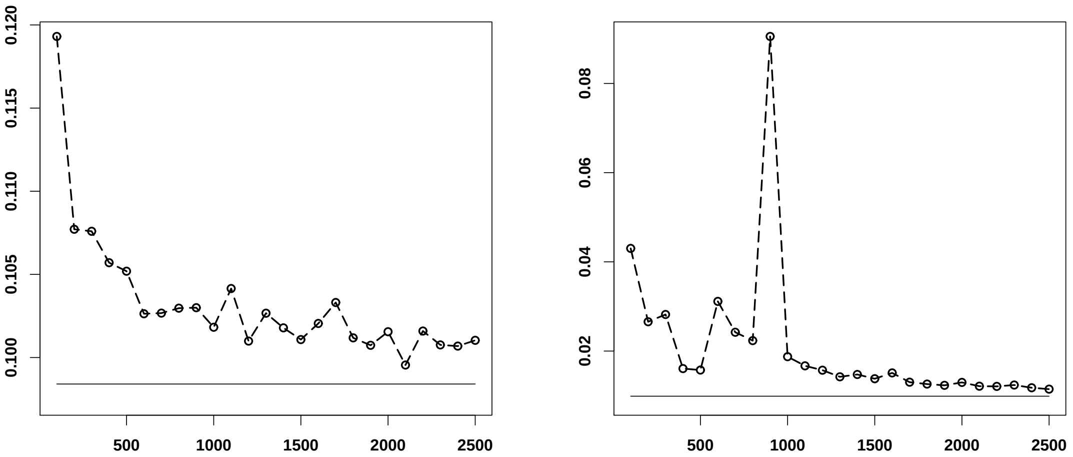

Figure 3.4: Convergence of NIS variance towards optimal importance sampling variance for $d = 1$ (left) and $d = 4$ (right) for Example 1. All figures are computed over 10,000 independent runs.

expectation $\mathbf{E}[F(Z)]$ of the discounted payoff $F(Z) = \exp(-rT)(S(T)-K)^+$. That is, the pricing problem reduces to the integration of a payoff function with respect to the standard normal distribution. Parametric importance sampling is a standard variance reduction technique for option pricing. A shifted standard normal distribution is often used as proposal. This approach is known as the change of drift technique. In our simple model, the (asymptotically) optimal drift is given by $\operatorname{argmax}_z \log F(z) - 0.5z^2$ (Glasserman, Heidelberger, and Shahabuddin 1999).

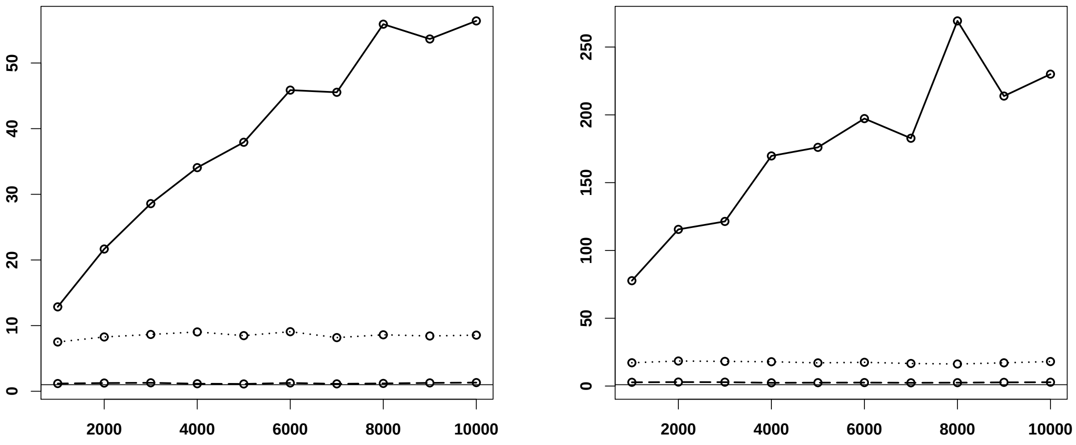

Figure 3.5: Relative efficiency of CDIS (dotted line), NIS (thick solid line), NSIS (dashed line), and crude Monte Carlo (thin solid line) for Example 2 (strike K_1 (left), strike K_2 (right)) and 1,000 $\leq N \leq$ 10,000. All figures are computed over 1,000 independent runs.

We state the simulation results for the optimal change-of-drift importance sampling (CDIS) as parametric benchmark.

For our simulation, we set $S(0) = 100$, $r = 0.1$, $\sigma = 0.2$, $T = 1$. The option price is estimated for the strikes $K_1 = 90$ and $K_2 = 130$. For K_1 the option is said to be in the money $(K_1 < S(0))$ where for K_2 it is called out-of-the money $(K_2 > S(0))$. The latter case is particularly suited for importance sampling techniques, as crude Monte Carlo fails to sample satisfactorily into the domain that affects the option price. q_0 is set to the uniform distribution on $[-5, 5]$ and bin width h is selected using the plug-in method. λ is set to the optimal value 4/9 for NIS and to 0.05 for NSIS.

The efficiency improvements of the importance sampling methods relative to crude Monte Carlo integration are shown in Figure 3.5. Whereas parametric importance sampling methods and NSIS yield constant reduction factors, NIS realizes increasing relative efficiency which coincides with its theoretical superior convergence rate. Particularly for the out-of-the money scenario, NIS achieves massive variance reduction. Establishing only slight variance reduction, NSIS is worst. This confirms our recommendation to avoid NSIS where NIS is applicable. Figure 3.6 shows the proposals used in the simulation for strike K_2. The optimal importance sampling proposal is single-moded and can be reasonably approximated by some Gaussian distribution. This explains the satisfying performance of importance sampling methods based on Gaussian proposals reported in the literature. However, NIS significantly outperforms CDIS. For more complex payoffs implying multimodal optimal proposals, the advantage of NIS should be even more pronounced (compare Chapter 4). Computation times for different sample sizes are reported in Table 3.2. First, notice that CDIS is much more expensive than crude Monte Carlo

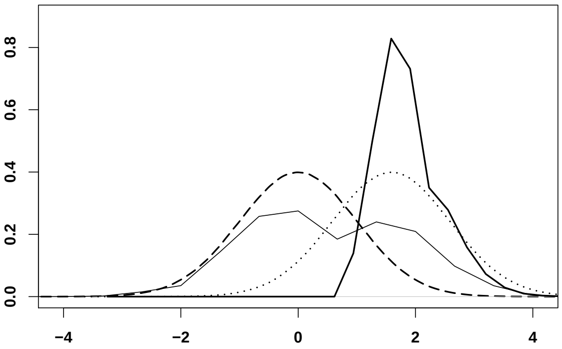

Figure 3.6: Standard normal distribution (dashed line), optimally shifted normal distribution (dotted line), linear blend frequency polygon estimates (N = 5,000) of the optimal proposals q_φ^{SIS} (thin solid line), and q_φ^{IS} (thick solid line) for Example 2.

	N = 1,000	N = 5,000	N = 10,000
Method	Time (ms)	Time (ms)	Time (ms)
MC	1.8	9.0	17.8
CDIS	6.0	27.8	54.5
NIS	13.7	29.2	48.9
NSIS	14.1	31.1	52.1

Table 3.2: CPU times for the option pricing example (Example 2) averaged over 1,000 independent runs.

as a result of the massive evaluation of the `exp` function while computing the likelihood ratios. Second, the computational burden of NIS increases sublinearly for our sample sizes. This is a result of the initial computation for the LBFP, which is roughly independent of N. Remarkably, NIS is computationally cheaper than CDIS for N = 10,000.

Example 3.

The last example is a two-dimensional benchmark integration problem discussed by Givens and Raftery (1996). The density of interest $p(x_1, x_2)$ is given by

$$X_1 \sim \mathcal{U}[-1, 4]$$

and

$$X_2|X_1 \sim \mathcal{N}(|X_1|, 0.09a^2).$$

Let's investigate the cases $a = 0.75$ and $a = 3.5$. This kind of density also occurs in work on whale modeling (Raftery, Givens, and Zeh 1995). Small values for a imply a strong nonlinear dependency between X_1 and X_2. As a becomes larger, the dependency vanishes in favor of a more diffuse relationship (see Figure 3.7). Following Givens and Raftery (1996), we use this scenario for comparing self-normalized importance sampling algorithms.

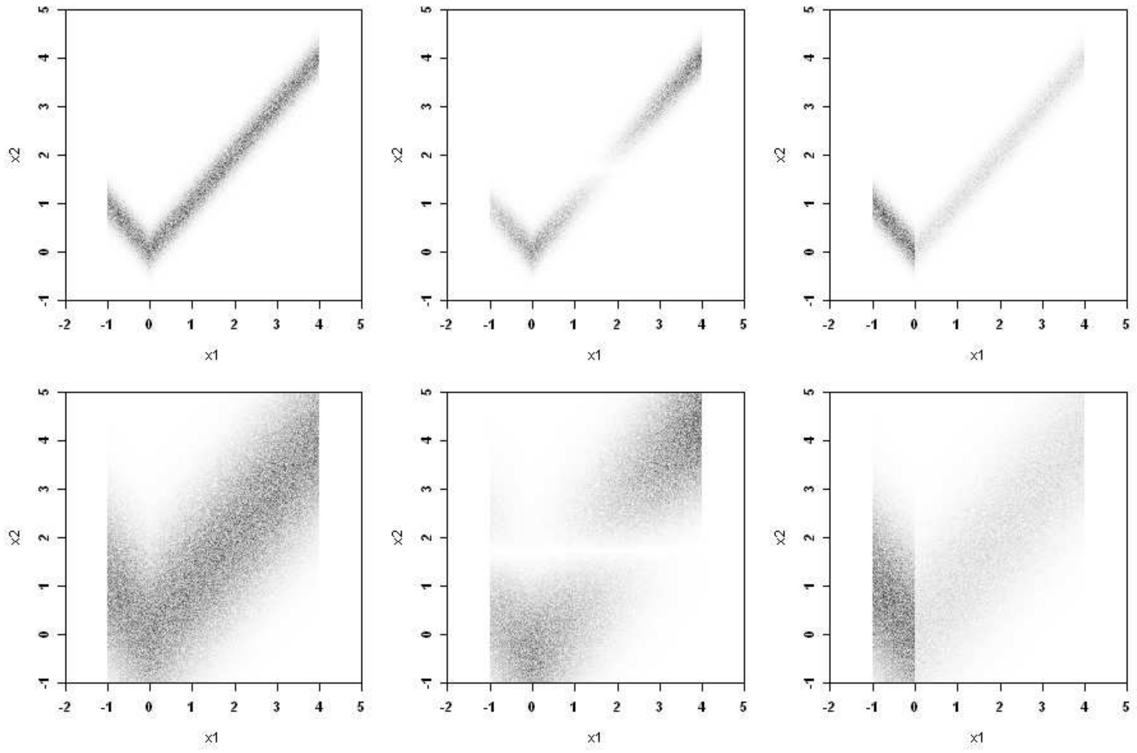

Figure 3.7: Example 3: The upper plots are for the case $a = 0.75$ and the lower plots for $a = 3.5$. From left to right we have density $p(x_1, x_2)$ and the optimal proposals $q_{\varphi_1}^{\mathrm{SIS}}$ and $q_{\varphi_2}^{\mathrm{SIS}}$.

NSIS is tested against self-normalized importance sampling with a proposal equal to the uniform distribution on $[-4, 7] \times [-4, 8]$. The same uniform distribution is used as trial distribution q_0 in the NSIS algorithm. We compute the expectation of functions $\varphi_1(x_1, x_2) = x_2$ and $\varphi_2(x_1, x_2) = \mathbf{1}_{\{x_1<0\}}(x_1, x_2)$. The parameters of NSIS are set as follows: $\lambda = 0.2$ and $h = 1.54$, 1.224, 1.09 (for $N =$ 1,250, 5,000, 10,000, respectively). For comparison, we also state the results of two other nonparametric algorithms, namely GAIS and LAIS (West 1992; Givens and Raftery 1996). GAIS and LAIS are adaptive nonparametric importance sampling methods, that estimate distribution p with adaptive envelope refinements based on nonparametric kernel estimators. Density p and the optimal self-normalized importance sampling proposals are shown in Figure 3.7. They are rather far away from the initial guess q_0. Table 3.3 shows the relative efficiency of NSIS, GAIS, and LAIS with respect to self-normalized importance sampling for the two functions and the two different values of a. The figures for GAIS and LAIS were reprocessed from Givens and Raftery (1996). For $N =$ 5,000, NSIS is clearly the method of choice.

3.6 Application: Spam Filter

We investigate spam filter queueing systems with real data. Queueing system are an active field of research (see, for instance, Lazowska 1984; Asmussen 2003). Numerous applications are readily available. The most basic queueing system, denoted briefly by M|M|1, consists of a single

		φ_1		φ_2	
Method	N	$a = 0.75$	$a = 3.5$	$a = 0.75$	$a = 3.5$
NSIS	1,250	1.59	2.89	0.58	3.82
GAIS	1,250	0.02	3.45	0.30	1.11
LAIS	1,250	0.75	0.99	1.92	0.58
NSIS	5,000	8.08	4.50	9.21	5.09
GAIS	5,000	5.88	0.67	0.96	0.36
LAIS	5,000	3.45	1.30	2.63	0.42
NSIS	10,000	9.38	4.75	11.06	5.77

Table 3.3: Relative efficiency of NSIS, GAIS, and LAIS compared with self-normalized importance sampling for Example 3. Figures for NSIS are computed over 1,000 independent runs. Figures for GAIS and LAIS are reprocessed from Table 2 in Givens and Raftery (1996).

server and a single waiting room (with infinite capacity). The interarrival and service times of the jobs are exponential distributed with parameter μ and ν, respectively. This model is well understood theoretically, but it is usually too restrictive for real world applications. In our case, e-mail arrives at a spam filter that decides whether a particular e-mail is spam. The data consist of interarrival times t_i (in seconds) and service times s_i (in milliseconds) for $n = 22,248$ e-mails. The data were recorded between 8 AM and 8 PM on eight business days in September 2008 and are available on request. (We are grateful to J. Kunkel for providing the data.) The system that produced the data is a single-queue, dual-server system (i.e. the e-mails are processed by two parallel spam filter threads). In the following, we investigate both the single- and the dual-server cases. The empirical distributions of the interarrival and service times are displayed in Figure 3.8. We can observe that the former is well approximated by an exponential distribution with parameter $\hat{\mu} = n/\sum_{i=1}^{n} t_i = 0.074$ (which is the maximum likelihood estimate). In contrast, for the service time distribution it is hard to find a parametric model. Therefore, we use a LBFP estimate. (Note that a kernel estimator is inappropriate because heavy sampling from the service time distribution is required.) The bin width was selected with the Gaussian reference rule for frequency polygons $\hat{h} = 2.15\hat{\sigma}n^{-1/5}$ (Terrell and Scott 1985), with $\hat{\sigma}$ being the standard deviation of the service times s_i.

We are interested in the probability that the queue length reaches a certain level K. This is a typical problem in queueing systems, with rare events being of particular interest. Importance sampling is a standard variance reduction technique for this task (see for instance Glynn and Iglehart 1989; Glasserman and Kou 1995; Kim, Roh, and Lee 2000). For estimating the probabilities, we simulate N busy periods and count the number of periods in which level K was reached. A busy period begins when an e-mail has arrived in an empty system and ends when either the system is empty again or the queue length has reached level K. Let ω_i be the sample path of the queue length in the ith busy period resulting from samples $\mathbf{x}_i^j$ and $\mathbf{y}_i^k$ drawn from the interarrival distribution p_t and service time distribution p_s, respectively. In the dual-server case, $\mathbf{y}_i^k$ represent the service times of both servers. The crude Monte Carlo estimate of the probability of interest is $\hat{I}_K = 1/N \sum_{i=1}^{N} \varphi(\omega_i)$, where $\varphi(\omega_i) = 1$ if ω_i reaches K, and $\varphi(\omega_i) = 0$ otherwise. Assume the number of e-mails that have been served during the ith busy period is

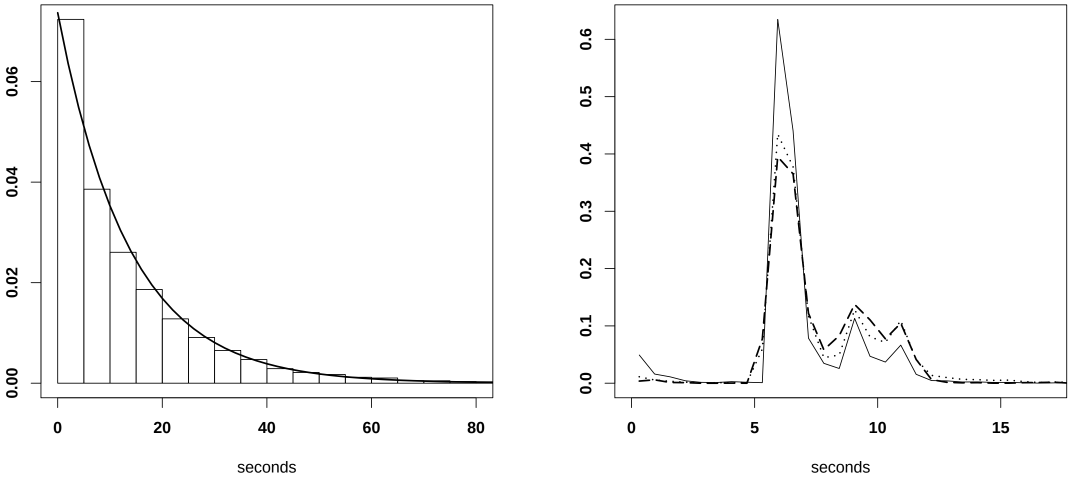

Figure 3.8: Spam filter application: Histogram of the empirical interarrival times and exponential distribution with parameter 0.074 (left). Linear blend frequency polygon estimates of the service time distribution (solid line) and of the optimal proposal q_φ^{opt} for the single server (dotted line) and dual server (dashed line) case for $K = 10$ (right).

L_i. Then there must be $K + L_i - 1$ arrivals during this period for the queue to reach level K. (Note, a busy period starts with one job in the queue.) Hence, if importance sampling is used, the estimator becomes

$$\hat{I}_K^{\text{IS}} = \frac{1}{N}\sum_{i=1}^{N} \varphi(\omega_i) l(\omega_i)$$

with likelihood ratio

$$l(\omega_i) = \prod_{j=1}^{K+L_i-1} \frac{p_t(\mathbf{x}_i^j)}{q_t(\mathbf{x}_i^j)} \prod_{k=1}^{L_i} \frac{p_s(\mathbf{y}_i^k)}{q_s(\mathbf{y}_i^k)}$$

and proposals q_t, q_s. Here, nonparametric importance sampling works as follows: We simulate M busy periods by sampling interarrival times $\tilde{\mathbf{x}}_i^j$ and service times $\tilde{\mathbf{y}}_i^k$ from trial distributions $q_{0,t}$ and $q_{0,s}$, respectively, and obtain sample paths $\tilde{\omega}_i$, $i = 1, \ldots, M$. Let $\mathcal{I} = \{i \in \{1, \ldots, M\}, \varphi(\tilde{\omega}_i) = 1\}$. For estimation of the optimal proposals, we use those times $\tilde{\mathbf{x}}_i^j$, $\tilde{\mathbf{y}}_i^k$, with $i \in \mathcal{I}$. The interarrival time proposal $\hat{q}_t$ is estimated parametrically by using an exponential distribution with parameter

$$\hat{\mu} = \sum_{i \in \mathcal{I}} \sum_{j=1}^{K+\tilde{L}_i-1} w_i^j \Big/ \sum_{i \in \mathcal{I}} \sum_{j=1}^{K+\tilde{L}_i-1} w_i^j \tilde{\mathbf{x}}_i^j \tag{3.7}$$

where $w_i^j = p_t(\tilde{\mathbf{x}}_i^j)/q_{0,t}(\tilde{\mathbf{x}}_i^j)$. The service time proposal $\hat{q}_s$ is estimated nonparametrically (as in Algorithm 1) based on samples $\tilde{\mathbf{y}}_i^k$ and weights $w_i^k = p_s(\tilde{\mathbf{y}}_i^k)/q_{0,s}(\tilde{\mathbf{y}}_i^k)$, $i \in \mathcal{I}$.

For our simulation, let's set $N = 1$ Mio. , $\lambda = 0.15$, and the trial distribution $q_{0,s}$ equal to the LBFP estimate of the service distribution. For M|M|1 systems it is well known that

	$K=5$		$K=10$		$K=20$		$K=30$	
Method	RE	CV	RE	CV	RE	CV	RE	CV
MC	1.0	0.001	1.0	0.14	-	-	-	-
IS	0.3	0.01	19.8	0.03	-	0.08	-	0.24
NIS	0.2	0.02	58.4	0.02	-	0.03	-	0.09

Table 3.4: Results for the spam filter queueing application (single-server case). Relative efficiency (RE) and coefficient of variation (CV) for the estimates of the probability that the queue length reaches level K. All figures are computed over 100 independent runs with 1 Mio. busy periods in each run.

	$K=4$		$K=6$		$K=8$	
Method	RE	CV	RE	CV	RE	CV
MC	1.0	0.007	1.0	0.044	1.0	0.34
IS	3.7	0.003	7.0	0.017	53.5	0.046
NIS	2.6	0.004	24.3	0.009	184.6	0.025

Table 3.5: Results for the spam filter queueing application (dual-server case). Relative efficiency (RE) and coefficient of variation (CV) for the estimates of the probability that the queue length reaches level K. All figures are computed over 100 independent runs with 1 Mio. busy periods in each run.

(asymptotically) optimal proposals are achieved by swapping the parameters μ and ν. For this reason, $q_{0,t}$ is set to the exponential distribution with parameter $\hat{\nu} = n / \sum_{i=1}^{n} s_i = 0.147$. As the parametric importance sampling benchmark, we consider the importance sampling scheme that carries out importance sampling for the interarrival times only. It uses the exponential distribution with parameter $\hat{\mu}$ defined in (3.7) as the proposal.

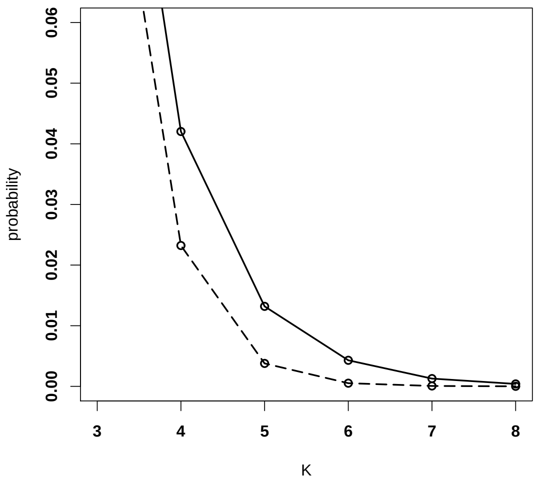

Figure 3.9: Results for spam filter application: Estimated probabilities of the queue length to reach level K for single server (heavy line) and dual server (dashed line) case.

We compare crude Monte Carlo, importance sampling, and nonparametric importance sampling in terms of the coefficient of variation (CV) and relative efficiency (RE). The former is defined as the ratio of the standard deviation to the mean of the probability estimate. Note that for the coefficient of variation, smaller values are favourable. The results are summarized in Tables 3.4 and 3.5.

Where no value is given, the crude Monte Carlo estimator was zero. We find that when the event of interest becomes rarer, nonparametric importance sampling becomes more favorable. This holds for both the single- and dual-server cases. The nonparametric importance sampling probability estimates for different queue levels K are shown in Figure 3.9. No error bounds are given because they are very small for the large number of busy periods used.

Real-world queueing systems typically involve complicated distributions, such as the service time distribution in this case. Therefore, it is often impossible to set up parametric importance sampling schemes for simulation. Here, nonparametric importance sampling has a distinct advantage.

Chapter 4

Nonparametric Partial Importance Sampling for Financial Derivative Pricing

4.1 Introduction

In the last decade, the complexity of the pricing models used for evaluation of financial products has experienced a distinct increase. As a consequence of this development, pure numerical methods became more and more inadequate for the high-dimensional integration tasks. Often, Monte Carlo integration is the only feasible method. This stems from the fact that the Monte Carlo convergence rate is independent of the problem dimension. However, crude Monte Carlo is often inefficient for practical sample sizes. Raising computing power and increasing the sample size is no solution. The need of efficient Monte Carlo methods is apparent.

Here, we consider importance sampling as a strategy to improve Monte Carlo simulation based derivative pricing. A key feature of importance sampling is that it can force the samples into the domain which is most important to the integrand. Intuitively, this is particularly useful for derivatives that rely on rare events. A deep out-of-the money option is an obvious example for rare event dependency. Crude Monte Carlo would only rarely produce samples which lead to non-zero payouts and, consequently, the Monte Carlo variance would be large. However, importance sampling is by far not limited to rare event cases. Compared with other variance reduction techniques (see Section 2.2) the usage of importance sampling is more involved, because the selection of a suitable proposal is generally difficult. But the additional effort is justified by the large potential of importance sampling to reduce the Monte Carlo variance.

Importance sampling has been successfully applied to derivative pricing based on Gaussian proposals. That is, the proposal was chosen from some class of Gaussian distributions. An important approach is based on a mean shift, which can be obtained through saddle point approximation (Glasserman, Heidelberger, and Shahabuddin 1999), adaptive stochastic optimization (Vazquez-Abad and Dufresne 1998; Su and Fu 2000, 2002), or least squares (Capriotti 2008).

This approach is also known as the "change-of-drift technique". In addition, Gaussian mixture distributions have been utilized for approximating the optimal proposal (Avramidis 2002). Summarizing, existing approaches are based on parametric importance sampling, that is the proposal is chosen from a certain class of distributions. For complex payouts it is hard to set up a class which contains a distribution that approximates the optimal proposal reasonably well.

We propose the usage of nonparametric importance sampling for derivative pricing. As shown in the preceding chapter, nonparametric importance sampling algorithms can be successfully applied to low-dimensional integration problems. However, high-dimensional integration tasks have not been considered until now. As a result of the curse of dimensionality and computational limitations nonparametric importance sampling cannot be applied directly to high-dimensional derivative pricing. The basic idea of our approach is to restrict nonparametric importance sampling to those coordinates which are of most importance to the integration problem. This approach can be justified by the concept of the effective dimension. To reduce the effective dimension and to identify the most relevant coordinates, principal component analysis is applied.

The advantage of nonparametric importance sampling compared with parametric importance sampling is its close approximation of the optimal proposal. We prove that the variance reduction factor of our nonparametric method increases with sample size converging to the – in some sense – optimal value. Parametric importance sampling methods achieve constant variance reduction factors. It is shown through simulations that the proposed algorithm is computationally more efficient than parametric importance sampling for well-known benchmark option pricing problems. In the case of low effective dimension, the algorithm not only outperforms in terms of mean square error but also in terms of computational costs. In other words, it is not only more accurate but also computationally cheaper. Nonparametric importance sampling and most parametric importance sampling methods share the property that they can be combined with other variance reduction techniques. This is demonstrated through the use of quasi-Monte Carlo (compare Section 2.6).

4.2 Derivative Pricing and Importance Sampling

Let's describe the evolution of the underlying asset through a stochastic differential equation of the form

$$dS(t) = rS(t)\ dt + \sigma(S(t))S(t)dW(t), \tag{4.1}$$

where $W(t)$ is a standard Brownian motion; r and σ are the risk-free interest rate and the volatility, respectively. Within this model, evaluating the price of a European option with payout function $C_K(S)$, strike level K, and expiry T means computing

$$\mathbf{E}[\exp(-rT)C_K(S)], \tag{4.2}$$

where the expectation is taken with respect to the risk neutral measure. Except of special cases, there is no explicit solution for stochastic differential equations like (4.1). Therefore, it is required to migrate to some discretization $\tilde{S}_{t_k}$ of the process $S(t)$, which is defined on a discrete-time grid

$0 = t_0 < t_1 < \cdots < t_d = T$. The first-order Euler discretization scheme yields

$$\tilde{S}_{t_{k+1}} = \tilde{S}_{t_k} + r\tilde{S}_{t_k}(t_{k+1} - t_k) + \sigma(\tilde{S}_{t_k})\tilde{S}_{t_k}\sqrt{t_{k+1} - t_k}Z_{t_k} \tag{4.3}$$

with standard normal innovations Z_{t_k}. In the following, we focus on an equally-spaced time grid, that is $t_i - t_{i-1} = \Delta t = \text{const}$. Based on this discretization, the option price (4.2) can be approximated through the integral

$$I_\varphi = \int_{\mathbb{R}^d} \varphi(\mathbf{x})p(\mathbf{x})d\mathbf{x},$$

where $\varphi(\mathbf{x}) = \exp(-rT)C_K(\tilde{S}(\mathbf{x}))$. p denotes the density of the multivariate Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ with $\mathbf{I}_d$ being the identity matrix of dimension d. By writing $\tilde{S}(\mathbf{x})$, it is meant that a trajectory of $\tilde{S}_{t_k}$ is built up based on the innovations $\mathbf{x} = (x_1, \ldots, x_d)^T$. To keep the discretization bias small, it is required to choose d considerably large which leads to a high-dimensional integration problem. Observe that we are precisely in the setting of sections 2.1 and 2.3. That is, the crude Monte Carlo estimator and the importance sampling estimator of the option price I_φ are given by (2.1) and (2.2), respectively. In addition, the optimal proposal q_φ^{IS} is given by (2.4).

4.3 Nonparametric Partial Importance Sampling

In this section, nonparametric partial importance sampling (NPIS) is introduced as a generalization of the nonparametric importance sampling algorithm discussed in Section 3.2.

Nonparametric importance sampling is a two-stage procedure. In the first stage, the optimal proposal is estimated nonparametrically based on samples drawn from a trial distribution q_0. In the second stage, this nonparametric density estimate is used as proposal for importance sampling. We pick up this approach, but instead of approximating the optimal proposal in the entire space, we focus on the optimal proposal in a certain subspace. That is, the nonparametric importance sampling procedure is restricted to a low-dimensional subproblem in order to avoid the curse of dimensionality. We decompose $\mathbf{x} = (\mathbf{x}_u, \mathbf{x}_{-u})$, where $u \subseteq \{1, 2, \ldots, d\}$, $\mathbf{x}_u = \{x_i; i \in u\}$, and $\mathbf{x}_{-u} = \{x_i; i \in \{1, 2, \ldots, d\} \setminus u\}$. The cardinality of u is denoted by $|u|$. Let's consider the marginalized optimal proposal obtained through integration with respect to $\mathbf{x}_{-u}$. It is given by

$$\check{q}_\varphi^{\mathrm{IS}}(\mathbf{x}_u) = \int_{\mathbb{R}^{d-|u|}} q_\varphi^{\mathrm{IS}}(\mathbf{x})d\mathbf{x}_{-u}.$$

Subspace u is chosen such that it covers those coordinates which are most important to the integrand (see Section 4.4). To limit the computational burden of the nonparametric method, u will be considerably small in practice ($1 \leq |u| \leq 3$). In the nonparametric partial importance sampling algorithm (NPIS) which is stated below $\check{q}_\varphi^{\mathrm{IS}}$ is estimated nonparametrically.

Algorithm: Nonparametric Partial Importance Sampling (NPIS)

Stage 1: Nonparametric estimation of the marginalized optimal proposal

- Select subset u, bin width h, trial distribution q_0, and sample sizes M and N.
- **For** $j = 1, \ldots, M$: Generate sample $\tilde{\mathbf{x}}^j \sim q_0$.
- Obtain nonparametric estimate $\hat{q}_\varphi^{\mathrm{IS}}$ of marginalized optimal proposal $\breve{q}_\varphi^{\mathrm{IS}}$

$$\hat{q}_\varphi^{\mathrm{IS}}(\mathbf{x}_u) = \frac{\hat{f}(\mathbf{x}_u)}{\frac{1}{M}\sum_{j=1}^{M} \omega^j},$$

where $\omega^j = |\varphi(\tilde{\mathbf{x}}^j)| p(\tilde{\mathbf{x}}^j) q_0(\tilde{\mathbf{x}}^j)^{-1}$ and

$$\begin{aligned} \hat{f}(\mathbf{x}_u) &= \frac{1}{M} \sum_{j_1,\ldots,j_{|u|}\in\{0,1\}} \left[\prod_{i\in u} \left(\frac{x_i - t_{k_i}}{h} \right)^{j_i} \left(1 - \frac{x_i - t_{k_i}}{h} \right)^{1-j_i} \right] \\ &\quad \times \sum_{j=1}^{M} \omega^j \mathbf{1}_{\prod_{i\in u}[t_{k_i+j_i} - h/2, t_{k_i+j_i} + h/2)}(\tilde{\mathbf{x}}^j) \end{aligned}$$

for $\mathbf{x}_u \in \prod_{i\in u}[t_{k_i}, t_{k_i} + h)$.

Stage 2: Partial Importance Sampling

- **For** $i = 1, \ldots, N$: Generate samples $\mathbf{x}_u^i \sim \hat{q}_\varphi^{\mathrm{IS}}(\mathbf{x}_u)$ and $\mathbf{x}_{-u}^i \sim p(\mathbf{x}_{-u})$.
- Evaluate

$$\hat{I}_\varphi^{\mathrm{NPIS}} = \frac{1}{N} \sum_{i=1}^{N} \varphi(\mathbf{x}^i) p(\mathbf{x}_u^i) \hat{q}_\varphi^{\mathrm{IS}}(\mathbf{x}_u^i)^{-1}.$$

The following theorem investigates the mean square error convergence properties of the NPIS algorithm to obtain the optimal value for bin width h.

Theorem 4.1. *Suppose that the assumptions given in Appendix A.7 hold, $\varphi \geq 0$, and $p(\mathbf{x}) = p(\mathbf{x}_u)p(\mathbf{x}_{-u})$. We denote $\breve{q} = \breve{q}_\varphi^{IS}$. Then, we obtain for $\hat{I}_{\varphi_M}^{NPIS}$ (as defined in Appendix A.7)*

$$\mathbf{E}[\hat{I}_{\varphi_M}^{NPIS} - I_\varphi]^2 = \frac{1}{N} \left[\int \frac{\nu(\mathbf{x})^2 p(\mathbf{x}_{-u})}{\breve{q}(\mathbf{x}_u)} d\mathbf{x} \ + \ I_\varphi^2 \left\{ h^4 H_1 + \frac{2^{|u|}}{3^{|u|} M h^{|u|}} H_2 \right\} \times (1 + o(1)) \right] \tag{4.4}$$

and the optimal bin width

$$h^{opt} = \left(\frac{|u| H_2 2^{|u|}}{4 H_1 3^{|u|}} \right)^{\frac{1}{4+|u|}} M^{-\frac{1}{4+|u|}},$$

where

$$H_1 = \frac{49}{2,880} \sum_{i\in u} \int \frac{(\partial_i^2 \breve{q})^2}{\breve{q}} + \frac{1}{64} \sum_{\substack{i,j\in u \\ i\neq j}} \int \frac{\partial_i^2 \breve{q} \partial_j^2 \breve{q}}{\breve{q}}, \quad H_2 = \int \frac{(q_\varphi^{IS})^2}{\breve{q}\ q_0}$$

and

$$\nu(\mathbf{x}) = \varphi(\mathbf{x})p(\mathbf{x}_u) - \int \varphi(\mathbf{x})p(\mathbf{x})d\mathbf{x}_{-u}.$$

Proof. See Appendix A.7.

The left and right term in the brackets in (4.4) can be interpreted as the variance caused by the components $\mathbf{x}_{-u}$ and $\mathbf{x}_u$, respectively. Note, subset u is chosen such that the left term is small compared with the right one. The expression in braces quantifies the mean square error of the nonparametric estimate, which depends on both $\breve{q}_\varphi^{\mathrm{IS}}$ and trial distribution q_0. For $h = h^{\mathrm{opt}}$ and $M/N \to \lambda \in (0, 1)$ $(M, N \to \infty)$ the theorem implies

$$\mathbf{E}[\hat{I}_{\varphi_M}^{\mathrm{NPIS}} - I_\varphi]^2 = \frac{1}{N}\int \frac{\nu(\mathbf{x})^2 p(\mathbf{x}_{-u})}{\breve{q}(\mathbf{x}_u)} d\mathbf{x} + \mathcal{O}(N^{-(8+|u|)/(4+|u|)}).$$

Hence, the variance caused by $\mathbf{x}_u$ is of lower order. In other words, the optimal variance (for partial importance sampling on coordinates u) is achieved asymptotically. As a consequence, compared with crude Monte Carlo and parametric importance sampling techniques, NPIS is expected to yield increasing efficiency as the sample size grows. Furthermore, if $|u| = d$ the mean square error converges as fast as $\mathcal{O}(N^{-(8+d)/(4+d)})$ which is precisely the case considered in Section 3.2. This is a massive improvement compared with the standard Monte Carlo rate $\mathcal{O}(N^{-1})$ for d which is small. Note, the results of this section also hold for distributions p other than the standard normal distribution.

Here, NPIS is only investigated for non-negative integrands. However, by decomposing the payout function $C = C^+ - C^-$, NPIS can also be applied to financial derivatives that have both positive and negative payouts.

4.4 Effective Dimension

The NPIS algorithm is based on the restriction on specific coordinates $\mathbf{x}_u$, where in high-dimensional integration problems $|u| \ll d$. This approach can be justified by the concept of the effective dimension. It is well known, that many integration problems in financial engineering, despite having a large nominal dimension, are low-dimensional in terms of the effective dimension. For a rigorous definition of the effective dimension, let's consider the functional analysis of variance (ANOVA) decomposition. Suppose $\int \varphi(\mathbf{x})^2 p(\mathbf{x})d\mathbf{x} < \infty$ and $p(\mathbf{x}) = \prod_{i=1}^d p(\mathbf{x}_i)$ is a product density. Then, φ can be written as a sum of 2^d orthogonal functions

$$\varphi(\mathbf{x}) = \sum_{u \subseteq \{1,2,\dots,d\}} \varphi_u(\mathbf{x}_u),$$

where the ANOVA functions φ_u are given recursively by

$$\varphi_u(\mathbf{x}_u) = \int_{\mathbb{R}^{d-|u|}} \varphi(\mathbf{x})p(\mathbf{x}_{-u})d\mathbf{x}_{-u} - \sum_{v \subset u} \varphi_v(\mathbf{x}_v).$$

Now, the fraction of the variance $\sigma^2 = \mathrm{Var}_p[\varphi]$, which is explained by certain lower-dimensional ANOVA functions, is considered. For this purpose, the variance of φ_u is defined by

$$\sigma_u^2 = \int_{\mathbb{R}^d} \varphi_u(\mathbf{x}_u)^2 p(\mathbf{x}) d\mathbf{x},$$

where $\sigma_\emptyset^2 = 0$. As the ANOVA decomposition is orthogonal, one has $\sigma^2 = \sum_u \sigma_u^2$. Hence, $\Gamma_u = \sum_{v \subseteq u} \sigma_v^2$ can be interpreted as the contribution of $\mathbf{x}_u$ to the total variance of φ. For a more detailed description of the ANOVA decomposition see, for instance, Takemura (1983) and Owen (1992). The following definition of the effective dimension is due to Caflisch, Morokoff, and Owen (1997).

Definition 4.2. The effective dimension (in the truncation sense) is the cardinality of the smallest subset u such that $\Gamma_u \geq \gamma\sigma^2$ with $0 < \gamma < 1$.

The threshold γ is chosen close to one. In our framework, we found $\gamma = 0.9$ reasonable. The effective dimension does not only allow to identify those coordinates which most effect the integral value but it also indicates how many coordinates are required to cover a certain amount of the variance.

Now, a Monte Carlo procedure that allows one to determine the effective dimension of a given problem is described (Wang and Fang 2003). It can be shown that the cumulated variances satisfy

$$\Gamma_u = \int_{\mathbb{R}^{2d-|u|}} \varphi(\mathbf{x})\varphi(\mathbf{x}_u, \mathbf{y}_{-u}) p(\mathbf{x}) p(\mathbf{y}) d\mathbf{x} d\mathbf{y}_{-u} - I_\varphi^2,$$

where both $\mathbf{x}$ and $\mathbf{y}$ are vectors in $\mathbb{R}^d$. Hence, the effective dimension can be computed based on the approximations

$$\hat{\Gamma}_u = \frac{1}{l} \sum_{i=1}^{l} \varphi(\mathbf{x}^i)\varphi(\mathbf{x}_u^i, \mathbf{y}_{-u}^i) - \hat{I}_\varphi^2 \qquad (i = 1, 2, \dots, d)$$

and

$$\hat{\sigma}^2(\varphi) = \frac{1}{l} \sum_{i=1}^{l} \varphi(\mathbf{x}^i)^2 - \hat{I}_\varphi^2$$

with $\hat{I}_\varphi = 1/l \sum \varphi(\mathbf{x}^i)$. The samples $\mathbf{x}^1$, $\mathbf{x}^2$, …, $\mathbf{x}^l$, $\mathbf{y}^1$, $\mathbf{y}^2$, …, $\mathbf{y}^l$ are drawn from p.

4.5 Gaussian Models

The purpose of this section is to show how NPIS can be applied to models that are based on the integration with respect to high-dimensional Gaussian distributions. As mentioned earlier, NPIS is inefficient as a result of the curse of dimensionality unless the effective dimension (and thus $|u|$) is small. For typical financial integration problems it is generally not advisable to apply NPIS with $|u|$ larger than 3, unless the number of paths to be sampled is huge or the domain of interest is very small (rare event case).

Suppose the task is to integrate with respect to $\mathcal{N}(0, \Sigma)$. Now, principal component analysis (PCA) is applied to transform the problem. The (positive-definite) covariance matrix Σ is written as

$$\Sigma = V \Lambda V^T,$$

with $\Lambda = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_d)$ and eigenvalues λ_i. The columns of V are the corresponding unit-length eigenvectors. Thus, one has

$$V \Lambda^{1/2} \mathbf{Z} \sim \mathcal{N}(0, \Sigma)$$

for $\mathbf{Z} \sim \mathcal{N}(0, I_d)$. Without loss of generality, it is assumed that the eigenvalues (and the corresponding eigenvectors) are sorted so that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$. The PCA construction of samples from $\mathcal{N}(0, \Sigma)$ is optimal in the sense that it provides an optimal lower-dimensional approximation (in the mean square error sense) to the random variable of interest. This means that the first k components of $\mathbf{Z}$ explain as much as possible of the total variance. More precisely, it can be shown that they explain the fraction $(\lambda_1 + \lambda_2 + \ldots + \lambda_k)/(\lambda_1 + \lambda_2 + \ldots + \lambda_d)$ of it.

The option pricing problem introduced in Section 4.2 leads to the construction of discretized Brownian motion paths based on samples from the multivariate Gaussian distribution. Paths are most easily built up through the random walk construction guided by (4.3). In this construction each component "counts roughly the same" rendering the restriction on a lower-dimensional subspace and hence the application of NPIS impractical. Note, the integral I_φ can be rewritten as $I_\varphi = \int \tilde{\varphi}(\mathbf{x}) p_{\mathcal{N}(\mathbf{0},\Sigma)}(\mathbf{x}) d\mathbf{x}$, where Σ is the covariance matrix of the discretized Brownian motion with entries $\Sigma_{ij} = \min\{t_i, t_j\}$. This suggests that PCA can be used to reduce the effective dimension. The PCA construction of discretized Brownian motion paths has a continuous limit known as Karhunen-Loève expansion of Brownian motion:

$$W(t) = \sum_{i=1}^{\infty} \sqrt{\lambda_i} \psi_i(t) Z_i, \qquad 0 \leq t \leq 1,$$

where $\psi_i(t) = \sqrt{2} \sin\{(i - 0.5)\pi t\}$, $\lambda_i = \{(i - 0.5)\pi\}^{-2}$, and $Z_i \sim \mathcal{N}(0, 1)$ (Adler 1990). Based on the expression for λ_i, it is easily shown that Z_i explains the fraction $2\lambda_i$ of the path's variability (which is approximately 81%, 9%, 3% for $i = 1, 2, 3$, respectively). These values are not only of asymptotic nature but also hold for a small number of discretization steps (with slight deviations). This astonishing result claims that very few PCA components suffice to determine most of the path's variation no matter how long or detailed it is. Particularly, the first PCA component plays a dominant role and has a nice geometrical interpretation. Roughly speaking, it determines the path's direction in the path space. This is visualized in Figure 4.1.

Another common method for the reduction of the effective dimension (of a discretized Brownian motion) is the Brownian Bridge technique. In this work, the focus is on PCA because of its optimality property. However, it is remarked that in certain situations Brownian Bridge techniques are superior to PCA. This may particularly be the case if the payout function only depends on the terminal value of the underlying. It is mentioned that NPIS can also be combined with Brownian Bridge techniques.

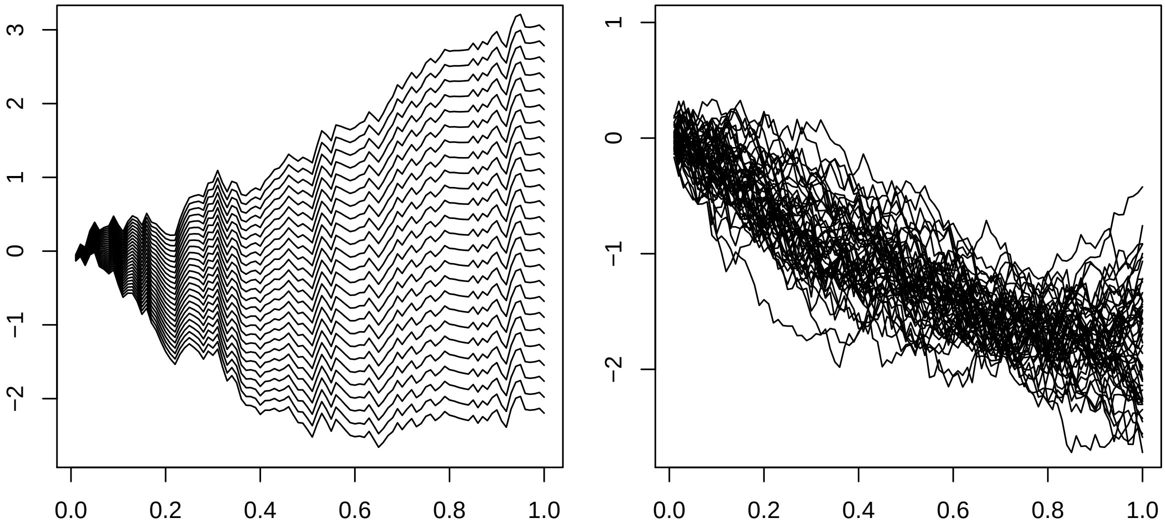

Figure 4.1: Discretized Brownian motion paths: first PCA component varies whereas other components are fixed to random values (left); first PCA component is fixed and other components vary randomly (right).

4.6 Quasi-Monte Carlo Integration

Quasi-Monte Carlo is often used to (further) improve Monte Carlo methods for derivative pricing. In contrast to Monte Carlo, quasi-Monte Carlo integration uses so-called low-discrepancy sequences instead of pseudo random numbers (see Section 2.6). From the well-known Koksma-Hlawka inequality (see Niederreiter (1992) is follows that quasi-Monte Carlo can massively outperform Monte Carlo in low-dimensional integration problems. In high dimensions the advantage of quasi-Monte Carlo should disappear. However, it has been shown that quasi-Monte Carlo may be effectively applied to high-dimensional problems in financial engineering (Paskov and Traub 1995; Ninomiya and Tezuka 1996; Traub and Werschulz 1998). This stems from the fact mentioned earlier that many problems in finance have rather low effective dimension compared with the nominal dimension. As the convergence properties of quasi-Monte Carlo become worse in higher dimensions, it is important to assign the first coordinates to the most relevant dimensions of the integration problem. In our setting, the relevant coordinates are those contained in u.

For the computation of the mean square error of a quasi-Monte Carlo based estimator it is required to randomize the deterministic low-discrepancy sequence used. In the simulations, we apply the random shift technique which is explained in Section 2.6.

4.7 Comparison with Parametric Importance Sampling

Until now, the application of importance sampling in finance was limited to parametric importance sampling. In particular, Gaussian or mixtures of Gaussian distributions have been applied. The variance of a parametric importance sampling estimator with proposal q_θ (and parameter

$\theta \in \Theta$) can be written as

$$\frac{\sigma_{\mathrm{IS}}^2}{N} = \frac{I_\varphi^2}{N}\left\{\int_{\mathbb{R}^d} \frac{q_\varphi^{\mathrm{IS}}(\mathbf{x})^2}{q_\theta(\mathbf{x})} d\mathbf{x} - 1\right\}, \tag{4.5}$$

where σ_{IS}^2 is defined as in Section 2.3. First, this suggests that, in contrast to NPIS, the variance reduction factor is constant because all terms are $\mathcal{O}(N^{-1})$. Second, the variance is critically affected by the tails of q_θ. Using Gaussian proposals, it is often hard to approximate the tails of q_φ^{IS} reasonably well. There lies a distinct advantage of nonparametric importance sampling methods. Most parametric importance sampling approaches aim at choosing θ so that (4.5) is minimized. We now discuss a variant of the least-squares importance sampling (LSIS) algorithm (Capriotti 2008) which is directly comparable to NPIS. It is based on the Gaussian proposal $\mathcal{N}(\mu, \mathbf{I}_d)$ with parameter $\mu \in \mathbb{R}^d$. Similar to NPIS, it is a two-stage algorithm. In the first stage, based on M samples from p, a least-squares problem is solved to estimate the optimal drift change μ. (The variance can also be adjusted through this procedure.) However, as the problem dimension grows the estimate of μ becomes unreliable. The variant of this algorithm which is suggested here applies LSIS to the coordinates $\mathbf{x}_u$, that are determined through principal component analysis and the effective dimension (analogous to NPIS). This makes the LSIS and the NPIS directly comparable. In Section 4.9, NPIS and this variant of LSIS are tested against each other through simulations.

Besides the superior convergence properties, NPIS has a computational advantage over parametric importance sampling which is of relevance in practice. For computing the importance sampling weights, parametric importance sampling typically needs to evaluate the `exp` function which is very expensive. Through the use of the LBFP estimator, these evaluations are reduced in the NPIS algorithm. This leads to a relevant reduction of the computational costs (compare Section 4.9).

Finally, we remark that combinations of parametric importance sampling and NPIS are possible. For instance, while applying NPIS to $\mathbf{x}_u$ one can carry out parametric importance sampling on the remaining coordinates $\mathbf{x}_{-u}$.

4.8 Implementation of the Algorithm

In this section, the details of practical implementation of the proposed NPIS algorithm are discussed. First, an overview of the required ingredients for the implementation is given.

Overview

u The subset u is chosen according to the effective dimension (with $\gamma = 0.9$), which can be computed with the algorithm given in Section 4.4. If PCA is used, the first few principal components are selected.

q_0 The choice of the trial distribution should be guided by the following two criteria: First, it should allow for efficient sampling and evaluation. Second, the marginal distributions of the coordinates contained in u should be overdispersed (heavy-tailed) compared with the standard normal distribution. An all-purpose trial distribution, which we found to work well in practice, is given below. Alternatively, one can use a parametric choice tailored to the specific integration problem or one can simply use the (multivariate) standard normal distribution. The latter is often not a good choice because of the importance of the tails of the proposal.

h A Gaussian reference rule for the bin width h can be computed in Stage 1 of the algorithm (the details are given below).

M For the simulations, we used $M = \max\{256, 0.25N\}$. In the special case when $|u| = d$ an optimal value for the proportion M/N can be derived (Theorem 3.1 in Section 3.2).

LBFP The details of the implementation of the LBFP estimator can be found in Section 3.4.2. A C++ implementation of the LBFP as well as the R-package `lbfp` are available (see Section 8.2).

We emphasize that, in contrast to most parametric importance sampling algorithms, all parameters are adjusted automatically, such that no trial-and-error parameter selection and no analytical computation are necessary in practice.

Trial Distribution

As trial distribution we propose a simple product density. It is composed of a uniform distribution on $[-\rho_M, \rho_M]^{|u|}$ and the multivariate Gaussian distribution $p(\mathbf{x}_{-u})$:

$$q_0(\mathbf{x}) = p(\mathbf{x}_{-u}) \times \frac{1}{(2\rho_M)^{|u|}} \prod_{i \in u} \mathbf{1}_{[-\rho_M, \rho_M]}(x_i),$$

where ρ_M is the $(1 + (1-\epsilon)^{1/M})/2$-quantile of $\mathcal{N}(0,1)$. $\epsilon > 0$ is very small, say $\epsilon = 10^{-4}$. Consequently, $\mathbf{P}(\max_{1 \le i \le M} |Z_i| > \rho_M) = \epsilon$ holds for standard normal distributed Z_i. This ensures that the bias caused by the bounded support of the uniform distribution is very small. In addition, the uniform distribution guaranties that the space of $\mathbf{x}_u$ is well explored even for a small sample size.

Practical Bin Width Selection

The expression for h^{opt} given in Theorem 4.1 is intractable analytically because of the unknown constants H_1 and H_2. The plug-in method suggested in Zhang (1996) also seems unsuitable for our integration problem as derivatives of the integrand are required. We propose to apply a Gaussian approximation of H_1 and H_2. Suppose $\check{q}_\varphi^{\text{IS}}$ is the density of a centered multivariate Gaussian distribution with covariance matrix $\text{diag}(\sigma_1^2, \sigma_2^2, \ldots, \sigma_{|u|}^2)$. Under this assumption, it can be shown that

$$H_1 = \frac{98}{2,880} \sum_{i \in u} \sigma_i^{-4}. \tag{4.6}$$

For the constant H_2 the mean of q_φ^{IS} plays the dominant role. Therefore, it is assumed that q_φ^{IS} is the density of $\mathcal{N}((\mu_1, \mu_2, \ldots, \mu_d)^T, \mathbf{I}_d)$. If the trial distribution is chosen as explained above, one yields

$$H_2 \approx \rho_M^{|u|} \exp[\sum_{i \notin u} \mu_i^2]. \tag{4.7}$$

In the algorithm, the expressions in (4.6) and (4.7) can be approximated based on the samples $\tilde{\mathbf{x}}^1, \ldots, \tilde{\mathbf{x}}^M$. This follows from the fact, that the samples $\tilde{\mathbf{x}}^j$ weighted with $\omega^j / \sum_{k=1}^M \omega^k$ approximate q_φ^{IS}.

4.9 Simulation Results

Different European option pricing scenarios are considered to compare the proposed algorithms (NPIS and the combination of NPIS and quasi-Monte Carlo (QNPIS)) with existing methods (crude Monte Carlo (MC), quasi-Monte Carlo (QMC), LSIS, and the combination of LSIS and quasi-Monte Carlo (QLSIS)). The performance of the algorithms is measured through the variance reduction factors (computed with respect to crude Monte Carlo) and the relative computational efficiency (RCE). The relative computational efficiency is defined as the ratio of the computational efficiency (defined in Section 2.2) of the method of interest to the computational efficiency of crude Monte Carlo. The computational costs are measured in seconds. All simulations are done for different sample sizes N in order to demonstrate the increasing variance reduction factors of NPIS.

Examples 1 through 3 consider different single- and multi-asset options within the standard Black-Scholes model. There, the price of an asset S at time t is given by

$$S(t) = S(0) \exp[(r - 0.5\sigma^2)t + \sigma\sqrt{t}Z]$$

with standard normal random variable Z. The simulations are based on the following setting: $S(0) = 100$, $\sigma = 0.3$, $r = 0.05$, and $T = 1$. In Example 4, the pricing of a cap within the CIR model is investigated to show the effectiveness of NPIS/QNPIS in a square-root diffusion model. For all algorithms, apart from crude Monte Carlo, the PCA path construction is used. The parameters u, q_0, and h are chosen according to the description in the preceding section. Note, Theorem 1 does not apply to quasi-Monte Carlo sampling. We found empirically that QNPIS requires a larger bin width. In the simulations, $3h^{\mathrm{opt}}$ is used. For LSIS and NPIS, M is set as suggested in the preceding section whereas for QNPIS and QLSIS $M = 1024$ is used throughout. The least squares estimates required in LSIS/QLSIS were computed with ten iterations of the Levenberg-Marquardt method (Press et al. 1992, pp. 683-688).

The computations were carried out on a Dell Precision T3400, Intel CPU 2.83GHz. All algorithms were coded in C++ (see Chapter 8 for more details). The Mersenne Twister 19937 (Matsumoto and Nishimura 1998) and the Sobol sequence (Sobol 1967) were used for pseudo- and quasi-random number generation, respectively. The Sobol sequence is randomized by the random shift technique.

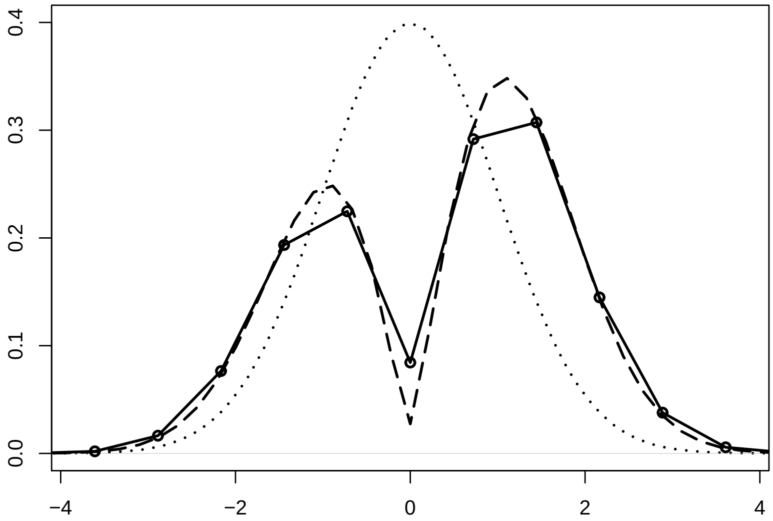

Figure 4.2: Standard normal distribution (dotted line), optimal proposal for a straddle option within the Black-Scholes model (dashed line), and an LBFP estimate of the optimal proposal (solid line). Model parameters: $S(0) = 100$, $\sigma = 0.3$, $r = 0.05$, $T = 1$, and $K = 100$.

Example 1. Straddle Option

The payout function of a straddle option is given by

$$C_K(S) = (S(T) - K)^+ + (K - S(T))^+.$$

In the Black-Scholes world the pricing of a straddle option is a one-dimensional integration problem with multi-modal optimal proposal. Gaussian proposals (such as drift changes) are severely inefficient for multi-modal payouts (Capriotti 2008). The optimal proposal and an LBFP estimate generated in the NPIS algorithm are shown in Figure 4.2. The LBFP estimate closely approximates the optimal proposal. To account for the bimodality, we used $2h^{\text{opt}}$ as bin width in the QNPIS algorithm. However, $3h^{\text{opt}}$ gives only slightly worse results. The simulation results for the strikes $K = 100$ and $K = 110$ are reported in Table 4.1. First notice, that NPIS significantly outperforms LSIS because of the better approximation of the optimal proposal. Second, the variance reduction factors for NPIS increase with sample size which agrees with Theorem 1. Third, the combination of NPIS and quasi-Monte Carlo leads to massive efficiency gains. Even after adjusting for the execution times the gains are enormous (see values for the RCE). Note, the increasing variance reduction factors for QLSIS and QNPIS are a result of the quasi-Monte Carlo sampling.

Example 2. Asian Options

An arithmetic Asian call with payout function

$$C_K(S) = \left(\frac{1}{d} \sum_{i=1}^{d} S(t_i) - K \right)^+$$

Parameters		VR (RCE)				
N	K	QMC	LSIS	NPIS	QLSIS	QNPIS
2^{10}	100	224 (380)	1.3 (0.4)	9 (0.9)	1,064 (127)	2.3 $\times 10^5$ (8,469)
	110	253 (548)	1 (0.4)	6 (0.8)	964 (150)	3.2 $\times 10^5$ (1.5 $\times 10^4$)
2^{11}	100	264 (557)	1.3 (0.5)	13 (1.7)	1,361 (384)	2.6 $\times 10^5$ (2.1 $\times 10^4$)
	110	290 (532)	1 (0.3)	8 (1)	1,092 (291)	3.1 $\times 10^5$ (2.4 $\times 10^4$)
2^{12}	100	460 (941)	1.3 (0.4)	17 (2.2)	2,209 (1,006)	6.8 $\times 10^5$ (8.6 $\times 10^4$)
	110	505 (953)	1 (0.3)	11 (1.3)	2,201 (965)	7.4 $\times 10^5$ (8.7 $\times 10^4$)

Table 4.1: The table reports the variance reduction (VR) factors and the relative computational efficiency (RCE) for a straddle option within the Black-Scholes model. Model parameters: $S(0) = 100$, $\sigma = 0.3$, $r = 0.05$, $T = 1$, and $|u| = d = 1$. All values are computed based on 1,000 independent runs.

is investigated. The optimal proposal is unimodal. This integration problem is well suited for NPIS/QNPIS because its effective dimension is one. The strikes $K = 100$, 130, and 175 are considered. For strike $K = 175$ the option price is approximately 0.018 (for d =16) representing a rare event option pricing framework (which is still of practical interest).

Table 4.2 shows the results for $d = 16$ and $d = 64$ discretization steps. The results of the Gaussian importance sampling algorithm (GIS) based on saddle point approximation (Glasserman, Heidelberger, and Shahabuddin 1999) are also reported. We emphasize that the variance reduction and the relative computational efficiency increase with both strike level K and the sample size. The variance reduction factors of GIS and LSIS are roughly constant. This coincides with the theoretical results. Particularly in the rare event cases, massive efficiency gains are achieved and NPIS/QNPIS improve significantly over their parametric competitors. In addition, the values for the relative computational efficiency establish that NPIS and QNPIS are computationally much more efficient than parametric importance sampling strategies. In the table, missing values indicate that the trial stage sometimes failed to generate paths with positive payouts. To explain the result's dependency on the strike level, the marginalized optimal proposal (of the first PCA component) for different strikes were plotted (Figure 4.3). One can observe that both the mean and the variance of the marginalized optimal proposals change with K. As a result of the shrinking variance (and the increasing skewness) of the marginalized optimal proposals, importance sampling approaches based on pure drift changes become worse (relatively to NPIS/QNPIS) as K increases.

Table 4.3 gives results for the case when the execution time is fixed such as in real-time applications. The sample sizes were chosen so that all algorithms needed approximately the same time for execution. The values suggest that the variance of NPIS is roughly ten times smaller than those of existing importance sampling techniques.

In Table 4.4, the values for an Asian option with a knock-out feature are shown. The option will pay nothing if the arithmetic average exceeds the knock-out level $\tilde{K}$. The payout function is given by

$$C_K(S) = \left(\frac{1}{d}\sum_{i=1}^{d} S(t_i) - K\right)^+ \mathbf{1}_{\{\frac{1}{d}\sum_{i=1}^{d} S(t_i) < \tilde{K}\}}.$$

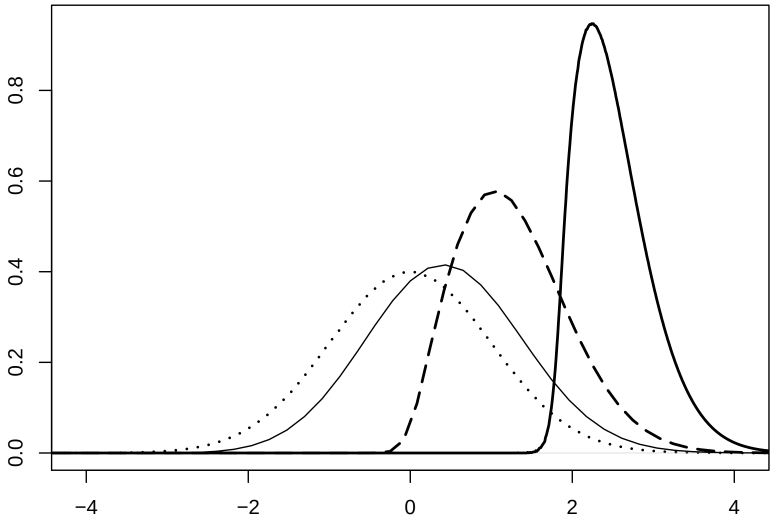

Figure 4.3: Standard normal distribution (dotted line), marginalized optimal proposal (of first principal component) for an Asian option with strike $K = 60$ (thin solid line), $K = 100$ (dashed line), and $K = 140$ (thick solid line). Model parameters: $S(0) = 100$, $\sigma = 0.3$, $r = 0.05$, $T = 1$, and $d = 16$.

The evaluation of this option is a difficult task because the relevant domain is very narrow. The strike $K = 140$ and the knock-out levels $\tilde{K} = 150$ and $\tilde{K} = 170$ are considered. The EDs are two and one for $\tilde{K} = 150$ and $\tilde{K} = 170$, respectively. Both LSIS and NPIS have problems to generate paths with positive payouts in the trial stage (which is reflected in the missing values in Table 4.4). Again, QNPIS significantly improves over QLSIS.

Finally, simulations for an Asian straddle option that pays

$$C_K(S) = (\frac{1}{d}\sum_{i=1}^{d} S(t_i) - K)^+ + (K - \frac{1}{d}\sum_{i=1}^{d} S(t_i))^+$$

are discussed. As for the standard straddle option, NPIS provides efficiency gains compared with LSIS (see Table 4.5). Although, the variance reduction factors and the relative computational efficiency of QNPIS are large, they are much smaller than those obtained for the standard straddle option.

Example 3. Multi-Asset Options

In this example, multi-asset options are considered. Suppose one deals with s assets that satisfy

$$S_i(t) = S_i(0)\exp[(r - 0.5\sigma^2)t + \sigma\sqrt{t}Z_i] \qquad i = 1, \ldots, s,$$

where the correlation matrix of $Z_1, \ldots, Z_s$ is denoted by Σ. To keep the setting simple, $S_i(0) = 100$ and $\text{corr}(Z_i, Z_j) = 0.3$ for $i, j = 1, \ldots, s$, $i \neq j$ is assumed. The effective dimension is reduced by applying PCA to the correlation matrix. We investigate two different payout structures. First,

Parameters				VR (RCE)					
N	d	K	ED	QMC	GIS	LSIS	NPIS	QLSIS	QNPIS
2^{10}	16	100	1	139 (139)	10 (3)	9 (2)	21 (11)	1,427 (113)	859 (187)
		140	1	17 (17)	55 (17)	50 (10)	200 (102)	4,778 (375)	5,462 (1,193)
		175	1	2 (2)	683 (202)	- (-)	3,809 (1,941)	4.3 $\times 10^4$ (3,326)	1.1 $\times 10^5$ (2.5 $\times 10^4$)
	64	100	1	145 (144)	8 (3)	8 (2)	20 (11)	1,409 (108)	909 (224)
		140	1	16 (16)	61 (19)	53 (10)	245 (138)	5,679 (434)	7,428 (1,828)
		175	1	2 (2)	902 (280)	- (-)	4,403 (2,501)	5.8 $\times 10^4$ (4,506)	1.0 $\times 10^5$ (2.5 $\times 10^4$)
2^{11}	16	100	1	171 (173)	9 (3)	9 (2)	28 (14)	1,535 (226)	908 (322)
		140	1	21 (22)	57 (17)	52 (10)	285 (146)	5,647 (823)	6,443 (2,267)
		175	1	3 (3)	680 (204)	- (-)	5,161 (2,646)	4.5 $\times 10^4$ (6,599)	1.3 $\times 10^5$ (4.4 $\times 10^4$)
	64	100	1	185 (185)	9 (3)	9 (2)	30 (17)	1,583 (225)	912 (360)
		140	1	21 (16)	69 (21)	55 (10)	329 (185)	5,951 (847)	8,027 (3,164)
		175	1	2 (2)	1,072 (332)	- (-)	7,255 (4,117)	6.2 $\times 10^4$ (8,757)	1.1 $\times 10^5$ (4.4 $\times 10^4$)
2^{12}	16	100	1	339 (339)	9 (3)	9 (2)	33 (17)	2,549 (647)	1,499 (767)
		140	1	42 (43)	56 (17)	59 (12)	324 (167)	8,742 (2,219)	1.0 $\times 10^4$ (5,212)
		175	1	5 (5)	756 (232)	- (-)	5,224 (2,696)	8.7 $\times 10^4$ (2.2 $\times 10^4$)	2.2 $\times 10^5$ (1.1 $\times 10^5$)
	64	100	1	354 (352)	10 (3)	10 (2)	35 (20)	2,743 (682)	1,627 (921)
		140	1	36 (36)	68 (21)	57 (11)	369 (209)	9,685 (2,407)	1.3 $\times 10^4$ (7,388)
		175	1	4 (4)	1,031 (318)	- (-)	7,414 (4,198)	9.7 $\times 10^4$ (2.4 $\times 10^4$)	1.8 $\times 10^5$ (1.0 $\times 10^5$)

Table 4.2: The table reports the variance reduction (VR) factors, the relative computational efficiency (RCE), and the effective dimension (ED) for an Asian option within the Black-Scholes model. Model parameters: $S(0) = 100$, $\sigma = 0.3$, $r = 0.05$, $T = 1$. All values are computed based on 1,000 independent runs.

		VR (N)			
Time	ED	MC	GIS	LSIS	NPIS
0.35	1	1 (2^{13})	16 ($\lfloor 2^{11.19} \rfloor$)	12 ($\lfloor 2^{10.68} \rfloor$)	168 (2^{12})
0.7	1	1 (2^{14})	16 ($\lfloor 2^{12.19} \rfloor$)	11 ($\lfloor 2^{11.68} \rfloor$)	175 (2^{13})
1.4	1	1 (2^{15})	17 ($\lfloor 2^{13.19} \rfloor$)	11 ($\lfloor 2^{12.68} \rfloor$)	158 (2^{14})

Table 4.3: The table reports the variance reduction (VR) factors, the sample sizes (N), and the effective dimension (ED) for an Asian option within the Black-Scholes model. The execution time is fixed to 0.35, 0.7, and 1.4 seconds, respectively. The sample sizes are chosen such that all algorithms approximately achieved the fixed execution time. Model parameters: $S(0) = 100$, $\sigma = 0.3$, $r = 0.05$, $T = 1$, $K = 140$, and $d = 16$. All values are computed based on 1,000 independent runs.

Parameters			VR (RCE)				
N	$\tilde{K}$	ED	QMC	LSIS	NPIS	QLSIS	QNPIS
2^{10}	150	2	5 (5)	- (-)	- (-)	69 (5)	110 (21)
	170	1	16 (16)	- (-)	37 (19)	1,003 (80)	1,362 (297)
2^{11}	150	2	6 (6)	- (-)	- (-)	68 (10)	123 (36)
	170	1	18 (18)	- (-)	134 (68)	1,168 (171)	1,613 (568)
2^{12}	150	2	6 (6)	- (-)	- (-)	82 (21)	163 (63)
	170	1	23 (24)	- (-)	106 (55)	1,530 (394)	1,883 (961)

Table 4.4: The table reports the variance reduction (VR) factors, the relative computational efficiency (RCE), and the effective dimension (ED) for an Asian option with a knock-out feature within the Black-Scholes model. Model parameters: $S(0) = 100$, $\sigma = 0.3$, $r = 0.05$, $T = 1$, $K = 140$, and $d = 16$. All values are computed based on 1,000 independent runs.

Parameters			VR (RCE)				
N	d	ED	QMC	LSIS	NPIS	QLSIS	QNPIS
2^{10}	16	1	193 (199)	1.2 (0.2)	6 (3)	300 (24)	323 (71)
	64	1	213 (214)	1.1 (0.2)	6 (4)	321 (25)	361 (90)
2^{11}	16	1	225 (233)	1.2 (0.2)	8 (4)	359 (53)	418 (151)
	64	1	256 (249)	1.2 (0.2)	9 (5)	397 (57)	410 (164)
2^{12}	16	1	425 (440)	1.2 (0.2)	10 (5)	634 (165)	711 (372)
	64	1	454 (455)	1.2 (0.2)	11 (6)	715 (179)	717 (406)

Table 4.5: The table reports the variance reduction (VR) factors, the relative computational efficiency (RCE), and the estimated effective dimension (ED) for an Asian straddle option within the Black-Scholes model. Model parameters: $S(0) = 100$, $\sigma = 0.3$, $r = 0.05$, $T = 1$, and $K = 100$. All values are computed based on 1,000 independent runs.

the price for an average option with payout

$$C_K(S_1, \ldots, S_s) = \left(\frac{1}{s} \sum_{i=1}^{s} S_i(T) - K \right)^+$$

is computed. The second option depends on the maximum of the underlyings' final values and has the payout function

$$C_K(S_1, \ldots, S_s) = \left(\max_{1 \le i \le s} \{S_i(T)\} - K \right)^+ .$$

From Table 4.6, one can observe that the results for the average option are qualitatively similar to those of the Asian option in Example 2. Particularly, the effective dimension is also equal to one.

The results for the second option with strikes $K = 150$ and $K = 200$ are reported in Tables 4.7 and 4.8, respectively. The pricing of the second option is a difficult problem because the effective dimension is equal to the nominal dimension. Although, for $K = 200$ QNPIS is superior to quasi-Monte Carlo and QLSIS for $s = 2$, 3, and 4 (in terms of the variance reduction factors), for $K = 150$ this only holds for $s = 2$ and 3. We emphasize on the massive efficiency gains obtained by QNPIS for strike $K = 200$. For $s > 2$ the sample size used was too small for NPIS to perform well. We conclude that the applicability of NPIS/QNPIS depends not only on the effective dimension of the problem but also on the sample size used. An LBFP estimate of the optimal proposal for the case $s = 2$ is plotted in Figure 4.4. Here, the PCA construction leads to a bimodal optimal proposal which can be closely approximated though an LBFP.

Example 4. Cap in the CIR Model

Finally, we consider the CIR interest rate model (Cox, Ingersoll, and Ross 1985). Here, interest rate r_t follows a square-root diffusion model

$$dr_t = \kappa(\theta - r_t)dt + \sigma\sqrt{r_t}dW_t.$$

Parameters			VR (RCE)				
N	K	ED	QMC	LSIS	NPIS	QLSIS	QNPIS
2^{10}	100	1	179 (346)	9 (4)	24 (24)	4,048 (620)	3,315 (1,384)
	140	1	19 (38)	43 (16)	212 (210)	5,171 (788)	6,269 (2,612)
	175	1	2 (3)	- (-)	3,277 (3,229)	2.5 $\times 10^4$ (3,856)	4.5 $\times 10^4$ (1.9 $\times 10^4$)
2^{11}	100	1	212 (409)	9 (3)	34 (33)	4,249 (1,197)	3,677 (2,475)
	140	1	26 (50)	48 (18)	338 (333)	5,438 (1,533)	6,932 (4,697)
	175	1	2 (4)	- (-)	4,637 (4,571)	2.9 $\times 10^4$ (8,313)	4.9 $\times 10^4$ (3.3 $\times 10^4$)
2^{12}	100	1	428 (830)	9 (3)	49 (48)	4,872 (2,403)	3,996 (3,948)
	140	1	49 (96)	52 (20)	372 (368)	6,373 (3,157)	7,720 (7,630)
	175	1	4 (9)	- (-)	5,953 (5,857)	4.3 $\times 10^4$ (2.1 $\times 10^4$)	6.5 $\times 10^4$ (6.4 $\times 10^4$)

Table 4.6: The table reports the variance reduction (VR) factors, the relative computational efficiency (RCE), and the estimated effective dimension (ED) for a multi-asset average option within the Black-Scholes model. Model parameters: $S_i(0) = 100$ ($i = 1, \ldots, s$), $\sigma = 0.3$, $r = 0.05$, $T = 1$, and $s = 16$. All values are computed based on 1,000 independent runs.

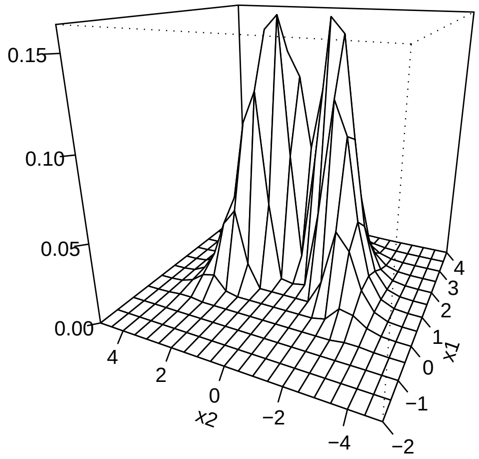

Figure 4.4: LBFP estimate of the optimal proposal for the multi-asset max option with strike $K = 150$. Model parameters: $S_i(0) = 100$ ($i = 1, 2$), $\sigma = 0.3$, $r = 0.05$, $T = 1$, and $|u| = 2$.

The first order Euler discretization yields

$$r_{t_{k+1}} = r_{t_k} + \kappa(\theta - r_{t_k})\Delta t + \sigma\sqrt{r_{t_k}} Z_{t_k},$$

with $Z_{t_k} \sim \mathcal{N}(0, 1)$ and $\Delta t = T/d$. The aim is to evaluate the price of an interest rate cap. It pays $(r_{t_k} - K)^+$ at time t_{k+1} ($k = 0, \ldots, d-1$) subject to strike K. The discounted payout is given by

$$\sum_{i=0}^{d-1} \exp[-\Delta t \sum_{j=0}^{i} r_{t_k}](r_{t_k} - K)^+.$$

The parameter values used in the simulations are $d = 16$, $r_0 = 0.07$, $\theta = 0.075$, $\kappa = 0.2$, $\sigma = 0.02$, $T = 1$ and 2, $K = 0.06$, 0.07, and 0.08. The results are reported in Table 4.9 and

Parameters			VR (RCE)				
N	s	ED	QMC	LSIS	NPIS	QLSIS	QNPIS
2^{11}	2	2	44 (68)	6 (1.8)	10 (0.6)	145 (33)	3,070 (228)
	3	3	26 (56)	3 (1.2)	0.3 (0.02)	52 (14)	72 (7)
	4	4	24 (41)	3 (1)	0.03 (0.002)	28 (7)	7 (0.5)
2^{12}	2	2	76 (136)	5 (1.9)	25 (1.6)	213 (91)	5,848 (620)
	3	3	42 (79)	3 (1.2)	0.3 (0.02)	72 (32)	148 (16)
	4	4	27 (45)	3 (1.1)	0.05 (0.002)	36 (16)	8 (0.7)
2^{13}	2	2	211 (391)	6 (2)	61 (4)	396 (270)	4.7 $\times 10^4$ (5,916)
	3	3	70 (119)	3 (1.1)	2 (0.08)	95 (65)	161 (20)
	4	4	35 (60)	3 (1)	0.01 (0.001)	42 (30)	7 (0.7)

Table 4.7: The table reports the variance reduction (VR) factors, the relative computational efficiency (RCE), and the estimated effective dimension (ED) for a multi-asset max option with strike $K = 150$. Model parameters: $S_i(0) = 100$ ($i = 1, \ldots, s$), $\sigma = 0.3$, $r = 0.05$, and $T = 1$. All values are computed based on 1,000 independent runs.

Parameters			VR (RCE)				
N	s	ED	QMC	LSIS	NPIS	QLSIS	QNPIS
2^{11}	2	2	8 (14)	- (-)	49 (3)	65 (17)	7,997 (652)
	3	3	4 (8)	2 (0.6)	0.2 (0.01)	17 (4)	165 (14)
	4	4	5 (8)	0.8 (0.3)	0.1 (0.006)	9 (2)	20 (1.5)
2^{12}	2	2	13 (23)	- (-)	163 (10)	82 (33)	1.8 $\times 10^4$ (1,837)
	3	3	7 (13)	4 (1.5)	3 (0.2)	22 (10)	259 (28)
	4	4	6 (9)	4 (1.3)	0.2 (0.01)	11 (4.7)	24 (2.2)
2^{13}	2	2	32 (58)	- (-)	304 (18)	95 (65)	1.1 $\times 10^5$ (1.4 $\times 10^4$)
	3	3	11 (19)	5 (1.7)	1.2 (0.06)	28 (19)	292 (36)
	4	4	8 (13)	4 (1.5)	0.05 (0.002)	13 (9.1)	27 (2.8)

Table 4.8: The table reports the variance reduction (VR) factors, the relative computational efficiency (RCE), and the estimated effective dimension (ED) for a multi-asset max option with strike $K = 200$. Model parameters: $S_i(0) = 100$ ($i = 1, \ldots, s$), $\sigma = 0.3$, $r = 0.05$, and $T = 1$. All values are computed based on 1,000 independent runs.

Table 4.10. Again the effective dimension is equal to one, which explains the good performance of NPIS/QNPIS. In particular, QNPIS strongly outperforms QLSIS for small strikes.

Parameters				VR (RCE)				
N	T	K	ED	QMC	LSIS	NPIS	QLSIS	QNPIS
2^{11}	1	.05	1	230 (231)	2 (0.4)	0.7 (0.4)	396 (58)	2,313 (814)
		.06	1	271 (280)	3 (0.6)	3 (1.4)	798 (119)	2,828 (1,021)
		.07	1	233 (236)	9 (1.8)	12 (6)	287 (42)	256 (91)
		.08	1	9 (9)	51 (10)	36 (19)	458 (67)	219 (78)
	2	.05	1	232 (235)	3 (0.5)	1.2 (0.6)	486 (71)	2,754 (977)
		.06	1	297 (298)	5 (1)	4 (2)	820 (120)	2,555 (905)
		.07	1	240 (247)	10 (1.9)	13 (7)	281 (42)	288 (104)
		.08	1	25 (25)	25 (5)	11 (6)	300 (44)	157 (56)
2^{12}	1	.05	1	479 (489)	2 (0.4)	1.1 (0.6)	820 (210)	5,235 (2,717)
		.06	1	582 (588)	3 (0.6)	4 (2)	1,621 (414)	4,961 (2,081)
		.07	1	415 (426)	9 (1.8)	13 (7)	388 (100)	332 (174)
		.08	1	15 (15)	49 (10)	43 (22)	588 (151)	283 (146)
	2	.05	1	484 (492)	2 (0.4)	1.9 (1)	1,007 (257)	5,723 (2,957)
		.06	1	626 (634)	5 (0.9)	6 (3)	1,377 (352)	4,182 (2,143)
		.07	1	422 (433)	9 (1.9)	14 (7)	375 (97)	360 (188)
		.08	1	44 (45)	26 (5)	18 (9)	374 (96)	214 (111)

Table 4.9: The table reports the variance reduction (VR) factors, the relative computational efficiency (RCE), and the estimated effective dimension (ED) for a cap within the CIR model. Model parameters: $r_0 = 0.07$, $\theta = 0.075$, $\kappa = 0.2$, $\sigma = 0.02$, and $d = 16$. All values are computed based on 1,000 independent runs.

Parameters				VR (RCE)				
N	T	K	ED	QMC	LSIS	NPIS	QLSIS	QNPIS
2^{11}	1	.05	1	238 (239)	3 (0.5)	0.9 (0.5)	451 (65)	2,924 (1,164)
		.06	1	284 (284)	4 (0.7)	3 (1.8)	940 (135)	4,225 (1.677)
		.07	1	263 (263)	10 (2)	15 (8)	414 (59)	335 (133)
		.08	1	9 (9)	48 (9)	30 (17)	536 (77)	336 (133)
	2	.05	1	240 (239)	3 (0.5)	1.4 (0.8)	552 (79)	3,760 (1,483)
		.06	1	309 (308)	7 (1.2)	5 (3)	1,013 (145)	3,345 (1,325)
		.07	1	270 (269)	11 (2)	15 (8)	410 (58)	402 (158)
		.08	1	28 (27)	25 (5)	21 (12)	411 (59)	162 (64)
2^{12}	1	.05	1	471 (472)	2 (0.4)	1.3 (0.7)	870 (218)	7,202 (4,101)
		.06	1	571 (571)	3 (0.6)	5 (3)	1,808 (453)	7,422 (4,219)
		.07	1	491 (491)	10 (2)	16 (9)	532 (133)	475 (271)
		.08	1	17 (17)	43 (8)	34 (19)	674 (168)	346 (196)
	2	.05	1	477 (474)	2 (0.4)	2 (1.2)	1,074 (266)	9,622 (5,442)
		.06	1	627 (624)	6 (1.1)	7 (4)	1,371 (342)	5,875 (3,328)
		.07	1	507 (503)	11 (2)	16 (9)	522 (130)	531 (299)
		.08	1	51 (51)	24 (5)	15 (9)	470 (117)	177 (100)

Table 4.10: The table reports the variance reduction (VR) factors, the relative computational efficiency (RCE), and the estimated effective dimension (ED) for a cap within the CIR model. Model parameters: $r_0 = 0.07$, $\theta = 0.075$, $\kappa = 0.2$, $\sigma = 0.02$, and $d = 64$. All values are computed based on 1,000 independent runs.

Chapter 5

Nonparametric Particle Filtering and Smoothing

5.1 Introduction

Let's consider the filtering and smoothing of the state variable $\mathbf{X}_t$ within a general state-space model which is given by the transition densities (2.7) and observation densities (2.8). As mentioned in Section 2.4, the basic particle particle suffers from weight degeneration which makes a resampling step necessary. However, resampling is problematic for at least three reasons: It leads to sample depletion (which means the particles' variety is reduced), it is time-consuming, and it causes additional variance. Therefore, it is worth to put some effort on the choice of a good proposal which can help to reduce the resampling frequency.

In this chapter we propose a nonparametric particle filter and a nonparametric particle smoother which are based on a sequential version of nonparametric importance sampling. The idea is to approximate the marginally optimal proposal nonparametrically. Typically, a nonparametrically constructed proposal can improve over parametric choices in low dimensions, because it is closer to the optimal proposal. As a consequence, the nonparametric particle filter and the nonparametric particle smoother provide better approximations of the distributions of interest compared with existing algorithms. A key feature is that they do not suffer from weight degeneration which makes resampling unnecessary. As a result of the nonparametric estimation, no analytical investigation of the problem at hand is required for identifying a suitable proposal. In addition, the nonparametric particle filter and the nonparametric particle smoother can be combined with quasi-Monte Carlo sampling which is not possible with standard particle filters and smoothers. Furthermore, it is shown that the quadratic costs of the likelihood approximation algorithm in Hürzeler and Künsch (2001) can be reduced by using nonparametric techniques. This gives a computationally efficient parameter estimation procedure. All algorithms proposed in this chapter have computational costs which are almost linear in the number of particles for low-dimensional state-spaces. This is achieved by the usage of the LBFP for nonparametric estimation. We emphasize that most existing particle smoothers as well as all existing sequential Monte Carlo methods which incorporate quasi-Monte Carlo have quadratic complexity. As an application of our methods, the filtering and smoothing of stochastic volatility models for multivariate high-frequency financial data is investigated.

5.2 A Nonparametric Particle Filter

Suppose the task is to approximate the filtering density $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ using importance sampling in the marginal space of $\mathbf{x}_t$. As a result of

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \propto p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = p(\mathbf{y}_t|\mathbf{x}_t)\int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})d\mathbf{x}_{t-1},$$

the proposal of the form $q(\mathbf{x}_t|\mathbf{y}_{1:t})$, that minimizes the variance of the importance weights

$$\omega(\mathbf{x}_t) \propto \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})}{q(\mathbf{x}_t|\mathbf{y}_{1:t})},$$

is the filtering density itself. The idea is to approximate the optimal proposal (that is the filtering density) nonparametrically. This approach has two advantages over existing techniques which are based on parametric proposals. First, at least in low dimensions, this gives a proposal which is closer to the optimal proposal. Second, this allows the use of quasi-Monte Carlo sampling (see Section 2.6).

We follow the ideas from Chapter 3 and utilize an LBFP for nonparametric density estimation. In the setting of this chapter, the (unnormalized) histogram estimator underlying the LBFP is computed based on weighted samples $\{\mathbf{z}^i, \omega^i\}_{i=1}^N$, that is we have

$$\tilde{f}^{\mathrm{H}}_{k_1,\ldots,k_d} = \frac{1}{Nh^d}\sum_{i=1}^{N}\omega^i \mathbf{1}_{\prod_{l=1}^d [t_{k_l}-h/2, t_{k_l}+h/2)}(\mathbf{z}^i).$$

For $\mathbf{z} \in \prod_{l=1}^d [t_{k_l}, t_{k_l}+h)$ the LBFP estimator is, analogous to (3.1), defined as $\hat{f}(\mathbf{z}) = \tilde{f}(\mathbf{z}) \times N/(\sum_{i=1}^N \omega^i)$, where

$$\tilde{f}(\mathbf{z}) = \sum_{j_1,\ldots,j_d \in \{0,1\}} \left[\prod_{l=1}^{d}\left(\frac{z_l - t_{k_l}}{h}\right)^{j_l}\left(1-\frac{z_l - t_{k_l}}{h}\right)^{1-j_l}\right]\tilde{f}^{\mathrm{H}}_{k_1+j_1,\ldots,k_d+j_d}.$$

As derived in Section 3.4.3 the LBFP has complexity of $\mathcal{O}(2^d d^2 N^{(d+5)/(d+4)})$ for N evaluations or the generation of N independent draws. All algorithms developed in this chapter have complexity $\mathcal{O}(2^d d^2 N^{(d+5)/(d+4)} \times T)$ for T time steps. This is close to linear for low-dimensional state-spaces. In the following, all densities assigned with a hat denote LBFP estimates. We first state the algorithm which in then discussed in detail.

Algorithm: Nonparametric Particle Filter (NPF)

Initialization: (for $t = 0$)

- For $i = 1, \ldots, N$: Sample $\mathbf{x}_0^i \sim p(\mathbf{x}_0)$ and set $\omega_0^i = 1$.

Nonparametric importance sampling: (for $t \geq 1$)

(i) For $i = 1, \ldots, N$: Sample $\breve{\mathbf{x}}_t^i \sim p(\mathbf{x}_t|\mathbf{x}_{t-1}^i)$.

(ii) Obtain LBFP estimate of the prediction density $\hat{p}(\mathbf{x}_t|\mathbf{y}_{1:t-1})$ based on $\{\breve{\mathbf{x}}_t^i, \omega_{t-1}^i\}_{i=1}^N$.

(iii) For $i = 1, \ldots, N$: Sample from proposal $\tilde{\mathbf{x}}_t^i \sim q(\mathbf{x}_t|\mathbf{y}_{1:t})$ and compute importance weights

$$\tilde{\omega}_t^i \propto \frac{p(\mathbf{y}_t|\tilde{\mathbf{x}}_t^i)\hat{p}(\tilde{\mathbf{x}}_t^i|\mathbf{y}_{1:t-1})}{q(\tilde{\mathbf{x}}_t^i|\mathbf{y}_{1:t})}.$$

(iv) Obtain LBFP estimate of the optimal proposal $\hat{p}(\mathbf{x}_t|\mathbf{y}_{1:t})$ using $\{\tilde{\mathbf{x}}_t^i, \tilde{\omega}_t^i\}_{i=1}^N$.

(v) For $i = 1, \ldots, N$: Sample $\mathbf{x}_t^i \sim \hat{p}(\mathbf{x}_t|\mathbf{y}_{1:t})$ and compute importance weights

$$\omega_t^i \propto \frac{p(\mathbf{y}_t|\mathbf{x}_t^i)\hat{p}(\mathbf{x}_t^i|\mathbf{y}_{1:t-1})}{\hat{p}(\mathbf{x}_t^i|\mathbf{y}_{1:t})}.$$

The output of the NPF consists of the particles $\{\mathbf{x}_t^i, \omega_t^i\}_{i=1}^N$ which approximate the filtering density $p(\mathbf{x}_t|\mathbf{y}_{1:t})$. We emphasize, that resampling is not required because no weight degeneration occurs. This is a result of the sampling from the proposal $\hat{p}(\mathbf{x}_t|\mathbf{y}_{1:t})$ which is close to optimal.

In the steps (i) and (ii), an estimate of the prediction density $\hat{p}(\mathbf{x}_t|\mathbf{y}_{1:t-1})$ is computed. It is required for the evaluation of the importance weights in steps (iii) and (v). In step (iii), auxiliary particles $\{\tilde{\mathbf{x}}_t^i, \tilde{\omega}_t^i\}_{i=1}^N$ are generated based on the proposal $q(\mathbf{x}_t|\mathbf{y}_{1:t})$. They are used to obtain the nonparametric estimate of the optimal proposal in step (iv). If the likelihood is not very peaked one can set $q(\mathbf{x}_t|\mathbf{y}_{1:t}) = \hat{p}(\mathbf{x}_t|\mathbf{y}_{1:t-1})$. In cases of a peaked likelihood a reasonable alternative is $q(\mathbf{x}_t|\mathbf{y}_{1:t}) = p(\mathbf{x}_t|\mathbf{y}_t)$. Then, step (iii) is related to the independent particle filter (Lin et al. 2005). The matching problem (which typically increases the complexity of the algorithm) discussed by Lin et al. (2005) is not an issue here because an approximation of the prediction density is available which can be evaluated in almost linear time.

In cases of peaked likelihood or severely nonlinear state transitions $\hat{p}(\mathbf{x}_t|\mathbf{y}_{1:t})$ obtained in step (iv) may only be a rough estimate of the filtering distribution. However, because $\hat{p}(\mathbf{x}_t|\mathbf{y}_{1:t})$ is just used as a proposal (and not as an approximation of the filtering distribution) it does not need to be a precise estimate. That is, it typically suffices if a few particles sampled from $q(\mathbf{x}_t|\mathbf{y}_{1:t})$ lie in the relevant domain of $p(\mathbf{x}_t|\mathbf{y}_{1:t})$. Note that the use of quasi-Monte Carlo sampling ensures that the space is well explored (compare Section 5.5). Summarizing, steps (i) through (iv) are basically carried out to obtain a nonparametric estimate of the optimal proposal. The actual approximation of the filtering distribution is done through importance sampling in step (v).

We emphasize that, in contrast to MCMC move steps (see, for instance, Gilks and Berzuini 2001), step (v) carries out importance sampling with proposal $\hat{p}(\mathbf{x}_t|\mathbf{y}_{1:t})$ and does not just add noise to increase the sample variety. In addition, our method is computationally much more efficient than MCMC steps (see Section 5.7.1).

5.3 A Nonparametric Particle Smoother

The posterior distribution $p(\mathbf{x}_{0:T}|\mathbf{y}_{1:T})$ can be decomposed as

$$p(\mathbf{x}_{0:T}|\mathbf{y}_{1:T}) = p(\mathbf{x}_T|\mathbf{y}_{1:T}) \prod_{t=0}^{T-1} p(\mathbf{x}_t|\mathbf{x}_{t+1:T}, \mathbf{y}_{1:T})$$

and the Markov property of the general state-space model implies

$$p(\mathbf{x}_t|\mathbf{x}_{t+1:T}, \mathbf{y}_{1:T}) \quad = \quad p(\mathbf{x}_t|\mathbf{x}_{t+1}, \mathbf{y}_{1:t}) \propto p(\mathbf{x}_t|\mathbf{y}_{1:t})p(\mathbf{x}_{t+1}|\mathbf{x}_t).$$

Based on this relation, a backward simulation particle smoother has been suggested (Godsill, Doucet, and West 2004). The basic idea is to apply a particle filter in order to obtain an approximation of $p(\mathbf{x}_t|\mathbf{y}_{1:t})$, $t = 1, \ldots, T$, and then proceed backwards in time. Unfortunately, this algorithm has quadratic complexity $\mathcal{O}(dN^2 \times T)$. We propose a backward simulation smoother with almost linear costs. It applies nonparametric importance sampling in the marginal space and makes use of the well-known smoothing formula

$$p(\mathbf{x}_t|\mathbf{y}_{1:T}) = p(\mathbf{x}_t|\mathbf{y}_{1:t}) \int \frac{p(\mathbf{x}_{t+1}|\mathbf{x}_t)p(\mathbf{x}_{t+1}|\mathbf{y}_{1:T})}{p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t})} d\mathbf{x}_{t+1}.$$

Let's define the density $\nu(\mathbf{x}_t)$ through

$$\nu(\mathbf{x}_t) \propto \int \frac{p(\mathbf{x}_{t+1}|\mathbf{x}_t)p(\mathbf{x}_{t+1}|\mathbf{y}_{1:T})}{p(\mathbf{x}_{t+1}|\mathbf{y}_{1:t})} d\mathbf{x}_{t+1},$$

where it is assumed that

$$\int p(\mathbf{x}_{t+1}|\mathbf{x}_t) d\mathbf{x}_t < \infty. \tag{5.1}$$

This is a weak assumption which is usually satisfied. For the state-space models considered in Section 5.7 it holds trivially. The case when it does not hold is discussed later. The idea of our algorithm is to approximate the smoothing density using the relation

$$p(\mathbf{x}_t|\mathbf{y}_{1:T}) \propto p(\mathbf{x}_t|\mathbf{y}_{1:t})\nu(\mathbf{x}_t),$$

and an LBFP estimate of $\nu(\mathbf{x}_t)$. The algorithm works as follows. First, the NPF is used to obtain particles that approximate the filtering densities. Second, the algorithm proceeds backwards in time using importance sampling based on an approximation of the marginally optimal proposal which is $p(\mathbf{x}_t|\mathbf{y}_{1:T})$. The LBFP estimates of both the optimal proposal and $\nu(\mathbf{x}_t)$ are computed based on the filtering particles from the NPF and suitably adjusted weights.

We emphasize that this is the first particle smoother with almost linear complexity which allows for quasi-Monte Carlo sampling.

```
Algorithm: Nonparametric Particle Smoother (NPS)
Filtering: (for $t = 1, \dots, T$) Generate filter particles $\{\mathbf{x}_t^i, \omega_t^i\}_{i=1}^N$ using the NPF.
    • For $i = 1, \dots, N$:   Set $\breve{\mathbf{x}}_T^i = \mathbf{x}_T^i$ and $\breve{\omega}_T^i = \omega_T^i$.
Backward simulation: (for $t = T-1, \dots, 0$)
    (i) For $i = 1, \dots, N$:   Compute weights $\tilde{\omega}_t^i \propto \omega_t^i \breve{\omega}_{t+1}^i p(\breve{\mathbf{x}}_{t+1}^i|\mathbf{x}_t^i)/\hat{p}(\breve{\mathbf{x}}_{t+1}^i|\mathbf{y}_{1:t})$.
    (ii) Obtain LBFP estimate of the optimal proposal $\hat{p}(\mathbf{x}_t|\mathbf{y}_{1:T})$ based on $\{\mathbf{x}_t^i, \tilde{\omega}_t^i\}_{i=1}^N$.
    (iii) Obtain LBFP estimate $\hat{\nu}(\mathbf{x}_t)$ based on $\{\mathbf{x}_t^i, \tilde{\omega}_t^i/(p(\mathbf{y}_t|\mathbf{x}_t^i)\hat{p}(\mathbf{x}_t^i|\mathbf{y}_{1:t-1}))\}_{i=1}^N$.
    (iv) For $i = 1, \dots, N$:   Sample $\breve{\mathbf{x}}_t^i \sim \hat{p}(\mathbf{x}_t|\mathbf{y}_{1:T})$ and compute importance weights
             $\breve{\omega}_t^i \propto \frac{p(\mathbf{y}_t|\breve{\mathbf{x}}_t^i)\hat{p}(\breve{\mathbf{x}}_t^i|\mathbf{y}_{1:t-1})\hat{\nu}(\breve{\mathbf{x}}_t^i)}{\hat{p}(\breve{\mathbf{x}}_t^i|\mathbf{y}_{1:T})}.$
```

The smoother's output consists of the sets of smoothing particles $\{\breve{\mathbf{x}}_t^i, \breve{\omega}_t^i\}_{i=1}^N$, which approximate the smoothing densities $p(\mathbf{x}_t|\mathbf{y}_{1:T})$, $t = 0, \dots, T$. Note, the NPS first runs the NPF. This implies that the LBFP estimates of the prediction densities $\hat{p}(\mathbf{x}_t|\mathbf{y}_{1:t-1})$ computed in the NPF can be reused. It is mentioned that the nonparametric importance sampling not only reduces the computational costs but also increases the variety and quality of the smoothing particles. Like the NPF, the NPS does not require resampling at any stage.

In the given form, the NPS cannot be applied if the assumption (5.1) does not hold, because then the density $\nu(\mathbf{x}_t)$ does not exist. However, the NPS can still be applied if the computation of the importance weights in step (iv) is replaced by

$$\breve{\omega}_t^i \propto \breve{\omega}_{t+1}^i \frac{p(\mathbf{y}_t|\breve{\mathbf{x}}_t^i)\hat{p}(\breve{\mathbf{x}}_t^i|\mathbf{y}_{1:t-1})p(\breve{\mathbf{x}}_{t+1}^i|\breve{\mathbf{x}}_t^i)}{\hat{p}(\breve{\mathbf{x}}_t^i|\mathbf{y}_{1:T})}.$$

Now, as a result of the single matching of $\breve{\mathbf{x}}_t^i$ with $\breve{\mathbf{x}}_{t+1}^i$, the weights $\breve{\omega}_t^i$ degenerate over time. This makes resampling necessary albeit not in every iteration.

5.4 On-Line Maximum Likelihood Parameter Estimation

Suppose the general state-space model depends on an unknown parameter vector $\theta \in \Theta$. The maximum likelihood estimator $\hat{\theta}$ maximizes the likelihood function

$$L(\theta) = p_\theta(\mathbf{y}_{1:T}) = \prod_{t=1}^{T} p_\theta(\mathbf{y}_t|\mathbf{y}_{1:t-1}).$$

Typically, the likelihood function cannot be computed analytically. Here, an approximation based on particles generated by a particle filter is discussed. In principle, $L(\theta)$ can be approximated pointwise through

$$L(\theta) = \prod_{t=1}^{T} \int p_\theta(\mathbf{y}_t|\mathbf{x}_t) p_\theta(\mathbf{x}_t|\mathbf{y}_{1:t-1}) d\mathbf{x}_t \approx \prod_{t=1}^{T} \left[\frac{1}{N} \sum_{i=1}^{N} p_\theta(\mathbf{y}_t|\breve{\mathbf{x}}_t^i) \right]$$

using prediction particles $\{\breve{\mathbf{x}}_t^i, 1/N\}_{i=1}^N$ which approximate $p(\mathbf{x}_t|\mathbf{y}_{1:t-1})$. For every parameter value θ new particles need to be generated. The major disadvantage of this approach is the independence of the Monte Carlo errors, which results in a non-smooth approximation. As discussed by Hürzeler and Künsch (2001), it is possible to obtain an approximation of the likelihood function for different θ based on a single set of particles, which are generated with respect to an initial parameter value θ_0. This gives a smooth approximation of the likelihood function. However, the algorithm proposed by Hürzeler and Künsch has quadratic complexity $\mathcal{O}(dN^2 \times T)$ making it inconvenient for practical applications. We propose a variant of this algorithm which has complexity $\mathcal{O}(2^d d^2 N^{(d+5)/(d+4)} \times T)$. The complexity reduction is achieved through the use of the LBFP estimator. The algorithm is based on

$$p_\theta(\mathbf{y}_t|\mathbf{y}_{1:t-1}) = \int p_\theta(\mathbf{y}_t|\mathbf{x}_t) \widetilde{\omega}_{t,\theta,\theta_0}(\mathbf{x}_t) p_{\theta_0}(\mathbf{x}_t|\mathbf{y}_{1:t-1}) d\mathbf{x}_t, \tag{5.2}$$

where

$$\begin{aligned} \widetilde{\omega}_{t,\theta,\theta_0}(\mathbf{x}_t) &= \frac{p_\theta(\mathbf{x}_t|\mathbf{y}_{1:t-1})}{p_{\theta_0}(\mathbf{x}_t|\mathbf{y}_{1:t-1})} & (5.3) \\ &= \frac{\int p_\theta(\mathbf{x}_t|\mathbf{x}_{t-1}) \omega_{t-1,\theta,\theta_0}(\mathbf{x}_{t-1}) p_{\theta_0}(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) d\mathbf{x}_{t-1}}{\int p_{\theta_0}(\mathbf{x}_t|\mathbf{x}_{t-1}) p_{\theta_0}(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) d\mathbf{x}_{t-1}} & (5.4) \end{aligned}$$

are the (parameter) prediction weights and

$$\omega_{t,\theta,\theta_0}(\mathbf{x}_t) = \frac{p_\theta(\mathbf{y}_t|\mathbf{x}_t)}{p_{\theta_0}(\mathbf{y}_t|\mathbf{x}_t)} \widetilde{\omega}_{t,\theta,\theta_0}(\mathbf{x}_t) \frac{p_{\theta_0}(\mathbf{y}_t|\mathbf{y}_{1:t-1})}{p_\theta(\mathbf{y}_t|\mathbf{y}_{1:t-1})}$$

the (parameter) filter weights. The algorithm of Hürzeler and Künsch uses (5.4) to compute the prediction weights. This causes the quadratic complexity, because for each evaluation of $\widetilde{\omega}_{t,\theta,\theta_0}(\mathbf{x}_t)$ the integrals in (5.4) need to be approximated. To make these evaluations computationally more efficient, we suggest that one uses LBFP estimates $\hat{p}_\theta(\mathbf{x}_t|\mathbf{y}_{1:t-1})$ and $\hat{p}_{\theta_0}(\mathbf{x}_t|\mathbf{y}_{1:t-1})$ to compute (5.3). Note, that $\hat{p}_{\theta_0}(\mathbf{x}_t|\mathbf{y}_{1:t-1})$ is also computed in the NPF. We state the NPF combined with the efficient maximum likelihood estimation procedure (NPF+ML). Note, the NPF is part of the NPS.

Algorithm: NPF with Maximum Likelihood Parameter Estimation (NPF+ML)

Initialization: (for $t = 0$)

- Choose initial parameter $\theta_0 \in \Theta$.
- For $i = 1, \dots, N$: Sample $\mathbf{x}_0^i \sim p(\mathbf{x}_0)$, set $\omega_0^i = 1$, and $\omega_{0,\theta,\theta_0}^i = 1$ for $\theta \in \Theta$.

Nonparametric importance sampling with likelihood approximation: (for $t = 1, \dots, T$)

- For $i = 1, \dots, N$: Sample $\breve{\mathbf{x}}_t^i \sim p_{\theta_0}(\mathbf{x}_t|\mathbf{x}_{t-1}^i)$.
- Obtain LBFP estimate $\hat{p}_{\theta_0}(\mathbf{x}_t|\mathbf{y}_{1:t-1})$ based on $\{\breve{\mathbf{x}}_t^i, \omega_{t-1}^i\}_{i=1}^N$.
- For $\theta \in \Theta \setminus \{\theta_0\}$: Obtain LBFP estimate $\hat{p}_\theta(\mathbf{x}_t|\mathbf{y}_{1:t-1})$ based on

$$\left\{ \breve{\mathbf{x}}_t^i, \frac{p_\theta(\breve{\mathbf{x}}_t^i|\mathbf{x}_{t-1}^i)\omega_{t-1,\theta,\theta_0}^i \omega_{t-1}^i}{p_{\theta_0}(\breve{\mathbf{x}}_t^i|\mathbf{x}_{t-1}^i)} \right\}_{i=1}^N .$$

- Use the NPF to generate particles $\{\mathbf{x}_t^i, \omega_t^i\}_{i=1}^N$ approximating $p_{\theta_0}(\mathbf{x}_t|\mathbf{y}_{1:t})$.
- For $\theta \in \Theta$ (beginning with θ_0):
  - For $i = 1, \dots, N$: Compute $\widetilde{\omega}_{t,\theta,\theta_0}^i = \hat{p}_\theta(\mathbf{x}_t^i|\mathbf{y}_{1:t-1}) / \hat{p}_{\theta_0}(\mathbf{x}_t^i|\mathbf{y}_{1:t-1})$.
  - Approximate $p_\theta(\mathbf{y}_t|\mathbf{y}_{1:t-1})$ through

$$a_{t,\theta} = \frac{1}{N} \sum_{i=1}^{N} \frac{p_\theta(\mathbf{y}_t|\mathbf{x}_t^i)\widetilde{\omega}_{t,\theta,\theta_0}^i \omega_t^i}{p_{\theta_0}(\mathbf{y}_t|\mathbf{x}_t^i)}. \tag{5.5}$$

  - For $i = 1, \dots, N$: Compute

$$\omega_{t,\theta,\theta_0}^i = \frac{p_\theta(\mathbf{y}_t|\mathbf{x}_t^i)}{p_{\theta_0}(\mathbf{y}_t|\mathbf{x}_t^i)} \widetilde{\omega}_{t,\theta,\theta_0}^i \frac{a_{t,\theta_0}}{a_{t,\theta}}.$$

  - Obtain new maximum likelihood estimate $\hat{\theta}_t = \text{argmax}_{\theta \in \Theta} \{\prod_{k=1}^t a_{k,\theta}\}$.

The approximation (5.5) follows from (5.2) and the observation that $\{\mathbf{x}_t^i, \omega_t^i / p_{\theta_0}(\mathbf{y}_t|\mathbf{x}_t^i)\}_{i=1}^N$ approximates $p_{\theta_0}(\mathbf{x}_t|\mathbf{y}_{1:t-1})$. The algorithm can be iterated with respect to θ_0 in order to improve the parameter estimate. We emphasize that our maximum likelihood procedure does not rely on the NPF. It can be combined with any other particle filter. Finally, it is mentioned that other parameter estimation techniques such as the EM algorithm (see Section 2.5) can also be combined with the NPF/NPS.

5.5 Quasi-Monte Carlo Sampling

As discussed in Section 2.6, quasi-Monte Carlo sampling is based on low-discrepancy sequences which are constructed to fill the space more evenly than (pseudo-) random numbers. Both the NPF and the NPS can be easily combined with quasi-Monte Carlo sampling. The quasi-Monte Carlo sampling has the advantage that even a small number of particles suffices to well represent the distribution given by the LBFP. It can be used in our algorithms whenever samples are drawn from an LBFP. This is a result of the fact that the inversion method (which is used to sample from an LBFP as explained in Section 3.4.2) preserves the structure of the low-discrepancy sequence. In order to avoid dependencies the original low-discrepancy sequence needs to be randomized, whenever the sampling distribution is changed. The random shift technique (see Section 2.6) was used in the simulations of this work.

Particle filters that incorporate quasi-Monte Carlo sampling were proposed earlier (Fearnhead 2005; Guo and Wang 2006). However, in contrast to the NPF, these particle filters have quadratic complexity. We emphasize that the computational costs of the NPF is not increased through usage of quasi-Monte Carlo.

5.6 Bin Width Selection

The major difficulty of applying nonparametric estimators lies in the selection of the smoothing parameter. For the LBFP, the bin width of the underlying histogram needs to be chosen. In the following, the theoretically optimal bin width is derived. Note, the results for the optimal bin width obtained in sections 3.2 and 3.3 do not apply. In addition, a Gaussian approximation is discussed which can be used in practice.

Optimal Bin Width

Suppose we have

$$f(\mathbf{z}) = \int g(\mathbf{z}, \tilde{\mathbf{z}}) d\tilde{\mathbf{z}} = \int \frac{g(\mathbf{z}, \tilde{\mathbf{z}})}{g_0(\mathbf{z}, \tilde{\mathbf{z}})} g_0(\mathbf{z}, \tilde{\mathbf{z}}) d\tilde{\mathbf{z}}$$

for some densities f, g, and g_0. The task is to obtain an LBFP estimate of f based on samples from the proposal density g_0, which are weighted proportional to g/g_0. Under the following assumptions, the optimal bin width h^* can be derived.

Assumption 1 f has three continuous and square integrable derivatives on supp(f).

Assumption 2 $\int\int g(\mathbf{z}, \tilde{\mathbf{z}})^2 / g_0(\mathbf{z}, \tilde{\mathbf{z}}) d\tilde{\mathbf{z}} d\mathbf{z}$ is finite on supp(f).

Assumption 3 As sample size $N \to \infty$, bin width h satisfies $h \to 0$ and $Nh^d \to \infty$.

Proposition 5.1. *Suppose that the assumptions 1 through 3 hold. Let* $\hat{f}_N$ *be the LBFP estimate (as defined in Appendix A.8) based on samples* $\{\mathbf{z}^i, \tilde{\mathbf{z}}^i\}_{i=1}^N$ *from* $g_0(\mathbf{z}, \tilde{\mathbf{z}})$ *and weights* $\omega^i \propto g(\mathbf{z}^i, \tilde{\mathbf{z}}^i)/g_0(\mathbf{z}^i, \tilde{\mathbf{z}}^i)$, $i = 1, \dots, N$. *Then we obtain*

$$\int \mathbf{E}[\hat{f}_N(\mathbf{z}) - f(\mathbf{z})]^2 d\mathbf{z} \quad = \quad \left\{ h^4 H_1 + \frac{1}{Nh^d} H_2 \right\} \times (1 + o(1))$$

and the optimal bin width

$$h^* = \left(\frac{dH_2}{4H_1} \right)^{\frac{1}{d+4}} N^{-\frac{1}{d+4}},$$

where

$$H_1 = \frac{49}{2880} \sum_{i=1}^d \int (\partial_i^2 f(\mathbf{z}))^2 d\mathbf{z} + \frac{1}{64} \sum_{i \neq j} \int \partial_i^2 f(\mathbf{z}) \partial_j^2 f(\mathbf{z}) d\mathbf{z}, \; H_2 = \frac{2^d}{3^d} \int \int \frac{g(\mathbf{z}, \tilde{\mathbf{z}})^2}{g_0(\mathbf{z}, \tilde{\mathbf{z}})} d\tilde{\mathbf{z}} d\mathbf{z}.$$

Proof. See Appendix A.8.

From this proposition, we immediately obtain the optimal bin widths for the LBFP estimates in the algorithms. Let

$$\tilde{p}(\mathbf{x}_t | \mathbf{y}_{1:t}) = \sum_{i=1}^N \omega_t^i \delta_{\mathbf{x}_t^i}(d\mathbf{x}_t) / \sum_{j=1}^N \omega_t^j$$

be the particle approximation of the filtering density with δ being the Dirac delta function. Conditional on $\tilde{p}(\mathbf{x}_{t-1} | \mathbf{y}_{1:t-1})$ and $\hat{p}(\mathbf{x}_{t-1} | \mathbf{y}_{1:t-1})$ one yields for the NPF

$$\begin{aligned} \hat{f}(\mathbf{x}_t) = \hat{p}(\mathbf{x}_t | \mathbf{y}_{1:t-1}): \quad & g(\mathbf{x}_t, \mathbf{x}_{t-1}) = p(\mathbf{x}_t | \mathbf{x}_{t-1}) \tilde{p}(\mathbf{x}_{t-1} | \mathbf{y}_{1:t-1}), \\ & g_0(\mathbf{x}_t, \mathbf{x}_{t-1}) = p(\mathbf{x}_t | \mathbf{x}_{t-1}) \hat{p}(\mathbf{x}_{t-1} | \mathbf{y}_{1:t-1}), \end{aligned}$$

and conditional on $\hat{p}(\mathbf{x}_t | \mathbf{y}_{1:t-1})$

$$\begin{aligned} \hat{f}(\mathbf{x}_t) = \hat{p}(\mathbf{x}_t | \mathbf{y}_{1:t}): \quad & g(\mathbf{x}_t) \propto p(\mathbf{y}_t | \mathbf{x}_t) \hat{p}(\mathbf{x}_t | \mathbf{y}_{1:t-1}), \\ & g_0(\mathbf{x}_t) = q(\mathbf{x}_t | \mathbf{y}_{1:t}). \end{aligned}$$

Conditional on $\tilde{p}_{\theta_0}(\mathbf{x}_{t-1} | \mathbf{y}_{1:t-1})$, $\hat{p}_{\theta_0}(\mathbf{x}_{t-1} | \mathbf{y}_{1:t-1})$, and $\omega_{t-1,\theta,\theta_0}(\mathbf{x}_{t-1})$ we obtain for the NPF+ML

$$\begin{aligned} \hat{f}(\mathbf{x}_t) = \hat{p}_\theta(\mathbf{x}_t | \mathbf{y}_{1:t-1}): \quad & g(\mathbf{x}_t, \mathbf{x}_{t-1}) = p_\theta(\mathbf{x}_t | \mathbf{x}_{t-1}) \omega_{t-1,\theta,\theta_0}(\mathbf{x}_{t-1}) \tilde{p}_{\theta_0}(\mathbf{x}_{t-1} | \mathbf{y}_{1:t-1}), \\ & g_0(\mathbf{x}_t, \mathbf{x}_{t-1}) = p_{\theta_0}(\mathbf{x}_t | \mathbf{x}_{t-1}) \omega_{t-1,\theta,\theta_0}(\mathbf{x}_{t-1}) \hat{p}_{\theta_0}(\mathbf{x}_{t-1} | \mathbf{y}_{1:t-1}), \end{aligned}$$

and conditional on $\tilde{p}(\mathbf{x}_t | \mathbf{y}_{1:t})$, $\tilde{p}(\mathbf{x}_{t+1} | \mathbf{y}_{1:T})$, $\hat{p}(\mathbf{x}_{t+1} | \mathbf{y}_{1:t})$, $\hat{p}(\mathbf{x}_t | \mathbf{y}_{1:t})$, and $\hat{p}(\mathbf{x}_{t+1} | \mathbf{y}_{1:T})$ we have for the NPS

$$\begin{aligned} \hat{f}(\mathbf{x}_t) = \hat{p}(\mathbf{x}_t | \mathbf{y}_{1:T}): \quad & g(\mathbf{x}_t, \mathbf{x}_{t+1}) \propto \tilde{p}(\mathbf{x}_t | \mathbf{y}_{1:t}) p(\mathbf{x}_{t+1} | \mathbf{x}_t) \tilde{p}(\mathbf{x}_{t+1} | \mathbf{y}_{1:T}) / \hat{p}(\mathbf{x}_{t+1} | \mathbf{y}_{1:t}), \\ & g_0(\mathbf{x}_t, \mathbf{x}_{t+1}) = \hat{p}(\mathbf{x}_t | \mathbf{y}_{1:t}) \hat{p}(\mathbf{x}_{t+1} | \mathbf{y}_{1:T}). \end{aligned}$$

Practical Bin Width Selection

To obtain a reasonable approximation of the optimal bin width h^*, estimates of the unknown constants H_1 and H_2 given in Proposition 5.1 are required. In the algorithms, samples $\{\mathbf{z}^i, \tilde{\mathbf{z}}^i\}_{i=1}^N$ from $g_0(\mathbf{z}, \tilde{\mathbf{z}})$ and weights $\omega^i \propto g(\mathbf{z}^i, \tilde{\mathbf{z}}^i)/g_0(\mathbf{z}^i, \tilde{\mathbf{z}}^i)$ are generated. Hence, an approximation of H_2 is conveniently given by

$$H_2 \approx \frac{2^d \sum_{i=1}^N (\omega^i)^2}{3^d \left(\sum_{i=1}^N \omega^i\right)^2}.$$

Constant H_1 is less tractable, because of its dependence on the second partial derivatives of density f. We suggest that one approximates f by means of a Gaussian distribution. Note, that H_1 does not depend on the location of f. Consequently, we can restrict to centered Gaussian distributions. Additionally, to ease estimation only diagonal covariance matrices are allowed. Based on the weighted samples, an estimator of the variance of the kth dimension is given by

$$\hat{\sigma}_k^2 = \sum_{i=1}^N \breve{\omega}^i (\mathbf{z}_k^i - \overline{\mathbf{z}}_k)^2,$$

with $\overline{\mathbf{z}}_k = \sum_{i=1}^N \breve{\omega}^i \mathbf{z}_k^i$, $\breve{\omega}^i = \omega^i / \sum_{j=1}^N \omega^j$, and $\mathbf{z}_k^i$ being the kth component of $\mathbf{z}^i$. For f being the Gaussian distribution $\mathcal{N}(\mathbf{0}, \operatorname{diag}(\hat{\sigma}_1^2, \ldots, \hat{\sigma}_d^2))$, it can be shown that

$$H_1 = \frac{1}{8d\pi^{d/2}} \left(\frac{49}{2880} \sum_{i=1}^d \frac{3}{\hat{\sigma}_i^5 \hat{\sigma}_{-i}} + \frac{1}{64} \sum_{i \neq j} \frac{1}{\hat{\sigma}_i^3 \hat{\sigma}_j^3 \hat{\sigma}_{-\{i,j\}}} \right),$$

where $\hat{\sigma}_{-i} = \prod_{j \neq i} \hat{\sigma}_j$.

5.7 Simulations

The coding of the algorithms is straightforward, given an implementation of the LBFP estimator. A detailed description of how to implement the LBFP estimator can be found in Section 3.4.2. The computations were carried out on a Dell Precision T3400, Intel CPU 2.83GHz. All algorithms were coded in C++ (see Chapter 8 for details). The Mersenne Twister 19937 (Matsumoto and Nishimura 1998) and the Sobol sequence (Sobol 1967) were used for pseudo- and quasi-random number generation, respectively.

Typically, the root mean square error of the estimated filtering/smoothing mean computed with respect to the “true” state (RMSE1) has been used for measuring the performance of filtering/smoothing algorithms. However, the RMSE1 does not converge to zero for increasing sample size, which makes it hard to interpret. It even may give misleading results. A better criterion is the root mean square error computed with respect to the mean of the “true” filtering/smoothing density (RMSE2). However, sequential Monte Carlo methods seek to approximate entire distributions, which is neither captured by the RMSE1 nor by the RMSE2. For the one-dimensional case, we suggest that one measures the difference between the target distribution and its particle approximation based on the squared distance of their cumulative distribution functions, which is given by

$$\mathcal{D} = \sum_{t=1}^{T} \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\mathbf{z}} p(\mathbf{x}_t|\mathbf{y}_{1:t}) d\mathbf{x}_t - \sum_{i=1}^{N} \omega_t^i \mathbf{x}_t^i \mathbf{1}_{(-\infty,\mathbf{z}]}(\mathbf{x}_t^i) / \sum_{j=1}^{N} \omega_t^j \right)^2 d\mathbf{z}$$

in the filtering setting.

5.7.1 Benchmark Model

Let's consider the general state-space model given by

$$\begin{aligned} \mathbf{X}_t|\mathbf{X}_{t-1} &\sim \mathcal{N}(0.5\mathbf{x}_{t-1} + 25\mathbf{x}_{t-1}/(1+\mathbf{x}_{t-1}^2) + 8\cos(1.2t), 10), \\ \mathbf{Y}_t|\mathbf{X}_t &\sim \mathcal{N}(\mathbf{x}_t^2/20, \sigma^2), \end{aligned}$$

and $\mathbf{X}_0 \sim \mathcal{N}(0, 10)$. This model is highly nonlinear with bimodal target densities. It has been studied extensively (for instance Kitagawa 1987; Doucet, Godsill, and Andrieu 2000; Godsill, Doucet, and West 2004). The two cases $\sigma^2 = 1$ and $\sigma^2 = 0.25^2$ are considered. In the second case the likelihood is rather peaked. The NPF and the NPF with quasi-Monte Carlo (NPF+QMC) are compared with the bootstrap particle filter with Metropolis-Hastings moves (SIRMH) (Gordon, Salmond, and Smith 1993; de Freitas et al. 2001) and the auxiliary particle filter (APF) proposed in Pitt and Shephard (1999). We use the version of the APF which is described by Fearnhead (2005). The NPF and NPF+QMC are applied with $q(\mathbf{x}_t|\mathbf{y}_{1:t}) = \hat{p}(\mathbf{x}_t|\mathbf{y}_{1:t-1})$. The NPS and the NPS with quasi-Monte Carlo (NPS+QMC) are tested against the backward simulation particle smoother (BSPS), proposed in Godsill, Doucet, and West (2004), and the simple particle smoother (SPS) (Kitagawa 1996; Fearnhead, Wyncoll, and Tawn 2008). Note, it can be easily verified that the assumption (5.1) holds for the present general state-space model. We produce 100 independent realizations of the model with $T = 100$ time steps. The filters and smoothers are applied with small and large sample sizes. The sample sizes are chosen such that all algorithms need approximately the same time for execution. To be able to compute the measures RMSE2 and $\mathcal{D}$, we approximate the filtering and smoothing densities with 50,000 particles using the bootstrap particle filter and the NPS, respectively.

The cumulative distribution functions of the filtering densities for several time steps are shown in Figure 5.1. The bimodality of some densities is clearly apparent. We can observe, that the NPF and the NPF+QMC approximate the filtering densities more closely than the APF.

In tables 5.1 and 5.2, the simulation results for the filters are reported. The NPF improves significantly over the APF and SIRMH for both values of σ^2 in terms of all criterions. In particular for the large sample sizes the NPF clearly performs better. The results for the measure $\mathcal{D}$ suggest that it approximates the filtering densities more closely. In addition, the quasi-Monte Carlo sampling further improves the gains of the NPF. Note, the sample sizes for the SIRMH are rather small. This is a result of the Metropolis-Hastings step being computationally very expensive.

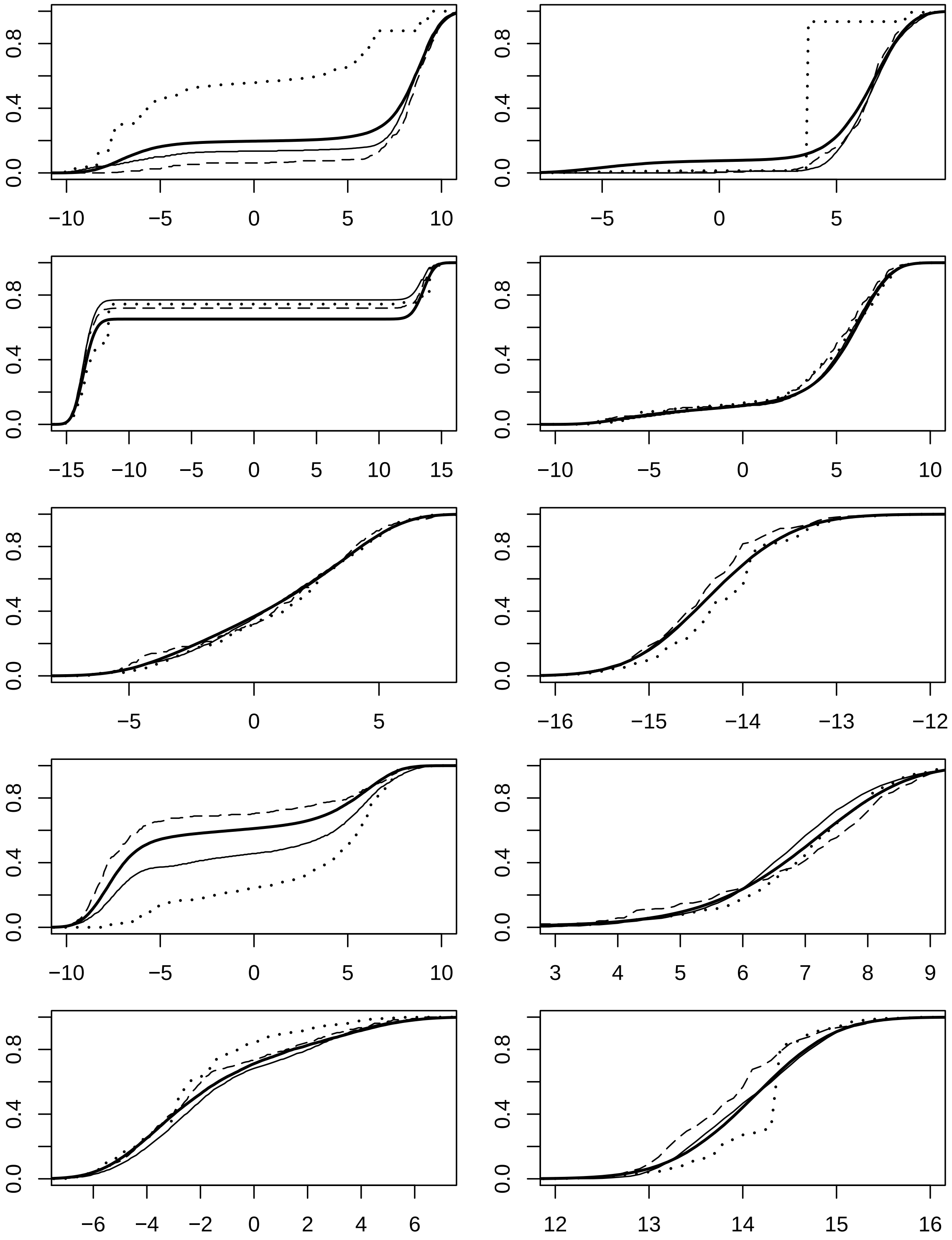

Figure 5.1: The estimated cumulative distribution functions of the filtering densities for times t = 10, 20, ..., 100 of one realization of the benchmark model. Filters: Bootstrap particle filter with 200,000 particles (heavy line); APF with 300 particles (dotted line); NPF with 250 particles (dashed line); NPF+QMC with 250 particles (solid line).

Algorithm	Sample Size N	RMSE1	RMSE2	$\mathcal{D}$	Time (sec)
APF	350	5.07	2.39	5.1×10^{-3}	0.34
SIRMH	175	4.77	1.50	2.4×10^{-3}	0.35
NPF	250	4.62	1.10	1.3×10^{-3}	0.35
NPF+QMC	250	4.64	0.92	1.1×10^{-3}	0.35
APF	8000	5.08	1.99	3.5×10^{-3}	7.98
SIRMH	1500	4.69	0.94	9.9×10^{-4}	9.31
NPF	6000	4.62	0.25	6.1×10^{-5}	8.77
NPF+QMC	6000	4.62	0.22	4.5×10^{-5}	8.83

Table 5.1: The simulation results for the benchmark model with 100 time steps and high observation noise ($\sigma^2 = 1$). Algorithms: The auxiliary particle filter (APF), the bootstrap filter with Metropolis-Hastings move step (SIRMH), the nonparametric PF (NPF), the nonparametric PF with quasi-Monte Carlo (NPF+QMC). The sample sizes were chosen such that all algorithms needed approximately the same time for execution. For the definitions of the measures RMSE1, RMSE2, and $\mathcal{D}$ see the text. All figures were computed/averaged over 100 independent runs.

Algorithm	Sample Size N	RMSE1	RMSE2	$\mathcal{D}$	Time (sec)
APF	350	6.25	4.51	1.7×10^{-2}	0.34
SIRMH	175	5.41	3.20	9.5×10^{-3}	0.36
NPF	250	4.97	2.31	5.5×10^{-3}	0.35
NPF+QMC	250	4.77	1.75	3.5×10^{-3}	0.35
APF	8000	5.70	3.90	1.3×10^{-2}	7.98
SIRMH	1500	4.44	1.52	2.4×10^{-3}	10.41
NPF	6000	4.26	0.42	2.9×10^{-4}	8.61
NPF+QMC	6000	4.26	0.40	2.5×10^{-4}	8.61

Table 5.2: The simulation results for the benchmark model with 100 time steps and low observation noise ($\sigma^2 = 0.25^2$). Algorithms: The auxiliary particle filter (APF), the bootstrap filter with Metropolis-Hastings move step (SIRMH), the nonparametric PF (NPF), the nonparametric PF with quasi-Monte Carlo (NPF+QMC). The sample sizes were chosen such that all algorithms needed approximately the same time for execution. For the definitions of the measures RMSE1, RMSE2, and $\mathcal{D}$ see the text. All figures were computed/averaged over 100 independent runs.

The results for the smoothers are given in the tables 5.3 and 5.4. It can be observed that the NPS and the NPS+QMC clearly outperform their competitors. The values for RMSE2 and $\mathcal{D}$ indicate that, in particular for the large sample sizes, the smoothing distributions are much more closely approximated. Note, the samples sizes of the smoothers differ significantly because the SPS and the BSPS have linear and quadratic costs, respectively.

Finally, we further investigate the computational costs of the NPF compared with the APF and the bootstrap particle filter by recording the execution times for different samples sizes (Figure 5.2). We can observe, that the times for the NPF grow superlinearly which agrees with the theoretical results for the costs of sampling from LBFPs. Surprisingly, for the univariate stochastic volatility model (see the following section) the NPF improves over the APF. This is explained by the following facts. The sampling and evaluation of an LBFP is based on simple arithmetic operations which are very cheap on computer systems. In contrast, the APF requires more frequent evaluations of the `exp` function which is very expensive.

Algorithm	Sample Size N	RMSE1	RMSE2	$\mathcal{D}$	Time (sec)
SPS	2000	2.26	1.55	1.0×10^{-2}	1.72
BSPS	70	2.83	1.93	7.2×10^{-3}	1.67
NPS	500	1.93	0.58	1.3×10^{-3}	1.66
NPS+QMC	500	1.85	0.45	0.8×10^{-3}	1.68
SPS	24000	2.05	0.80	5.4×10^{-3}	21.16
BSPS	250	2.01	1.13	2.1×10^{-3}	22.25
NPS	5000	1.68	0.12	5.5×10^{-5}	20.91
NPS+QMC	5000	1.68	0.11	4.9×10^{-5}	19.86

Table 5.3: The simulation results for the benchmark model with 100 time steps and high observation noise ($\sigma^2 = 1$). Algorithms: The simple particle smoother (SPS), the backward simulation PS (BSPS), the nonparametric PS (NPS), and the NPS with quasi-Monte Carlo (NPS+QMC). The sample sizes were chosen such that all algorithms needed approximately the same time for execution. For the definitions of the measures RMSE1, RMSE2, and $\mathcal{D}$ see the text. All figures were computed/averaged over 100 independent runs.

Algorithm	Sample Size N	RMSE1	RMSE2	$\mathcal{D}$	Time (sec)
SPS	2000	2.13	1.77	8.6×10^{-3}	1.70
BSPS	70	4.42	4.22	2.1×10^{-2}	1.62
NPS	500	1.73	1.07	3.1×10^{-3}	1.73
NPS+QMC	500	1.45	0.80	2.0×10^{-3}	1.75
SPS	24000	1.49	1.09	5.1×10^{-3}	21.09
BSPS	250	2.38	1.85	6.2×10^{-3}	22.02
NPS	5000	1.00	0.19	4.3×10^{-4}	21.85
NPS+QMC	5000	0.96	0.13	9.7×10^{-5}	21.36

Table 5.4: The simulation results for the benchmark model with 100 time steps and low observation noise ($\sigma^2 = 0.25^2$). Algorithms: The simple particle smoother (SPS), the backward simulation PS (BSPS), the nonparametric PS (NPS), and the NPS with quasi-Monte Carlo (NPS+QMC). The sample sizes were chosen such that all algorithms needed approximately the same time for execution. For the definitions of the measures RMSE1, RMSE2, and $\mathcal{D}$ see the text. All figures were computed/averaged over 100 independent runs.

5.7.2 High-Frequency Stochastic Volatility Application

The volatility of security prices is defined as the standard deviation of their first differences (which are known as returns). Here, different univariate and multivariate stochastic volatility (Jacquier, Polson, and Rossi 1994) models are applied to high-frequency transaction data. High-frequency volatility is a central quantity in risk management, trading, and derivative pricing. Because of their flexibility, stochastic volatility models became very popular as alternatives to GARCH models. Recently, several multivariate variants have been proposed (see Asai, McAleer, and Yu (2006) for an overview).

We extracted the transaction data of the symbols C (Citigroup) and JPM (JPMorgan Chase & Co) for the 5th September 2007 from the TAQ data base. To improve the data quality, we only use the transactions from the two major exchanges (NYSE and NASDAQ). The data are sampled at a frequency of 15 seconds giving a total number of 1560 returns for each stock (the exchanges open at 9:30 AM and close at 4 PM). By investigating the autocorrelations of both the returns and the absolute values of the returns, we find microstructure effects such as the bid-ask

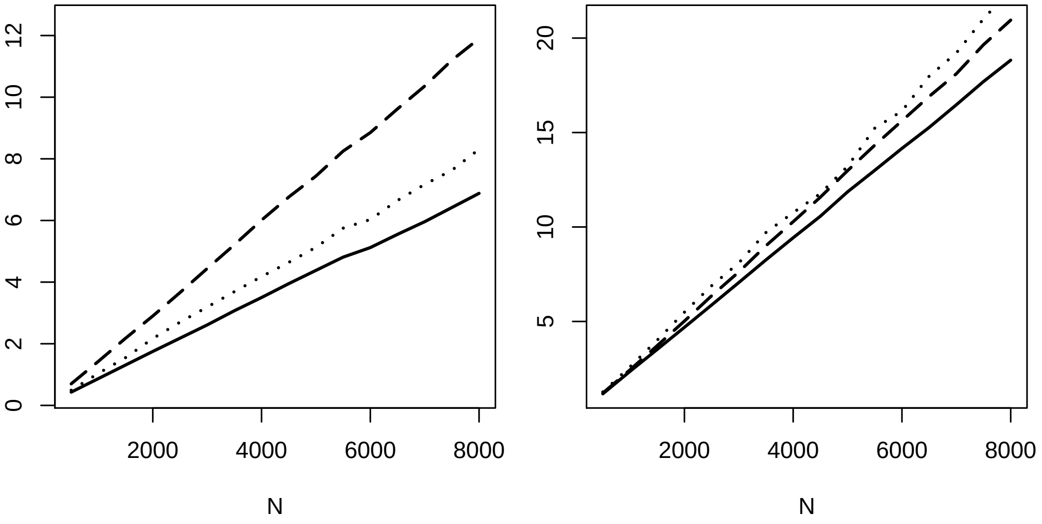

Figure 5.2: The execution times in seconds of the NPF (dashed line), the APF (dotted line), and the bootstrap filter (solid line) for different sample sizes N. Results are shown for the benchmark model (left) and for the univariate stochastic volatility model (right). For both models 100 time steps were used. All figures are averaged over 50 independent runs.

bounce to play a negligible role at this frequency. A rounding effect is the only microstructure feature that is present (compare upper plot in Figure 5.3).

First, the volatilities of both C and JPM are estimated separately within an univariate stochastic volatility model, which is given by

$$\begin{aligned} X_t|X_{t-1} &\sim \mathcal{N}(\phi x_{t-1}, \sigma^2), \\ Y_t|X_t &\sim \mathcal{N}(0, \beta^2 \exp(x_t)). \end{aligned}$$

The parameters ϕ and σ are estimated using the NPF+ML algorithm described in Section 5.4. As initial parameters we used $\theta_0 = (0.97, 0.18)^T$ following Pitt and Shephard (1999), who fitted the univariate stochastic volatility model to low-frequency data. Parameter β can be interpreted as the average volatility. Therefore, β is estimated from the data directly, as the empirical standard deviation of the returns. The parameter estimates are $(\phi, \sigma, \beta) = (0.93, 0.22, 0.0162)$ for C and $(\phi, \sigma, \beta) = (0.98, 0.11, 0.0174)$ for JPM.

The model is filtered and smoothed by the NPF and the NPS, respectively, using $N = 1{,}000$ particles. Figure 5.3 shows the data and results based on the estimated smoothing densities. It can be seen, that the stochastic volatilities of both stocks move together and that the confidence bounds of the smoothing densities have a broad common support. As a consequence, it seems reasonable to consider multivariate stochastic volatility models which allow that one studies stochastic volatility comovements. Especially for trading, on-line volatility estimation is of great importance. Therefore, the filtering and smoothing densities are compared with each other (Figure 5.4). While the confidence bounds for the filtering densities are wider, the filtering mean is close to the smoothing mean.

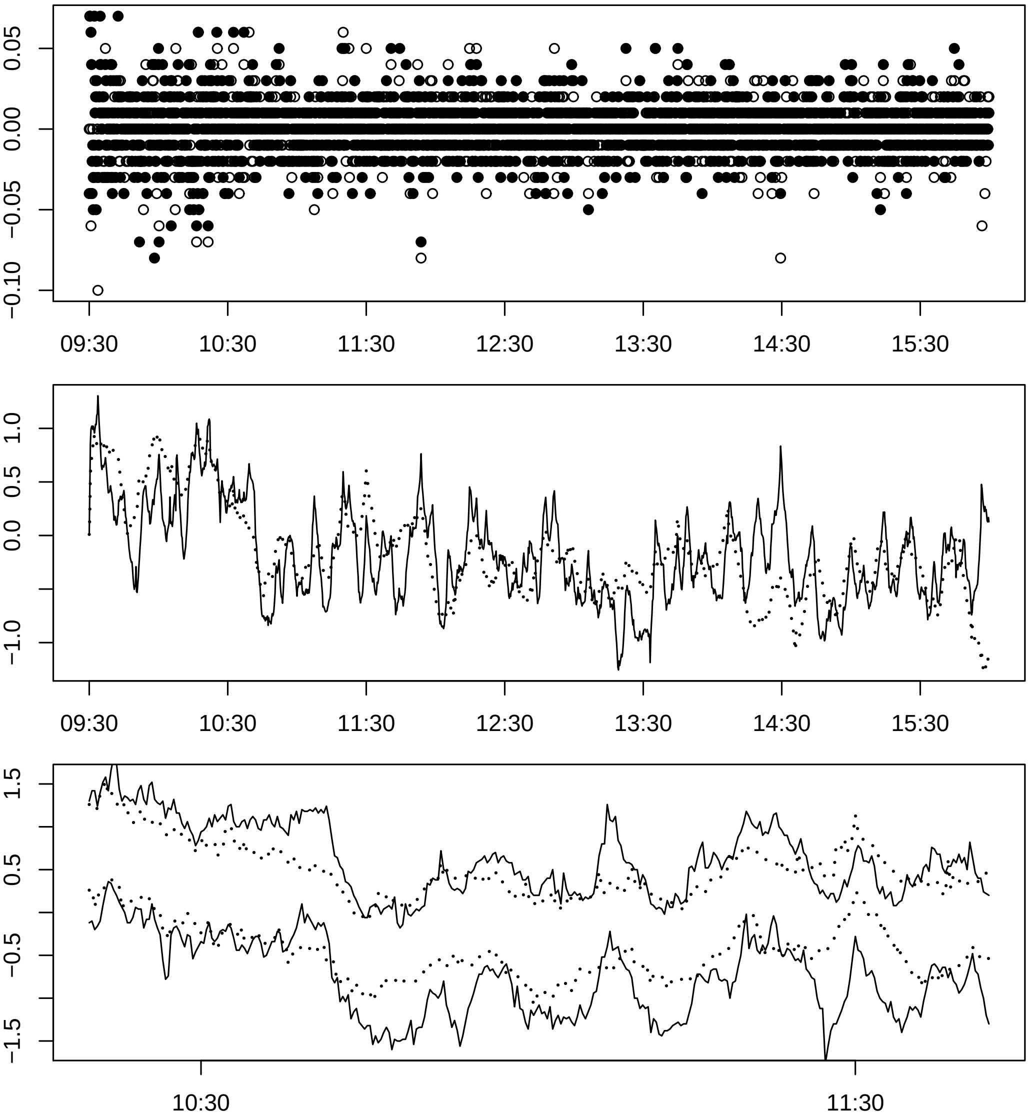

Figure 5.3: The returns of C (circles) and JPM (solid circles) sampled at a frequency of 15 seconds (upper plot). The means (middle plot) and the 95% confidence bounds (lower plot) of the stochastic volatility smoothing densities estimated within the univariate stochastic volatility model for C (solid line) and JPM (dotted line). Note, that the lower plot only shows a fraction of the trading day.

The first multivariate stochastic volatility model is a factor model (Aguilar and West 2000). It allows that one estimates a common volatility component and it is given by

$$\begin{aligned} X_t|X_{t-1} &\sim \mathcal{N}(\phi x_{t-1}, \sigma^2), \\ \mathbf{Y}_t|X_t &\sim \mathcal{N}(\mathbf{0}, H_t V), \end{aligned}$$

where $H_t = \mathrm{diag}\{\beta_1^2 \exp(x_t), \beta_2^2 \exp(x_t)\}$. V is the correlation matrix, that is $V_{11} = V_{22} = 1$ and $V_{12} = V_{21}$ equal to the correlation of the returns. From the data we computed $V_{12} = 0.491$ and β_1, β_2 as for the univariate stochastic volatility model. For the parameters we obtained

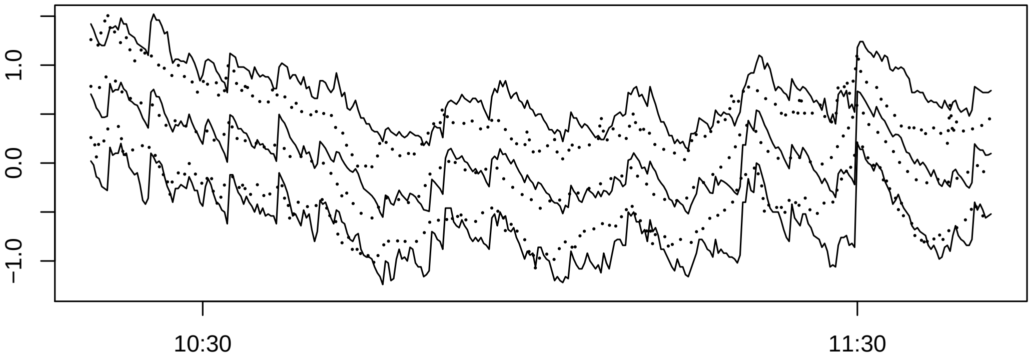

Figure 5.4: The means and the 95% confidence bounds of the stochastic volatility filtering (solid line) and the stochastic volatility smoothing densities (dotted line) for JPM. Note, the plot only shows a fraction of the trading day. The results for C are very similar.

$(\phi, \sigma) = (0.92, 0.24)$ using NPF+ML. An alternative multivariate stochastic volatility model is defined through

$$\begin{aligned} \mathbf{X}_t|\mathbf{X}_{t-1} &\sim \mathcal{N}(\Phi \mathbf{x}_{t-1}, \Sigma), \\ \mathbf{Y}_t|\mathbf{X}_t &\sim \mathcal{N}(\mathbf{0}, H_t V), \end{aligned}$$

where $H_t = \text{diag}\{\beta_1^2 \exp(\mathbf{x}_{1,t}), \beta_2^2 \exp(\mathbf{x}_{2,t})\}$ and $\mathbf{X}_t = (\mathbf{X}_{1,t}, \mathbf{X}_{2,t})^T$. Obviously, this model is more flexible than the factor model. However, it has the disadvantage of a larger number of parameters. To limit the number of parameters, we restricted Φ to be a diagonal matrix which is a common assumption. The parameter matrices Φ and Σ were estimated with the NPF+ML algorithm.

Figure 5.5 compares the stochastic volatility estimates of the three different models in terms of quantil-quantil plots and (kernel) density estimates of the normalized returns. First, note that the empirical distribution of the original returns (left plots) exhibit heavy tails. Second, one can observe that all three models provide normalized returns which have very similar empirical distributions. These distributions are close to the standard Gaussian distribution in the tails. However, at the origin they have a strange behaviour which is caused by the rounding feature of the original data. We emphasize that these results indicate that a single stochastic volatility factor suffices to capture the stochastic part of the volatilities of the two stocks.

Alternative stochastic volatility models that account for the discreteness of the price movements can be constructed. For instance, one can use $\mathcal{N}(0, \beta^2 \exp(x_t))$ rounded to the nearest cent, as observation model. However, in practice the presented stochastic volatility models should suffice, because of the fact that the normalized returns' empirical distributions have close-to-Gaussian tails. This allows that one constructs reliable confidence intervals (for the returns) which is a major application of stochastic volatility estimates.

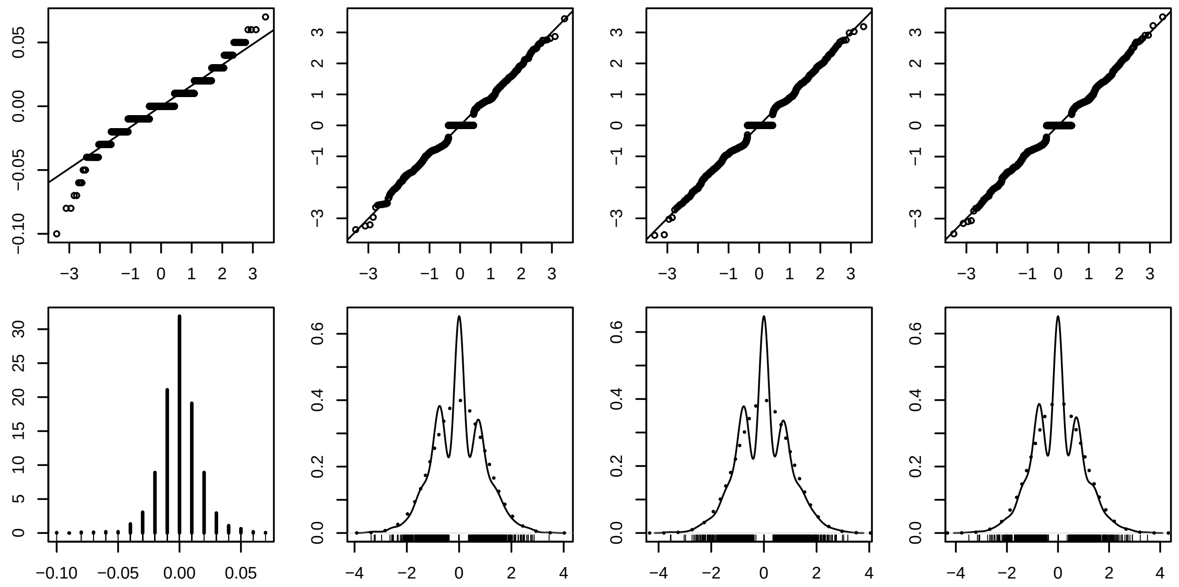

Figure 5.5: Quantil-quantil plots and density estimates for the returns of C (left) and the returns of C normalized with respect to the stochastic volatilities, which were estimated within the univariate, the factor, and the bivariate stochastic volatility model (from left to right). The results for JPM are very similar.

Chapter 6

Particle Filter-Based On-Line Estimation of Spot Cross-Volatility

6.1 Introduction

Nowadays, financial transaction data (tick-by-tick data) are widely available and modern computer systems allow tracking the trading process in real-time even in case of the transactions occurring on a millisecond basis. For high-frequency trading the on-line estimation of the spot cross-volatility (covariance matrix of the returns) based on tick-by-tick data is an important task. It is challenging because of the non-synchronous trading and the presence of market microstructure noise. The goal is to develop a new method which works on-line and updates the cross-volatility (covariance) estimate immediately when a new transaction comes in.

Until recently, the main focus in the literature has been on the estimation of the integrated (cross-)volatility. This task has been studied extensively under various assumptions on the market microstructure noise (Zhou 1996; Hayashi and Yoshida 2005; Zhang, Mykland, and Aït-Sahalia 2005; Andersen, Bollerslev, and Meddahi 2006; Bandi and Russell 2006, 2008; Hansen and Lunde 2006; Voev and Lunde 2007; Barndorff-Nielsen et al. 2008a, 2008b; Kalnina and Linton 2008; Robert and Rosenbaum 2008; Zhang 2008; Christensen, Podolskij, and Vetter 2009). Some authors suggested that estimates of the spot (cross-)volatility can be obtained through localized versions of estimators for the integrated (cross-)volatility (Foster and Nelson 1996; Fan and Wang 2008; Bos, Janus, and Koopman 2009; Kristensen 2009). In contrast to these existing methods which are essentially off-line procedures, our approach allows on-line estimation.

In this work, the efficient log-price processes of different securities are treated as latent states in a nonlinear state-space model with non-synchronously evolving components. The relation between the efficient prices and the transaction prices are described through a new market microstructure noise model. A new particle filter is developed which allows the estimation of the filtering distributions of the efficient log-prices given the observed transaction prices. Based on the filtering distributions the (time-varying) covariance matrices are estimated using a new sequential Expectation-Maximization (EM) type algorithm. The method is easy to implement and suitable for real-time application because of its computational efficiency.

The chapter is organized into three parts. In the first part (sections 6.2 through 6.5) the estimation of (univariate) spot volatility in the presence of market microstructure noise is considered. The second part (sections 6.6 through 6.8) generalizes this univariate method to the multivariate case of cross-volatility estimation where the non-synchronous trading times further complicate the situation. In the third part (sections 6.9 and 6.11) details on the implementation are given and empirical results for simulated and real data are presented followed by a discussion on our methods.

We model transaction data as noisy observations of a latent efficient log-price process X_t. It is assumed that transaction prices Y_{t_j} are observed at times $t_1 < t_2 < \ldots < t_T$. The evolution of the efficient log-price process is modeled by a random walk in transaction time with possibly time-varying volatility σ_{t_j}, that is

$$X_{t_j} = X_{t_{j-1}} + Z_{t_j} \tag{6.1}$$

with $Z_{t_j} \sim \mathcal{N}(0, \sigma^2_{t_j})$, or alternatively by a diffusion model in clock time – see Section 6.4. Drift terms are ignored because their effect is of lower order with high-frequency data.

We make a clear distinction between volatility per time unit and volatility per transaction and provide estimators for both. We start with a model in transaction time instead of clock time leading to an estimator of the spot volatility per transaction. In Section 6.4, a transformation from transaction time volatility to clock time volatility is given leading to a subsequent estimator of the volatility per time unit. In addition, we give a direct clock time estimator. In our opinion a model in transaction time has at least two advantages: First, the distribution of asset log-returns in a transaction time model can be modeled in most situations quite well by a Gaussian distribution, and second, volatility in transaction time is more constant than volatility in clock time making the algorithm more stable (Ané and Geman 2000; Plerou et al. 2001; Gabaix et al. 2003 - see also the discussion in sections 6.4, 6.10, and 6.11).

The relation between the efficient (log-)prices and the observed transaction prices is described through a general nonlinear market microstructure noise model which is completely different from the models considered so far. It depends on the (observed or unobserved) order book or market maker quotes and it can be expressed through a nonlinear equation

$$Y_{t_j} = g_{t_j}\big(\exp[X_{t_j}]\big) = g_{t_j;Y_{t_{1:j-1}}}\big(\exp[X_{t_j}]\big), \tag{6.2}$$

where the function g_{t_j} may also depend on past observations $Y_{t_{1:j-1}} := \{Y_{t_1}, \ldots, Y_{t_{j-1}}\}$ (see case 3 in Section 6.2). The function g_{t_j} is time-inhomogeneous and it can be interpreted as a generalized rounding function. The details of this model along with its economic motivation are given in Section 6.2.

The state equation (6.1) and the observation equation (6.2) form a nonlinear state-space model. The volatility is considered as a parameter of this state-space model. The estimation is done through a particle filter and a new sequential EM-type algorithm. Very roughly speaking our volatility estimator can be viewed as a localized realized volatility estimator based upon the particles of the particle filter. In detail the situation is however more complicated because we need a back and forth between particle filter and volatility estimator to obtain a decent on-line estimator. Bias improvements and an adaptive parameter choice complicate the situation even further.

In the second part of the chapter, the univariate model is extended to the case of multiple securities. That is, we consider efficient price processes

$$X_{t_j^{(s)},s} = X_{t_{j-1}^{(s)},s} + Z_{t_j^{(s)},s}, \qquad s = 1, \dots, S,$$

where the returns $Z_{t_j^{(s)},s}$ are correlated (for details see Section 6.6.1). In the multivariate case the non-synchronous trading becomes an issue. We propose a new transaction time model for non-synchronously trading securities which leads to a non-standard state-space model. For the estimation we develop a new particle filter which can cope with this non-standard state-space model.

We mention that our methods are not restricted to the above model but can also be applied with other microstructure noise models. Contrary to several other papers we do not assume that the transaction times are equidistant nor do we use interpolated prices.

6.2 A New Nonlinear Market Microstructure Noise Model

In most existing market microstructure models the efficient log-price is assumed to be corrupted by additive stationary noise (Aït-Sahalia, Mykland, and Zhang 2005; Zhang, Mykland, and Aït-Sahalia 2005; Hansen and Lunde 2006; Barndorff-Nielsen et al. 2008a). The noise variables are typically independent of the efficient log-price process. The major weakness of these models is the fact that they cannot reproduce the discreteness of the transaction prices. More adequate models which incorporate rounding noise have also been considered (Ball 1988; Large 2007; Li and Mykland 2007, 2008; Robert and Rosenbaum 2008). A popular model is based on additive noise followed by rounding according to the smallest tick size. A drawback of most existing models is the dependence on parameters and on distributional assumptions.

Now, a general market microstructure noise model is proposed which differs significantly from existing models. We are convinced that it is more suitable to explain microstructure features of real data. The model is based on the following simple assumption on the filtering distribution $p(\exp[x_{t_j}]|y_{t_1}, \dots, y_{t_j})$ of the unknown efficient price $\exp[X_{t_j}]$ given the observed transaction prices $Y_{t_1} = y_{t_1}, \dots, Y_{t_j} = y_{t_j}$.

Model assumption 1: The support A_{t_j} of the filtering distribution $p(\exp[x_{t_j}]|y_{t_1}, \dots, y_{t_j})$ is bounded and known.

It follows that the support of the filtering distribution of the efficient log-price $p(x_{t_j}|y_{t_1}, \dots, y_{t_j})$ is given by $\log A_{t_j}$.

This assumption is rather weak because we make no assumption on the distribution of Y_t at all. The clue is that given the model of the efficient log-price process (6.1) this assumption already leads to the identifiability of the distribution $p(x_{t_j}|y_{t_1}, \dots, y_{t_j})$ (see Proposition 6.1 below). It is shown later that this distribution can be approximated through a particle filter. A real data

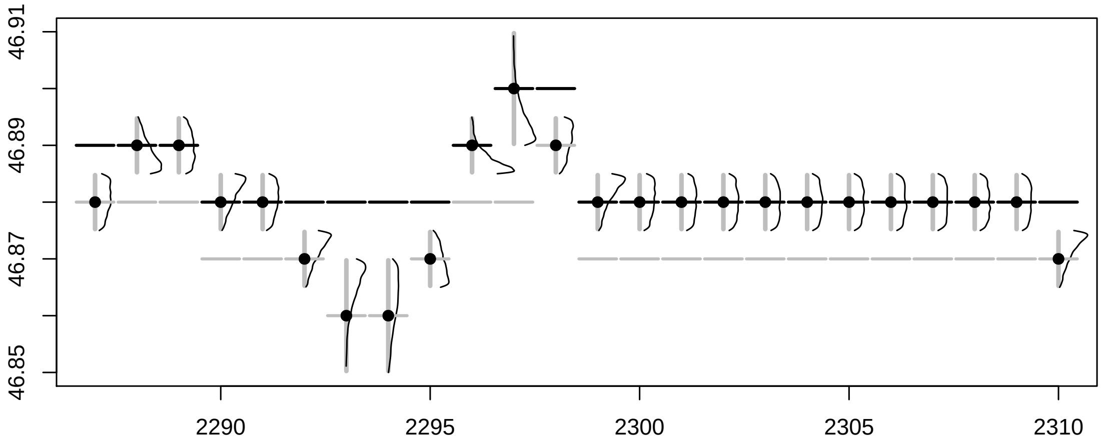

Figure 6.1: A real data example of estimated filtering distributions based on our market microstructure noise model for the case when market maker quotes are available in addition to the transaction data. The details are provided in Section 6.10.2. The plot shows some transaction prices (circles) along with kernel density estimates of the filtering distributions of the efficient prices (black lines) based on the particles produced by our particle filter. The gray vertical lines indicate the assumed support of the filtering distributions. The bid and ask market maker quotes are displayed by gray and black horizontal lines, respectively. The x-axis shows transaction time.

example is given in Figure 6.1. It shows the supports A_{t_j} (gray vertical lines) and kernel density estimates of the filtering distributions of the efficient prices (black lines) which are computed based on the output of the particle filter. In this example, market maker quotes are available (see case 2 below) which are indicated by gray and black horizontal lines. The details of this example are provided in Section 6.10.2.

The above model assumption is, for instance, fulfilled in the following three cases: In cases 1 and 2, limit order book data and market maker quotes are available, respectively, in addition to the transaction data leading to the support A_{t_j}. In case 3, only transaction data are available and a method to construct the A_{t_j} is suggested.

Case 1: (order book data)
Let's assume that at each transaction time t_j the exchange provides a limit order book with bid and ask levels given by $\alpha^k_{t_j}$ and $\beta^k_{t_j}$, $k = 1, 2, \ldots, K$, respectively. The order book levels satisfy $\alpha^K_{t_j} < \ldots < \alpha^2_{t_j} < \alpha^1_{t_j} < \beta^1_{t_j} < \beta^2_{t_j} < \ldots < \beta^K_{t_j}$ and we denote

$$\mathcal{M}_{t_j} = \{\alpha^K_{t_j}, \ldots, \alpha^2_{t_j}, \alpha^1_{t_j}, \beta^1_{t_j}, \beta^2_{t_j}, \ldots, \beta^K_{t_j}\}.$$

$\mathcal{M}_{t_j}$ represents the state of the order book immediately before the transaction at time t_j occurs. Clearly, $y_{t_j} \in \mathcal{M}_{t_j}$. The support of the filtering distribution at time t_j is defined through

$$A_{t_j} = \{x \in \mathbb{R} : \text{argmin}_{\gamma \in \mathcal{M}_{t_j}} |x - \gamma| = y_{t_j}\}.$$

Thus, the transaction price at time t_j is that price in the set $\mathcal{M}_{t_j}$ with the smallest Euclidean distance to the efficient price. Note, that A_{t_j} is simply an interval of the real line. The economic intuition behind this model is that the efficient price at time t_j should be closer to the observed price y_{t_j} than to any other order book level. Of course, this cannot be guaranteed. However, it

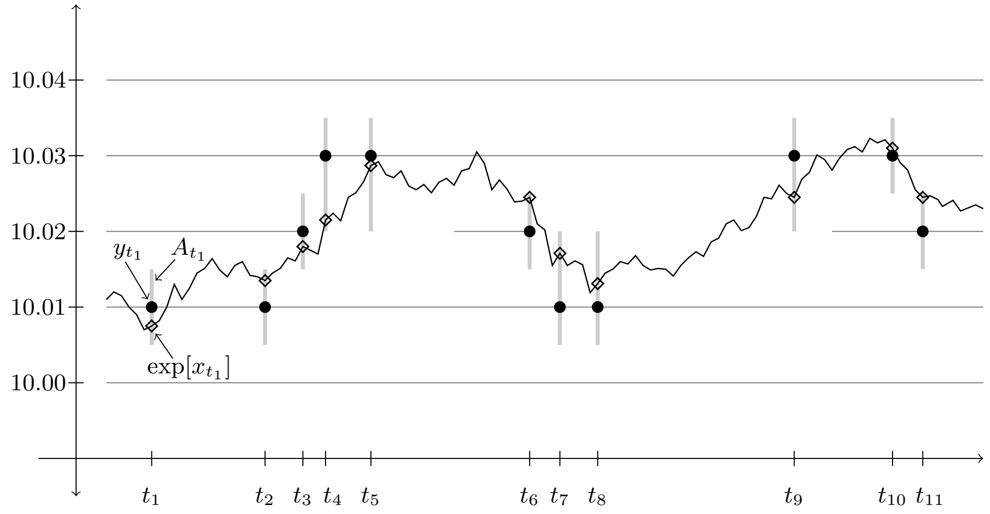

Figure 6.2: An example of our market microstructure noise model for the case when order book data are available. The figure shows the transaction prices (circles), the (in practice unknown) efficient prices in transaction time (diamonds), the latent efficient price process in clock time (black line), the order book levels (gray horizontal lines), and the supports of the filtering distributions of the efficient prices (gray vertical lines).

seems to be more realistic assumption than many other microstructure noise models leading at the same time to quite strong results.

An example of this market microstructure model is visualized in Figure 6.2. The supports of the filtering distributions are indicated by thick vertical lines. Observe that sometimes the bid-ask spread widens the support of the filtering distribution.

Case 2: (market maker quotes)
In the case where market maker quotes are available (instead of order book data), we only have a single bid and a single ask level α_{t_j} and β_{t_j}, respectively, which satisfy $\alpha_{t_j} < \beta_{t_j}$. That is, y_{t_j} is either equal to α_{t_j} or equal to β_{t_j}. The supports A_{t_j} are then defined through

$$A_{t_j} = [y_{t_j} - \Delta_{t_j}, y_{t_j} + \Delta_{t_j}),$$

where $\Delta_{t_j} = 0.5(\beta_{t_j} - \alpha_{t_j})$. The economic intuition given in case 1 applies similarly.

Case 3: (transaction data only)
For the case where no order book data or market maker quotes are available we now suggest a method for defining the supports of the filtering distributions solely based on the observed transaction prices. Conditional on $y_{t_1}, \dots, y_{t_j}$, we set

$$A_{t_j} = [y_{t_j} - \Delta_{t_j}, y_{t_j} + \Delta_{t_j}),$$

where

$$\Delta_{t_j} = \begin{cases} 0.5|y_{t_j} - y_{t_{j-1}}| & \text{if } y_{t_j} \neq y_{t_j-1}, \\ \Delta_{t_{j-1}} & \text{else.} \end{cases}$$

Note that Δ_{t_j} can be seen as an estimate of half the bid-ask spread at time t_j.

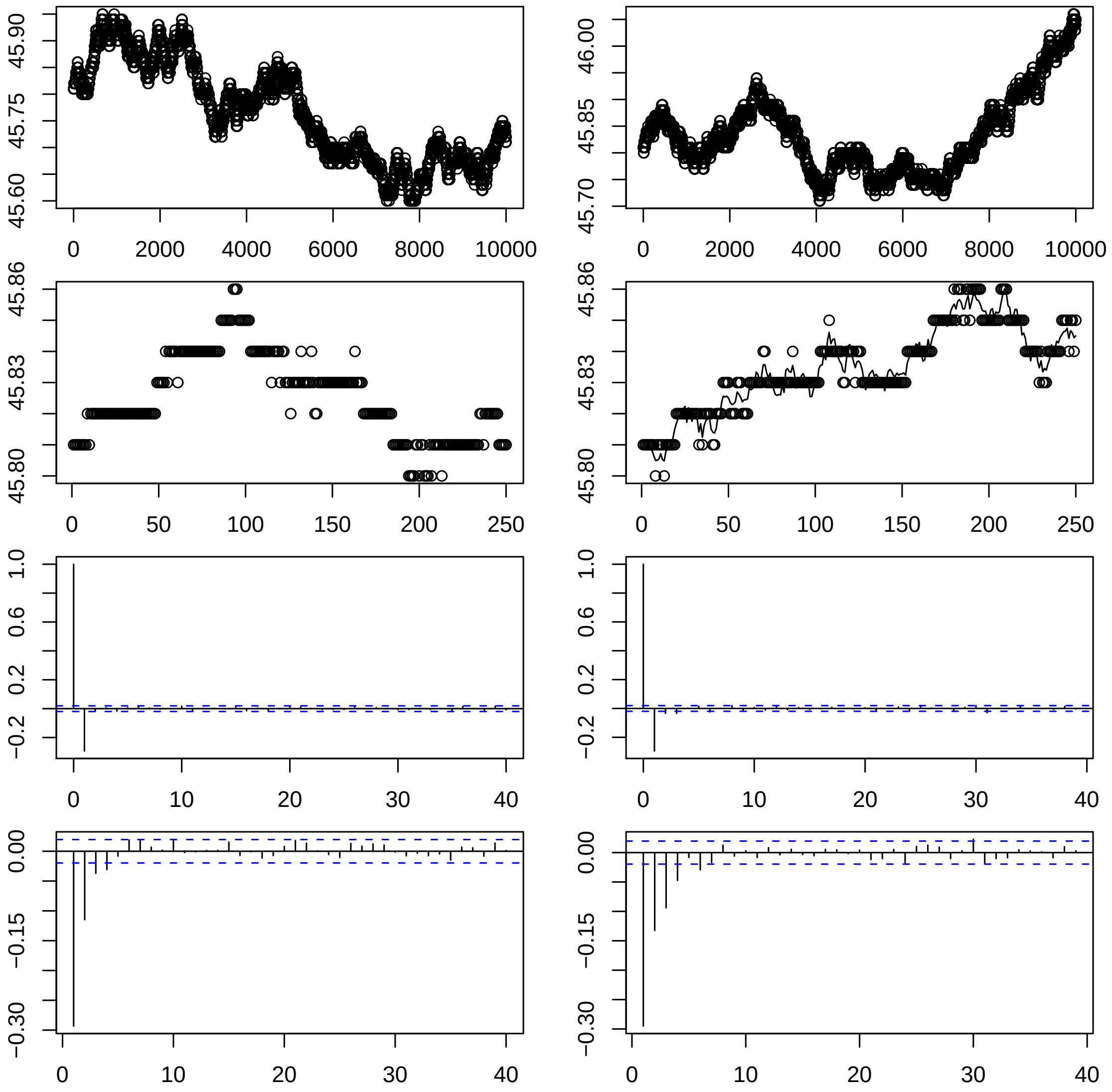

Figure 6.3: Comparison of real transaction data for Citigroup (left column) with simulated data from our market microstructure noise model (right column). The plots show (from top to bottom): 10,000 transaction prices; the first 250 transaction prices and the efficient price process of the simulated data; the autocorrelations and the partial autocorrelations of the returns of the transaction prices.

In practice, the intervals A_{t_j} will be similar for all three cases. Consequently, the estimation results will not differ much. It is mentioned that we do not need to explicitly specify the unknown nonlinear function g_{t_j} in the observation equation (6.2). The model assumption can be regarded as an assumption on the inverse mapping $g_{t_j}^{-1}$, namely $g_{t_j}^{-1}(y_{t_j}) = \{x|g_{t_j}(x) = y_{t_j}\} = A_{t_j}$ (conditional on $y_{t_1}, \dots, y_{t_{j-1}}$). That is, the observed price y_{t_j} determines the possible values of the associated efficient price.

We strongly believe that our model better describes the real world market microstructure than most existing models. Data simulated from our model reproduce the major stylized facts of high-frequency data, such as price discreteness and (first-order) negative autocorrelation of the returns.

Therefore, it seems to be an adequate model. As an example, transaction data of Citigroup are compared with data simulated from a special case of our model (see Figure 6.3). The figure shows the simulated efficient prices and the observations. The observations are the efficient prices rounded to the nearest cent (that is $\Delta_{t_j} \equiv 0.5$ cents). The efficient log-prices were generated according to (6.1) such that the observations have approximately the same volatility as the Citigroup data. We emphasize on the large number of zero returns. It is not surprising that the trajectories of the transaction processes look completely different. The important point, however, is the fact that our market microstructure noise model automatically introduces autocorrelations and partial autocorrelations of the returns which are very similar to those of the real Citigroup data.

We emphasize that our estimation method is not limited to this market microstructure noise model. It can be applied (after a suitable modification of the particle filter) to many microstructure noise models which comply with the general nonlinear observation equation

$$Y_{t_j} = g_{t_j}(X_{t_j}, U_{t_j}),$$

where g_{t_j} is a (nonlinear) function and U_{t_j} a noise variable. As mentioned earlier in this section a popular model describes market microstructure noise through additive noise followed by rounding. It is given by the equation

$$Y_{t_j} = \text{round}(\exp[X_{t_j} + U_{t_j}]), \tag{6.3}$$

where the U_{t_j} are, for instance, i.i.d. Gaussian distributed.

6.3 On-Line Estimation of Spot Volatility Based on a Particle Filter and Sequential EM-Type Algorithms

We now present on-line algorithms for the estimation of the spot volatility. Because all results also hold in the multivariate case with synchronous trading times we formulate this section for multivariate security prices. We are aware of the fact that the main challenge in the multivariate case are non-synchronous trading times. The present results are, however, the basis for the method for non-synchronous trading developed in the second part of the chapter.

We therefore consider in this section the estimation of the covariance matrix Σ_{t_j} which gives the volatilities of the individual efficient log-price processes $X_{t,s}$, $s = 1, \ldots, S$, as well as their cross-volatilities. The algorithms for the spot volatility are obtained by setting $\Sigma_{t_j} = \sigma^2_{t_j}$.

6.3.1 A Nonlinear State-Space Model

The multivariate version of the nonlinear state-space model (6.2) and (6.1) is given by

$$\begin{aligned} \mathbf{Y}_{t_j} &= g_{t_j}(\exp[\mathbf{X}_{t_j}]), && (6.4) \\ \mathbf{X}_{t_j} &= \mathbf{X}_{t_{j-1}} + \mathbf{Z}_{t_j}, && (6.5) \end{aligned}$$

where $\mathbf{X}_t = (X_{t,1}, \dots, X_{t,S})^T$, $g_{t_j}(\exp[\mathbf{X}_{t_j}]) = \big(g_{t_j^{(1)}}(\exp[X_{t_j,1}]), \dots, g_{t_j^{(S)}}(\exp[X_{t_j,S}])\big)^T$ with the $g_{t_j^{(s)}}$ possibly depending on $Y_{t_{1:j-1}}$, and $\mathbf{Z}_{t_j} \sim \mathcal{N}(\mathbf{0}, \Sigma_{t_j})$. The set $\mathbf{A}_{t_j}$ from Model assumption 1 usually is of the form $\mathbf{A}_{t_j} = A_{t_j,1} \times \cdots \times A_{t_j,S}$ with the $A_{t_j,s}$ being intervals (although this is not used). For simplicity we assume as an initial condition that given $Y_{t_1,s}$ the efficient prices $\exp[X_{t_1,s}]$ are uniformly distributed on $A_{t_1,s}$.

Model assumption 2: Σ_{t_j} is assumed to be either constant or slowly varying in time, that is we assume some smoothness for Σ_{t_j}.

The smoothness assumption needs not to be specified any further because we do not use it formally. However, without this assumption the estimation procedure developed in Section 6.3.3 would not make sense. A detailed specification of this assumption would become necessary if we tried to prove consistency (see Section 6.11).

We remark that (6.4) and (6.5) constitute a slightly generalized state-space model because the observations $\mathbf{Y}_{t_j}$ are not conditional independent of $\mathbf{Y}_{t_{1:j-1}}$ given $\mathbf{X}_{t_j}$ as in standard state-space models. This dependency on past observations is induced by our market microstructure noise model (see case 3 in Section 6.2). In the following section a particle filter is derived which can cope with this setting.

Our objective is the estimation of the covariance matrix Σ_{t_j} based on the observed prices $\mathbf{Y}_{t_{1:j}} = \mathbf{y}_{t_{1:j}}$. Because of the nonlinear market microstructure noise this is difficult. It is well known that crude estimators that ignore the noise lead to severely biased estimates (see, for instance, Voev and Lunde 2007). The idea of our estimation procedure is to approximate the conditional distribution of the efficient log-prices $\mathbf{X}_{t_j}$ given all observed transaction prices $\mathbf{y}_{t_{1:j}}$ up to time t_j (which is known as filtering). Based on this approximation a localized EM-type algorithm is used to construct an estimator of Σ_{t_j}. An efficient particle filter that allows the approximation of the target distributions is described in the following section.

6.3.2 An Efficient Particle Filter

A particle filters which approximates the posterior (joint filtering) distributions $p(\mathbf{x}_{t_{1:j}}|\mathbf{y}_{t_{1:j}})$ with clouds of particles $\{\mathbf{x}^i_{t_{1:j}}, \omega^i_{t_j}\}_{i=1}^N$ is developed. As a result of the violated conditional independence property mentioned earlier, the decomposition (2.11) does not hold for the present state-space model. Instead one obtains

$$p(\mathbf{x}_{t_{1:j}}|\mathbf{y}_{t_{1:j}}) = \frac{p(\mathbf{y}_{t_j}|\mathbf{y}_{t_{1:j-1}}, \mathbf{x}_{t_j})\, p(\mathbf{x}_{t_j}|\mathbf{x}_{t_{j-1}})}{p(\mathbf{y}_{t_j}|\mathbf{y}_{t_{1:j-1}})}\, p(\mathbf{x}_{t_{1:j-1}}|\mathbf{y}_{t_{1:j-1}}). \tag{6.6}$$

In contrast to standard state-space models $p(\mathbf{y}_{t_j}|\mathbf{y}_{t_{1:j-1}}, \mathbf{x}_{t_j})$ does not simplify to $p(\mathbf{y}_{t_j}|\mathbf{x}_{t_j})$.

As discussed in Section 2.4, the choice of the proposal is crucial to the filter's efficiency. In our framework it is possible to sample from the proposal $p(\mathbf{x}_{t_j}|\mathbf{y}_{t_{1:j}}, \mathbf{x}_{t_{j-1}})$ which is the optimal proposal in the sense that it minimizes the variance of the importance sampling weights (Doucet, Godsill, and Andrieu 2000). This gives the following algorithm:

Assume weighted particles $\{\mathbf{x}^i_{t_{1:j-1}}, \omega^i_{t_{j-1}}\}_{i=1}^N$ approximating $p(\mathbf{x}_{t_{1:j-1}}|\mathbf{y}_{t_{1:j-1}})$ are given; then

- For $i = 1, \dots, N$:
    - Sample $\mathbf{x}^i_{t_j} \sim p(\mathbf{x}_{t_j}|\mathbf{y}_{t_{1:j}}, \mathbf{x}^i_{t_{j-1}})$.
    - Compute importance weights

$$\breve{\omega}^i_{t_j} \propto \omega^i_{t_{j-1}} \frac{p(\mathbf{y}_{t_j}|\mathbf{y}_{t_{1:j-1}}, \mathbf{x}^i_{t_j})\, p(\mathbf{x}^i_{t_j}|\mathbf{x}^i_{t_{j-1}})}{p(\mathbf{x}^i_{t_j}|\mathbf{y}_{t_{1:j}}, \mathbf{x}^i_{t_{j-1}})} = \omega^i_{t_{j-1}}\, p(\mathbf{y}_{t_j}|\mathbf{y}_{t_{1:j-1}}, \mathbf{x}^i_{t_{j-1}}).$$

- For $i = 1, \dots, N$:
    - Normalize importance weights $\omega^i_{t_j} = \breve{\omega}^i_{t_j} / (\sum_{k=1}^N \breve{\omega}^k_{t_j})$.
- Obtain particles $\{\mathbf{x}^i_{t_{1:j}}, \omega^i_{t_j}\}_{i=1}^N$ which approximate $p(\mathbf{x}_{t_{1:j}}|\mathbf{y}_{t_{1:j}})$.

In addition, a resampling step needs to be introduced to resolve the problem of weight degeneracy. Because resampling is time consuming, it is carried out only if the effective sample size is below some threshold (see Section 2.4 for details).

To apply the particle filter to the state-space model given by (6.4) and (6.5) it is necessary to specify the optimal proposal and the computation of the importance weights. The following result shows that both take a very simple form. Furthermore, it gives the uniqueness of the joint filtering distribution $p(\mathbf{x}_{t_{1:j}}|\mathbf{y}_{t_{1:j}})$. This implies that in our microstructure noise model the knowledge of the support $\mathbf{A}_{t_j}$ of $p(\exp[\mathbf{x}_{t_j}]|\mathbf{y}_{t_1}, \dots, \mathbf{y}_{t_j})$ already is sufficient for the identifiability of the efficient (log-)price distribution conditional on the observations.

Proposition 6.1. *The joint filtering distribution $p(\mathbf{x}_{t_{1:j}}|\mathbf{y}_{t_{1:j}})$ is uniquely determined by the supports $\log \mathbf{A}_{t_k}$ of the filtering distributions $p(\mathbf{x}_{t_k}|\mathbf{y}_{t_1}, \dots, \mathbf{y}_{t_k})$, $k = 1, \dots, j$. The optimal proposal is a truncated multivariate normal distribution given by*

$$p(\mathbf{x}_{t_j}|\mathbf{y}_{t_{1:j}}, \mathbf{x}_{t_{j-1}}) \propto \mathcal{N}(\mathbf{x}_{t_j}|\mathbf{x}_{t_{j-1}}; \Sigma_{t_j})\big|_{\log \mathbf{A}_{t_j}}$$

with $\log \mathbf{A}_{t_j} = \log A_{t_j,1} \times \cdots \times \log A_{t_j,S}$ and the importance weights can be computed through

$$\breve{\omega}^i_{t_j} \propto \omega^i_{t_{j-1}} \int_{\log \mathbf{A}_{t_j}} \mathcal{N}(\mathbf{x}_{t_j}|\mathbf{x}^i_{t_{j-1}}; \Sigma_{t_j})\, d\mathbf{x}_{t_j}. \tag{6.7}$$

Proof. See Appendix A.9.

Remark: For the market microstructure noise model (6.3) the optimal proposal cannot be computed easily. In this case we propose to modify the particle filter as follows. Assume that $\mathbf{U}_{t_j}$ is i.i.d. $\mathcal{N}(0, \Sigma^U)$ distributed. As proposal we use

$$p(\mathbf{x}_{t_j}|\mathbf{x}_{t_{j-1}}) = \mathcal{N}(\mathbf{x}_{t_j}|\mathbf{x}_{t_{j-1}}; \Sigma_{t_j})$$

which gives the importance weights

$$\breve{\omega}^i_{t_j} \propto \omega^i_{t_{j-1}} p(\mathbf{y}_{t_j}|\mathbf{y}_{t_{1:j-1}}, \mathbf{x}^i_{t_j}) = \omega^i_{t_{j-1}} \int_{\log \mathbf{A}_{t_j}} \mathcal{N}(\mathbf{y}|\mathbf{x}^i_{t_j}; \Sigma^U) d\mathbf{y}.$$

6.3.3 Sequential EM-Type Algorithms

In this section, we discuss the estimation of Σ_{t_j} in the time-constant and time-varying case.

A stochastic EM algorithm can be used to obtain the maximum likelihood estimator in the time-constant case $\Sigma_{t_j} = \Sigma$ (compare Section 2.5). The EM algorithm maximizes the likelihood $p_\Sigma(\mathbf{y}_{t_{1:T}})$ by iteratively carrying out an E-step and an M-step. In the E-step, the expectation

$$\begin{aligned}\mathcal{Q}(\Sigma|\hat{\Sigma}^{(m)}) &= \mathbf{E}_{\hat{\Sigma}^{(m)}}\big[\log p_\Sigma(\mathbf{X}_{t_{1:T}}, \mathbf{y}_{t_{1:T}})|\mathbf{y}_{t_{1:T}}\big] \\ &= \sum_{j=1}^{T} \mathbf{E}_{\hat{\Sigma}^{(m)}}\big[\log p(\mathbf{y}_{t_j}|\mathbf{y}_{t_{1:j-1}}, \mathbf{X}_{t_j})|\mathbf{y}_{t_{1:T}}\big] + \mathbf{E}_{\hat{\Sigma}^{(m)}}\big[\log p(\mathbf{X}_{t_1})|\mathbf{y}_{t_{1:T}}\big] \\ &\quad + \sum_{j=2}^{T} \mathbf{E}_{\hat{\Sigma}^{(m)}}\big[\log p_\Sigma(\mathbf{X}_{t_j}|\mathbf{X}_{t_{j-1}})|\mathbf{y}_{t_{1:T}}\big] \end{aligned} \tag{6.8}$$

needs to be approximated, where $\hat{\Sigma}^{(m)}$ is the current estimator. Note, it is sufficient to consider the sum in (6.8) because the random variables $\log p(\mathbf{y}_{t_j}|\mathbf{y}_{t_{1:j-1}}, \mathbf{X}_{t_j})$ and $p(\mathbf{X}_{t_1})$ do not depend on Σ. In the M-step, a new parameter estimate $\hat{\Sigma}^{(m+1)}$ is obtained by maximizing $\mathcal{Q}(\Sigma|\hat{\Sigma}^{(m)})$.

If Σ_{t_j} is time-varying some regularization is needed. For example $\hat{\Sigma}_{t_j}^{(m+1)}$ can be obtained by maximizing some localized version of (6.8), e.g.

$$\mathcal{Q}_{t_j}(\Sigma|\hat{\Sigma}_{t_{1:T}}^{(m)}) = \frac{1}{T}\sum_{k=j-T}^{j-2} \frac{1}{b}K\Big(\frac{k}{bT}\Big)\, \mathbf{E}_{\hat{\Sigma}_{t_{1:T}}^{(m)}}\big[\log p_\Sigma(\mathbf{X}_{t_{j-k}}|\mathbf{X}_{t_{j-k-1}})|\mathbf{y}_{t_{1:T}}\big] \tag{6.9}$$

with a kernel $K(\cdot)$ and a bandwidth b.

An approximation of $\mathcal{Q}(\Sigma|\hat{\Sigma}^{(m)})$ and $\mathcal{Q}_{t_j}(\Sigma|\hat{\Sigma}_{t_{1:T}}^{(m)})$ can be computed based on the smoothing particles

$$\{\mathbf{x}_{t_{1:T}}^i, \omega_{t_T}^i\}_{i=1}^N$$

from our particle filter or (with higher precision) from existing particle smoothing algorithms (Godsill, Doucet, and West 2004; Neddermeyer 2010b; Briers, Doucet, and Maskell 2010). The smoothing particles give the approximation

$$\begin{aligned}&\mathbf{E}_{\hat{\Sigma}_{t_{1:T}}}[\log p_\Sigma(\mathbf{X}_{t_{j-k}}|\mathbf{X}_{t_{j-k-1}})|\mathbf{y}_{t_{1:T}}] \\ &\qquad \approx \sum_{i=1}^{N} \omega_{t_T}^i \frac{1}{2}\Big[S\log 2\pi + \log|\Sigma| + \operatorname{tr}\Big\{\Sigma^{-1}\big(\mathbf{x}_{t_{j-k}}^i - \mathbf{x}_{t_{j-k-1}}^i\big)\big(\mathbf{x}_{t_{j-k}}^i - \mathbf{x}_{t_{j-k-1}}^i\big)^T\Big\}\Big] \end{aligned} \tag{6.10}$$

which leads, with

$$\breve{\Sigma}_{t_j}(\omega_{t_T}) := \sum_{i=1}^{N} \omega_{t_T}^i\big(\mathbf{x}_{t_j}^i - \mathbf{x}_{t_{j-1}}^i\big)\big(\mathbf{x}_{t_j}^i - \mathbf{x}_{t_{j-1}}^i\big)^T, \tag{6.11}$$

to the maximizers

$$\hat{\Sigma}^{(m+1)} = \frac{1}{T-1}\sum_{j=2}^{T} \breve{\Sigma}_{t_j}(\omega_{t_T}) \tag{6.12}$$

and

$$\hat{\Sigma}_{t_j}^{(m+1)} = \Big[\sum_k K\Big(\frac{k}{bT}\Big)\Big]^{-1} \sum_k K\Big(\frac{k}{bT}\Big)\, \breve{\Sigma}_{t_{j-k}}(\omega_{t_T}) \tag{6.13}$$

of (6.8) and (6.9), respectively (note that the particles and, therefore, also $\breve{\Sigma}$ depend on m.)

Instead of these estimates, one will prefer in most situations an on-line algorithm which updates the estimates when a new observation comes in. This requires on the one hand the use of filtering particles instead of smoothing particles and on the other hand an integration of the E-step into the algorithm.

We now develop such an algorithm step-by-step. Note that the recursion developed in 1) below is not an on-line algorithm. It is just discussed to demonstrate the relation of the on-line algorithms in (6.20) and (6.21) to the estimates (6.12) and (6.13), respectively. Note, in the following steps the notation $\hat{\Sigma}_{t_j}$ is used for different estimates.

1) A "recursive" solution for the above situation (both for time-constant and time-varying Σ_{t_j}) is

$$\mathcal{Q}_{t_j}(\Sigma|\hat{\Sigma}_{t_{1:T}}) := \{1-\lambda_j\}\, \mathcal{Q}_{t_{j-1}}(\Sigma|\hat{\Sigma}_{t_{1:T}}) + \lambda_j\, \mathbf{E}_{\hat{\Sigma}_{t_{1:T}}}\big[\log p_\Sigma(\mathbf{X}_{t_j}|\mathbf{X}_{t_{j-1}})|\mathbf{y}_{t_{1:T}}\big] \tag{6.14}$$

with $\mathcal{Q}_{t_2}(\Sigma|\hat{\Sigma}_{t_{1:T}}) = \mathbf{E}_{\hat{\Sigma}_{t_{1:T}}}\big[\log p_\Sigma(\mathbf{X}_{t_2}|\mathbf{X}_{t_1})|\mathbf{y}_{t_{1:T}}\big]$ leading to

$$\begin{aligned}\mathcal{Q}_{t_j}(\Sigma|\hat{\Sigma}_{t_{1:T}}) = {} & \sum_{k=0}^{j-3}\Big[\prod_{\ell=0}^{k-1}(1-\lambda_{j-\ell})\Big]\lambda_{j-k}\, \mathbf{E}_{\hat{\Sigma}_{t_{1:T}}}\big[\log p_\Sigma(\mathbf{X}_{t_{j-k}}|\mathbf{X}_{t_{j-k-1}})|\mathbf{y}_{t_{1:T}}\big] \\ & + \Big[\prod_{\ell=0}^{j-3}(1-\lambda_{j-\ell})\Big]\, \mathbf{E}_{\hat{\Sigma}_{t_{1:T}}}\big[\log p_\Sigma(\mathbf{X}_{t_2}|\mathbf{X}_{t_1})|\mathbf{y}_{t_{1:T}}\big].\end{aligned} \tag{6.15}$$

With the "constant parameter setting" $\lambda_j := 1/(j-1)$, where $\Big[\prod_{\ell=0}^{k-1}(1-\lambda_{j-\ell})\Big]\lambda_{j-k} = \frac{1}{j-1}$, this gives the classical (quasi-) likelihood

$$\frac{1}{j-1}\sum_{k=0}^{j-2}\mathbf{E}_{\hat{\Sigma}_{t_{1:T}}}\big[\log p_\Sigma(\mathbf{X}_{t_{j-k}}|\mathbf{X}_{t_{j-k-1}})|\mathbf{y}_{t_{1:T}}\big],$$

that is (6.8) for $j = T$. Furthermore, the maximizer of (6.15) is, with the smoother-approximation as in (6.10) and $\breve{\Sigma}_{t_j}(\omega_{t_T})$ as in (6.11), given by

$$\hat{\Sigma}_{t_j}^{(m+1)} = \sum_{k=0}^{j-3}\Big[\prod_{\ell=0}^{k-1}(1-\lambda_{j-\ell})\Big]\lambda_{j-k}\, \breve{\Sigma}_{t_{j-k}}(\omega_{t_T}) + \Big[\prod_{\ell=0}^{j-3}(1-\lambda_{j-\ell})\Big]\, \breve{\Sigma}_{t_2}(\omega_{t_T}). \tag{6.16}$$

This can be written as the recursion

$$\hat{\Sigma}_{t_j}^{(m+1)} = \{1-\lambda_j\}\, \hat{\Sigma}_{t_{j-1}}^{(m+1)} + \lambda_j\, \breve{\Sigma}_{t_j}(\omega_{t_T})$$

with $\hat{\Sigma}_{t_2}^{(m+1)} = \breve{\Sigma}_{t_2}(\omega_{t_T})$. Again, we obtain with the "constant parameter setting" $\lambda_j := 1/(j-1)$ that $\hat{\Sigma}_{t_j}^{(m+1)}$ coincides with the estimate in (6.12) for $j = T$.

2) On-line algorithms: The above algorithm is not an on-line algorithm because the conditional expectation in (6.14) depends on all observations. Therefore, we replace the conditioning set of variables $\{\mathbf{y}_{t_{1:T}}\}$ by $\{\mathbf{y}_{t_{1:j}}\}$ meaning that we pass from the smoothing distribution to the filtering distribution. More precisely,

$$\mathbf{E}_{\hat{\Sigma}_{t_{1:T}}}\big[\log p_\Sigma(\mathbf{X}_{t_j}|\mathbf{X}_{t_{j-1}})|\mathbf{y}_{t_{1:T}}\big]$$

is replaced by

$$\mathbf{E}_{\hat{\Sigma}_{t_{1:j-1}}}\big[\log p_\Sigma(\mathbf{X}_{t_j}|\mathbf{X}_{t_{j-1}})|\mathbf{y}_{t_{1:j}}\big]$$

(we need at this point an estimate for Σ_{t_j} - see the comment at the end of this section) leading to the on-line algorithm

$$\mathcal{Q}_{t_j}(\Sigma|\hat{\Sigma}_{t_{1:j-1}}) := \{1-\lambda_j\}\,\mathcal{Q}_{t_{j-1}}(\Sigma|\hat{\Sigma}_{t_{1:j-2}}) + \lambda_j\,\mathbf{E}_{\hat{\Sigma}_{t_{1:j-1}}}\big[\log p_\Sigma(\mathbf{X}_{t_j}|\mathbf{X}_{t_{j-1}})|\mathbf{y}_{t_{1:j}}\big] \qquad (6.17)$$

with $\mathcal{Q}_{t_2}(\Sigma|\hat{\Sigma}_{t_1}) = \mathbf{E}_{\hat{\Sigma}_{t_1}}\big[\log p_\Sigma(\mathbf{X}_{t_2}|\mathbf{X}_{t_1})|\mathbf{y}_{t_{1:2}}\big]$. (6.15) holds analogously and we now obtain analogous to (6.16) the estimate

$$\hat{\Sigma}_{t_j} = \sum_{k=0}^{j-3}\Big[\prod_{\ell=0}^{k-1}(1-\lambda_{j-\ell})\Big]\lambda_{j-k}\,\breve{\Sigma}_{t_{j-k}}(\omega_{t_{j-k}}) + \Big[\prod_{\ell=0}^{j-3}(1-\lambda_{j-\ell})\Big]\,\breve{\Sigma}_{t_2}(\omega_{t_2}) \qquad (6.18)$$

now with

$$\breve{\Sigma}_{t_j}(\omega_{t_j}) := \sum_{i=1}^{N}\omega_{t_j}^i\big(\mathbf{x}_{t_j}^i - \mathbf{x}_{t_{j-1}}^i\big)\big(\mathbf{x}_{t_j}^i - \mathbf{x}_{t_{j-1}}^i\big)^T$$

based on the filtering particles $\{\mathbf{x}_{t_{j-1:j}}^i, \omega_{t_j}^i\}_{i=1}^N$. This estimate can be obtained from the on-line recursion

$$\hat{\Sigma}_{t_j} = \{1-\lambda_j\}\,\hat{\Sigma}_{t_{j-1}} + \lambda_j\,\breve{\Sigma}_{t_j}(\omega_{t_j}) \quad \text{with} \quad \hat{\Sigma}_{t_2} = \breve{\Sigma}_{t_2}(\omega_{t_2}). \qquad (6.19)$$

Observe that the estimated covariance matrix $\hat{\Sigma}_{t_j}$ is positive (semi-) definite by construction.

The new parameter estimate $\hat{\Sigma}_{t_j}$ is used afterwards to calculate the next filtering particles and their weights $\{\mathbf{x}_{t_{j+1}}^i, \omega_{t_{j+1}}^i\}_{i=1}^N$ followed by the calculation of $\hat{\Sigma}_{t_{j+1}}$ via another application of (6.19) etc. In contrast to the standard EM algorithm, our sequential variant therefore updates the covariance estimate (which in turn is used in the next step of the particle filter) in every time step. In the "new E-step", $\mathcal{Q}_{t_j}(\Sigma|\hat{\Sigma}_{t_{1:j-1}})$ is approximated through

$$\begin{aligned}\hat{\mathcal{Q}}_{t_j}(\Sigma|\hat{\Sigma}_{t_{1:j-1}}) &= \{1-\lambda_j\}\,\hat{\mathcal{Q}}_{t_{j-1}}(\Sigma|\hat{\Sigma}_{t_{1:j-2}}) \\ &\quad - \lambda_j\,\frac{1}{2}\sum_{i=1}^{N}\omega_{t_j}^i\Big[S\log 2\pi + \log|\Sigma| + \operatorname{tr}\Big\{\Sigma^{-1}\big(\mathbf{x}_{t_j}^i - \mathbf{x}_{t_{j-1}}^i\big)\big(\mathbf{x}_{t_j}^i - \mathbf{x}_{t_{j-1}}^i\big)^T\Big\}\Big]\end{aligned}$$

using the particles $\{\mathbf{x}_{t_{j-1:j}}^i, \omega_{t_j}^i\}_{i=1}^N$ which are generated by the particle filter described in the preceding section. In the "new M-step", the maximization of $\hat{\mathcal{Q}}_{t_j}(\Sigma|\hat{\Sigma}_{t_{1:j-1}})$ gives the on-line estimator defined in (6.19).

3) Time-constant covariance matrices: If Σ_{t_j} is time-constant the first idea is to apply the algorithm (6.19) with the "constant parameter setting" $\lambda_j = 1/(j-1)$. However, the situation is

different from the classical case in that the "old" estimate $\hat{\Sigma}_{t_{j-1}}$ has in addition some bias due to the use of particles generated with an estimated covariance instead of the true one. Therefore we need to put less weight on the first term in (6.19). The situation has been carefully investigated for a similar algorithm in the i.i.d.-case by Cappé and Moulines (2009). Following their recommendation we use in our situation the on-line algorithm

$$\hat{\Sigma}_{t_j} = \{1-(j-1)^{-\gamma}\}\,\hat{\Sigma}_{t_{j-1}} + (j-1)^{-\gamma}\,\breve{\Sigma}_{t_j}(\omega_{t_j}) \tag{6.20}$$

with $\gamma \in (\frac{1}{2},1)$. Cappé and Moulines prove consistency and asymptotic normality of their estimate for weights $\lambda_j := \lambda_0 j^{\,-\gamma}$ and $\gamma \in (\frac{1}{2},1)$ and also for $\gamma = 1$ under some restrictions on λ_0 (Theorem 2). Furthermore, in their simulations it turned out that a value of $\gamma = 0.6$ and $\lambda_0 = 1$ has lead to good estimates. From our experience we prefer the choice $\gamma = 0.8$ and $\lambda_0 = 1$ (see Section 6.9). Even-Dar and Mansour (2003) obtained an optimal value of about 0.85 in a related estimation problem. We emphasize that the choice of γ needs more investigations - both theoretical and practical.

4) Time-varying covariance matrices: If Σ_{t_j} is time-varying it is necessary to put more weight on recent observations. In this case, the traditional solution is to use the algorithms (6.14), (6.17) and (6.19) with time-constant $\lambda_j \equiv \lambda$ instead of a decaying λ_j. To achieve a better degree of adaptation our λ_j will still be time-varying (see Section 6.5.1) but with the intuition that the λ_j fluctuate around some constant value of λ. That is we use in the time-varying case

$$\hat{\Sigma}_{t_j} = \{1-\lambda_j\}\,\hat{\Sigma}_{t_{j-1}} + \lambda_j\,\breve{\Sigma}_{t_j}(\omega_{t_j}) \quad \text{with} \quad \hat{\Sigma}_{t_2} = \breve{\Sigma}_{t_2}(\omega_{t_2}). \tag{6.21}$$

The choice of the λ_j is discussed in Section 6.5.1. For a deeper understanding we stress the following heuristics: If $\lambda_j \equiv \lambda$ and $t_j = j\,\delta$ (e.g. $\delta = \frac{1}{T}$) then we have with $b := \frac{\delta}{\lambda}$ for $\delta \to 0$

$$\Big[\prod_{\ell=0}^{k-1}(1-\lambda_{j-\ell})\Big]\lambda_{j-k} = (1-\lambda)^k\lambda = \frac{\delta}{b}\Big(1-\frac{\delta}{b}\Big)^{\frac{1}{\delta}k\delta} \approx \frac{\delta}{b}\Big(e^{-\frac{1}{b}}\Big)^{k\delta} = \frac{\delta}{b}K\Big(\frac{k\delta}{b}\Big) \tag{6.22}$$

where $K(x) := e^{-x}$. That is $\mathcal{Q}_{t_j}(\Sigma|\hat{\Sigma}_{t_{1:T}})$ from (6.17) is basically the kernel likelihood given in (6.13) with the one-sided exponential kernel, and $\hat{\Sigma}_{t_j}$ given by (6.21) with constant weights $\lambda_j = \lambda$ is basically the kernel estimate

$$\hat{\Sigma}_{t_j} = \Big[\sum_k K\Big(\frac{k}{bT}\Big)\Big]^{-1}\sum_k K\Big(\frac{k}{bT}\Big)\sum_{i=1}^N \omega^i_{t_{j-k}}\big(\mathbf{x}^i_{t_{j-k}} - \mathbf{x}^i_{t_{j-k-1}}\big)\big(\mathbf{x}^i_{t_{j-k}} - \mathbf{x}^i_{t_{j-k-1}}\big)^T.$$

6.3.4 Summary

Our estimation method consists of three components:

(i) The state-space model with a new market microstructure noise model and the transaction time model for the efficient log-price (Section 6.3.1);

(ii) A particle filter which sequentially approximates the filtering distributions of the efficient log-prices given the observed transaction prices (Section 6.3.2);

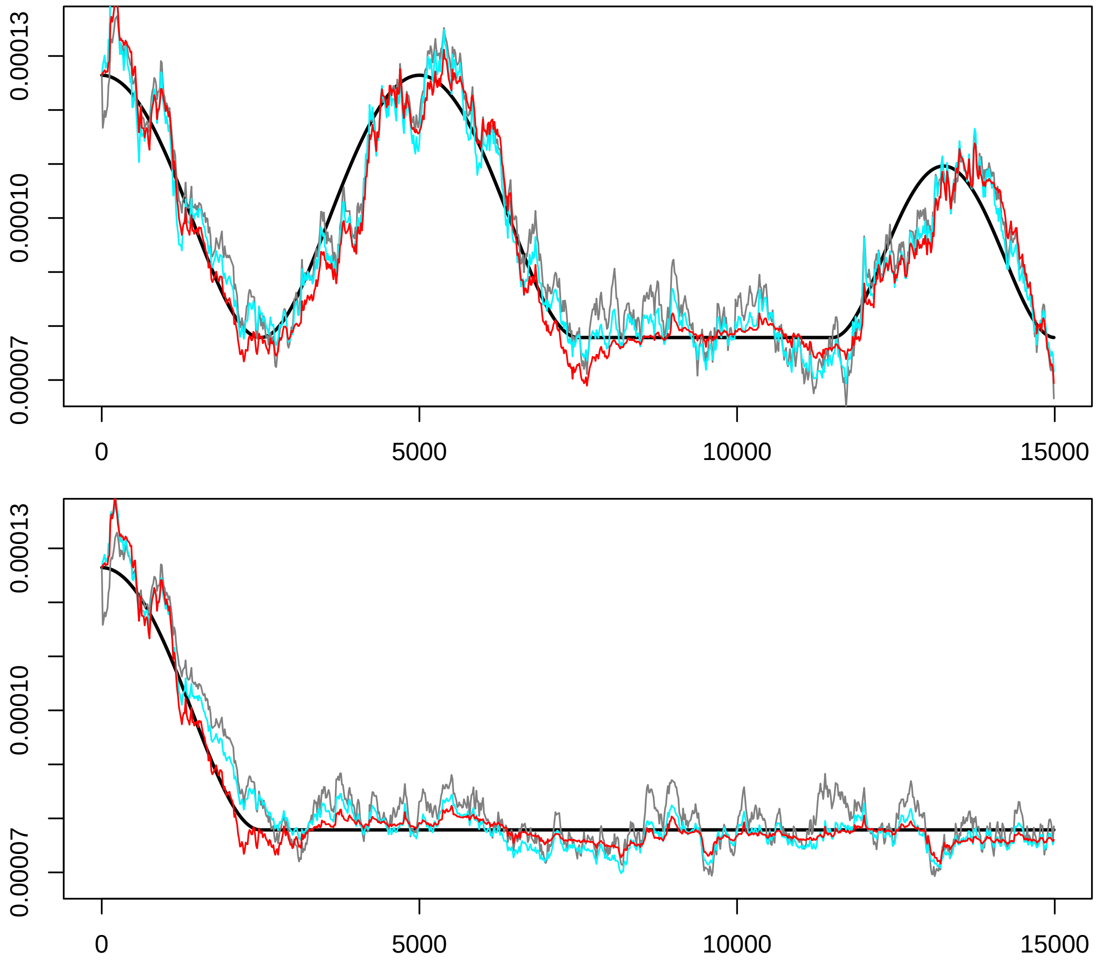

Figure 6.4: Estimation of two time-varying volatility curves given by the black lines based on simulated data. Estimators: $\hat{\Sigma}_{t_j}$ (turquoise line), $\tilde{\Sigma}^*_{t_j|t_j}$ (red line), benchmark estimator (gray line). For details see Section 6.10.1.

(iii) The on-line EM-type estimator $\hat{\Sigma}_{t_j}$ given by (6.20) or (6.21) which estimates Σ_{t_j} based on the particle approximation of the filtering distribution obtained from the particle filter (Section 6.3.3). This estimator is improved in the time-varying case to $\tilde{\Sigma}^*_{t_j|t_j}$ in Section 6.5.

A key aspect of the method is the back and forth between the particle filter and the EM-type estimator. To propagate the particles from time t_j to time t_{j+1} the particle filter requires an estimator of $\Sigma_{t_{j+1}}$ which we denote by $\hat{\Sigma}^{\mathrm{pf}}_{t_{j+1}}$. A simple solution is to use $\hat{\Sigma}^{\mathrm{pf}}_{t_{j+1}} := \hat{\Sigma}_{t_j}$ from the previous EM-type step. A more sophisticated solution is to use the estimator $\hat{\Sigma}^{\mathrm{pf}}_{t_{j+1}} := \tilde{\Sigma}^*_{t_{j+1}|t_j}$ from Section 6.5.2 based on a prediction argument. The EM-type estimator then in turn updates the covariance estimate based on the new particles for time t_{j+1} generated by the particle filter.

Estimation results of our estimators $\hat{\Sigma}_{t_j}$, $\tilde{\Sigma}^*_{t_j|t_j}$ (see Section 6.5), and a benchmark estimator (see Section 6.10.1) are presented in Figure 6.4. Details and a discussion are given in Section 6.10.1.

6.4 From Transaction Time to Clock Time

6.4.1 Clock Time Spot Volatility Estimation

In the preceding section, we have derived an algorithm for the estimation of the covariance matrix $\Sigma_{t_j} = \Sigma(t_j)$ in a transaction time model. If one prefers a clock time model all results of this chapter continue to hold with some modifications. In this case one may consider as the underlying model the stochastic differential equation

$$d\mathbf{X}(t) = \Gamma(t)\, d\mathbf{W}(t) \qquad \text{where} \quad \Gamma(t)\, \Gamma^T(t) = \Sigma^c(t) \tag{6.23}$$

and $\mathbf{W}(t)$ is a multivariate Brownian motion. $\Sigma^c(t)$ is the volatility curve in the clock time model. Loosely speaking, it denotes volatility per time unit while $\Sigma(t)$ denotes the volatility per transaction at time t. The relation between the two curves should be given by (6.25) (of course this depends on the mathematical definition of $\Sigma(t)$ and $\Sigma^c(t)$). If we set $\mathbf{X}_{t_j} = \mathbf{X}(t_j)$ we obtain the same state space model as in (6.4) and (6.5) but now with the log-returns $\mathbf{Z}_{t_j} = \mathbf{X}_{t_j} - \mathbf{X}_{t_{j-1}}$ approximately distributed as

$$\mathbf{Z}_{t_j} \sim \mathcal{N}\big(\mathbf{0}, |t_j - t_{j-1}|\; \Sigma^c(t_j)\big).$$

This is the only change needed in the state-space model (6.4), (6.5). As an estimate $\hat{\Sigma}^c_{t_j}$ we can use the on-line estimates (6.20) and (6.21) but now with the update matrix $\breve{\Sigma}_{t_j}(\omega_{t_j})$ replaced by

$$\breve{\Sigma}^c_{t_j}(\omega^c_{t_j}) := \sum_{i=1}^{N} \omega^{ci}_{t_j} \frac{\big(\mathbf{x}^{ci}_{t_j} - \mathbf{x}^{ci}_{t_{j-1}}\big)\big(\mathbf{x}^{ci}_{t_j} - \mathbf{x}^{ci}_{t_{j-1}}\big)^T}{|t_j - t_{j-1}|} \tag{6.24}$$

based on the modified filtering particles $\{\mathbf{x}^{ci}_{t_{j-1:j}}, \omega^{ci}_{t_j}\}_{i=1}^N$. In Section 6.5, we discuss bias correction, adaptive and time-varying selection of the step size λ_j, and prediction of future volatilities. All methods can also be applied to $\Sigma^c(t)$ which is briefly summarized at the end of Section 6.5.2.

6.4.2 An Alternative Estimator for Clock Time Spot Volatility

In the diffusion model (6.23) the spot volatility in clock time is

$$\Sigma^c(t) = \lim_{\Delta t \to 0} \frac{\int_t^{t+\Delta t} \Sigma^c(s)\, ds}{\Delta t} = \lim_{\Delta t \to 0} \frac{\mathrm{Var}\big(\mathbf{X}(t+\Delta t) - \mathbf{X}(t)\big)}{\Delta t}.$$

To clarify the relation to the transaction time volatility $\Sigma(t)$ we assume for a moment that the transaction times t_j are realizations of a stochastic point process with intensity function $\lambda_I(t)$ (transaction rate) which is independent of the efficient and observed prices. We then have

$$\begin{aligned}
\lim_{\Delta t \to 0} \frac{\mathrm{Var}\big(\mathbf{X}(t+\Delta t) - \mathbf{X}(t)\big)}{\Delta t} &= \lim_{\Delta t \to 0} \mathbf{E}\, \frac{\sum_{j\,:\, t<t_j \le t+\Delta t} \Sigma(t_j)}{\Delta t} \\
&= \lim_{\Delta t \to 0} \mathbf{E}\, \frac{\sum_{j\,:\, t<t_j \le t+\Delta t} \Sigma(t_j)}{\big|\{j\,:\, t < t_j \le t+\Delta t\}\big|}\; \frac{\big|\{j\,:\, t < t_j \le t+\Delta t\}\big|}{\Delta t} \\
&= \Sigma(t)\; \lambda_I(t)
\end{aligned}$$

that is

$$\Sigma^c(t) = \Sigma(t)\,\lambda_I(t). \tag{6.25}$$

We stress that this relation is primarily a nonparametric relation ("variance per time unit = variance per transaction × expected number of transactions per time unit") and it depends on the underlying model whether this coincides with the definition of $\Sigma^c(t)$ and $\Sigma(t)$ given in the model. A model which exactly leads to this formula is the subordinated differential equation $d\mathbf{X}(t) = \Gamma(t)\,d\mathbf{W}_{N(t)}$ with a point process $N(t)$ with intensity $\lambda_I(t)$ (cf. Howison and Lamper 2001). The unit of Δt (e.g. milliseconds) is also the unit of $\Sigma(t)$ (e.g. variance per millisecond) and of the intensity (e.g. expected number of transactions per millisecond). An obvious estimate of the clock time volatility therefore is $\hat{\hat{\Sigma}}^c(t_j) = \hat{\Sigma}_{t_j} \times |\{\ell : t_j - \Delta t < t_\ell \leq t_j\}| \,/\, \Delta t$ with some Δt.

Here we advocate a different estimation method of the intensity function $\lambda_I(t)$ which is closer related to our on-line scheme, namely the estimation of $\lambda_I(t)$ by the inverse of the averaged duration times $\bar{\delta}_j$ defined by the recursion

$$\bar{\delta}_j = (1 - \lambda_j)\,\bar{\delta}_{j-1} + \lambda_j\,\big(t_j - t_{j-1}\big) \quad \text{with} \quad \bar{\delta}_2 = t_2 - t_1$$

leading with (6.18) to the alternative clock time volatility estimator

$$\hat{\Sigma}^c_{\text{alt}}(t_j) := \frac{\hat{\Sigma}_{t_j}}{\bar{\delta}_j} = \frac{\sum_{k=0}^{j-3}\Big[\prod_{\ell=0}^{k-1}(1-\lambda_{j-\ell})\Big]\lambda_{j-k}\,\breve{\Sigma}_{t_{j-k}}(\omega_{t_{j-k}}) + \Big[\prod_{\ell=0}^{j-3}(1-\lambda_{j-\ell})\Big]\,\breve{\Sigma}_{t_2}(\omega_{t_2})}{\sum_{k=0}^{j-3}\Big[\prod_{\ell=0}^{k-1}(1-\lambda_{j-\ell})\Big]\lambda_{j-k}\big(t_{j-k} - t_{j-k-1}\big) + \Big[\prod_{\ell=0}^{j-3}(1-\lambda_{j-\ell})\Big]\big(t_2 - t_1\big)}$$

(or better with $\hat{\Sigma}_{t_j}$ replaced by $\tilde{\Sigma}^*_{t_j|t_j}$ from Section 6.5). This estimator has a remarkable property: Because $\breve{\Sigma}_{t_\ell}(\omega_{t_\ell}) \approx \big(t_\ell - t_{\ell-1}\big)\,\breve{\Sigma}^c_{t_\ell}(\omega^c_{t_\ell})$ the estimator is of the form

$$\hat{\Sigma}^c_{\text{alt}}(t_j) \approx \frac{\sum_{k=0}^{j-2} w_k \breve{\Sigma}^c_{t_{j-k}}(\omega^c_{t_{j-k}})}{\sum_{k=0}^{j-2} w_k}$$

that is $\hat{\Sigma}^c_{\text{alt}}(t_j)$ is a weighted average of the $\breve{\Sigma}^c_{t_\ell}(\omega^c_{t_\ell})$ and therefore also a decent estimator in the clock time model (the "≈" signs stem from the fact that in $\breve{\Sigma}_{t_\ell}(\omega_{t_\ell})$ and $\breve{\Sigma}^c_{t_\ell}(\omega^c_{t_\ell})$ two different particle filters are used - the effect of this is not clear!). Notice that the denominator $t_{j-k} - t_{j-k-1}$ in $\breve{\Sigma}^c_{t_{j-k}}(\omega^c_{t_{j-k}})$ cancels out leading therefore to a more stable estimator (for example the sharp green peaks in Figures 6.11 and 6.12 are caused by small values of $t_{j-k} - t_{j-k-1}$).

The above argument contains a pitfall: While $\Sigma(t)$ usually is smooth thus requiring small values of λ_j, the intensity of the point process $\lambda_I(t)$ changes considerably over time thus requiring larger values of λ_j. For that reason we use different sequences λ_j for the estimators $\hat{\Sigma}_{t_j}$ and $\bar{\delta}_j$. More specific we can use the same adaptation procedure as described in Section 6.5.1 also for $\bar{\delta}_j$ with the only difference that we determine the equivalent to the unbiased estimator $\tilde{\Sigma}_{t_j|t_j}$ as given by (6.35) and (6.36). (The formula (6.37) for the minimal mean squared error estimate does not transfer to $\bar{\delta}_j$ because the durations usually would not be independent.)

The estimators $\tilde{\Sigma}^{*c}_{t_j|t_j}$ (quasi mean squared error corrected version of $\hat{\Sigma}^{c}_{t_j}$ as defined in the following section) and $\tilde{\Sigma}^{c}_{\text{alt}}(t_j) = \tilde{\Sigma}^{*}_{t_j|t_j}/\bar{\delta}_j$ $\big($with $\tilde{\Sigma}^{*}_{t_j|t_j}$ as defined in (6.43)$\big)$ are plotted in figures 6.11 and 6.12 and discussed in Section 6.10.2. In this example a constant step size $\lambda_j \equiv \lambda$ turned out to be sufficient for the estimator $\bar{\delta}_j$.

6.5 Fine-Tuning of the Volatility Estimator in the Time-Varying Case

In this section we present a method for the adaptive choice of the time-varying step size λ_j and an on-line bias correction for the estimator $\hat{\Sigma}_{t_j}$ given by (6.18) through (6.19). The basic idea for bias correction is to calculate two estimators with different step sizes in parallel and to balance the two on-line. The resulting estimator is the estimator $\tilde{\Sigma}^{*}_{t_j|t_j}$ from Figure 6.4. We continue to use the notation with Σ although we only discuss the univariate case (the basic formula (6.35) also holds in the multivariate case with synchronous trading times). We also present an on-line method for quasi mean squared error minimization, and a method for the prediction of future volatilities.

6.5.1 Adaptive Step Size Selection

For constant λ we have the equivalence of the on-line estimator with a kernel estimator with kernel $K(x) = e^{-x}$ as described in Section 6.3.3 under 4). For kernel estimators the adaptive (off-line) choice of the bandwidth has been discussed extensively and most of these results could be transferred to the present setting. However, there does not exist any equivalence between our on-line estimator with time-varying λ_j and kernel estimators with local bandwidths: The weight λ_j at time t_j only applies to the last observation and not to a longer stretch of data.

We are not aware of any rigorous results on adaptive choices for a sequence λ_j for exponential smoothing estimators. This means that the method proposed below may also be of relevance in other on-line estimation settings.

Here is an overview of the method:

1. We start with the ad-hoc proposal based on the logistic function (to ensure $0<\lambda_j<1$)

$$\lambda_j := \frac{\exp[\alpha + \beta h_{t_{j-1}}]}{1 + \exp[\alpha + \beta h_{t_{j-1}}]}, \tag{6.26}$$

   where

$$h_{t_{j-1}} := \left| \frac{\log \hat{\Sigma}_{t_{j-1}} - \log \hat{\Sigma}^{(1/2)}_{t_{j-1}}}{\overline{j-1} \, - \, \overline{j-1}^{(1/2)}} \right|^2 . \tag{6.27}$$

   (6.26) was proposed by Taylor (2004) with a different $h_{t_{j-1}}$. The above $h_{t_{j-1}}$ is motivated at the end of Section 6.5.2. For the definition of the expressions in $h_{t_{j-1}}$ see (6.33) and below. α and β are adaptively determined in step 4.

2. At each time t_j we calculate on-line two different estimators: First $\hat{\Sigma}_{t_j}$ as defined in (6.19) and second $\hat{\Sigma}_{t_j}^{(1/2)}$ which is the same as $\hat{\Sigma}_{t_j}$ but with all λ_j replaced by $\lambda_j/2$. Thus we have at each time step two on-line estimators available - one with a larger step size sequence (with less smoothing) and one with a smaller step size sequence (with stronger smoothing).

3. We then consider arbitrary linear combinations of these estimators and determine at each time step t_j the optimal linear combination with respect to the optimal quasi mean squared error, or alternatively with respect to unbiasedness resulting in the estimators $\tilde{\Sigma}^*_{t_j|t_j}$ or $\tilde{\Sigma}_{t_j|t_j}$. The advantage of this method is that it can be performed on-line for each t_j.

4. The mean squared error of the estimator $\tilde{\Sigma}^*_{t_j|t_j}$ resulting from the whole procedure 1. through 3. is finally minimized with respect to α and β by the cross-validation type criterion

$$\operatorname{crit}(\alpha,\beta) := \sum_{j=2}^{T-1} \big(\tilde{\Sigma}^*_{t_j|t_j} - \breve{\Sigma}_{t_{j+1}}(\omega_{t_{j+1}})\big)^2. \tag{6.28}$$

   This cannot be done on-line. In practice, one will use in an on-line setting the values of α and β from past experience. The expectation of the above criterion is approximately

$$\sum_{j=2}^{T-1} \Big[\big(\mathbf{E}\tilde{\Sigma}^*_{t_j|t_j} - \Sigma_{t_j}\big)^2 + \operatorname{Var}\big(\tilde{\Sigma}^*_{t_j|t_j}\big) + \operatorname{Var}\big(\breve{\Sigma}_{t_{j+1}}(\omega_{t_{j+1}})\big)\Big].$$

   Because the last term does not depend on α and β we correctly minimize the approximate mean squared error.

We do not know anything about the theoretical properties of the procedure as a whole. We feel however that the degree of adaption is high as a result of the minimization in step 3 (correcting somehow for the limitations of the ad-hoc proposal in step 1) and the final minimization with respect to α and β. This is confirmed by our simulations.

Remark: A simpler alternative is to use a fixed step size $\lambda_j \equiv \lambda$ and to minimize the mean squared error (6.28) with respect to λ. Steps 2 and 3 can be kept as they are in this case.

6.5.2 On-line Bias Correction and Mean Squared Error Minimization

We now describe steps 2 and 3 in detail. We stress that these steps can be done for arbitrary step size sequences λ_j, that is we do not need the specific choice from step 1.

Let $\tau : [0,\infty) \to [0,\infty)$ be the mapping that maps transaction time to clock time, i.e. $\tau(j) = t_j$ (we assume that τ is defined on the whole positive real line). We define

$$\dot{\Sigma}(s) := \frac{\partial}{\partial s}\Sigma\big(\tau(s)\big) = \Sigma'\big(\tau(s)\big)\;\tau'(s)$$

leading to the linear approximation

$$\Sigma(t_j) = \Sigma\big(\tau(j)\big) \approx \Sigma(t_i) + (j-i)\,\dot{\Sigma}(i) \tag{6.29}$$

$\big($for the meaning of the "$\approx$"-sign see Section 6.11; for example in the equidistant case $t_i = i\delta$ we have $\tau'(i) = \delta$ and $(j-i)\,\dot{\Sigma}(i) = (j-i)\,\delta\Sigma'(t_i)$ is small for small $\delta\big)$. By using the approximation

$$\mathbf{E}\,\breve{\Sigma}_{t_j}(\omega_{t_j}) = \mathbf{E}\sum_{i=1}^{N}\omega_{t_j}^i(\mathbf{x}_{t_j}^i - \mathbf{x}_{t_{j-1}}^i)(\mathbf{x}_{t_j}^i - \mathbf{x}_{t_{j-1}}^i)^T \approx \mathbf{E}\,(\mathbf{X}_{t_j} - \mathbf{X}_{t_{j-1}})(\mathbf{X}_{t_j} - \mathbf{X}_{t_{j-1}})^T = \Sigma_{t_j} \quad (6.30)$$

we obtain from (6.18) (with some i close to j)

$$\begin{aligned}\mathbf{E}\,\hat{\Sigma}_{t_j} &\approx \sum_{k=0}^{j-3}\Big[\prod_{\ell=0}^{k-1}(1-\lambda_{j-\ell})\Big]\lambda_{j-k}\,\Sigma\big(t_{j-k}\big) + \Big[\prod_{\ell=0}^{j-3}(1-\lambda_{j-\ell})\Big]\,\Sigma\big(t_2\big)\\ &\approx \sum_{k=0}^{j-3}\Big[\prod_{\ell=0}^{k-1}(1-\lambda_{j-\ell})\Big]\lambda_{j-k}\,\Big[\Sigma(t_i) - \big(i-(j-k)\big)\dot{\Sigma}(i)\Big] \qquad (6.31)\\ &\qquad + \Big[\prod_{\ell=0}^{j-3}(1-\lambda_{j-\ell})\Big]\Big[\Sigma(t_i) - \big(i-2\big)\dot{\Sigma}(i)\Big]\\ &= \Sigma(t_i) - \big(i-\bar{j}\big)\,\dot{\Sigma}(i) \approx \Sigma(t_i) - \big(i-\bar{j}\big)\,\dot{\Sigma}(\bar{j}) \approx \Sigma\big(t_{\bar{j}}\big) \qquad (6.32)\end{aligned}$$

with

$$\bar{j} := \sum_{k=0}^{j-3}\Big[\prod_{\ell=0}^{k-1}(1-\lambda_{j-\ell})\Big]\lambda_{j-k}\,(j-k) + \Big[\prod_{\ell=0}^{j-3}(1-\lambda_{j-\ell})\Big]\,2\,. \qquad (6.33)$$

We note that $\bar{j}$ can be obtained via the on-line recursion

$$\bar{j} = (1-\lambda_j)\;\overline{j-1} + \lambda_j\,j \qquad \text{with} \quad \bar{2} = 2. \qquad (6.34)$$

This means that we are estimating $\Sigma(t)$ essentially at time $t_{\bar{j}} < t_j$. This is a result of the one-sidedness of the recursive method $\big($for example in the equidistant case $t_j = j\delta$ and $\lambda_j \equiv \lambda$ we obtain $\bar{j} \approx j + 1 - 1/\lambda$ and $t_{\bar{j}} \approx (j+1-1/\lambda)\,\delta$ - see also (6.22) $\big)$. In order to correct for this bias or to construct even approximately unbiased estimators of future volatilities we now take a linear combination of the two estimators $\hat{\Sigma}_{t_j}$ and $\hat{\Sigma}_{t_j}^{(1/2)}$ where the latter is the same as $\hat{\Sigma}_{t_j}$ in (6.18) and (6.19) but with all λ_j replaced by $\lambda_j/2$. Analogously we define $\bar{j}^{\,(1/2)}$ as in (6.33) and (6.34) but again with all λ_j replaced by $\lambda_j/2$. The new estimator now is defined by the extrapolation

$$\tilde{\Sigma}_{t_i|t_j} := \big(1+\kappa_{i|j}\big)\,\hat{\Sigma}_{t_j} - \kappa_{i|j}\,\hat{\Sigma}_{t_j}^{(1/2)} \qquad (6.35)$$

with time-varying weights

$$\kappa_{i|j} := \frac{i-\bar{j}}{\bar{j}-\bar{j}^{\,(1/2)}}\,. \qquad (6.36)$$

We immediately obtain

$$\mathbf{E}\,\tilde{\Sigma}_{t_i|t_j} \approx \Sigma(t_i) - \Big[\big(1+\kappa_{i|j}\big)\,\big(i-\bar{j}\big) - \kappa_{i|j}\,\big(i-\bar{j}^{\,(1/2)}\big)\Big]\,\dot{\Sigma}(i) = \Sigma(t_i)$$

and for $i = j$ we therefore have a bias-corrected estimator of $\Sigma(t_j)$. Because the estimator extrapolates the two estimators $\hat{\Sigma}_{t_j}$ and $\hat{\Sigma}_{t_j}^{(1/2)}$ we have to watch particularly the variance which may become large. From a statistical view a better choice is the estimator with a minimal mean squared error. In Appendix A.10, we calculate the quasi mean squared error (with the unknown efficient log-prices used instead of the filter particles) and show that this mean squared error is minimized by

$$\kappa_{\min} \approx \frac{\big(i-\bar{j}\big)\big(\bar{j}-\bar{j}^{\,(1/2)}\big)\,\big[\frac{\partial}{\partial t}\log\Sigma(t)_{|t=t_{\bar{j}}}\,\tau'\big(\bar{j}\big)\big]^2 - 2\,(v_{1,j}-v_{3,j})}{\big(\bar{j}-\bar{j}^{\,(1/2)}\big)^2\,\big[\frac{\partial}{\partial t}\log\Sigma(t)_{|t=t_{\bar{j}}}\,\tau'\big(\bar{j}\big)\big]^2 + 2\,(v_{1,j}+v_{2,j}-2v_{3,j})} \tag{6.37}$$

with $v_{1,j}$, $v_{2,j}$ and $v_{3,j}$ obtained from the recursions

$$v_{1,j} = \big(1-\lambda_j\big)^2\,v_{1,j-1} + \lambda_j^2\,, \qquad v_{1,2} = 1; \tag{6.38}$$

$$v_{2,j} = \big(1-\frac{\lambda_j}{2}\big)^2\,v_{2,j-1} + \frac{\lambda_j^2}{4}\,, \qquad v_{2,2} = 1; \tag{6.39}$$

$$v_{3,j} = \big(1-\lambda_j\big)\big(1-\frac{\lambda_j}{2}\big)\,v_{3,j-1} + \frac{\lambda_j^2}{2}\,, \qquad v_{3,2} = 1. \tag{6.40}$$

$\frac{\partial}{\partial t}\log\Sigma(t)_{|t=t_{\bar{j}}}$ and $\tau'\big(\bar{j}\big)$ are unknown. In order to get an adaptive choice of κ we replace these terms by estimators. From (6.31) we know that $\hat{\Sigma}_{t_j}$ and $\hat{\Sigma}_{t_j}^{(1/2)}$ are essentially estimators of $\Sigma(t)$ at times $t_{\bar{j}}$ and $t_{\bar{j}^{\,(1/2)}}$, respectively. We therefore use

$$\frac{\log\hat{\Sigma}_{t_j} - \log\hat{\Sigma}_{t_j}^{(1/2)}}{t_{\bar{j}} - t_{\bar{j}^{\,(1/2)}}}\;\frac{t_{\bar{j}} - t_{\bar{j}^{\,(1/2)}}}{\bar{j}-\bar{j}^{\,(1/2)}} \tag{6.41}$$

as an estimate of $\frac{\partial}{\partial t}\log\Sigma(t)_{|t=t_{\bar{j}}}\,\tau'\big(\bar{j}\big)$ leading to

$$\kappa_{i|j}^{*} := \frac{\frac{i-\bar{j}}{\bar{j}-\bar{j}^{\,(1/2)}}\,\Big[\log\hat{\Sigma}_{t_j} - \log\hat{\Sigma}_{t_j}^{(1/2)}\Big]^2 - 2\,(v_{1,j}-v_{3,j})}{\Big[\log\hat{\Sigma}_{t_j} - \log\hat{\Sigma}_{t_j}^{(1/2)}\Big]^2 +\; 2\,(v_{1,j}+v_{2,j}-2v_{3,j})} \tag{6.42}$$

and the corresponding estimator

$$\tilde{\Sigma}_{t_i|t_j}^{*} := \big(1+\kappa_{i|j}^{*}\big)\,\hat{\Sigma}_{t_j} - \kappa_{i|j}^{*}\,\hat{\Sigma}_{t_j}^{(1/2)}. \tag{6.43}$$

In practice, the values of $\kappa_{i|j}^{*}$ will be restricted to the interval $[-1,1]$ because other values do not make sense (smaller values than -1 may occur because $\hat{\Sigma}_{t_j}$ and $\hat{\Sigma}_{t_j}^{(1/2)}$ are correlated - however such values yield an extrapolation in the wrong time direction).

An example of this estimator for simulated data is given in Figure 6.5. The bias of $\hat{\Sigma}_{t_j}$ and $\hat{\Sigma}_{t_j}^{(1/2)}$ and the bias correction of $\tilde{\Sigma}_{t_j|t_j}^{*}$ are clearly visible. For details see Section 6.10.1.

It is easy to prove that $(v_{1,j}+v_{2,j}-2v_{3,j}) \geq 0$. "Usually" also $v_{1,j}-v_{3,j} \geq 0$ $\big($for example for constant $\lambda_j \equiv \lambda$ $v_{1,j}$ and $v_{3,j}$ converge to the fixpoints of the recursion $v_1 = \frac{\lambda}{2-\lambda}$ and $v_3 = \frac{\lambda}{3-\lambda}$ with $v_{1,j}-v_{3,j} > 0\big)$. For this reason we usually have $\kappa_{i|j}^{*} < \kappa_{i|j}$.

For the recursion described in Section 6.3.4 (where $\Sigma(t_{j+1})$ is needed in the next step of the particle filter) we think that the mean squared error choice $\tilde{\Sigma}_{t_{j+1}|t_j}^{*}$ with $\kappa_{j+1|j}^{*}$ is the best choice. On the contrary as an estimate for $\Sigma(t_j)$ of financial log-returns the unbiased estimator with

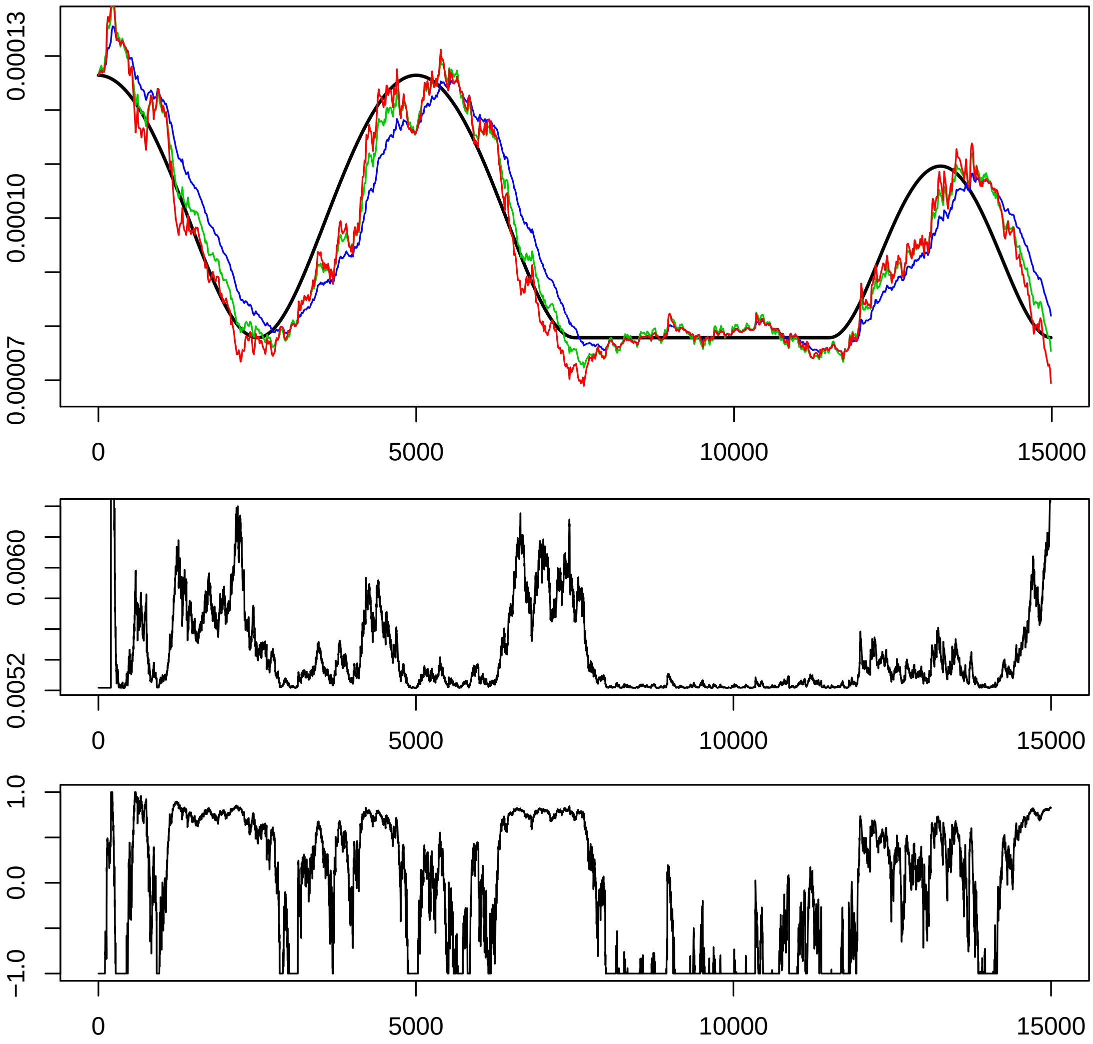

Figure 6.5: Estimation of the time-varying volatility curve given by the black line in the upper plot based on simulated data. Upper plot: $\tilde{\Sigma}^*_{t_j|t_j}$ (red line), $\hat{\Sigma}_{t_j}$ (green line), $\hat{\Sigma}^{(1/2)}_{t_j}$ (blue line); middle plot: step size sequence λ_j; lower plot: sequence $\kappa^*_{j|j}$. For details see Section 6.10.1.

$\kappa_{j|j}$ may be more interesting (it is less smoothed and contains in some sense more information). Perhaps in a practical application both estimators (with $\kappa_{j|j}$ and $\kappa^*_{j|j}$) should be plotted.

We finally motivate the choice of λ_j and $h_{t_{j-1}}$ in step 1: In the case of constant $\lambda_j = \lambda$ we obtain from (6.31) and (A.14) for the mean squared error

$$\mathbf{E}\Big(\hat{\Sigma}_{t_j} - \Sigma(t_j)\Big)^2 \approx 1/\lambda^2\,\dot{\Sigma}(\bar{j})^2 + \lambda\,\Sigma(t_{\bar{j}})^2$$

which gets minimal for

$$\lambda = \left|\sqrt{2}\,\frac{\partial}{\partial t}\log\Sigma(t)_{|t=t_{\bar{j}}}\,\tau'\big(\bar{j}\big)\right|^{2/3}.$$

Together with the restriction $0 < \lambda_j < 1$ (leading to the use of the logistic function) and the estimate (6.41) this has motivated the local choice of λ_j as in (6.26) with

$$h_{t_{j-1}} := \left| \frac{\log \hat{\Sigma}_{t_{j-1}} - \log \hat{\Sigma}^{(1/2)}_{t_{j-1}}}{\overline{j-1} - \overline{j-1}^{(1/2)}} \right|^{\rho}$$

where α and β are determined as described in step 4. We have simulated the mean squared error of the whole procedure 1. through 4. for several values of ρ leading finally to the choice $\rho = 2$ as in (6.27). Nevertheless, the choice of λ_j and $h_{t_{j-1}}$ as given in (6.26) and (6.27) remains to be an ad-hoc suggestion.

Bias correction in clock time models: A similar algorithm for adaption and bias correction can be established in the clock time setting from Section 6.4. Instead of the approximation (6.29) we start with

$$\Sigma^c(t_j) = \Sigma^c(t_i) + (t_j - t_i)\, {\Sigma^c}'(t_i)$$

and define instead of $\bar{j}$

$$\bar{t}_j := \sum_{k=0}^{j-3} \Big[\prod_{\ell=0}^{k-1} (1 - \lambda_{j-\ell}) \Big] \lambda_{j-k}\, t_{j-k} + \Big[\prod_{\ell=0}^{j-3} (1 - \lambda_{j-\ell}) \Big]\, t_2$$

given by the on-line recursion

$$\bar{t}_j = (1 - \lambda_j)\ \bar{t}_{j-1} + \lambda_j\, t_j \quad \text{with} \quad \bar{t}_2 = t_2.$$

Analogously we obtain the estimator

$$\tilde{\Sigma}^c_{t_i|t_j} := \big(1 + \kappa_{t_i|t_j}\big)\, \hat{\Sigma}^c_{t_j} - \kappa_{t_i|t_j}\ \hat{\Sigma}^{c\,(1/2)}_{t_j}$$

with

$$\kappa_{t_i|t_j} := \frac{t_i - \bar{t}_j}{\bar{t}_j - \bar{t}_j^{(1/2)}}$$

as the approximately unbiased estimator and $\tilde{\Sigma}^{*\,c}_{t_i|t_j}$ with

$$\kappa^*_{t_i|t_j} \approx \frac{\frac{t_i - \bar{t}_j}{\bar{t}_j - \bar{t}_j^{(1/2)}} \left[\log \hat{\Sigma}^c_{t_j} - \log \hat{\Sigma}^{c\,(1/2)}_{t_j} \right]^2 - 2\,(v_{1,j} - v_{3,j})}{\left[\log \hat{\Sigma}^c_{t_j} - \log \hat{\Sigma}^{c\,(1/2)}_{t_j} \right]^2 +\ 2\,(v_{1,j} + v_{2,j} - 2v_{3,j})}$$

as the estimator with approximately optimal quasi mean squared error.

Prediction: The estimators $\tilde{\Sigma}_{t_i|t_j}$ and $\tilde{\Sigma}^*_{t_i|t_j}$ can be used (with $i > j$) for prediction of future volatilities. In particular in combination with a predictor for future durations (e.g. with an ACD model - cf. Engle and Russell (1998)) this may lead to new predictors. One should keep in mind that these predictions are based on linear extrapolation. However, it should be possible to adapt the methods of this work also to other prediction models such as in Meddahi, Renault, and Werker (2006). By plugging the relation

$$\frac{t_i - \bar{t}_j}{\bar{t}_j - \bar{t}_j^{(1/2)}} \approx \frac{\big(i - \bar{j}\big)\,\tau'\big(\bar{j}\big)}{\big(\bar{j} - \bar{j}^{\,(1/2)}\big)\,\tau'\big(\bar{j}\big)} = \frac{i - \bar{j}}{\bar{j} - \bar{j}^{\,(1/2)}}$$

into (6.36) and (6.42) and replacing afterwards t_i by t we can also obtain predictors for arbitrary time points t. Similarly the above estimators from the clock time model can be used for prediction.

6.6 Spot Cross-Volatility Estimation: New Modeling Aspects

In the preceding sections the estimation of spot volatility as well as the estimation of spot cross-volatility in the simplified case of synchronous trading were considered. Now, the realistic multivariate case with non-synchronous trading times is treated. For this purpose we first propose a new model for non-synchronous tick-by-tick data.

6.6.1 A New Transaction Time Model for Non-Synchronous Data

The model we propose is a random walk model in transaction time in each component (with time-varying volatilities) plus an interpolation given by a stochastic differential equation. It allows to handle the different transaction times and the definition and estimation of the covariances of the log-returns. In our model, each component evolves in an individual transaction time which agrees with the economic intuition that each security price responds to its own information flow.

More precisely, let's consider the discrete time log-price processes $X_{t_j^{(s)},s}$, $s = 1, \ldots, S$. For notational convenience $X_{t_j^{(s)},s}$ is denoted briefly by $X_{t_j^{(s)}}$ if there is no danger of confusion. We assume that

$$X_{t_j^{(s)}} = X_{t_{j-1}^{(s)}} + Z_{t_j^{(s)}} \tag{6.44}$$

with log-returns $Z_{t_j^{(s)}} \sim \mathcal{N}\big(0, (\Sigma_{t_j^{(s)}})_{ss}\big)$. If the transactions times $t_j^{(s)}$ are different in each component s (which they usually are) there is no natural definition of the covariance of the log-returns. To overcome this problem we assume at this point that all components are interpolated between the transaction times according to the stochastic differential equation

$$d\mathbf{X}(t) = \operatorname{diag}\big(|t_{j_1}^{(1)} - t_{j_1-1}^{(1)}|^{-1/2}, \ldots, |t_{j_S}^{(S)} - t_{j_S-1}^{(S)}|^{-1/2}\big)\ \Gamma(t)\, d\mathbf{W}(t) \tag{6.45}$$

if $t \in [t_{j_s-1}^{(s)}, t_{j_s}^{(s)})$ for all s. We have $\mathbf{X}(t) = (X_1(t), \ldots, X_S(t))$, diag denotes a diagonal matrix, $\mathbf{W}(t)$ is a multivariate Brownian motion, and $\Gamma(t)\Gamma^T(t) = \Sigma(t)$ is the (time-varying) covariance matrix of the log-returns. Note, we set $X_{t_j^{(s)}} = X_s(t_j^{(s)})$ (and analogous for other variables).

It is easy to check that in this model

$$\operatorname{Var}(Z_{t_j^{(s)}}) \;=\; \frac{1}{|t_j^{(s)} - t_{j-1}^{(s)}|} \int_{t_{j-1}^{(s)}}^{t_j^{(s)}} (\Sigma(t))_{ss}\, dt \tag{6.46}$$

$$\approx\; (\Sigma_{t_j^{(s)}})_{ss} \tag{6.47}$$

and

$$\begin{aligned}
&\operatorname{Cov}(Z_{t_j^{(s_1)}}, Z_{t_k^{(s_2)}}) \\
&= \frac{1}{|t_j^{(s_1)} - t_{j-1}^{(s_1)}|^{1/2} |t_k^{(s_2)} - t_{k-1}^{(s_2)}|^{1/2}} \int_{[t_{j-1}^{(s_1)}, t_j^{(s_1)}) \cap [t_{k-1}^{(s_2)}, t_k^{(s_2)})} (\Sigma(t))_{s_1 s_2}\, dt \qquad (6.48) \\
&\approx \frac{|[t_{j-1}^{(s_1)}, t_j^{(s_1)}) \cap [t_{k-1}^{(s_2)}, t_k^{(s_2)})|}{|t_j^{(s_1)} - t_{j-1}^{(s_1)}|^{1/2} |t_k^{(s_2)} - t_{k-1}^{(s_2)}|^{1/2}} \left(\Sigma_{\min\{t_j^{(s_1)}, t_k^{(s_2)}\}}\right)_{s_1 s_2}. \qquad (6.49)
\end{aligned}$$

Model assumptions:

(i) Given the time-varying covariance matrix $\Sigma(t)$ we assume that (6.47) and (6.49) hold.

(ii) We assume that $\Sigma(t)$ evolves slowly in time, that is we assume some smoothness on $\Sigma(t)$.

As mentioned earlier the smoothness assumption (ii) needs not to be specified any further because we do not make any use of it formally. However, without this assumption the approximate relations in (6.47) and (6.49) would not be correct.

Remarks: An alternative view is to assume that (6.46) and (6.48) hold, and to use (6.47) and (6.49) as a numerical approximation in our calculations. A drawback of the above model is that it depends on the observation times $t_j^{(s)}$. In our opinion this cannot be avoided if one wants to work in transaction time rather than in clock time.

If one prefers a clock time model all results (given later) continue to hold with some modifications. In this case one will start with the stochastic differential equation $d\mathbf{X}(t) = \Gamma(t) d\mathbf{W}(t)$ which gives

$$\operatorname{Var}(Z_{t_j^{(s)}}) \approx |t_j^{(s)} - t_{j-1}^{(s)}|\, (\Sigma_{t_j^{(s)}})_{ss} \qquad (6.50)$$

and

$$\operatorname{Cov}(Z_{t_j^{(s_1)}}, Z_{t_k^{(s_2)}}) \approx |[t_{j-1}^{(s_1)}, t_j^{(s_1)}) \cap [t_{k-1}^{(s_2)}, t_k^{(s_2)})|\, \left(\Sigma_{\min\{t_j^{(s_1)}, t_k^{(s_2)}\}}\right)_{s_1 s_2} \qquad (6.51)$$

instead of (6.47) and (6.49). The estimation method for the covariances presented later can also be applied to this clock time model (see Section 6.8).

6.6.2 Non-Standard State-Space Models for Non-Synchronous Data

A multivariate extension of the observation equation (6.2) and the state equation (6.44) can be combined to form a non-standard state-space model (see Section 6.7.1). The components of the state and observation processes evolve non-synchronously in different (discrete) times. To the author's knowledge such kind of state-space models have not been considered before. The properties of this non-standard state-space model differ significantly from those of standard state-space models. In particular, the Markov property of the state process does not hold. As a consequence, standard methods for filtering such as particle filters are not applicable for non-synchronous state-space models. In the course of this work, a new particle filter which can cope with this situation is developed. The results obtained in Section 6.7 (including the particle filter) do not only hold for our specific state-space model. They can be easily transfered to other non-synchronous state-space models. In particular, the generalization to linear, Gaussian state equations is straightforward.

6.7 On-Line Estimation of Spot Cross-Volatility

6.7.1 A State-Space Model with Non-Synchronous Observations and States

Now, the non-synchronous state-space model defined through the equations (6.2) and (6.44) is considered. For convenience, it is restated as

$$\begin{aligned} Y_{t_j^{(s)}} &= g_{t_j^{(s)}}(\exp[X_{t_j^{(s)}}]), &(6.52)\\ X_{t_j^{(s)}} &= X_{t_{j-1}^{(s)}} + Z_{t_j^{(s)}}, &(6.53)\end{aligned}$$

for $s = 1, \ldots, S$ and $Z_{t_j^{(s)}} \sim \mathcal{N}(0, (\Sigma_{t_j^{(s)}})_{ss})$. The initial efficient prices $\exp[X_{t_1^{(s)}}]$ are assumed to be uniformly distributed on $A_{t_1^{(s)}}$. The covariances of the log-returns $Z_{t_j^{(s)}}$ (defined in (6.48)) are rewritten as

$$\mathrm{Cov}(Z_{t_j^{(s_1)}}, Z_{t_k^{(s_2)}}) = f(t_j^{(s_1)}, t_k^{(s_2)})(\Sigma_{\min\{t_j^{(s_1)}, t_k^{(s_2)}\}})_{s_1 s_2}. \tag{6.54}$$

The function f, which is defined through

$$f(t_j^{(s_1)}, t_k^{(s_2)}) = \frac{|[t_{j-1}^{(s_1)}, t_j^{(s_1)}) \cap [t_{k-1}^{(s_2)}, t_k^{(s_2)})|}{|t_j^{(s_1)} - t_{j-1}^{(s_1)}|^{1/2} |t_k^{(s_2)} - t_{k-1}^{(s_2)}|^{1/2}}, \tag{6.55}$$

gives the “normalized overlapping time” of the log-returns $Z_{t_j^{(s_1)}}$ and $Z_{t_k^{(s_2)}}$. We mention that in the continuous time model discussed in Section 6.6.1 the variances and covariances of the log-returns are only approximately equal to $(\Sigma_{t_j^{(s)}})_{ss}$ and $f(t_j^{(s_1)}, t_k^{(s_2)})(\Sigma_{\min\{t_j^{(s_1)}, t_k^{(s_2)}\}})_{s_1 s_2}$, respectively. However, in the following we only treat the discrete time state-space model given above. For simplicity we therefore assume that the equalities hold.

As mentioned earlier, the equations (6.52) and (6.53) form a non-standard state-space model because the components of the state and observation processes evolve non-synchronously in different discrete times. For two securities, the setup is visualized in Figure 6.6. The time assignment $\min\{t_j^{(s_1)}, t_k^{(s_2)}\}$ in (6.54) is ad hoc (one could also choose $t_j^{(s_1)}$ or $t_k^{(s_2)}$). Note, this is a result of Σ_t being slowly varying. The components of the covariance matrix Σ_t are defined in certain (average) transaction times. More precisely, the diagonal entry $(\Sigma_t)_{ss}$ (that is the variance of the log-returns of security s) is defined with respect to the unit time step of the transaction time $\{t_j^{(s)}\}_{j=1}^{T_s}$. For the off-diagonal entry $(\Sigma_t)_{s_1 s_2}$, $s_1 \neq s_2$, the situation is more subtle. $(\Sigma_t)_{s_1 s_2}$ is obtained from the interpolation given in (6.45). The resulting formula (6.55) can be interpreted as a specific overlapping geometric average of linear interpolated times in each component. Consider the situation shown in Figure 6.6. To calculate $f(t_4^{(1)}, t_2^{(2)})$ one needs the overlapping linear interpolated times of the log-returns $z_{t_4^{(1)}}$ and $z_{t_2^{(2)}}$ which are given by

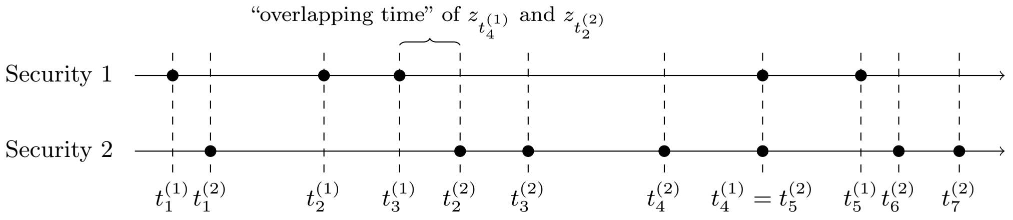

Figure 6.6: An example of non-synchronous trading times for two securities.

$(t_2^{(2)} - t_3^{(1)})/(t_4^{(1)} - t_3^{(1)})$ and $(t_2^{(2)} - t_3^{(1)})/(t_2^{(2)} - t_1^{(2)})$, respectively. The geometric average then gives (6.55).

6.7.2 An Original Particle Filter for Non-Synchronous State-Space Models

For non-synchronous state-space models, standard particle filters do not apply because the components of the state variable cannot be updated simultaneously. Now, a new particle filter is developed.

For notational convenience the following is only shown for two securities. However, the general case works analogous. Assume $t_j^{(1)}$ and $t_k^{(2)}$ are the most recent transaction times of the securities 1 and 2, respectively. In addition, suppose $t_j^{(1)} \geq t_k^{(2)}$. With this setting, the filtering distribution is given by $p(x_{t_{1:j}^{(1)}}, x_{t_{1:k}^{(2)}} | y_{t_{1:j}^{(1)}}, y_{t_{1:k}^{(2)}})$. The goal is to construct a particle filter which approximates the filtering distribution through particles

$$\left\{ x^i_{t_{1:j}^{(1)}}, x^i_{t_{1:k}^{(2)}}, \omega^i_{t_j^{(1)}} \right\}_{i=1}^N .$$

For this purpose we consider the following decomposition of the filtering distribution. It is easy to check that

$$\begin{aligned} & p(x_{t_{1:j}^{(1)}}, x_{t_{1:k}^{(2)}} | y_{t_{1:j}^{(1)}}, y_{t_{1:k}^{(2)}}) \\ & \quad \propto \; p(y_{t_j^{(1)}} | y_{t_{1:j-1}^{(1)}}, x_{t_j^{(1)}}) \; p(x_{t_j^{(1)}} | x_{t_{1:j-1}^{(1)}}, x_{t_{1:k}^{(2)}}) p(x_{t_{1:j-1}^{(1)}}, x_{t_{1:k}^{(2)}} | y_{t_{1:j-1}^{(1)}}, y_{t_{1:k}^{(2)}}). \end{aligned} \tag{6.56}$$

In standard state-space models two simplifications are possible. First, the likelihood

$$p(y_{t_j^{(1)}} | y_{t_{1:j-1}^{(1)}}, x_{t_j^{(1)}})$$

simplifies to $p(y_{t_j^{(1)}} | x_{t_j^{(1)}})$. Because of dependencies between the observed transaction prices induced by the market microstructure noise model this is not the case here (compare Section 6.3.2). Second and more importantly, the set of conditioning variables in the transition prior

$$p(x_{t_j^{(1)}} | x_{t_{1:j-1}^{(1)}}, x_{t_{1:k}^{(2)}})$$

can not be reduced as in standard state-space models. This follows from the fact that in non-synchronous state-space models the Markov property of the state transition does not hold. In general, the transition prior incorporates the full history of the states. However, there are some cases when simplifications of the transition prior for non-synchronous state-space models are possible which are summarized in the following proposition.

Proposition 6.2. *(i) If $t^{(1)}_{j-1} \geq t^{(2)}_k$ then the transition prior simplifies to*

$$p(x_{t^{(1)}_j}|x_{t^{(1)}_{1:j-1}}, x_{t^{(2)}_{1:k}}) = p(x_{t^{(1)}_j}|x_{t^{(1)}_{j-1}}).$$

(ii) Let l_1 and l_2 be (the largest) integers such that $t^{(1)}_{l_1} = t^{(2)}_{l_2}$, $1 \leq l_1 < j$ and $1 \leq l_2 \leq k$, then

$$p(x_{t^{(1)}_j}|x_{t^{(1)}_{1:j-1}}, x_{t^{(2)}_{1:k}}) = p(x_{t^{(1)}_j}|x_{t^{(1)}_{l_1:j-1}}, x_{t^{(2)}_{l_2:k}}).$$

Our particle filter generates particles sequentially in time making use of the relation (6.56) and sequential importance sampling. Analogous to the case discussed in Section 6.3.2 it is possible to sample from the optimal proposal which is given by $p(x_{t^{(1)}_j}|y_{t^{(1)}_{1:j}}, x_{t^{(1)}_{1:j-1}}, x_{t^{(2)}_{1:k}})$ in this setting. This gives the following particle filter:
Suppose weighted particles

$$\{x^i_{t^{(1)}_{1:j-1}}, x^i_{t^{(2)}_{1:k}}, \omega^i_{\max\{t^{(1)}_{j-1}, t^{(2)}_k\}}\}_{i=1}^N$$

approximating

$$p(x_{t^{(1)}_{1:j-1}}, x_{t^{(2)}_{1:k}}|y_{t^{(1)}_{1:j-1}}, y_{t^{(2)}_{1:k}})$$

are given; then

- For $i = 1, \ldots, N$:
  - Sample $x^i_{t^{(1)}_j} \sim p(x_{t^{(1)}_j}|y_{t^{(1)}_{1:j}}, x^i_{t^{(1)}_{1:j-1}}, x^i_{t^{(2)}_{1:k}})$.
  - Compute importance weights

$$\begin{aligned} \breve{\omega}^i_{t^{(1)}_j} &\propto \omega^i_{\max\{t^{(1)}_{j-1}, t^{(2)}_k\}} \frac{p(y_{t^{(1)}_j}|y_{t^{(1)}_{1:j-1}}, x^i_{t^{(1)}_j})\; p(x_{t^{(1)}_j}|x^i_{t^{(1)}_{1:j-1}}, x^i_{t^{(2)}_{1:k}})}{p(x^i_{t^{(1)}_j}|y_{t^{(1)}_{1:j}}, x^i_{t^{(1)}_{1:j-1}}, x^i_{t^{(2)}_{1:k}})} \\ &= \omega^i_{\max\{t^{(1)}_{j-1}, t^{(2)}_k\}} p(y_{t^{(1)}_j}|y_{t^{(1)}_{1:j-1}}, x^i_{t^{(1)}_{1:j-1}}, x^i_{t^{(2)}_{1:k}}). \end{aligned}$$

- For $i = 1, \ldots, N$:
  - Normalize importance weights $\omega^i_{t^{(1)}_j} = \breve{\omega}^i_{t^{(1)}_j} / (\sum_{k=1}^N \breve{\omega}^k_{t^{(1)}_j})$.
- Obtain particles $\{x^i_{t^{(1)}_{1:j}}, x^i_{t^{(2)}_{1:k}}, \omega^i_{t^{(1)}_j}\}_{i=1}^N$ which approximate $p(x_{t^{(1)}_{1:j}}, x_{t^{(2)}_{1:k}}|y_{t^{(1)}_{1:j}}, y_{t^{(2)}_{1:k}})$.

Again, a resampling step is required tackling the weight degeneracy.

Now, the concrete state-space model given by (6.52) and (6.53) is considered. To derive the optimal proposal and the computation of the importance weights we consider the distribution of the log-returns $Z_{t^{(s)}_j} = X_{t^{(s)}_j} - X_{t^{(s)}_{j-1}}$. It is easy to establish that the joint distribution is given by

$$p(z_{t^{(1)}_{2:j}}, z_{t^{(2)}_{2:k}}) = \mathcal{N}\Big((z_{t^{(1)}_{2:j}}, z_{t^{(2)}_{2:k}})^T|\mathbf{0}; \mathbf{S}\Big), \tag{6.57}$$

where the entries of $\mathbf{S}$ are defined through (6.54). Let's introduce the notation $\mathbf{z} = (z_{t^{(1)}_{2:j-1}}, z_{t^{(2)}_{2:k}})^T$ and

$$\mathbf{S} = \begin{pmatrix} S_{11} & S_{12} \\ S_{12}^T & S_{22} \end{pmatrix},$$

where $S_{11} = (\Sigma_{t_j^{(1)}})_{11}$.

Proposition 6.3. *The optimal proposal is a truncated conditional normal distribution given by*

$$p(x_{t_j^{(1)}} | y_{t^{(1)}_{1:j}}, x_{t^{(1)}_{1:j-1}}, x_{t^{(2)}_{1:k}}) \propto \mathcal{N}(x_{t_j^{(1)}} | \overline{\mu} + x_{t^{(1)}_{j-1}}; \overline{\Sigma})\big|_{\log A_{t_j^{(1)}}} \tag{6.58}$$

with $\overline{\mu} = S_{12}S_{22}^{-1}\mathbf{z}$ *and* $\overline{\Sigma} = S_{11} - S_{12}S_{22}^{-1}S_{12}^T$. *If* $A_{t_j^{(1)}}$ *is an interval, then the importance weights are computed through*

$$\breve{\omega}^i_{t_j^{(1)}} \propto \omega^i_{\max\{t^{(1)}_{j-1}, t^{(2)}_k\}} \Big\{ \Phi(\sup \log A_{t_j^{(1)}} | \overline{\mu} + x^i_{t^{(1)}_{j-1}}; \overline{\Sigma}) - \Phi(\inf \log A_{t_j^{(1)}} | \overline{\mu} + x^i_{t^{(1)}_{j-1}}; \overline{\Sigma}) \Big\}, \tag{6.59}$$

where $\Phi(\cdot|\mu, \sigma^2)$ *denotes the distribution function of* $\mathcal{N}(\mu, \sigma^2)$.

Proof. To derive the optimal proposal it suffices to show that $p(z_{t_j^{(1)}} | z_{t^{(1)}_{2:j-1}}, z_{t^{(2)}_{2:k}}) = \mathcal{N}(z_{t_j^{(1)}} | \overline{\mu}; \overline{\Sigma})$. This follows directly from (6.57). The expression for the importance weights is obtained similar to that in Proposition 6.1.

Note that the covariance matrix S_{22} is of dimension $j + k - 3$ which grows over time. This renders the algorithm impractical because S_{22} needs to be inverted to compute $\overline{\mu}$ and $\overline{\Sigma}$ in every iteration of the particle filter. Hence, a reduction of the dimension of S_{22} is required. For this purpose, we replace (6.58) by $p(x_{t_j^{(1)}} | y_{t^{(1)}_{1:j}}, x_{t^{(1)}_{l_1:j-1}}, x_{t^{(2)}_{l_2:k}})$, where l_1 and l_2 are close to $j - 1$ and k, respectively. The indices l_1 and l_2 are selected by applying one of the following three rules.

Rule 1: If $t^{(1)}_{j-1} \geq t^{(2)}_k$ then $p(x_{t_j^{(1)}} | y_{t^{(1)}_{1:j}}, x_{t^{(1)}_{1:j-1}}, x_{t^{(2)}_{1:k}}) = p(x_{t_j^{(1)}} | y_{t^{(1)}_{1:j}}, x_{t^{(1)}_{j-1}})$.

Rule 2: Let l_1, and l_2 be the largest integers such that $t^{(1)}_{l_1} = t^{(2)}_{l_2}$, $1 \leq l_1 < j$ and $1 \leq l_2 \leq k$, then $p(x_{t_j^{(1)}} | y_{t^{(1)}_{1:j}}, x_{t^{(1)}_{1:j-1}}, x_{t^{(2)}_{1:k}}) = p(x_{t_j^{(1)}} | y_{t^{(1)}_{1:j}}, x_{t^{(1)}_{l_1:j-1}}, x_{t^{(2)}_{l_2:k}})$.

Rule 3: Whenever the cardinality of the set $\{x_{t^{(1)}_{l'_1:j-1}}, x_{t^{(2)}_{l'_2:k}}\}$ is larger than K, the set is reduced to $\{x_{t^{(1)}_{l_1:j-1}}, x_{t^{(2)}_{l_2:k}}\}$. The reduction is done by removing log-returns from the set (in clock time order) as long as the cardinality is larger than K and the conditions $l_1 < j$ and $l_2 \leq k$ are satisfied. The conditions imply that at least one log-return of each security remains in the set.

The reductions in the first two rules follow directly from Proposition 6.2. They are exact and, therefore, are applied with priority. In the situation shown in Figure 6.6, rules 1 and 2 can be applied to obtain, for instance, $p(x_{t_3^{(2)}} | y_{t^{(2)}_{1:3}}, x_{t^{(1)}_{1:3}}, x_{t^{(2)}_{1:2}}) = p(x_{t_3^{(2)}} | y_{t^{(2)}_{1:3}}, x_{t_2^{(2)}})$ and $p(x_{t_6^{(2)}} | y_{t^{(2)}_{1:6}}, x_{t^{(1)}_{1:5}}, x_{t^{(2)}_{1:5}}) = p(x_{t_6^{(2)}} | y_{t^{(2)}_{1:6}}, x_{t^{(1)}_{4:5}}, x_{t_5^{(2)}})$, respectively. The third rule is an approximation which is used in the case when both rule 1 and rule 2 do not apply. Rule 3 can be justified by the fact that the influence of older log-returns on the most recent log-return vanishes rapidly so that the bias introduced will be small if K is reasonably large. The choice of K is discussed in Section 6.9.

6.7.3 EM-Type Algorithms for Non-Synchronous Observations and States

Again, the following is only presented for two securities. The generalization is straightforward.

First, a non-sequential EM algorithm for the estimation of a constant covariance matrix Σ is discussed. We consider the distribution of the log-returns instead of the transition prior (see (6.8)) which gives

$$\begin{aligned}\mathcal{Q}(\Sigma|\hat{\Sigma}^{(m)}) &= \mathbf{E}_{\hat{\Sigma}^{(m)}}[\log p_\Sigma(X_{t^{(1)}_{1:T_1}}, X_{t^{(2)}_{1:T_2}}, y_{t^{(1)}_{1:T_1}}, y_{t^{(2)}_{1:T_2}})|y_{t^{(1)}_{1:T_1}}, y_{t^{(2)}_{1:T_2}}] \\ &= \text{const} \ + \ \mathbf{E}_{\hat{\Sigma}^{(m)}}[\log p_\Sigma(Z_{t^{(1)}_{2:T_1}}, Z_{t^{(2)}_{2:T_2}})|y_{t^{(1)}_{1:T_1}}, y_{t^{(2)}_{1:T_2}}].\end{aligned}$$

An approximation of this conditional expectation can be obtained based on smoothing particles

$$\left\{x^i_{t^{(1)}_{1:T_1}}, x^i_{t^{(2)}_{1:T_2}}, \omega^i_{\max\{t^{(1)}_{T_1}, t^{(2)}_{T_2}\}}\right\}_{i=1}^N$$

which approximate the smoothing distribution $p(x_{t^{(1)}_{1:T_1}}, x_{t^{(2)}_{1:T_2}}|y_{t^{(1)}_{1:T_1}}, y_{t^{(2)}_{1:T_2}})$. As a result of (6.57) this leads to

$$\begin{aligned}&\mathbf{E}_{\hat{\Sigma}^{(m)}}[\log p_\Sigma(Z_{t^{(1)}_{2:T_1}}, Z_{t^{(2)}_{2:T_2}})|y_{t^{(1)}_{1:T_1}}, y_{t^{(2)}_{1:T_2}}] \\ &= \mathbf{E}_{\hat{\Sigma}^{(m)}}[\log p_{\mathbf{S}}(Z_{t^{(1)}_{2:T_1}}, Z_{t^{(2)}_{2:T_2}})|y_{t^{(1)}_{1:T_1}}, y_{t^{(2)}_{1:T_2}}] \\ &\approx \sum_{i=1}^N \omega^i_{\max\{t^{(1)}_{T_1}, t^{(2)}_{T_2}\}} \frac{1}{2}\left[S\log 2\pi + \log|\mathbf{S}| + \text{tr}\left\{\mathbf{S}^{-1}(z^i_{t^{(1)}_{2:T_1}}, z^i_{t^{(2)}_{2:T_2}})^T(z^i_{t^{(1)}_{2:T_1}}, z^i_{t^{(2)}_{2:T_2}})\right\}\right], \quad (6.60)\end{aligned}$$

where $z^i_{t^{(s)}_h} = x^i_{t^{(s)}_h} - x^i_{t^{(s)}_{h-1}}$.

From (6.60) it is clear that the particle approximation $\hat{\mathcal{Q}}(\Sigma|\hat{\Sigma}^{(m)})$ can be easily maximized with respect to $\mathbf{S}$ if we ignore the fact that certain entries of $\mathbf{S}$ are equal up to a proportionality factor. This maximization leads to the following estimate of $\mathbf{S}$

$$\hat{\mathbf{S}}^{(m+1)} = \sum_{i=1}^N \omega^i_{\max\{t^{(1)}_{T_1}, t^{(2)}_{T_2}\}}(z^i_{t^{(1)}_{2:T_1}}, z^i_{t^{(2)}_{2:T_2}})^T(z^i_{t^{(1)}_{2:T_1}}, z^i_{t^{(2)}_{2:T_2}}). \quad (6.61)$$

The estimators derived below also hold for more than two securities. That is, from now on we assume that we have S securities. From (6.61) it follows that a natural estimator of the covariance matrix Σ is given, componentwise, by

$$(\hat{\Sigma}^{(m+1)})_{s_1 s_2} = \frac{1}{|H_{s_1 s_2}|}\sum_{(h_1,h_2)\in H_{s_1 s_2}}\sum_{i=1}^N \omega^i_{\max\{t^{(s_1)}_{T_{s_1}}, t^{(s_2)}_{T_{s_2}}\}} \frac{z^i_{t^{(s_1)}_{h_1}} z^i_{t^{(s_2)}_{h_2}}}{f(t^{(s_1)}_{h_1}, t^{(s_2)}_{h_2})}, \quad (6.62)$$

where $H_{s_1 s_2}$ is a set of paired time stamps defined through

$$H_{s_1 s_2} = \left\{(h_1, h_2) : [t^{(s_1)}_{h_1 - 1}, t^{(s_1)}_{h_1}) \cap [t^{(s_2)}_{h_2 - 1}, t^{(s_2)}_{h_2}) \neq \emptyset\right\}$$

and $s_1, s_2 \in \{1, 2, \dots, S\}$. Notice, the summands in (6.62) are scaled according to the inverse of the associated normalized overlapping times. In general $\hat{\Sigma}^{(m+1)}$ only approximately maximizes $\hat{\mathcal{Q}}(\Sigma|\hat{\Sigma}^{(m)})$. The estimator which maximizes $\hat{\mathcal{Q}}(\Sigma|\hat{\Sigma}^{(m)})$ cannot be obtained in closed form. It could be computed through a high-dimensional numerical optimization procedure. Because this is computationally inefficient we suggest to use the easy-to-compute estimator (6.62).

Analogous to Section 6.3.3 we now propose a localized variant of (6.62) which can be computed recursively. Again, the crucial step is the transition from the smoothing particles to filtering particles.

Let's define the joint transaction time $\{t_v\}_{v=1}^T$ as the ordered set of time stamps

$$t_1 = \min\{t_1^{(1)}, \dots, t_S^{(S)}\} \qquad \text{and} \qquad \{t_2, \dots, t_T\} = \{t_{2:T_1}^{(1)}, \dots, t_{2:T_S}^{(S)}\}.$$

Because multiple times are included only once, $t_v < t_{v+1}$ holds. For notational convenience the joint transaction time includes only one of the first transaction times $t_1^{(1)}, \dots, t_S^{(S)}$. This implies that the covariance estimate can be updated (for the first time) at time t_2 because then at least one security traded two times (which implies that one log-return is available).

It is easy to see (compare Section 6.3.3) that based on filtering particles a recursive version of (6.62) is given by

$$(\hat{\Sigma}_{t_v})_{s_1 s_2} = (1 - \lambda_{v,s_1,s_2})(\hat{\Sigma}_{t_{v-1}})_{s_1 s_2} + \lambda_{v,s_1,s_2}(\breve{\Sigma}_{t_v})_{s_1 s_2}, \tag{6.63}$$

for $v = 2, 3, \dots, \max\{w : t_w \leq t_{T_{s_1}}^{(s_1)} \wedge t_w \leq t_{T_{s_2}}^{(s_2)}\}$, where

$$(\breve{\Sigma}_{t_v})_{s_1 s_2} = \begin{cases} \sum_{i=1}^N \omega^i_{\max\{t_{h_1^v}^{(s_1)}, t_{h_2^v}^{(s_2)}\}} \frac{z^i_{t_{h_1^v}^{(s_1)}} z^i_{t_{h_2^v}^{(s_2)}}}{f(t_{h_1^v}^{(s_1)}, t_{h_2^v}^{(s_2)})} & \text{if} \quad t_{h_1^v}^{(s_1)} = t_v \ \text{ or } \ t_{h_2^v}^{(s_2)} = t_v \\ (\hat{\Sigma}_{t_{v-1}})_{s_1 s_2} & \text{else} \end{cases} \tag{6.64}$$

with $h_s^v = \min\{h_s : t_{h_s}^{(s)} \geq t_v\}$. To comply with (6.62) the initial covariance estimate $\hat{\Sigma}_{t_1}$ needs to be set to the zero matrix of dimension S.

From (6.64) it follows that the estimate of the covariance of the securities s_1 and s_2 is updated at time t_v if one (or both) of the securities trades at time t_v. If none of the two securities trades at time t_v (6.63) and (6.64) imply $(\hat{\Sigma}_{t_v})_{s_1 s_2} = (\hat{\Sigma}_{t_{v-1}})_{s_1 s_2}$. A practical issue is that in an on-line application the update $(\breve{\Sigma}_{t_v})_{s_1 s_2}$ cannot always be computed at time t_v. Assume $t_j^{(1)} = t_v$, $t_j^{(1)} > t_k^{(2)}$, and $t_j^{(1)} < t_{k+1}^{(2)}$. Then, $(\breve{\Sigma}_{t_v})_{s_1 s_2}$ cannot be computed because the transaction at time $t_{k+1}^{(2)}$ is not available yet. In practice, the particle filter always uses the most recent available estimate to simulate the transitions (see Section 6.9 and compare Section 6.3.4).

It remains to specify the step size λ_{v,s_1,s_2}. For time-constant covariance matrices we propose to use

$$\lambda_{v,s_1,s_2} = \left|\left\{t \in \{t_{2:T_{s_1}}^{(s_1)}, t_{2:T_{s_2}}^{(s_2)}\} : t \leq t_v\right\}\right|^{-\gamma},$$

analogous to (6.20), with $\gamma \in (\frac{1}{2}, 1)$. Note that the set $\{t_{2:T_{s_1}}^{(s_1)}, t_{2:T_{s_2}}^{(s_2)}\}$ basically contains the joint transaction times of the securities s_1 and s_2 up to time t_v (multiple times are included only once). The magnitude of this set gives the number of updates for the component $(\Sigma_t)_{s_1 s_2}$ (up to time t_v). In the case of two securities the set is equal to the set of the joint transaction times of all securities $\{t_2, t_3, \ldots, t_v\}$ (besides t_1). Thus, for two securities we obtain $\lambda_{v,s_1,s_2} = (v-1)^{-\gamma}$ (compare (6.20)). We mention that if we use the smoothing particles to compute $\hat{\Sigma}_{t_T}$ through (6.63) and set $\gamma = 1$ then $\hat{\Sigma}_{t_T}$ is equal to (6.62).

In the time-varying covariance case a constant step size $\lambda_{v,s_1,s_2} = \lambda_{s_1,s_2}$ or a step size that fluctuates around some constant value λ_{s_1,s_2} will be used as in (6.21). For more details on the practical choice of the step size see Section 6.9. We remark that the fine-tuning of the volatility estimator in the time-varying case proposed in Section 6.5 is not directly transferable to the non-synchronous cross-volatility estimator (6.63). Adaptive step size selection as well as bias (or quasi mean squared error) correction for (6.63) should be considered in further research.

Finally, we mention that in contrast to the synchronous trading case (6.19), the covariance estimates given by (6.62) and (6.63) are not necessarily positive (semi-) definite. A practical solution of this problem is to increase the diagonal entries in case of non-positive definiteness.

6.8 Clock Time Covariance Estimation

In the preceding section an algorithm for the estimation of transaction time covariance matrices Σ_t was derived. This method can also be used to estimate the covariance in clock time by making a slight change to the definition of the covariance of the log-returns (6.54). The function f defined in (6.55) needs to be replaced by

$$\tilde{f}(t_j^{(s_1)}, t_k^{(s_2)}) = |[t_{j-1}^{(s_1)}, t_j^{(s_1)}) \cap [t_{k-1}^{(s_2)}, t_k^{(s_2)})|.$$

$\tilde{f}$ gives the overlapping time of two log-returns in clock time (compare (6.50) and (6.51)). In contrast to f, it is not normalized. Note that the particle filter and the EM-type algorithm do not change except that f is replaced by $\tilde{f}$ (compare Section 6.4.1). Although this gives a plausible estimator for the clock time covariance, we are sceptical about its applicability is practice. We expect it to be highly unstable as a results of the high variability of the durations $t_j^{(s)} - t_{j-1}^{(s)}$. In particular, the positive definiteness of the covariance estimate will be an issue.

It is desirable to construct an alternative clock time estimator similar to $\hat{\Sigma}^c_{\text{alt}}(t_j)$ (see Section 6.4.2) for the non-synchronous multivariate case. A straightforward adaptation of $\hat{\Sigma}^c_{\text{alt}}(t_j)$ leads to

$$\big(\hat{\Sigma}^c_{\text{alt}}(t_v)\big)_{s_1 s_2} = \frac{(\hat{\Sigma}_{t_v})_{s_1 s_2}}{\big(\bar{\delta}_v^{(s_1)}\big)^{1/2}\big(\bar{\delta}_v^{(s_2)}\big)^{1/2}}$$

with averaged duration times $\bar{\delta}_v^{(s)}$ which are computed through $\bar{\delta}_v^{(s)} = (1-\lambda_{v,s})\bar{\delta}_{v-1}^{(s)} + \lambda_{v,s}\breve{\delta}_v^{(s)}$ and

$$\breve{\delta}_v^{(s)} = \begin{cases} t_{h_s^v}^{(s)} - t_{h_s^v-1}^{(s)} & \text{if} \quad t_{h_s^v}^{(s)} = t_v, \\ \bar{\delta}_{v-1}^{(s)} & \text{else.} \end{cases}$$

We mention that the case discussed here is significantly more complicated than the situation of Section 6.4.2. Therefore, the reasoning given there does not transfer directly to the present case.

A closer investigation of these clock time estimators is beyond the scope of this dissertation. However, this is highly desirable and should be considered in future research.

6.9 Implementation Details

Algorithms

In sections 6.2 through 6.5, our estimation method was developed for the (artificial) multivariate case with synchronous trading times. Here, this algorithm is summarized for (univariate) spot volatility estimation (SVE) in transaction time. It is mentioned that the SVE algorithm (in its simplest version without bias correction and adaptive step size selection) is a special case of the algorithm for spot cross-volatility estimation (SCVE) given later. However, it is worth to be stated separately because it is much easier to implement. In fact, it just needs a few lines in R. For the available computer code see Chapter 8. (For notational convenience we use the matrix notation also in the univariate case, that is $\Sigma_t = \sigma_t^2$ below.)

Algorithm: On-line Spot Volatility Estimation (SVE)

Initialization:

- Set the initial volatility estimate $\hat{\Sigma}_{t_2} = \hat{\Sigma}^{\text{pf}}_{t_2}$, the number of particles N, and the step size λ (or α and β) for time-varying volatility estimation or γ for time-constant volatility estimation.
- For $i = 1, \dots, N$: Generate sample $x^i_{t_1}$ such that $\exp[x^i_{t_1}]$ is uniformly distributed on A_{t_1}.

On-line spot volatility estimation: (for $j = 2, \dots, T$)

- For $i = 1, \dots, N$:
  - Generate $x^i_{t_j}$ from the optimal proposal $\mathcal{N}(x_{t_j}|x^i_{t_{j-1}}; \hat{\Sigma}^{\text{pf}}_{t_j})\big|_{\log A_{t_j}}$.
  - Compute the importance weight
  $$\breve{\omega}^i_{t_j} \propto \omega^i_{t_{j-1}} \Big\{ \Phi\big(\sup \log A_{t_j} | x^i_{t_{j-1}}; \hat{\Sigma}^{\text{pf}}_{t_j}\big) - \Phi\big(\inf \log A_{t_j} | x^i_{t_{j-1}}; \hat{\Sigma}^{\text{pf}}_{t_j}\big) \Big\}.$$
- For $i = 1, \dots, N$: Normalize the importance weight $\omega^i_{t_j} = \breve{\omega}^i_{t_j} / \sum_{k=1}^N \breve{\omega}^k_{t_j}$.
- Update the volatility estimate and obtain $\hat{\Sigma}^{\text{pf}}_{t_{j+1}}$ for the next iteration.
- If the effective sample size $\text{ESS}(\{\omega^i_{t_j}\}_{i=1}^N) < 0.2N$, then resample the particles using, for instance, the residual resampling scheme (Douc, Cappé, and Moulines 2005).

In the time-varying case the particle filter uses $\hat{\Sigma}^{\text{pf}}_{t_j} = \tilde{\Sigma}^*_{t_j|t_{j-1}}$ (see Section 6.3.4). As the estimator of Σ_{t_j} we usually use $\tilde{\Sigma}^*_{t_j|t_j}$ from (6.43) (if not otherwise stated) and sometimes $\tilde{\Sigma}_{t_j|t_j}$ from (6.35). In addition, the adaptation procedure described in Section 6.5.1 is applied. α and

β are used from past experience or determined as described in step 4. In the time-constant estimation case we simply use $\hat{\Sigma}^{\mathrm{pf}}_{t_j} = \hat{\Sigma}_{t_{j-1}}$. More details on the choice and initialization of the parameters are given later.

We emphasize that the whole algorithm is computationally very efficient because the complexity of one iteration is linear in the number of particles N.

Now, the algorithm for spot cross-volatility estimation (SCVE) developed in Section 6.7 is stated.

Algorithm: On-line Spot Cross-Volatility Estimation (SCVE)

Initialization:

- Set the initial covariance estimate $\hat{\Sigma}_{t_2} = \hat{\Sigma}^{\mathrm{pf}}_{t_2}$, the number of particles N, the constant K, and the step sizes λ_{v,s_1,s_2} for time-varying covariance estimation or γ for time-constant covariance estimation.
- For $i = 1, \dots, N$ and $s = 1, \dots, S$: Generate sample $x^i_{t_1^{(s)},s}$ such that $\exp[x^i_{t_1^{(s)},s}]$ is uniformly distributed on $A_{t_1^{(s)}}$.

On-line covariance estimation: (for $v = 2, \dots, T$)

- Determine the securities $s_1, \dots, s_W$ that were traded at time t_v.
- For $w = 1, \dots, W$: (the following three steps are conditional on $\{x^i_{t_v,s_1}, \dots, x^i_{t_v,s_{w-1}}\}_{i=1}^N$)
  - Determine $l_1, \dots, l_S$ by applying the rules 1 through 3 described in Section 6.7.2 and compute the optimal proposal (6.58), where Σ_{t_v} is replaced by $\hat{\Sigma}^{\mathrm{pf}}_{t_v}$.
  - For $i = 1, \dots, N$: Generate sample $x^i_{t_v,s_w}$ from the optimal proposal.
  - For $i = 1, \dots, N$: Compute the importance weight $\tilde{\omega}^i_{t_v}$ according to (6.59), where Σ_{t_v} is again replaced by $\hat{\Sigma}^{\mathrm{pf}}_{t_v}$.
  - For $i = 1, \dots, N$: Normalize the importance weight $\omega^i_{t_v} = \tilde{\omega}^i_{t_v} / \sum_{j=1}^N \tilde{\omega}^j_{t_v}$.
- Compute the update matrices according to (6.64). Obtain the covariance estimate $\hat{\Sigma}^{\mathrm{pf}}_{t_{v+1}}$ for the next iteration according to (6.63).
- If $\hat{\Sigma}^{\mathrm{pf}}_{t_{v+1}}$ is not positive definite, then iteratively increase the diagonal entries until positive definiteness is achieved.
- If the effective sample size $\mathrm{ESS}(\{\omega^i_{t_v}\}_{i=1}^N) < 0.2N$, then resample the particles using, for instance, the residual resampling scheme (Douc, Cappé, and Moulines 2005).

As mentioned in Section 6.7.3, the covariance estimate $\hat{\Sigma}_{t_v}$ can often not be computed at time t_v. The estimate $\hat{\Sigma}^{\mathrm{pf}}_{t_v}$ used in the particle filter is the most recent available estimate which is, at best, $\hat{\Sigma}_{t_{v-1}}$ (compare Section 6.7.3). Note that for the multivariate case prediction estimates (such as $\tilde{\Sigma}^{*}_{t_j|t_{j-1}}$) are not discussed in this work. However, they could be constructed similarly to the univariate case.

We mention that at a time t_v it may be required to update the covariance estimate more than one time. This is the case because a transaction can have a positive overlapping time with multiple transactions of another security.

The complexity of one iteration of the SCVE algorithm is $\mathcal{O}(NK + K^3)$ and the storage requirement is $\mathcal{O}(NK)$. The NK term follows from the computation of $\overline{\mu}$ which is required for every particle. The K^3 term is a result of the matrix inversion required for the computation of $\overline{\Sigma}$ (see Proposition 6.3). Therefore, the SCVE algorithm is computationally efficient as long as K is small (see below).

Parameter Initialization

$\hat{\Sigma}^{\mathrm{pf}}_{t_2}$ Our experience from many data sets is that both algorithms stabilize quickly provided that reasonable starting values are used – e.g. $\hat{\Sigma}_{t_2} = \hat{\Sigma}^{(1/2)}_{t_2} = \hat{\Sigma}^{\mathrm{pf}}_{t_2} = \hat{\Sigma}$ with $\hat{\Sigma}$ from prior knowledge or with $\hat{\Sigma}$ being a rough initial estimate. In the SVE algorithm one will use $v_{1,3} = \frac{\lambda_3}{2-\lambda_3}$, $v_{2,3} = \frac{\lambda_3}{4-\lambda_3}$ and $v_{3,3} = \frac{\lambda_3}{3-\lambda_3}$ with (say) $\lambda_3 = 1/500$ (these are the fix points of the recursions (6.38) through (6.40)). More sophisticated starting values are obtained as follows (only for the univariate case): One uses our procedure over the first 500 transactions in reversed time order leading to values $\hat{\Sigma}^{\mathrm{rev}}_{t_1}$, $\hat{\Sigma}^{\mathrm{rev}(1/2)}_{t_1}$, $\bar{1}^{\mathrm{rev}}$, $\bar{1}^{\mathrm{rev}(1/2)}$ and starts the algorithm then with the following values obtained by extrapolation:

$$\bar{2} := -\bar{1}^{\mathrm{rev}} + 3\,; \qquad \bar{2}^{(1/2)} := -\bar{1}^{\mathrm{rev}(1/2)} + 3;$$

$$\hat{\Sigma}_{t_2} := \Big(1 + \frac{2 \times \bar{1}^{\mathrm{rev}} - 2}{\bar{1}^{\mathrm{rev}(1/2)} - \bar{1}^{\mathrm{rev}}}\Big)\, \hat{\Sigma}^{\mathrm{rev}}_{t_1} - \frac{2 \times \bar{1}^{\mathrm{rev}} - 2}{\bar{1}^{\mathrm{rev}(1/2)} - \bar{1}^{\mathrm{rev}}}\, \hat{\Sigma}^{\mathrm{rev}(1/2)}_{t_1};$$

$$\hat{\Sigma}^{(1/2)}_{t_2} := \Big(1 + \frac{\bar{1}^{\mathrm{rev}} + \bar{1}^{\mathrm{rev}(1/2)} - 2}{\bar{1}^{\mathrm{rev}(1/2)} - \bar{1}^{\mathrm{rev}}}\Big)\, \hat{\Sigma}^{\mathrm{rev}}_{t_1} - \frac{\bar{1}^{\mathrm{rev}} + \bar{1}^{\mathrm{rev}(1/2)} - 2}{\bar{1}^{\mathrm{rev}(1/2)} - \bar{1}^{\mathrm{rev}}}\, \hat{\Sigma}^{\mathrm{rev}(1/2)}_{t_1};$$

$v_{1,2} = \frac{\lambda_1^{\mathrm{rev}}}{2-\lambda_1^{\mathrm{rev}}}$; $v_{2,2} = \frac{\lambda_1^{\mathrm{rev}}}{4-\lambda_1^{\mathrm{rev}}}$; $v_{3,2} = \frac{\lambda_1^{\mathrm{rev}}}{3-\lambda_1^{\mathrm{rev}}}$. We then obtain e.g. for the bias-corrected estimator $\big($where $\kappa_{2|2} = \frac{2-\bar{2}}{\bar{2}-\bar{2}^{(1/2)}} = \frac{\bar{1}^{\mathrm{rev}}-1}{\bar{1}^{\mathrm{rev}(1/2)}-\bar{1}^{\mathrm{rev}}} = \kappa^{\mathrm{rev}}_{1|1}\big)$ after some calculations (see Appendix A.11)

$$\tilde{\Sigma}_{t_2|t_2} = \big(1 + \kappa_{2|2}\big)\, \hat{\Sigma}_{t_2} - \kappa_{2|2}\, \hat{\Sigma}^{(1/2)}_{t_2} = \ldots = \tag{6.65}$$

$$= \Big(1 + \frac{\bar{1}^{\mathrm{rev}} - 1}{\bar{1}^{\mathrm{rev}(1/2)} - \bar{1}^{\mathrm{rev}}}\Big)\, \hat{\Sigma}^{\mathrm{rev}}_{t_1} - \frac{\bar{1}^{\mathrm{rev}} - 1}{\bar{1}^{\mathrm{rev}(1/2)} - \bar{1}^{\mathrm{rev}}}\, \hat{\Sigma}^{\mathrm{rev}(1/2)}_{t_1} = \tilde{\Sigma}^{\mathrm{rev}}_{t_1|t_1}$$

(note that because of $\mathbf{X}_{t_1} - \mathbf{X}_{t_2} = -\mathbf{Z}_{t_2}$ with $\mathbf{Z}_{t_2} \sim \mathcal{N}(\mathbf{0}, \Sigma_{t_2})$ we have $\Sigma^{\mathrm{rev}}_{t_1} = \Sigma_{t_2}$). The particle filter could be started with $\hat{\Sigma}^{\mathrm{pf}}_{t_2} := \tilde{\Sigma}_{t_2|t_2}$. λ_3 can then be calculated from the above formulas. The reversed method works nicely as can be seen from Figure 6.9 below. In order to exclude the effect of starting values we have used in the simulations (except from Figure 6.7) the true matrix Σ_{t_2} as the starting value (i.e. $\hat{\Sigma}^{\mathrm{pf}}_{t_2} = \hat{\Sigma}_{t_2} = \hat{\Sigma}^{(1/2)}_{t_2} = \Sigma_{t_2}$).

N As a result of the efficiency of our particle filters the number of particles N is not a critical quantity. Typically, a few hundred (say 500) particles suffice to achieve a reasonable precision in the case of a small number of securities considered (see Figure 6.7). However, it may be necessary to increase the number of particles significantly, if the number of securities considered is much larger than three.

K In the multivariate case K defined in Section 6.7.2 needs to be chosen. We find empirically that (at least for our data) K as small as 20 suffices for two securities. Of course, K should be increased when more than two securities are considered or when the trading frequencies of the securities differ much. As discussed above the complexity of the SCVE algorithm heavily depends on K. Hence, it should be chosen as small as possible. In practice, a reasonable value can be obtained by running our algorithm with different values for K and comparing the results.

λ, α, β In the time-varying covariance case the step size parameter λ or α and β need to be specified. In the univariate case we apply the adaptive step size selection (with α and β) as described in Section 6.5.1. In the multivariate case this method cannot be directly applied and, therefore, we propose to use a constant step size λ which can be optimized with respect to a criterion related to (6.28). We think that in the multivariate case the optimal step size depends much on the application. For instance, if one is interested in the cross-volatility (correlation) of two securities one can optimize λ for the estimation of a particular covariance component (analogous to (6.28)). We mention that it is also possible to consider individual step sizes for the different components of the covariance matrix, that is $\lambda_{v,s_1,s_2} = \lambda_{s_1,s_2} = \lambda_{s_2,s_1}$. However, the (optimal) step size selection in the multivariate case remains an open question which should be tackled in future research.

γ In the time-constant covariance case the step size only depends on γ. In the simulation study we obtain that $\gamma \approx 0.8$ is a good choice in practice (see Figure 6.7). However, a more rigorous choice based on theoretical results is desirable.

6.10 Simulations and Applications

6.10.1 Results for Simulated Data

Estimation of time-constant spot volatility

We first consider the estimation of time-constant spot volatility. An efficient log-price process is simulated from t_1 to t_{5000} with squared volatility equal to $\Sigma_t = 0.00005^2$. The initial efficient price $\exp[X_{t_1}]$ is sampled from a uniform distribution on $[50-0.005, 50+0.005)$. The transaction prices are obtained by rounding the efficient prices to the nearest cent which constitutes a special case of our market microstructure noise model. Our algorithm for time-constant spot volatility estimation (6.20) is applied with different numbers of particles N and different values of γ. The

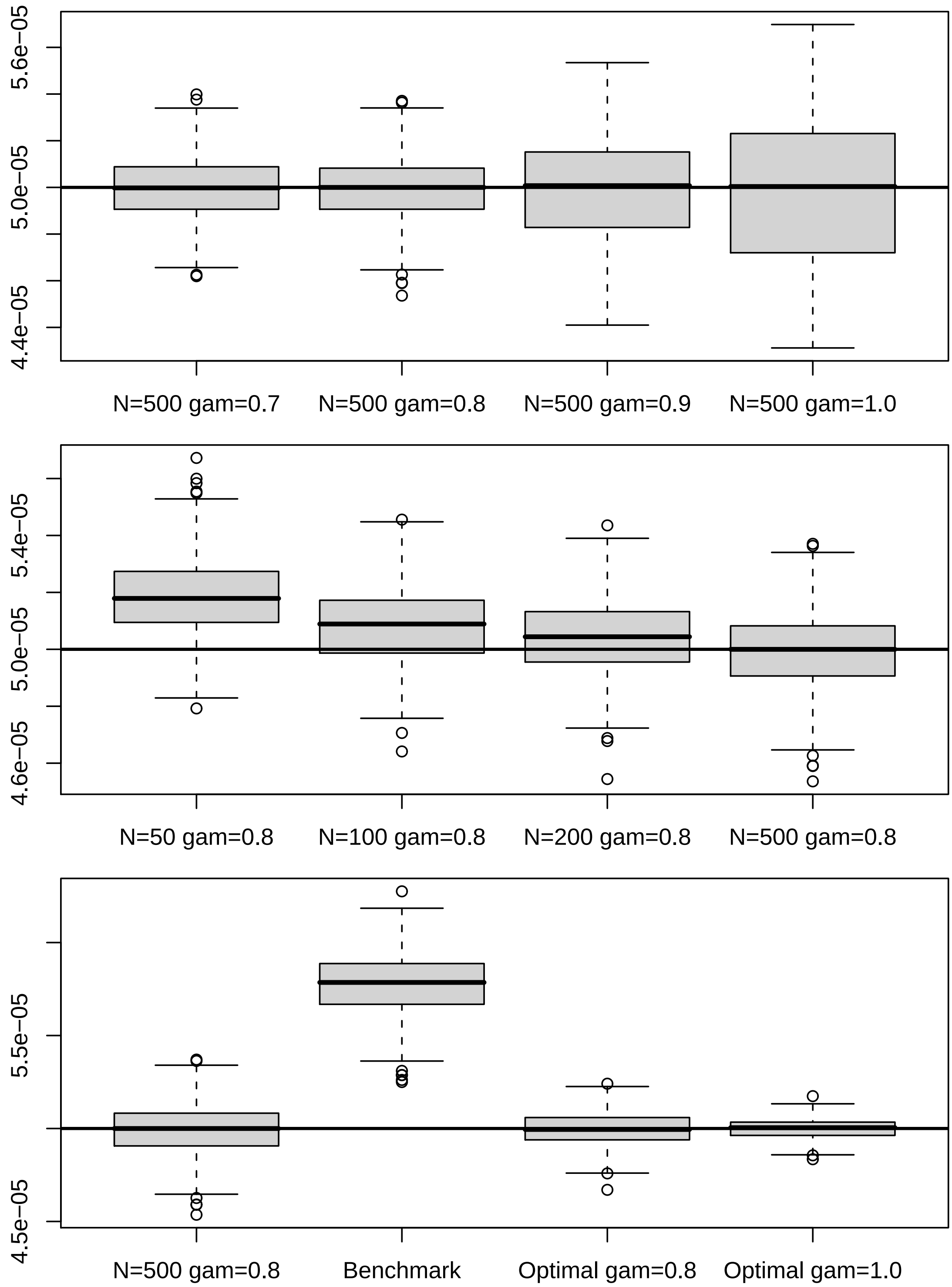

Figure 6.7: Box plots for the results of the estimation of a time-constant volatility of 0.00005 based on simulated data (5,000 transactions). Our estimator (6.20) is applied with different numbers of particles N and different values for γ. The initial volatility is drawn from a uniform distribution on $(0.00004, 0.00006)$. For comparison the results of the benchmark estimator ("Benchmark") and the optimal estimator ("Optimal") are also reported. Note that the optimal estimator is not available in practice. The box plots are computed based on 500 independent runs.

starting value $\hat{\Sigma}_{t_2}^{\mathrm{pf}} = \hat{\Sigma}_{t_2}$ is drawn from a uniform distribution on $(0.00004^2, 0.00006^2)$. For comparison the results of two benchmark algorithms are also reported. The first benchmark method ("Benchmark" in Figure 6.7) is a recursive estimator with a simpler microstructure noise correction. It is related to the method in Zumbach, Corsi, and Trapletti (2002) and it is based on the market microstructure model $\log Y_{t_j} = X_{t_j} + U_{t_j}$, where the noise variables U_{t_j} are i.i.d. with Var $U_{t_j} = \eta^2$. The recursive estimator is given by

$$\hat{\Sigma}_{t_j}^{\mathrm{B}} := \big\{1 - \frac{1}{j-1}\big\}\big(\hat{\Sigma}_{t_{j-1}}^{\mathrm{B}} + \max\{0, 2\hat{\eta}_{t_{j-1}}^2\}\big) + \frac{1}{j-1}\,(\log y_{t_j} - \log y_{t_{j-1}})^2 - \max\{0, 2\hat{\eta}_{t_j}^2\} \quad (6.66)$$

where $\hat{\eta}_{t_j}^2 := \{1 - \frac{1}{j-2}\}\hat{\eta}_{t_{j-1}}^2 - \frac{1}{j-2}\,\big(\log y_{t_j} - \log y_{t_{j-1}}\big)\big(\log y_{t_{j-1}} - \log y_{t_{j-2}}\big)$ (here $\frac{1}{j-2}$ is used instead of $\frac{1}{j-1}$ because the algorithm starts one time point later). The term $\max\{0, 2\hat{\eta}_{t_j}^2\}$ corrects for the market microstructure noise. This follows from the fact that

$$\mathrm{Cov}\big(\log Y_{t_j} - \log Y_{t_{j-1}}, \log Y_{t_{j-1}} - \log Y_{t_{j-2}}\big) = -\eta^2.$$

The second benchmark method is, in some sense, the optimal estimator ("Optimal" in Figure 6.7). It is unavailable in practice because it uses the latent efficient log-prices. It is computed analogous to (6.20) but instead of the particles it employs the efficient log-prices leading to

$$\hat{\Sigma}_{t_j}^{\mathrm{Opt}} = \{1 - (j-1)^{-\gamma}\}\hat{\Sigma}_{t_{j-1}}^{\mathrm{Opt}} + (j-1)^{-\gamma}(x_{t_j} - x_{t_{j-1}})^2. \quad (6.67)$$

The simulation results are given in terms of box plots which are obtained by 500 independent runs (Figure 6.7). The box plots suggest that our volatility estimator is asymptotically unbiased and that $\gamma = 0.8$ is a reasonable value. We can also conclude that about 500 particles are sufficient which makes our algorithm computationally efficient and suitable for real-time applications. In addition, it can be observed that the benchmark estimator is biased.

Estimation of time-varying spot volatility

We now compare our algorithms (6.21) and (6.43) for the time-varying spot volatility estimators $\hat{\Sigma}_{t_j}$ and $\tilde{\Sigma}_{t_j|t_j}^*$, respectively, with a benchmark estimator. The efficient log-prices are generated with respect to the time-varying volatility given by the black lines in Figure 6.4 (the first case (upper plot) is more challenging while the second case (lower plot) is more realistic for a volatility curve in transaction time - see the real data example in Figure 6.9). The volatility curves used are given by the square roots of the variance curves

$$\Sigma(t) = 0.000105^2 \times \begin{cases} 1 + 0.45\cos(3\pi t/7500) & \text{for} \quad 0 < t \le 7500, \\ 0.55 & \text{for} \quad 7500 < t \le 11500, \\ 0.82 + 0.27\cos(\pi + 2\pi t/3500) & \text{for} \quad 11500 < t \le 15000, \end{cases}$$

and

$$\Sigma(t) = 0.000105^2 \times \begin{cases} 1 + 0.45\cos(\pi t/2500) & \text{for} \quad 0 < t \le 2500, \\ 1 + 0.45\cos(3\pi) & \text{for} \quad 2500 < t \le 15000. \end{cases}$$

		Figures 6.4 and 6.5			Figure 6.4			Figure 6.9	
		(upper plot)			(lower plot)				
Estimator	line color	α	β	MSE	α	β	MSE	α	β
$\hat{\Sigma}_{t_j}$	turquoise	-4.46	150,000	1.21 $\times 10^{-18}$	-4.95	289,000	3.20 $\times 10^{-19}$	-5.23	27,121
$\tilde{\Sigma}^*_{t_j\|t_j}$	red	-5.25	261,000	1.14 $\times 10^{-18}$	-5.36	431,000	1.77 $\times 10^{-19}$	-5.08	27,700
$\hat{\Sigma}^{\mathrm{B}}_{t_j}$	gray	-5.42	9,100	1.34 $\times 10^{-18}$	-6.35	13,900	6.76 $\times 10^{-19}$	-5.86	50,900

Table 6.1: Parameters α and β optimized with respect to (6.28). The mean squared errors (MSE) are computed as described in Section 6.10.1.

In both cases $\exp[X_{t_1}] \sim \mathcal{U}[50-0.005, 50+0.005)$. Again transaction prices (observations) are obtained by rounding the efficient prices to the nearest cent. 15,000 transactions are generated which is typical for one trading day of a liquid stock. The particle filter is applied with $N = 500$ particles. The estimator $\tilde{\Sigma}^*_{t_j|t_j}$ is calculated as described in sections 6.5 and 6.9. $\hat{\Sigma}_{t_j}$ also uses the time-varying step sizes (6.26) where α and β are obtained by minimizing the criterion (6.28) as for $\tilde{\Sigma}^*_{t_j|t_j}$. (A simpler strategy avoiding the calculation of $\hat{\Sigma}^{(1/2)}_{t_j}$ is to use a constant step size λ obtained by minimizing (6.28).) Analogous to (6.66) we consider the benchmark estimator given by

$$\hat{\Sigma}^{\mathrm{B}}_{t_j} := \{1-\lambda_j\}\big(\hat{\Sigma}^{\mathrm{B}}_{t_{j-1}} + \max\{0, 2\hat{\eta}^2_{t_{j-1}}\}\big) + \lambda_j\big(\log y_{t_j} - \log y_{t_{j-1}}\big)^2 - \max\{0, 2\hat{\eta}^2_{t_j}\} \tag{6.68}$$

with $\hat{\eta}^2_{t_j} := \{1-\frac{1}{j-2}\}\hat{\eta}^2_{t_{j-1}} - \frac{1}{j-2}\,\big(\log y_{t_j} - \log y_{t_{j-1}}\big)\big(\log y_{t_{j-1}} - \log y_{t_{j-2}}\big)$. For a fair comparison we also use the time-varying step sizes (6.26) where α and β are obtained by minimizing the criterion

$$\sum_{j=2}^{T-1} \big(\hat{\Sigma}^{\mathrm{B}}_{t_j} + \max\{0, 2\hat{\eta}^2_{t_j}\} - (\log y_{t_{j+2}} - \log y_{t_{j+1}})^2\big)^2 \tag{6.69}$$

(the terms $\hat{\Sigma}^{\mathrm{B}}_{t_j} + \max\{0, 2\hat{\eta}^2_{t_j}\}$ and $(\log y_{t_{j+2}} - \log y_{t_{j+1}})^2$ are independent in the additive microstructure noise model $\log Y_{t_j} = X_{t_j} + U_{t_j}$ with U_{t_j} i.i.d. - thus by using $(\log y_{t_{j+2}} - \log y_{t_{j+1}})^2$ (6.69) becomes a decent estimate of the mean squared error (plus a term constant in α and β)). For $\hat{\eta}^2_{t_j}$ we use the step sizes $\frac{1}{j-2}$ because η^2_t should be close to a constant function.

All estimators use the true volatility as starting value. Typical outcomes of the estimators are given in Figure 6.4. Note that the volatility is plotted (instead of the squared volatility). Because the true $\Sigma(t_j)$ is known we can compute the mean squared error $\Sigma_{j=2}^{T-1}\big(\hat{\Sigma}(t_j) - \Sigma(t_j)\big)^2$ for all estimators. The obtained mean squared errors and the optimized parameters α and β can be found in Table 6.1. In both plots, $\tilde{\Sigma}^*_{t_j|t_j}$ significantly outperforms the other estimators.

We have tried to improve the benchmark estimator by a bias correction similar to Section 6.5. Surprisingly, this has lead only to minor improvements. (We have refrained from plotting this estimator.) The reason for this is not clear: We think that the rounding in the values y_{t_j} is responsible for the bad quality in that it leads to a (local) bias and higher fluctuations. Perhaps the estimator may be improved a bit by modifying (6.27).

To further investigate the estimator $\tilde{\Sigma}^*_{t_j|t_j}$ we have also plotted in Figure 6.5 $\hat{\Sigma}_{t_j}$ and $\hat{\Sigma}^{(1/2)}_{t_j}$ from (6.43) (i.e. with the α and β used to optimize $\tilde{\Sigma}^*_{t_j|t_j}$) as well as the sequences λ_j and $\kappa^*_{j|j}$. The bias of $\hat{\Sigma}_{t_j}$ and $\hat{\Sigma}^{(1/2)}_{t_j}$ is clearly visible. Furthermore, it can be seen how the estimator

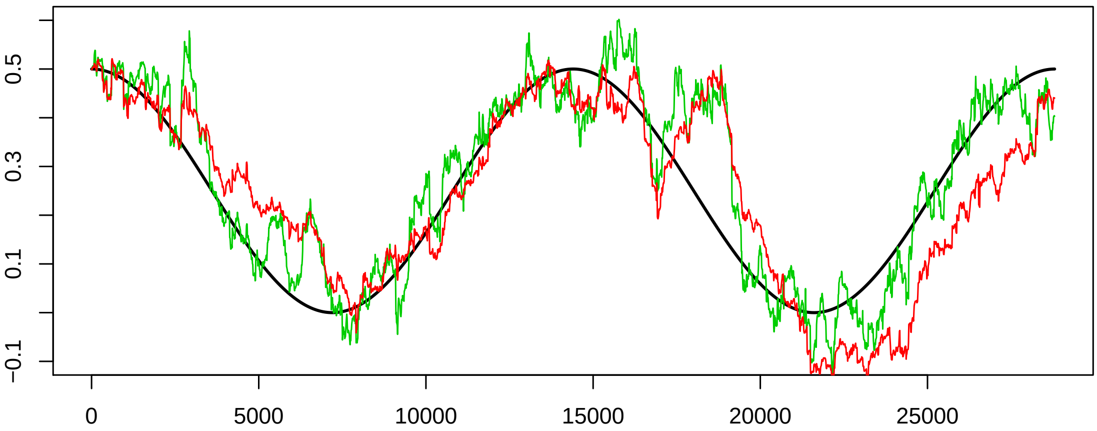

Figure 6.8: Estimation of the time-varying cross-volatility based on two simulated price processes with real non-synchronous trading times. The transaction prices are generated with respect to the volatility curve and the correlation curve given by the black lines in Figure 6.8 (lower plot) and in this figure, respectively. Estimators: $\hat{\Sigma}_{t_v}$ (red line), optimal estimator $\hat{\Sigma}^{\text{opt}}_{t_v}$ (green line) which is not available in practice. Note that only the correlation estimates are plotted. The x-axis shows the joint transaction time.

$\tilde{\Sigma}^{*}_{t_j|t_j}$ extrapolates these raw estimates to improve on the bias. During the period of constant volatility the step size λ_j gets low because (6.27) is close to zero. Furthermore, $\kappa^{*}_{j|j}$ gets close to -1 which implies that $\tilde{\Sigma}^{*}_{t_j|t_j} \approx \hat{\Sigma}^{(1/2)}_{t_j}$ (which is the smoother estimate). During periods of volatility changes the step size λ_j gets large and $\tilde{\Sigma}^{*}_{t_j|t_j}$ adapts more quickly to Σ_{t_j}.

Estimation of time-varying spot cross-volatility

Finally, a time-varying covariance matrix is estimated. Two efficient log-price processes are generated with respect to the time-varying volatility and the time-varying correlation given by the black lines in Figure 6.4 (lower plot) and Figure 6.8, respectively. To make the simulation setting realistic real non-synchronous trading times are used. We take the time stamps of BAC and C for the 4th September 2007 which gives 16,444 and 13,323 transactions for the two simulated processes, respectively (for details on the data see below). The initial prices $X_{t_1^{(s)}}$, $s = 1, 2$, are again sampled from a uniform distribution on $[50 - 0.005, 50 + 0.005)$ and the transaction prices are obtained by rounding to the nearest cent.

The aim is to compare our cross-volatility estimator $\hat{\Sigma}_{t_v}$ with the optimal estimator $\hat{\Sigma}^{\text{opt}}_{t_v}$ which, analogous to (6.67), uses the latent efficient prices instead of the particles. For the optimal estimator the update (6.64) is defined by

$$(\breve{\Sigma}^{\text{opt}}_{t_v})_{s_1 s_2} = \begin{cases} \dfrac{z_{t^{(s_1)}_{h^v_1}} z_{t^{(s_2)}_{h^v_2}}}{f(t^{(s_1)}_{h^v_1}, t^{(s_2)}_{h^v_2})} & \text{if} \quad t^{(s_1)}_{h^v_1} = t_v \ \text{ or } \ t^{(s_2)}_{h^v_2} = t_v \\ (\hat{\Sigma}^{\text{opt}}_{t_{v-1}})_{s_1 s_2} & \text{else} \end{cases}$$

with $z_{t^{(s)}_{h_s}} = x_{t^{(s)}_{h_s}} - x_{t^{(s)}_{h_s - 1}}$. As a result of the non-synchronous trading there is no simple benchmark estimator (such as (6.66)) available in the multivariate case. Our algorithm is applied with the setting $N = 500$ and $K = 20$. Both algorithms use the true covariance matrix as starting value. We use constant step sizes which are chosen to minimize the criterion (6.28) with respect to the cross-volatility component of the covariance matrix as suggested in Section 6.9. This gives $\lambda = 0.00325$ and $\lambda = 0.0031$ for $\hat{\Sigma}_{t_v}$ and $\hat{\Sigma}_{t_v}^{\text{opt}}$, respectively. The estimation results are shown in Figure 6.8 in terms of the correlation estimates. Note that even the optimal estimator is quite unstable suggesting that on-line estimation of time-varying correlation is a very hard problem. The volatility estimates of $\hat{\Sigma}_{t_v}$ (not reported) are slightly worse compared to the univariate case (lower plot in Figure 6.8) because the step size is optimized for the cross-volatility component. In addition, neither time-varying step sizes nor bias correction are used.

6.10.2 Results for Real Data

The data

We use stock data from the TAQ data base. Transactions and market maker quotes of the symbols BAC (Bank of America Corporation), C (Citigroup), and JPM (JPMorgan Chase & Co) for the 3rd and 4th September 2007 were extracted from the TAQ data base. To improve the data quality we carried out the following data cleaning and transformation.

Cleaning A: Delete all transactions (quotes) with time stamps outside the main trading period (9:30 AM to 4 PM).

Cleaning B: Delete all transactions (quotes) that are not originating from the NYSE.

Cleaning C: Delete all transactions with abnormal sale condition or corrected prices (see the TAQ User's Guide for details).

Data transformation: If multiple transactions have the same time stamp (after the data cleaning) apply the following transformation. Assume $t_j = t_{j+1} = \ldots = t_{k-1} \neq t_k$. Replace t_l by $t'_l = t_j + (l-j)(t_k - t_j)/(k-j)$ for $l = j+1, \ldots, k-1$.

After the data cleaning the following numbers of transactions remained for the symbols BAC, C, and JPM, respectively: 16,219, 16,287, 13,400 for the 3rd September and 16,444, 13,323, 18,569 for the 4th September. The transformation replaces identical time stamps with time stamps that are equally spaced. This transformation is necessary because the time stamp precision of our data is limited to one second. See the remark below for an alternative approach.

Unfortunately, the quality of the TAQ data is to poor to match easily the transactions with the market maker quotes. Note that it is necessary for our method that the transaction and quote data are perfectly matched. Therefore, our simulations are mainly focused on transaction data.

Remark: In the literature, it was suggested to delete multiple transactions with the same time stamp and to use the median price (e.g. Barndorff-Nielsen et al. 2009). This is problematic for at least two reasons. First, one loses much of the data. For instance, by removing the multiple

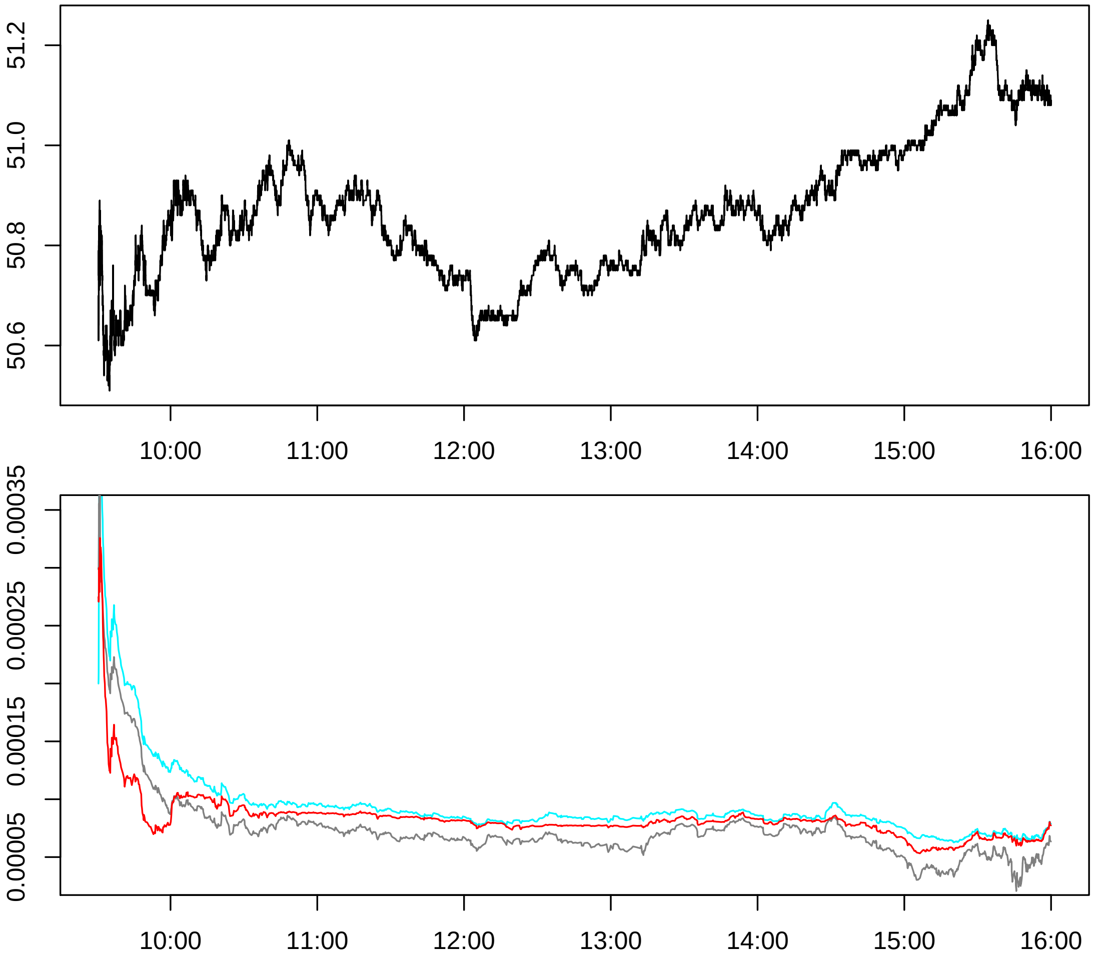

Figure 6.9: Real data example: Estimation of time-varying spot volatility in transaction time. The upper plot shows the transaction data of the symbol BAC for the 3rd September 2007. The lower plot gives our volatility estimators $\hat{\Sigma}_{t_j}$ (turquoise line) and $\tilde{\Sigma}^*_{t_j|t_j}$ (red line) and the benchmark estimator $\hat{\Sigma}^{\mathrm{B}}_{t_j}$ (gray line).

transactions with the same time stamp for BAC (3rd September) the number of transactions will reduce to 8,696. That is, the number will almost be cut by half. Second, by using the median price one will introduce spurious positive first order autocorrelations between consecutive returns. In addition, the definition of the transaction time will not make sense any more.

Estimation results for real market maker quotes

In order to show how our method works in the case when market maker quotes are available (case 2 in Section 6.2) we matched by hand (through an adjustment of the time stamps) the quotes and transactions of symbol C for a fraction of the trading day. As mentioned earlier, the quality of our data is to poor to do this automatically. Our particle filter is used with $N = 5,000$ particles to estimate the filtering distributions of the unknown efficient (log-)prices. Figure 6.1 gives kernel density estimates of filtering distributions of some efficient prices which are computed based on the particle approximations. The market maker quotes, the transaction prices, and supports of the filtering distributions are also shown. From the figure it can be seen that some filtering distributions are highly skewed. In addition, consecutive zero returns lead to very uninformative filtering distributions (see transactions 2,300 through 2,309).

Transaction time spot volatility estimation

We apply our estimators $\hat{\Sigma}_{t_j}$ and $\tilde{\Sigma}^*_{t_j|t_j}$ with $N = 500$ particles and the benchmark method $\hat{\Sigma}^{\mathrm{B}}_{t_j}$ (6.68) to estimate the spot volatility for BAC. To obtain a good initialization for the estimator $\tilde{\Sigma}^*_{t_j|t_j}$ the initialization algorithm which proceeds in reversed time order is applied to the first 500 transactions (see Section 6.9). For $\hat{\Sigma}_{t_j}$ and $\hat{\Sigma}^{\mathrm{B}}_{t_j}$ an initial volatility of 0.0002 is used. The optimized values for α and β are reported in Table 6.1.

The transaction data of BAC and the volatility estimators are shown in Figure 6.9. At the beginning of the trading day the volatility is large and highly varying. Later, the volatility settles down and seems to be almost constant. Therefore, the localized step size selection from (6.26) is clearly advantageous compared to fixed step sizes. Again the benchmark estimator is rougher than our estimators. Practically, the volatility in transaction time is almost constant after 10:00.

Transaction time spot cross-volatility estimation

Time-varying estimates for the BAC/C/JPM data sets of the 4th September 2007 are plotted in Figure 6.10. Our algorithm $\hat{\Sigma}_{t_v}$ is applied with the following setup: $N = 500$, $K = 30$, and starting value $\hat{\Sigma}^{\mathrm{pf}}_{t_2} = \mathrm{diag}(0.0003^2, 0.0003^2, 0.0003^2)$. The initial covariance matrix is far-off the true one. The step size is obtained by minimizing the sum of the squared prediction errors of the cross-volatilities, which are computed analogous to (6.28), yielding $\lambda = 0.0065$. We observe that the first 30 minutes of the trading day are characterized by high volatilities and low correlations. We emphasize that this is not an effect of the choice of the initial covariance matrix because our algorithm can adapt very quickly. For the rest of the trading day the volatilities are roughly constant. (Note that we use a constant step size $\lambda_{v,s_1,s_2} = \lambda$ which is optimized for the estimation of the cross-volatilities (correlations). Therefore, this step size may not be optimal for the plotted volatility estimates.) In contrast, the correlations show a high variability during the whole trading day. For instance, the correlation of BAC/JPM (green line in Figure 6.10) fluctuates between approximately 0.05 and 0.7 within less than two hours. This is an important result for risk management and high-frequency trading.

Clock time spot volatility estimation

We now compare our two approaches for the estimation of spot volatility in clock time for symbol BAC. The first estimator $\tilde{\Sigma}^{*c}_{t_j|t_j}$ is applied as described in sections 6.4.1 and 6.5.2. α and β are optimized with respect to (6.28) where $\breve{\Sigma}_{t_{j+1}}(\omega_{t_{j+1}})$ is replaced with $\breve{\Sigma}^c_{t_{j+1}}(\omega^c_{t_{j+1}})$ (the results are reported Table 6.2). A plot of this estimator is quite poor – apart from some strong spikes caused by very small values of $t_j - t_{j-1}$ and therefore very large values of $\breve{\Sigma}^c_{t_j}(\omega^c_{t_j})$ in (6.24), the

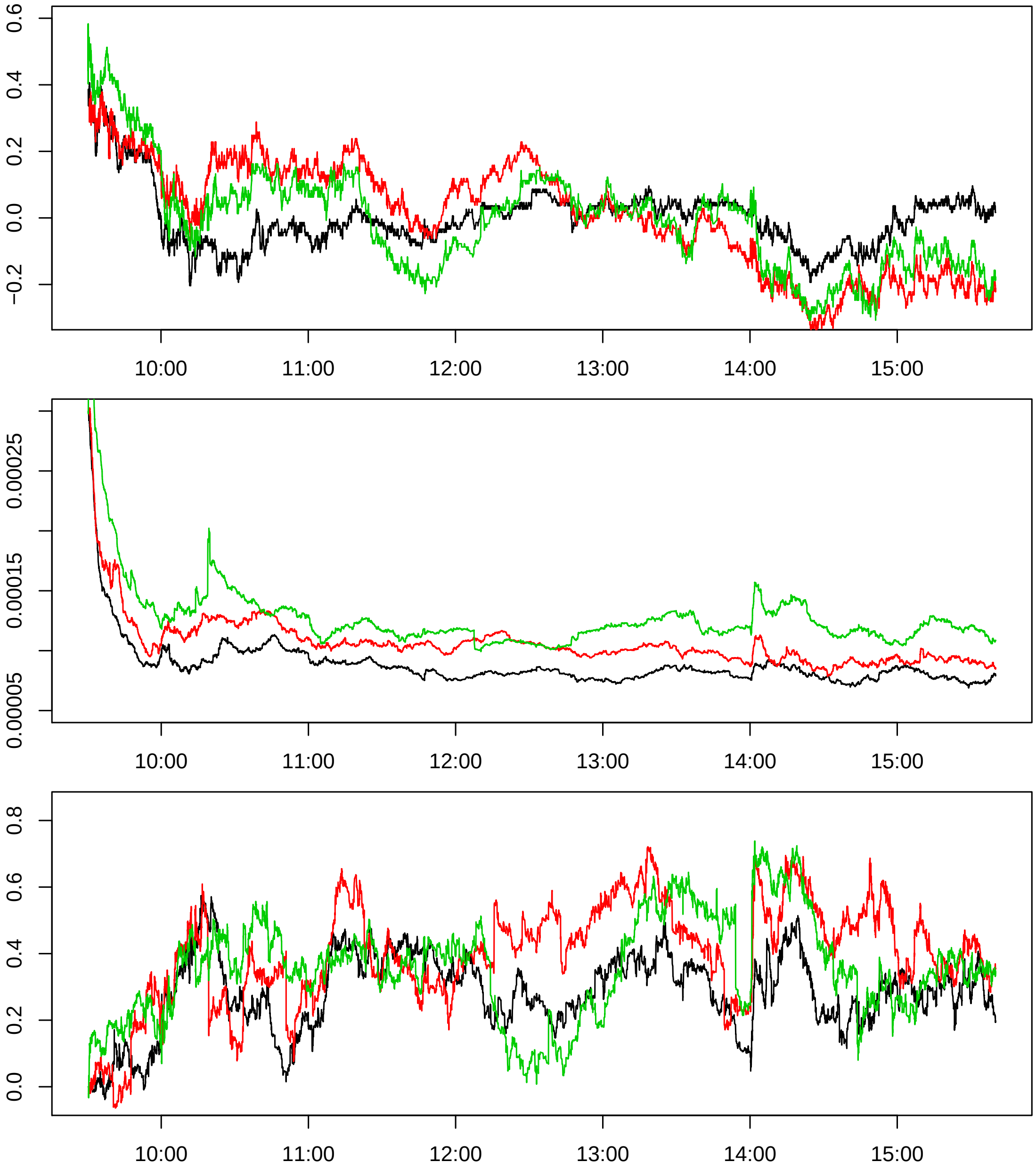

Figure 6.10: Real data example: Estimation of time-varying spot cross-volatilities. Upper plot: Transaction data of the 4th September 2007 for the symbols BAC (black line), C (red line), and JPM (green line) plotted with offsets. Middle plot: Volatility estimates in transaction time (colors of upper plot apply). Lower plot: Correlation estimates for BAC/C (black line), BAC/JPM (green line), and C/JPM (red line).

volatility seems to be strongly oversmoothed. This is caused by the MSE-type criterion in (6.28) in combination with the very large values of $\check{\Sigma}^c_{t_j}(\omega^c_{t_j})$ acting like outliers and leading to small λ_j. We therefore intuitively took $2\lambda_j$ leading to the estimate which is plotted in Figure 6.11. The second estimator is the alternative estimator $\hat{\Sigma}^c_{\text{alt}}(t_j) = \tilde{\Sigma}^*_{t_j|t_j}/\bar{\delta}_j$ proposed in Section 6.4.2 with the transaction time estimator $\tilde{\Sigma}^*_{t_j|t_j}$ from Figure 6.9 (red line). For the duration estimator

		Figures 6.11 and 6.12	
Estimator	line color	α	β
$\tilde{\Sigma}^{*c}_{t_j\|t_j}$	green	-2.97	20,100
$\tilde{\Sigma}^{*}_{t_j\|t_j}$ $(\hat{\Sigma}^c_{\text{alt}}(t_j))$	red	-5.08	27,700

Table 6.2: Parameters α and β optimized with respect to (6.28).

$\bar{\delta}_j$ we found empirically that a constant step size suffices (because the duration curve roughly has constant smoothness over the trading day). The used step size for $\bar{\delta}_j$ is determined by minimizing the prediction error $\Sigma_{j=2}^{T-1}\{\bar{\delta}_j - (t_{j+1} - t_j)\}^2$ leading to $\lambda = 0.1025$. (We mention that because of the dependence of the durations $\bar{\delta}_j$ and $(t_{j+1} - t_j)$ usually are not independent and the minimization of the above criterion therefore is not approximately the same as the minimization of the mean squared error. Despite of this we think that the resulting λ is reasonable. However, this should be investigated further.)

The estimation results are provided in Figures 6.11 and 6.12. First we state that both estimators roughly coincide (which was not clear beforehand). From the upper plot of Figure 6.11 we observe that $\tilde{\Sigma}^{*c}_{t_j|t_j}$ (green line) produces some large spikes during the trading day (due to small values of $t_j - t_{j-1}$). The variability of $\hat{\Sigma}^c_{\text{alt}}(t_j) = \tilde{\Sigma}^{*}_{t_j|t_j}/\bar{\delta}_j$ is mainly a result of the variability of the duration estimator $\bar{\delta}_j$ (plotted in the lower plot) because the transaction time estimator $\tilde{\Sigma}^{*}_{t_j|t_j}$ is almost constant (apart from the beginning of the trading day - see Figure 6.9). The fluctuation of the duration estimator is very high during the whole day.

Figure 6.12 compares the transaction data and the volatility estimates for a small time period. The different behavior of the two estimators is apparent. We regard the strong spikes of $\tilde{\Sigma}^{*c}_{t_j|t_j}$ as artificial due to small values of $t_j - t_{j-1}$. Furthermore, the estimator needs about one minute to settle down again after the occurrence of a spike. On the other hand the small spikes of $\hat{\Sigma}^c_{\text{alt}}(t_j)$ are caused by small $\underline{\text{averaged}}$ durations. For this reason we have more confidence in the second estimator. In addition, it is theoretically more appealing (because the transaction time volatility is almost constant and the variability of the clock time volatility is mainly caused by the variability of the trading intensity).

The second estimator is also more stable for another reason: Because volatility in transaction time is less varying the particle filter in transaction time is more stable.

6.11 Discussion

The discussion below focuses on the univariate case because we provided a complete method for this case. The multivariate method presented can still be improved significantly (adaptive step size selection, bias correction, etc.). However, this will be even more challenging than is the univariate case because of the non-synchronicity. General conclusions are given at the end of the dissertation.

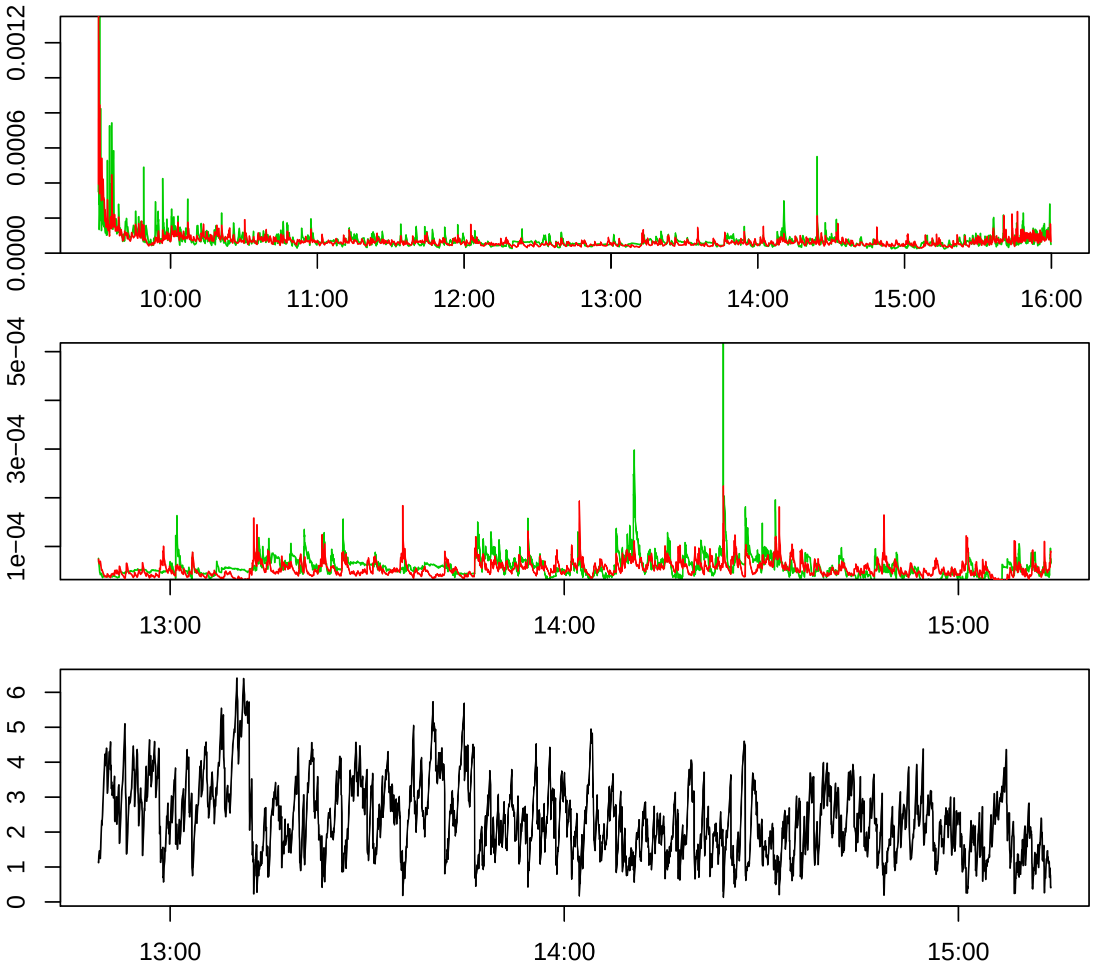

Figure 6.11: Real data example: Estimation of time-varying spot volatility in clock time based on the transactions of symbol BAC for the 3rd September 2007. The upper plot gives the volatility estimators $\tilde{\Sigma}^{*c}_{t_j|t_j}$ (green line) and $\hat{\Sigma}^{c}_{\text{alt}}(t_j)$ (red line). The middle plot shows the estimators for a fraction of the trading day. The averaged duration times $\bar{\delta}_j$ (for a fraction of the trading day) are given in the lower plot (the y-axis shows seconds).

Methodological Comments

We have presented a new technique for the on-line estimation of time-varying volatility based on noisy transaction data. Our algorithm is easy to implement and computationally efficient. It updates the volatility estimate immediately after the occurrence of a new transaction, and it therefore is as close to the market as possible. It also corrects for the bias which occurs as a result of the on-line estimation. It is straightforward to extend our method to more complicated price models (e.g. with a drift term) or other microstructure noise models.

Our work was guided by the goal to execute all calculations on-line in a high-frequency situation, and, at the same time, to base all methods on solid statistical principles. We feel that this goal has been achieved: Our algorithm is computationally efficient and it can be applied in real-time. On a recent personal computer an efficient implementation of our method requires

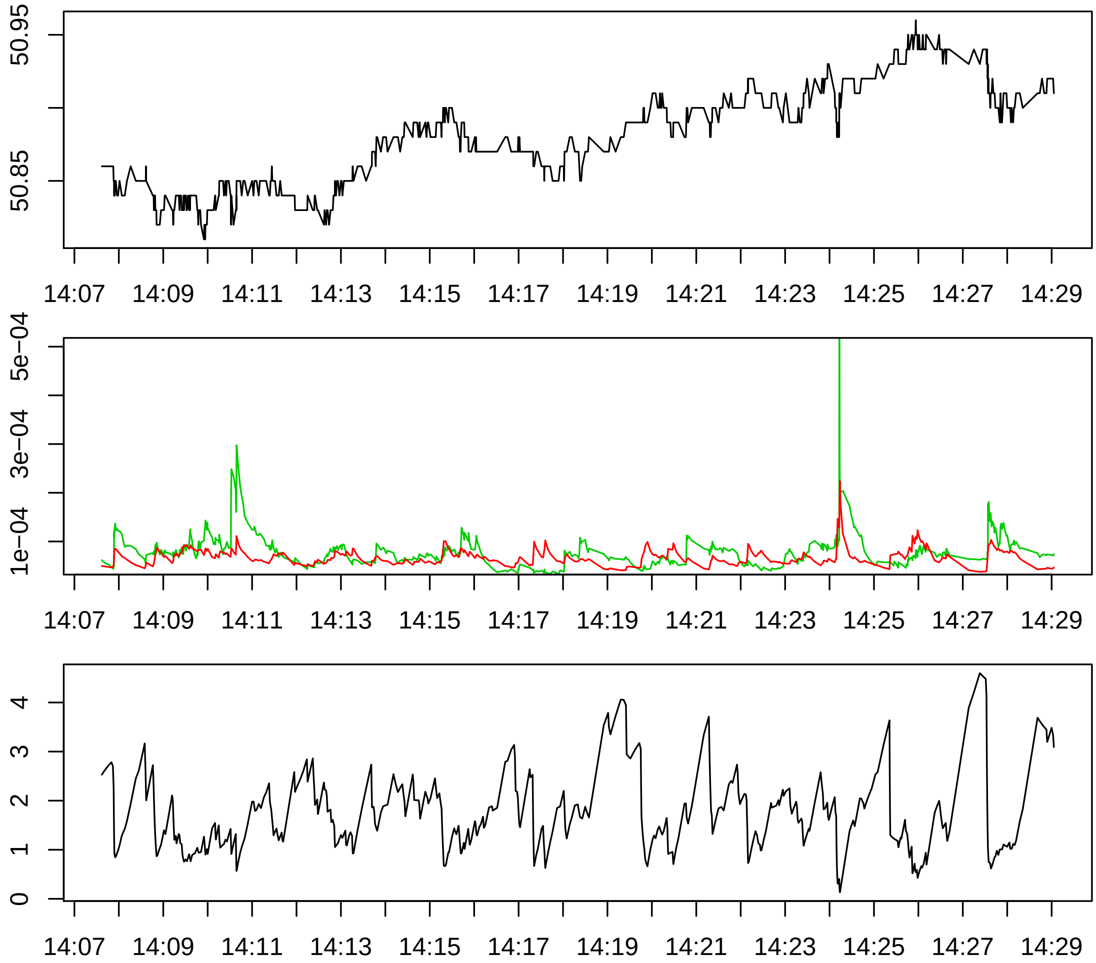

Figure 6.12: Real data example: Estimation of time-varying spot volatility in clock time based on the transactions of symbol BAC for the 3rd September 2007. The figure only gives the results for a small fraction of the trading day (compare Figure 6.11). The plots show (from top to bottom): transaction prices of BAC; our volatility estimators $\tilde{\Sigma}^{*c}_{t_j|t_j}$ (green line) and $\hat{\Sigma}^{c}_{\text{alt}}(t_j)$ (red line); the averaged duration times $\bar{\delta}_j$.

a few milliseconds for a single update of the estimator (including one iteration of the particle filter with 500 particles). At the same time we use established or new statistical methods such as particle filters in nonlinear state space models, EM-type algorithms, and adaptation by quasi mean squared error minimization.

The contribution of this work is manifold. First, we have proposed a nonlinear market microstructure noise model that covers bid-ask bounces, time-varying bid-ask spreads, and the discreteness of prices observed in real data. Second, the problem of on-line volatility estimation has been treated in a nonlinear state-space framework. It has been shown that the filtering distribution of the efficient price can be approximated with a particle filter and that the volatility can be estimated as a parameter of the filtering distribution. Third, we have presented a new bias-corrected sequential EM-type algorithm which allows the on-line estimation of time-varying volatility. Fourth, the problem of on-line adaptation has been treated satisfactorily (although still a bid ad-hoc from a theoretical viewpoint). The usefulness of the approach for real-time applications has been demonstrated through Monte Carlo simulations and applications to stock data.

Practical Aspects

Besides the new microstructure noise model we make a clear distinction between the (spot) volatility per time unit $\Sigma^c(t)$ and the volatility per transaction $\Sigma(t)$. Volatility in clock time usually is much more volatile than volatility in transaction time. We advocate the use of transaction time for modeling, i.e. to estimate $\Sigma(t)$, together with a subsequent transformation based on the trading intensity to obtain an estimator for $\Sigma^c(t)$. At least for our data sets it turned out that volatility in transaction time is almost constant (apart from the beginning of the trading day) and the fluctuation of clock time volatility is merely a result of fluctuation of the trading intensity (or the mean duration between subsequent trades). Thus a new focus in volatility estimation may be on the modeling of trading times. It is an interesting open question whether major external events do not only cause an increase in trading intensity but also an increase in transaction time volatility.

Furthermore, we are convinced that the distribution of asset returns in a transaction time model can be modeled in most situations quite well by a Gaussian distribution and many "jumps" observed in security prices sampled on an equally spaced clock time grid are due to a drastically increased number of transactions at that time. Our view is based on the investigation of several data sets (not reported in this work).

Another issue is the question for the correct goal in volatility estimation: We think that practitioners are more interested in a rapidly adapting (i.e. close to unbiased) and undersmoothed estimator instead of an oversmoothed estimator. In that case minimizing the mean squared error would not be the optimal strategy. We have presented with the approximately unbiased $\tilde{\Sigma}_{t_j|t_j}$ an estimator in this direction.

Mathematical Challenges

Of course it is desirable to have a complete mathematical theory on the methods of this work. However, we think that this is very hard to achieve. Here are a few comments in detail:

The results on the particle filter are mathematically exact given that the true volatility is known (i.e. with $\hat{\Sigma}^{\text{pf}}_{t_j} = \Sigma_{t_j}$) including the results from Proposition 6.1 on the optimal proposal and the importance weights. In particular it determines correctly the conditional distribution of the efficient prices given the observations.

Even in the case of constant volatility and for the simplest estimator $\hat{\Sigma}_{t_j}$ from (6.20) it seems to be very difficult to establish consistency and the asymptotic distribution. In the slightly simpler context of i.i.d.-observations convergence properties of recursive EM-type algorithms have been studied in Titterington (1984), Sato (2000), Wang and Zhao (2006), and Cappé and Moulines (2009) where also proofs of consistency and asymptotic normality are provided.

For strict mathematical results on local consistency or asymptotic normality some rescaling framework would be necessary. One approach could be to let the sampling frequency tend to infinity which would mean in the present setting of non-equally spaced observations that $\sup_j \tau'(j) \to 0$ where τ is defined as in Section 6.5.2. At the same time the maximal step size had to go to zero, i.e. $\sup_j \lambda_j \to 0$. Furthermore the assumption $\sup_j \tau'(j) \big/ \inf_j \lambda_j \to 0$ would be needed (this corresponds to the common assumption $n \to \infty$, $bn \to 0$ and $b \to 0$ for kernel estimates with bandwidth b). All "$\approx$"-signs in Appendix A.10 and most of the "$\approx$"-signs in Section 6.5.2 mean that the remaining terms are of lower order if these assumption were fulfilled.

An even bigger challenge is to determine the approximate mean squared error for the estimate (A.13) (with the particles instead of the efficient price as in Appendix A.10). This would require to prove the "$\approx$"-sign in (6.30) and (even harder) to prove the corresponding relation for the variance.

A strict mathematical result on bias reduction by combining two on-line algorithms with different step sizes (similar to (6.36) but with time-constant step sizes) has been proved in the context of time-varying ARCH models in Dahlhaus and Subba Rao (2007).

Finally, it is a mathematical challenge to put the interplay between transaction time volatility and clock time volatility on solid mathematical grounds - for example by proving consistency of the estimator $\hat{\Sigma}^c_{\text{alt}}(t_j) = \tilde{\Sigma}^*_{t_j|t_j}/\bar{\delta}_j$ in a subordinated differential equation model $d\mathbf{X}(t) = \Gamma(t)\, d\mathbf{W}_{N(t)}$ with an adequate point process $N(t)$.

Chapter 7

Bayesian Phase Estimation for Noisy Quasi-Periodic Time Series

7.1 Introduction

In the last decade, the estimation of the instantaneous phase of noisy oscillators experienced significant attraction in a broad range of fields including engineering (e.g. channel decoding), signal processing (e.g. signal denoising), physics (e.g. chaotic oscillators), and neuroscience (e.g. seizure detection). In the latter two fields, the estimated phase is often used for the detection of phase synchronization of oscillators (Rosenblum, Pikovsky, and Kurths 1996). In engineering, estimation of the carrier frequency is of interest. Another application of phase estimation is the detection of characteristic features and anomalies, for instance, in electrocardiogram (ECG) recordings (Clifford, Azuaje, and McSharry 2006). Traditional approaches for phase estimation are based on the Hilbert transform (Rosenblum, Pikovsky, and Kurths 1996), Wavelet transform (Grossmann, Kronland-Martinet, and Morletet 1989), and the periodogram (Hannan 1973). In the non-constant frequency case these methods are applied to rolling data windows. This renders them, at least theoretically, limited to the estimation of locally constant frequencies. In practice, they often fail not only in situations of fast varying frequency but also in cases when the signal is noisy or when it incorporates baseline changes.

We propose herein a new model for stationary time series with a quasi-periodic component. It is defined through

$$Y_t = A_t g(\phi_t) + B_t + \epsilon_t \tag{7.1}$$

with amplitude A_t, phase ϕ_t, baseline B_t, and i.i.d. Gaussian noise ϵ_t. Both, the amplitude and the baseline are allowed to be time-varying. g is a 2π-periodic function representing a fluctuation pattern. The phase ϕ_t is assumed to be a monotonically increasing process. (This complies with the usual definition of the phase as the integrated frequency.) In its full generality, the model allows to be fitted to a broad range of noisy quasi-periodic time series. An important special case is the cosine model

$$Y_t = A_t \cos(\phi_t) + \epsilon_t.$$

This work concerns the estimation of A_t, B_t, and ϕ_t given the observed time series $Y_t = y_t$. Our approach is based on a state-space representation of model (7.1). Within the state-space model, the amplitude, the phase, and the baseline are modelled as latent states. For the estimation, we propose an efficient Rao-Blackwellized particle smoother that combines the Kalman smoother and a particle smoother. As discussed in the preceding chapters particle smoothers (and particle filters) are Bayesian simulation methods for sequential estimation of the hidden states of a general state-space model. More precisely, given observations $y_{1:T} = \{y_1, \ldots, y_T\}$ the task is to approximate the filtering distributions $p(a_t, b_t, \phi_t|y_{1:t})$ or the smoothing distributions $p(a_t, b_t, \phi_t|y_{1:T})$ of the hidden states A_t, B_t, and ϕ_t. Estimates of the hidden states can be obtained as the means of these distributions.

In practice, the periodic function g is often unknown. For this setting, an original nonparametric EM algorithm is suggested which allows that one estimates g iteratively. It is mentioned that this nonparametric EM algorithm is not limited to our specific model but it is a general method for the nonparametric estimation of functions within non-linear state-space models.

We show empirically that our method allows to obtain reliable estimates in cases when the traditional methods fail. In addition, the proposed method can be applied in on-line settings.

7.2 A New State-Space Model for Quasi-Periodic Time Series

Here, we introduce a general setting which allows for on-line estimation of the model. The amplitude A_t, the baseline B_t, and the phase ϕ_t are modeled as unobserved Markov processes. By specifying the transition distributions, the problem can be written in terms of a general state-space model with state equations

$$\begin{aligned} \begin{pmatrix} A_t \\ B_t \end{pmatrix} &= H \begin{pmatrix} A_{t-1} \\ B_{t-1} \end{pmatrix} + \begin{pmatrix} \xi_t \\ \zeta_t \end{pmatrix}, & (7.2) \\ \phi_t &= f(\phi_{t-1}, \eta_t), & (7.3) \end{aligned}$$

and observation equation

$$Y_t = A_t g(\phi_t) + B_t + \epsilon_t,$$

where $(\xi_t, \zeta_t)^T \sim \mathcal{N}(\mathbf{0}, Q)$ and $\epsilon_t \sim \mathcal{N}(0, \sigma_\epsilon^2)$. H is a transition matrix. The evolution of the phase is modeled through function f and i.i.d. noise η_t. It is assumed that ϵ_t, η_t, and $(\xi_t, \zeta_t)^T$ are mutually and serially independent. In order to meet the requirement of ϕ_t to be non-decreasing, f and the distribution of η_t need to be chosen so that $f(\phi_{t-1}, \eta_t) \geq \phi_{t-1}$ is satisfied. We propose to model the phase differences $\Delta\phi_t = \phi_t - \phi_{t-1}$ as durations within an ACD(1,0) (autoregressive conditional duration) model. The ACD model was originally introduced by Engle and Russell (1998) as a model for the dependency structure of the durations between consecutive transactions in financial markets. The ACD(1,0) model is defined through

$$\Delta\phi_t = \breve{\psi}_t \eta_t \quad \text{where} \quad \breve{\psi}_t = \alpha + \beta \Delta\phi_{t-1},$$

with positive random increments η_t (e.g. Beta or Gamma distributed). In addition, it is assumed that $\mathbf{E}\eta_t = 1$. If the expectation of η_t is not normalized, η_t can be replaced by $\tilde{\eta}_t = \eta_t / \mathbf{E}\eta_t$.

The restrictions $\alpha, \beta > 0$ and $\beta < 1$ are imposed on the parameters. It can be shown that the (unconditional) mean of the phase increments is

$$\mathbf{E}[\Delta\phi_t] = \frac{\alpha}{1-\beta}, \tag{7.4}$$

which can be interpreted as the average frequency. A state-space representation of the ACD(1,0) model is given through

$$\begin{pmatrix} \phi_t \\ \psi_t \end{pmatrix} = \begin{pmatrix} \phi_{t-1} + (\alpha + \beta\psi_{t-1})\eta_t \\ (\alpha + \beta\psi_{t-1})\eta_t \end{pmatrix}, \tag{7.5}$$

where $\psi_t = \breve{\psi}_t \eta_t$ is an auxiliary state which carries the information of the duration model from time $t-1$ to t.

As can be seen from (7.2), the amplitude and the baseline evolutions are described through a VAR(1) model. It is assumed that the diagonal entries of H are close or equal to one and that the diagonal entries of Q are very small. That ensures that the amplitude and the baseline vary slowly. In practice, it usually suffices to assume that $H = \operatorname{diag}(1,1)$ and Q is a diagonal matrix. In the setting of constant (but unknown) amplitude and baseline, one will replace (7.2) by $(A_t, B_t)^T = (A_{t-1}, B_{t-1})^T$ which simplifies the estimation significantly.

A key feature of our model is that, conditional on the phase, it is a linear, Gaussian state-space model. In the following sections, it is shown that this allows the usage of the Kalman filter and the Kalman smoother for inference on the amplitude and the baseline.

7.3 The Estimation Method

7.3.1 Rao-Blackwellized Particle Filtering

We introduce a Rao-Blackwellized particle filter which generalizes the particle filter discussed in Section 2.4. The posterior distribution can be decomposed as

$$p(\mathbf{x}_{0:t}|y_{1:t}) = p(\phi_{0:t}, \psi_{0:t}|y_{1:t})p(a_{0:t}, b_{0:t}|y_{1:t}, \phi_{0:t}),$$

where $\mathbf{X}_t = (A_t, B_t, \phi_t, \psi_t)^T$ throughout this chapter. The basic idea of the Rao-Blackwellized particle filter is to compute $p(a_{0:t}, b_{0:t}|y_{1:t}, \phi_{0:t})$ with the well-known Kalman filter (Kalman 1960) while approximating $p(\phi_{0:t}, \psi_{0:t}|y_{1:t})$ using particles $\{(\phi^i_{0:t}, \psi^i_{0:t})^T, \omega^i_t\}_{i=1}^N$ generated by a particle filter. This gives the approximation

$$p(a_{0:t}, b_{0:t}, \phi_{0:t}|y_{1:t}) \approx \sum_{i=1}^N \omega^i_t p(a_{0:t}, b_{0:t}|y_{1:t}, \phi^i_{0:t})\delta_{\phi^i_{0:t}}(\phi_{0:t}). \tag{7.6}$$

The particle filter employs the relation

$$p(\phi_{0:t}, \psi_{0:t}|y_{1:t}) \propto p(\phi_{0:t-1}, \psi_{0:t-1}|y_{1:t-1})p(y_t|y_{1:t-1}, \phi_{0:t})p(\phi_t, \psi_t|\phi_{t-1}, \psi_{t-1}).$$

Note that contrary to the basic particle filter (see Section 2.4), the likelihood term $p(y_t|y_{1:t-1}, \phi_{0:t})$ does not reduce to $p(y_t|\phi_t)$. The relation (7.6) implies that the marginal densities $p(a_t, b_t|y_{1:t})$

are approximated by a mixture of Gaussian distributions

$$p(a_t, b_t|y_{1:t}) \approx \sum_{i=1}^{N} \omega_t^i \mathcal{N}\left(a_t, b_t|(a_t^i, b_t^i)^T, \Sigma_t^i\right),$$

where the means $(a_t^i, b_t^i)^T$ and covariances matrices Σ_t^i are computed by the Kalman filter. The following Rao-Blackwellized particle filter is similar to the algorithm in de Freitas (2001). For notational convenience, we set $C_t^i = (g(\phi_t^i), 1)$.

Algorithm: Rao-Blackwellized Particle Filter (RBPF)

Initialization (for $t = 0$)

- **For** $i = 1, \ldots, N$: Sample $(\phi_0^i, \psi_0^i)^T \sim p(\phi_0, \psi_0)$, set $\omega_0^i = 1$, and choose a_0^i, b_0^i, Σ_0^i according to prior knowledge.

Filtering (for $t = 1, 2, \ldots$)

1. *Kalman Prediction Step*

   - **For** $i = 1, \ldots, N$: Compute

$$\begin{aligned}(a_{t|t-1}^i, b_{t|t-1}^i)^T &= H(a_{t-1}^i, b_{t-1}^i)^T,\\ \Sigma_{t|t-1}^i &= H\Sigma_{t-1}^i H^T + Q.\end{aligned}$$

2. *Importance Sampling Step*

   - **For** $i = 1, \ldots, N$: Sample $(\phi_t^i, \psi_t^i)^T \sim p(\phi_t, \psi_t|\phi_{t-1}^i, \psi_{t-1}^i)$, compute $S_t^i = C_t^i \Sigma_{t|t-1}^i (C_t^i)^T + \sigma_\epsilon^2$ and evaluate importance weights

$$\breve{\omega}_t^i \propto \omega_{t-1}^i p(y_t|y_{1:t-1}, \phi_{0:t}^i) = \omega_{t-1}^i \mathcal{N}\left(y_t|C_t^i(a_{t|t-1}^i, b_{t|t-1}^i)^T, S_t^i\right).$$

   - **For** $i = 1, \ldots, N$: Normalize importance weights $\omega_t^i = \breve{\omega}_t^i / (\sum_{j=1}^N \breve{\omega}_t^j)$.

3. *Resampling Step*

   - **If** $\mathrm{ESS}(\{\omega_t^i\}_{i=1}^N) < 0.2N$: Resample $\{(\phi_{0:t}^i, \psi_{0:t}^i, a_{0:t-1}^i, a_{t|t-1}^i, b_{0:t-1}^i, b_{t|t-1}^i, \Sigma_{0:t-1}^i, \Sigma_{t|t-1}^i, S_t^i)^T, \omega_t^i\}_{i=1}^N$ with replacement and set $\omega_t^i = 1/N$ for $i = 1, \ldots, N$.

4. *Kalman Updating Step*

   - **For** $i = 1, \ldots, N$: Compute

$$\begin{aligned}(a_t^i, b_t^i)^T &= (a_{t|t-1}^i, b_{t|t-1}^i)^T + \Sigma_{t|t-1}^i (C_t^i)^T \left\{y_t - C_t^i(a_{t|t-1}^i, b_{t|t-1}^i)^T\right\} (S_t^i)^{-1},\\ \Sigma_t^i &= \Sigma_{t|t-1}^i - \left\{\Sigma_{t|t-1}^i (C_t^i)^T C_t^i \Sigma_{t|t-1}^i\right\} (S_t^i)^{-1}.\end{aligned}$$

The estimates are obtained through $\hat{a}_t = \sum_{i=1}^N \omega_t^i a_t^i$, $\hat{b}_t = \sum_{i=1}^N \omega_t^i b_t^i$, and $\hat{\phi}_t = \sum_{i=1}^N \omega_t^i \phi_t^i$ respectively.

7.3.2 Rao-Blackwellized Fixed-Lag Particle Smoothing

In the preceding section, the filtering distributions were used for inference on the hidden states. However, the estimates can be improved upon by using smoothing distributions. We propose to use fixed-lag smoothing with lag l. That is, the estimates for time t are computed based on all information, which is available up to time $t+l$. Consequently, the task is to approximate the (fixed-lag) smoothing distributions $p(\mathbf{x}_t|y_{1:t+l})$. In practice, lag l will be relatively small. It can be chosen such that it incorporates, say, approximately two periods of g. However, even much smaller values may suffice. Let's assume the signal is observed up to time $t+l$ and an approximation of the posterior distribution $p(\mathbf{x}_{0:t+l}|y_{1:t+l})$ is obtained from the RBPF. Then, the (marginal) smoothing distribution of the phase can be approximated through marginalization

$$p(\phi_t|y_{1:t+l}) \approx \sum_{i=1}^{N} \omega^i_{t+l}\delta_{\phi^i_t}(\phi_t).$$

For the amplitude and the baseline one yields

$$p(a_t,b_t|y_{1:t+l}) \approx \sum_{i=1}^{N} \omega^i_{t+l} p(a_t,b_t|y_{1:t+l},\phi^i_{0:t+l}) = \sum_{i=1}^{N} \omega^i_{t+l}\mathcal{N}\left(a_t,b_t|(\tilde{a}^i_t,\tilde{b}^i_t)^T,\tilde{\Sigma}^i_t\right),$$

where $(\tilde{a}^i_t,\tilde{b}^i_t)^T$ and $\tilde{\Sigma}^i_t$ are computed with the Kalman smoother. Smoothing by marginalization has been criticized for causing sample impoverishment (Doucet, Gordon, and Krishnamurthy 1999). While this is true in general, it is not an issue in our setting because lag l is small and the resampling frequency is rather low. In contrast to smoothing algorithms which proceed backwards in time (see, for instance, Godsill, Doucet, and West 2004), smoothing by marginalization has the advantage that it can be applied on-line. When the observation at time t comes in, the estimates of time $t-l$ can be updated using the fixed-lag smoothing density. In addition, it is computationally very cheap.

Rao-Blackwellized Fixed-Lag Particle Smoothing Step

5. *Kalman Smoothing Step* (for $k = t-1,\dots,\max\{t-l,0\}$)

   - **For** $i = 1,\dots,N$: Compute

$$\begin{aligned}
V^i_k &= \Sigma^i_k H^T(\Sigma^i_{k+1|k})^{-1}, \\
(\tilde{a}^i_k,\tilde{b}^i_k)^T &= (a^i_k,b^i_k)^T + V^i_k\left\{(\tilde{a}^i_{k+1},\tilde{b}^i_{k+1})^T - H(a^i_k,b^i_k)^T\right\}, \\
\tilde{\Sigma}^i_k &= \Sigma^i_k + V^i_k(\tilde{\Sigma}^i_{k+1} - \Sigma^i_{k+1|k})(V^i_k)^T, \\
\tilde{\Sigma}^i_{k,k-1} &= \Sigma^i_k(V^i_{k-1})^T + V^i_k(\tilde{\Sigma}^i_{k+1,k} - H\Sigma^i_k)(V^i_{k-1})^T,
\end{aligned}$$

   with initial values $(\tilde{a}^i_t,\tilde{b}^i_t)^T = (a^i_t,b^i_t)^T$, $\tilde{\Sigma}^i_t = \Sigma^i_t$, and $\tilde{\Sigma}^i_{t,t-1} = (I - K^i_tC^i_t)H\Sigma^i_{t-1}$.

6. *Result*

   - Obtain amplitude estimate $\hat{a}_k = \sum_{i=1}^N \omega_t^i \tilde{a}_k^i$, baseline estimate $\hat{b}_k = \sum_{i=1}^N \omega_t^i \tilde{b}_k^i$, and phase estimate $\hat{\phi}_k = \sum_{i=1}^N \omega_t^i \phi_k^i$ for time $k = \max\{t-l, 0\}$.

The Rao-Blackwellized fixed-lag particle smoother (RBPS) algorithm is obtained by combining the RBPF with the smoothing step. Note, the cross-covariances $\tilde{\Sigma}_{k,k-1}^i$ are only required in the parameter estimation step (see the following section). We emphasize on the computational efficiency of the RBPS. It has computational costs $\mathcal{O}(lNT)$ for smoothing T time steps. In each iteration only the particles for times $t-l-1, \ldots, t$ are required, implying a storage requirement of $\mathcal{O}(lN)$. As mentioned earlier, l will be rather small in practice.

7.3.3 A Stochastic EM Algorithm for Parameter Estimation

For application of the proposed RBPS in practice, it is required to estimate the state-space model's unknown parameter vector $\theta = (\alpha, \beta, \sigma_\epsilon^2, \mathrm{vec}(H), \mathrm{vec}(Q))^T$. We consider the estimation of θ based on a stochastic EM algorithm (compare Section 2.5). Let's assume signal y_t is received up to time T. The EM algorithm maximizes the likelihood $p_\theta(y_{1:T})$ iteratively. In the E-step, the expectation

$$\mathcal{Q}(\theta|\theta^{(m)}) = \mathbf{E}_{\theta^{(m)}}[\log p_\theta(\mathbf{X}_{0:T}, y_{1:T})|y_{1:T}]$$

is approximated, where $\theta^{(m)}$ is the current parameter estimate. This expectation can be decomposed as

$$\begin{aligned}\mathcal{Q}(\theta|\theta^{(m)}) &= \mathbf{E}_{\theta^{(m)}}[\log p(\phi_0, \psi_0)|y_{1:T}] + \sum_{t=1}^T \mathbf{E}_{\theta^{(m)}}[\log p_\theta(y_t|\mathbf{X}_t)|y_{1:T}] \\ &\quad + \sum_{t=1}^T \mathbf{E}_{\theta^{(m)}}[\log p_\theta(A_t, B_t|A_{t-1}, B_{t-1})|y_{1:T}] + \sum_{t=1}^T \mathbf{E}_{\theta^{(m)}}[\log p_\theta(\phi_t, \psi_t|\phi_{t-1}, \psi_{t-1})|y_{1:T}].\end{aligned}$$

It follows that $\mathcal{Q}(\theta|\theta^{(m)})$ can be approximated through smoothing particles, which were generated with respect to parameter value $\theta^{(m)}$. That is, we obtain

$$\begin{aligned}\hat{\mathcal{Q}}(\theta|\theta^{(m)}) &= \text{const} - \frac{1}{2}\sum_{t=1}^T \sum_{i=1}^N \tilde{\omega}_t^i \left[\log 2\pi + \log \sigma_\epsilon^2 + \frac{1}{\sigma_\epsilon^2}\left\{y_t^2 - 2C_t^i(\tilde{a}_t^i, \tilde{b}_t^i)^T y_t + C_t^i \tilde{S}_t^i (C_t^i)^T\right\}\right] \\ &\quad -\frac{1}{2}\sum_{t=1}^T \sum_{i=1}^N \tilde{\omega}_t^i \left[2\log 2\pi + \log|Q| + \mathrm{tr}\left\{Q^{-1}(\tilde{S}_t^i - H\tilde{S}_{t-1,t}^i - \tilde{S}_{t,t-1}^i H^T + H\tilde{S}_{t-1}^i H^T)\right\}\right] \\ &\quad + \sum_{t=1}^T \sum_{i=1}^N \tilde{\omega}_t^i \log p_{\alpha,\beta}(\phi_t^i, \psi_t^i|\phi_{t-1}^i, \psi_{t-1}^i),\end{aligned}$$

where $\tilde{\omega}_t^i = \omega_{\min\{t+l,T\}}^i$ are the smoothing weights and

$$\begin{aligned}\tilde{S}_t^i &= \tilde{\Sigma}_t^i + (\tilde{a}_t^i, \tilde{b}_t^i)^T(\tilde{a}_t^i, \tilde{b}_t^i), \\ \tilde{S}_{t,t-1}^i &= \tilde{\Sigma}_{t,t-1}^i + (\tilde{a}_t^i, \tilde{b}_t^i)^T(\tilde{a}_{t-1}^i, \tilde{b}_{t-1}^i) = (\tilde{S}_{t-1,t}^i)^T.\end{aligned}$$

In the M-step, a new parameter estimate $\theta^{(m+1)}$ is obtained by maximizing $\hat{\mathcal{Q}}(\theta|\theta^{(m)})$. Maximization with respect to σ_ϵ^2, H, and Q yield the estimates

$$\begin{aligned}
(\sigma_\epsilon^2)^{(m+1)} &= \frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{N}\tilde{\omega}_t^i\left\{y_t^2 - 2C_t^i(\tilde{a}_t^i,\tilde{b}_t^i)^T y_t + C_t^i\tilde{S}_t^i(C_t^i)^T\right\},\\
H^{(m+1)} &= \left(\sum_{t=1}^{T}\sum_{i=1}^{N}\tilde{\omega}_t^i\tilde{S}_{t,t-1}^i\right)\left(\sum_{t=1}^{T}\sum_{i=1}^{N}\tilde{\omega}_t^i\tilde{S}_{t-1}^i\right)^{-1},\\
Q^{(m+1)} &= \frac{1}{T}\left\{\sum_{t=1}^{T}\sum_{i=1}^{N}\tilde{\omega}_t^i\tilde{S}_t^i - H^{(m+1)}\sum_{t=1}^{T}\sum_{i=1}^{N}\tilde{\omega}_t^i\tilde{S}_{t-1,t}^i\right\}.
\end{aligned}$$

It is easy to see that the memory requirement can be reduced by computing the estimators recursively. For α and β, numerical maximization is required because no closed-form expression can be derived.

The RBPS produces the smoothing particles $\{(\phi_t^i,\psi_t^i)^T,\tilde{\omega}_t^i\}_{i=1}^N$, the means $\{(\tilde{a}_t^i,\tilde{b}_t^i)^T\}_{i=1}^N$, and the covariance matrices $\{\tilde{\Sigma}_t^i\}_{i=1}^N$, which approximate the fixed-lag smoothing distributions. Also the cross-covariance matrices $\{\tilde{\Sigma}_{t,t-1}^i\}_{i=1}^N$ are computed. However, approximations of the fixed-interval smoothing distributions $p_{\theta^{(m)}}(\mathbf{x}_t|y_{1:T})$ are actually required to approximate $\mathcal{Q}(\theta|\theta^{(m)})$. We argue that, for sufficiently large l, the fixed-lag and fixed-interval smoothing distributions are very similar. In practice, even for l that is small one can obtain a reliable approximation.

7. *(Parametric) EM Step*
   - Update parameter estimators $(\sigma_\epsilon^2)^{(m+1)}$, $H^{(m+1)}$, and $Q^{(m+1)}$.
   - Update numerical maximization of $\hat{\mathcal{Q}}_t(\alpha,\beta|\alpha^{(m)},\beta^{(m)})$ to obtain parameter estimates $\alpha^{(m+1)}$ and $\beta^{(m+1)}$.

There are two different approaches for switching to the next parameter estimate $\theta^{(m+1)}$. In a batch setting, one will restart the RBPS with $\theta^{(m+1)}$. In on-line applications, one can switch to the new parameter values after every T time steps.

7.4 Nonparametric Estimation of the Fluctuation Pattern

Here, we discuss the case when the function g is unknown. We propose a new nonparametric EM algorithm that estimates g iteratively. In the E-step

$$\begin{aligned}
\mathcal{Q}(g|g^{(m)}) &= \text{const} + \sum_{t=1}^{T}\mathbf{E}_{g^{(m)}}[\log p_g(y_t|\mathbf{X}_t)|y_{1:T}]\\
&\propto \text{const} - \sum_{t=1}^{T}\mathbf{E}_{g^{(m)}}[\{Y_t - A_t g(\phi_t) - B_t\}^2|y_{1:T}] \qquad (7.7)
\end{aligned}$$

is computed. The M-step consists of maximizing $\mathcal{Q}(g|g^{(m)})$ with respect to g. In the following we will derive an estimator for g which is based on the output of the RBPS. The basic idea is approximate the densities $p_{g^{(m)}}(\phi_t|y_{1:T})$ (which are contained in (7.7)) through kernel density estimates based on the smoothing particles. This yields

$$p_{g^{(m)}}(\phi_t|y_{1:T}) \approx \frac{1}{h_t}\sum_{i=1}^{N} \tilde{\omega}_t^i K\{(\phi_t - \phi_t^i)/h_t\} \tag{7.8}$$

with kernel function K and bandwidth h_t. The proposition given below shows that this leads to an estimator for g which is also based on kernel approximations. For the derivation of a convenient estimator we need to assume that the support of K is bounded (which is fulfilled, for instance, by the Epanechnikov kernel) and that h_t is chosen such that $K\{(\phi_t - \phi_t^i)/h_t\} = 0$ for $|\phi_t - \phi_t^i| > 2\pi$.

Proposition 7.1. *Assume that (i) particles generated by the RBPS with respect to $g^{(m)}$ are available, (ii) the assumptions on the kernel function K and bandwidths h_t given above hold, and (iii) g is 2π-periodic. Then, $\mathcal{Q}(g|g^{(m)})$ is approximately maximized by the estimate*

$$\hat{g}^{(m+1)}(\phi) = \frac{\sum_{t=1}^{T}\sum_{i=1}^{N} \tilde{\omega}_t^i K\{(\phi - \phi_t^i mod\ 2\pi)/h_t\}\{y_t\tilde{a}_t^i - (\tilde{S}_t^i)_{12}\}}{\sum_{t=1}^{T}\sum_{i=1}^{N} \tilde{\omega}_t^i K\{(\phi - \phi_t^i mod\ 2\pi)/h_t\}(\tilde{S}_t^i)_{11}}.$$

Proof. See Appendix A.12.

An important property of EM algorithms is that an EM iteration never reduces the likelihood. For our nonparametric EM algorithm this is shown in the following proposition. A more detailed analysis of the convergence properties of this algorithm in beyond the scope of this work.

Proposition 7.2. *The nonparametric EM algorithm never decreases the (log-)likelihood, that is $p_{g^{(m+1)}}(y_{1:T}) \geq p_{g^{(m)}}(y_{1:T})$.*

Proof. See Appendix A.13.

The RBPS combined with the nonparametric EM algorithm does not guarantee that all 2π-periodic structure is included in the estimated fluctuation pattern. Therefore, after each iteration the following transformations are applied which remove all 2π-periodic structures from the phase, amplitude, and baseline estimates and transfer them to the fluctuation pattern:

$$\begin{aligned}
\breve{\phi}_t^i &= 2\pi\{\hat{F}_\phi(\phi_t^i \mathrm{mod} 2\pi) + \lfloor \phi_t^i/(2\pi)\rfloor\} && (7.9)\\
\breve{a}_t^i &= \tilde{a}_t^i/\hat{a}(\breve{\phi}_t^i \mathrm{mod} 2\pi) && (7.10)\\
\breve{b}_t^i &= \tilde{b}_t^i - \hat{b}(\breve{\phi}_t^i \mathrm{mod} 2\pi) && (7.11)\\
\breve{g}^{(m+1)}(\phi) &= \hat{a}(\hat{F}_\phi^{-1}(\breve{\phi})) \times \hat{g}^{(m+1)}(\hat{F}_\phi^{-1}(\breve{\phi})) + \hat{b}(\hat{F}_\phi^{-1}(\breve{\phi})) && (7.12)
\end{aligned}$$

with $\breve{\phi} = (\phi \mathrm{mod} 2\pi)/(2\pi)$ and $\hat{F}_\phi$ being the distribution function of the 2π-folded phase which is given by

$$\hat{F}_\phi(y) = \int_0^y \hat{p}_\phi(\phi)d\phi, \tag{7.13}$$

where

$$\hat{p}_\phi(\phi) = \frac{1}{h} \sum_{t=1}^{T} \sum_{i=1}^{N} \tilde{\omega}_t^i K \left(\frac{\phi - \phi_t^i \mathrm{mod} 2\pi}{h_\phi} \right).$$

Note, the empirical distribution function cannot be used (instead of (7.13)) because it is not invertible. The functions $\hat{a}$ and $\hat{b}$ (in (7.10) and (7.11)) are kernel estimates of the 2π-folded amplitude and baseline. They are given by

$$\hat{a}(\phi) = \frac{\sum_{t=1}^{T} \sum_{i=1}^{N} \tilde{\omega}_t^i K\{(\phi - \phi_t^i \mathrm{mod} 2\pi)/h_a\} \tilde{a}_t^i}{\sum_{t=1}^{T} \sum_{i=1}^{N} \tilde{\omega}_t^i K\{(\phi - \phi_t^i \mathrm{mod} 2\pi)/h_a\}}$$

and

$$\hat{b}(\phi) = \frac{\sum_{t=1}^{T} \sum_{i=1}^{N} \tilde{\omega}_t^i K\{\phi - \phi_t^i \mathrm{mod} 2\pi)/h_b\} \tilde{b}_t^i}{\sum_{t=1}^{T} \sum_{i=1}^{N} \tilde{\omega}_t^i K\{(\phi - \phi_t^i \mathrm{mod} 2\pi)/h_b\}},$$

respectively.

In this section, various bandwidth parameters (h_t, h_ϕ, h_a, h_b) occurred which need to chosen in practice. We propose the convenient method of cross-validation to obtain reasonable values.

8. *Nonparametric EM Step*

   - Compute the estimator $\hat{g}^{(m+1)}$ of the fluctuation pattern g and obtain the transformed pattern $\breve{g}^{(m+1)}$ (which is used in the next iteration) according to (7.12).
   - Obtain the initial particles and weights $\breve{\phi}_0^i$, $\breve{a}_0^i$, $\breve{b}_0^i$, and $\tilde{\omega}_0^i$, $i = 1, \ldots, N$, for the next iteration.

Analogous to the parametric EM step one will switch to the new fluctuation pattern $\breve{g}^{(m+1)}$ after every T time steps. To start up the iteration an initial guess $\hat{g}^{(0)}$ is required. In Section 7.6.3, noisy ECG recordings are studied and it is shown that an uninformative function $\hat{g}^{(0)}$ may suffice in practice.

Remark: In practice the transformations (7.9), (7.10), and (7.11) need only be applied to the initial particles ϕ_0^i, a_0^i, b_0^i.

7.5 Discussion

The aim of this work was to propose a flexible model for quasi-periodic time series and an efficient procedure for its estimation. In addition, simulation results are provided which demonstrate the usefulness of the approach for practical applications (see Section 7.6). A rigorous theoretical analysis of the proposed model and methods is, however, beyond the scope of the present work. In particular, identifiability issues of the model and the convergence properties of the estimators are important questions which should be tackled in future work.

Now, certain aspects of our method which are relevant for its practical application are discussed. In a theoretical analysis these aspects should also be considered.

- The state equations (7.2) and (7.3) do not claim to be appropriate models for the actual amplitude, baseline, and phase processes. For instance, the amplitude should be positive which is not guaranteed by (7.2). The reason for the use of these models is their flexibility and ease of estimation. In this sense, the state equations are just a tool for estimating the components of the observation equation. It is mentioned that more complex models (e.g. nonlinear models for the amplitude and baseline) could be applied in our framework.

- An assumption of our model and the estimation method is that the amplitude and baseline processes vary slowly. In particular, large jumps may cause problems in practice. If the amplitude indeed varies slowly the estimated variance of the innovations ξ_t will be small. This, in turn, usually implies that the estimator produces positive estimates for the amplitude which is desired.

- In practice g should be a smooth function. If g is known a few discontinuities can be allowed if their number is small compared with the number of observations. However, if g is unknown and the nonparametric EM algorithm is used we implicitly assume that the kernel estimate (7.8) is a good approximation of the smoothing density which, in turn, leads to smoothness assumptions on g.

- When the nonparametric EM algorithm is used or the data is very noise, it is important to provide good initial parameter values for the phase, amplitude, and baseline models. This is necessary because no other prior information is given to the estimation method. Particularly, one should ensure that the average phase increment (7.4) is relatively close to the "true" value. This can be done by counting the number of cycles in the data (or in a small subset) and computing a rough estimate of the average phase increment.

7.6 Simulations

In this section, results of the proposed algorithms for benchmark problems and an application to human electrocardiogram recordings are presented.

7.6.1 Simulated Data

We consider a case when the true amplitude, baseline, and phase are available. We generate observations y_t, $t = 1, \dots, 1000$, from the general state-space model defined through (7.5) and

$$Y_t = A_t \cos(\phi_t) + B_t + \epsilon_t,$$

where $a_t = 0.2\sin(2\pi t/1000) + 0.4$ and $b_t = 0.4t/750\ \mathbf{1}_{t\leq 750} + (0.4 - 0.4(t-750)/250)\ \mathbf{1}_{t>750}$. The ACD model parameters are set to $\alpha = 0.2$ and $\beta = 0.99$. Two levels of the observation noise are investigated: $\sigma_\epsilon^2 = 0.01$ and $\sigma_\epsilon^2 = 0.16$. The parameters $(\alpha, \beta, \sigma_\epsilon^2, \text{vec}(Q))$ are estimated with the (parametric) EM algorithm and we set $H = \text{diag}(1, 1)$. For both noise levels, the EM algorithm obtains estimates $(\hat{\alpha}, \hat{\beta}, \hat{\sigma}_\epsilon^2)$ which were very close to the true values after a few iterations. For Q, we obtain $\text{diag}(10^{-4}, 5 \times 10^{-5})$.

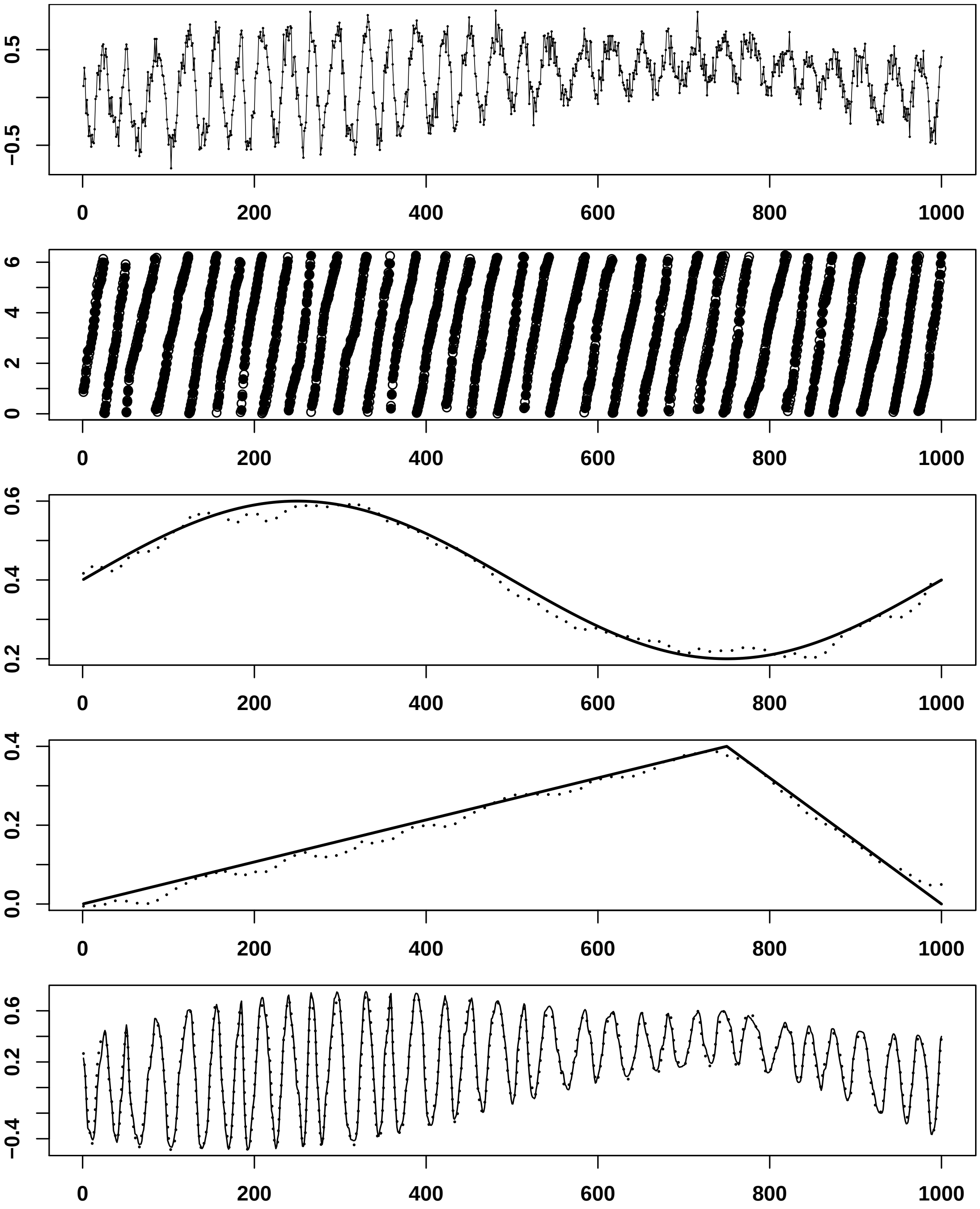

Figure 7.1: The estimation results of the RBPS for the simulated signal with $\mathcal{N}(0, 0.01)$ noise (from top to bottom): The simulated noisy observations; the folded estimated phase (circles) and the folded true phase (solid circles); the estimated amplitude (dotted line) and the true amplitude (solid line); the estimated baseline (dotted line) and the true baseline (solid line); the simulated non-noisy signal (solid line) and the denoised signal obtained from the RBPS estimates (dotted line).

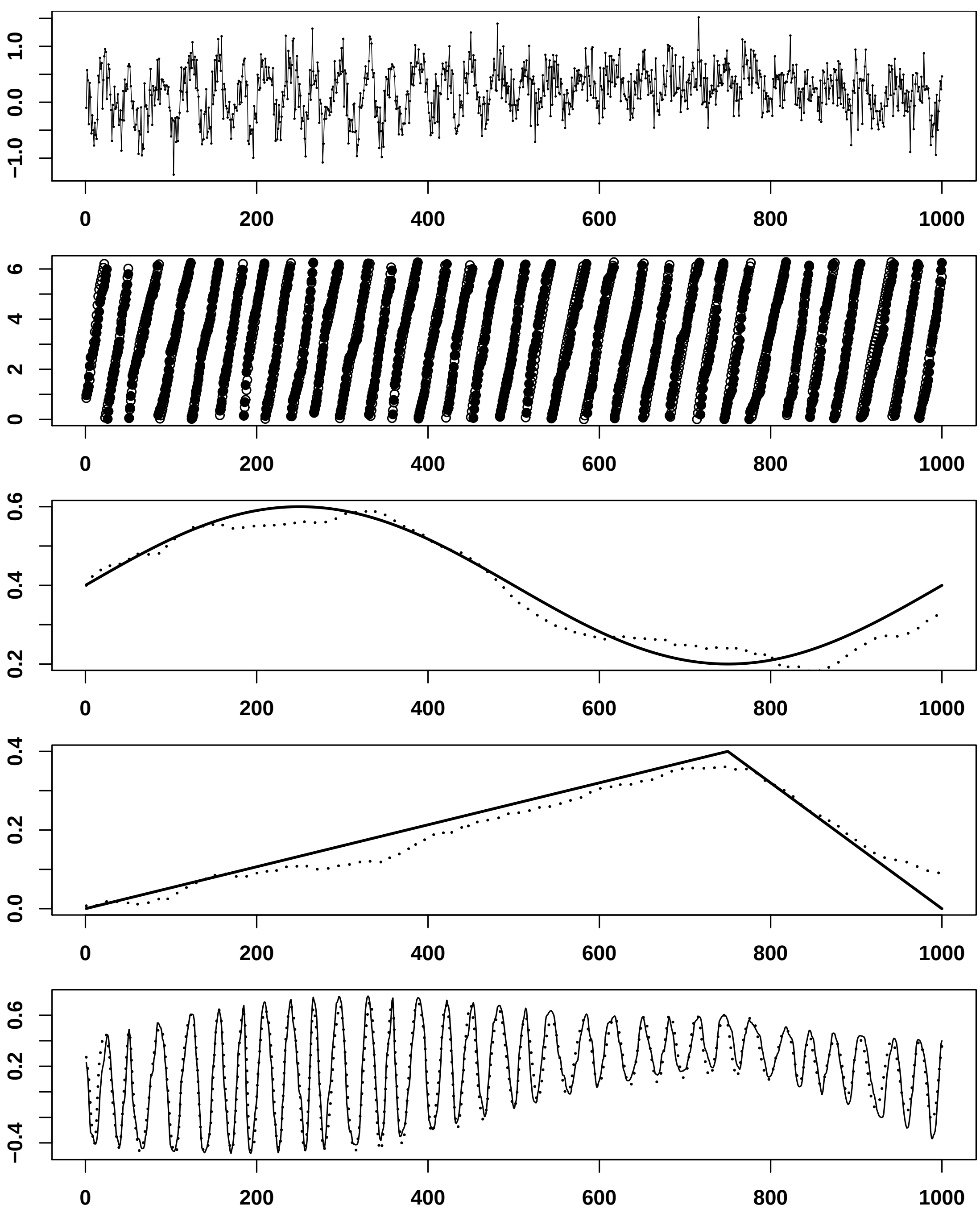

Figure 7.2: The estimation results of the RBPS for the simulated signal with $\mathcal{N}(0, 0.16)$ noise (from top to bottom): The simulated noisy observations; the folded estimated phase (circles) and the folded true phase (solid circles); the estimated amplitude (dotted line) and the true amplitude (solid line); the estimated baseline (dotted line) and the true baseline (solid line); the simulated non-noisy signal (solid line) and the denoised signal obtained from the RBPS estimates (dotted line).

The figures 7.1 and 7.2 show the true values and the estimated values for the two noise levels, respectively. The estimates of the amplitude, baseline, and (folded) phase are computed by the RBPS with $N = 500$ particles and lag $l = 100$. In addition, the figures display a signal reconstruction based on the estimates (see bottom plots), that is the estimated denoised observations $\hat{y}_t = \hat{a}_t \cos \hat{\phi}_t + \hat{b}_t$. For comparison, also the non-noisy observations $y_t - \epsilon_t$ are given. In the low-noise setting, it can be observed that the estimates are very accurate for all quantities. In the high-noise setting, the estimates are only slightly worse which is, to some extend, surprising given the low signal-to-noise ratio (particularly between times 600 to 900).

7.6.2 Noisy Rössler Attractor

Let's consider the Rössler attractor with the configuration

$$\begin{aligned}
\dot{x_1} &= -x_2 - x_3, \\
\dot{x_2} &= x_1 + .15x_2, \\
\dot{x_3} &= .4 + x_3(x_1 - 8.5).
\end{aligned}$$

The Rössler attractor and related systems are, for instance, used to model population dynamics (Blasius, Huppert, and Stone 1999; Lloyd and May 1999). We focus on the x_1 component for which the (folded) phase can be defined by means of

$$\arctan(x_{2,t}/x_{1,t}) \tag{7.14}$$

(see, for instance, Pikovsky et al. (1997)). It is assumed that $x_{1,t}$ is not observed directly but through $y_t = x_{1,t} + \epsilon_t$. One could replace $x_{1,t}$ in the denominator in (7.14) with the observations y_t to estimate the phase. However, at least in cases of large observation noise this would give very unstable estimates. Here, we apply the Hilbert transform and our method to estimate the phase.

As a result of the oscillation of x_1 being close to sinusoidal the cosine model

$$Y_t = A_t \cos(\phi_t) + \epsilon_t$$

can be used for estimation. The baseline does not need to be estimated because it is set to zero. This setting allows that we compare the phase estimate of the RBPS with the phase obtained from the Hilbert transform. The Hilbert transform is a well-known technique for phase estimation and it is often applied to (noisy) oscillators. Based on the Hilbert transform y_t^h of signal y_t the signal's phase ϕ_t can be defined through

$$\zeta_t = y_t + i y_t^H = a_t \exp(i\phi_t),$$

where ζ_t is called the analytic signal. The Hilbert transform y_t^H is defined as

$$y_t^H = \frac{1}{\pi} P \int \frac{y_s}{t-s} ds,$$

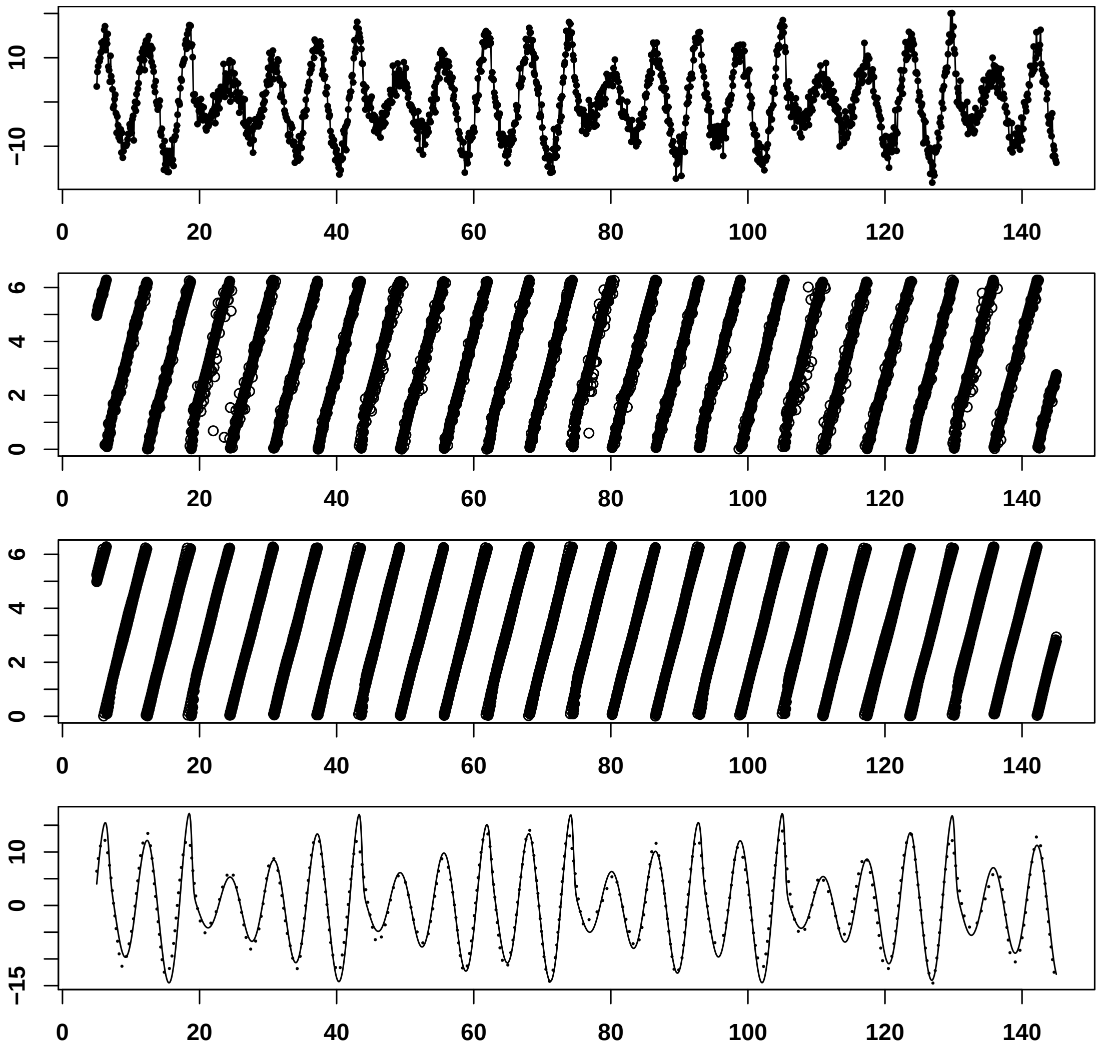

Figure 7.3: Estimation results for the noisy Rössler attractor. The plots show (from top to bottom): x_1-component of the Rössler attractor with additive i.i.d. $\mathcal{N}(0,4)$ noise; the folded Hilbert phase (circles) compared with the true folded phase (solid circles); the folded phase estimated with the RBPS (circles) compared with the true folded phase (solid circles); the (non-noisy) x_1-component of the Rössler attractor compared with the reconstructed (denoised) signal based on the amplitude and phase estimates of the RBPS.

with P being the Cauchy principal value. The Hilbert phase is computed from $y_t = \mathrm{Re}(\zeta_t) = a_t \cos(\phi_t)$.

We integrate the Rössler system with step size 0.1 using the Runge-Kutta method (Press et al. 1992, pp. 710-714) and we add i.i.d. Gaussian noise to the x_1-component. Again, two noise levels are considered: $\mathcal{N}(0,4)$ and $\mathcal{N}(0,40)$ (see top plots in figures 7.3 and 7.4). As parameter estimates we obtain $(\hat{\alpha},\hat{\beta})^T = (0.2, 0.02)^T$, $\hat{Q} = \mathrm{diag}(0.9, 0)$ (the second value is set to zero), and $\hat{\sigma}_\epsilon^2$ close to the true value. H was set to $\mathrm{diag}(1,0)$. The RBPS is applied with $N = 1000$ particles and lag $l = 200$. For the computation of the Hilbert phase a running window of 100 data points is used. The (folded) phase estimates of the Hilbert transform and our method for

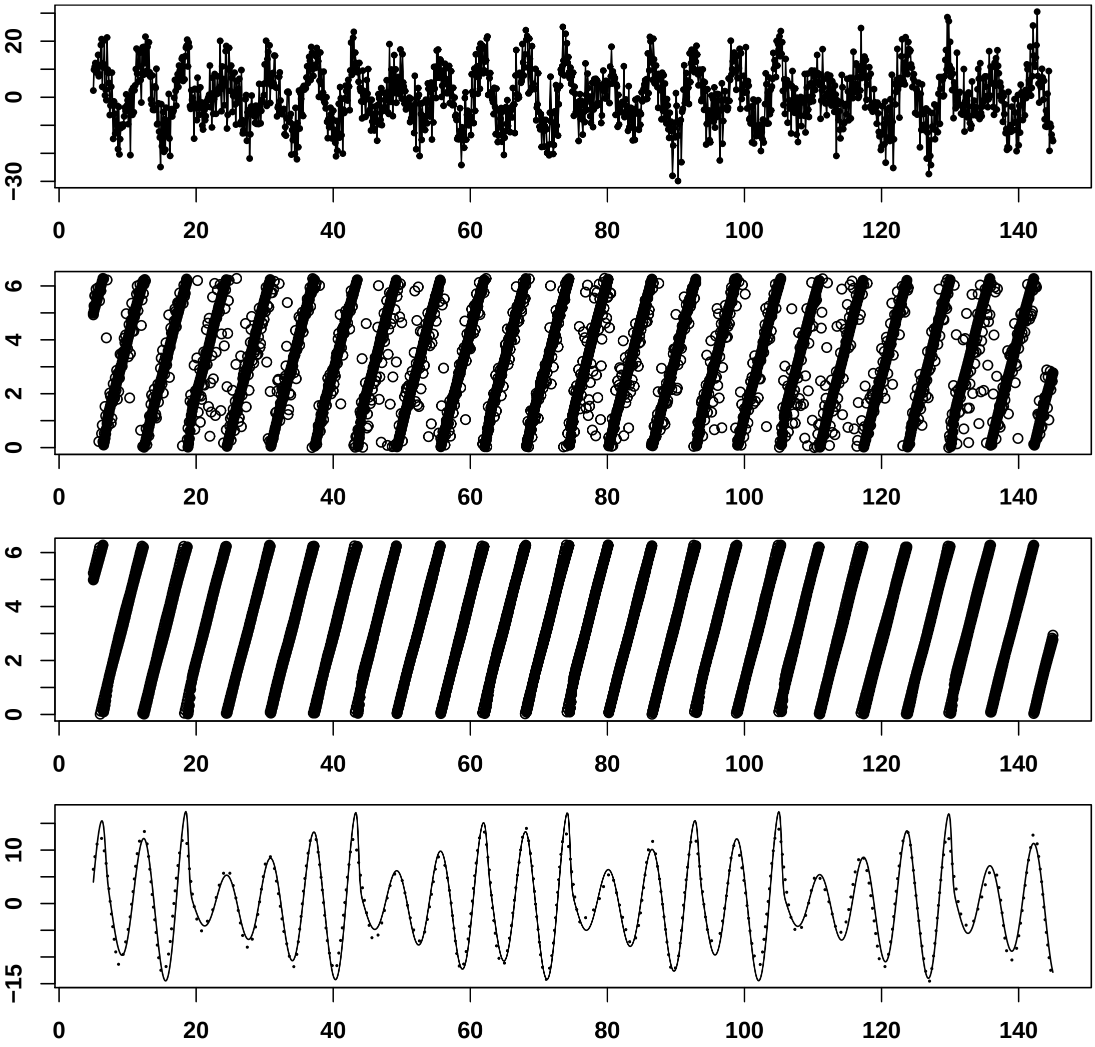

Figure 7.4: Estimation results for the noisy Rössler attractor. The plots show (from top to bottom): x_1-component of the Rössler attractor with additive i.i.d. $\mathcal{N}(0, 40)$ noise; the folded Hilbert phase (circles) compared with the true folded phase (solid circles); the folded phase estimated with the RBPS (circles) compared with the true folded phase (solid circles); the (non-noisy) x_1-component of the Rössler attractor compared with the reconstructed (denoised) signal based on the amplitude and phase estimates of the RBPS.

the two noise levels are presented in figures 7.3 and 7.4. It can be observed, that the phase estimate of the RBPS is much closer to the true phase than the Hilbert phase. The bottom plots show the (non-noisy) x_1-component of the Rössler attractor along with the denoised signal $\hat{y} = \hat{a}_t \cos(\hat{\phi}_t)$, where $\hat{a}_t$ and $\hat{\phi}_t$ are obtained from the RBPS. Note, that even in the high noise case, the denoised signal is very close to the true signal. In the light of the low signal-to-noise ratio this is a very satisfying result.

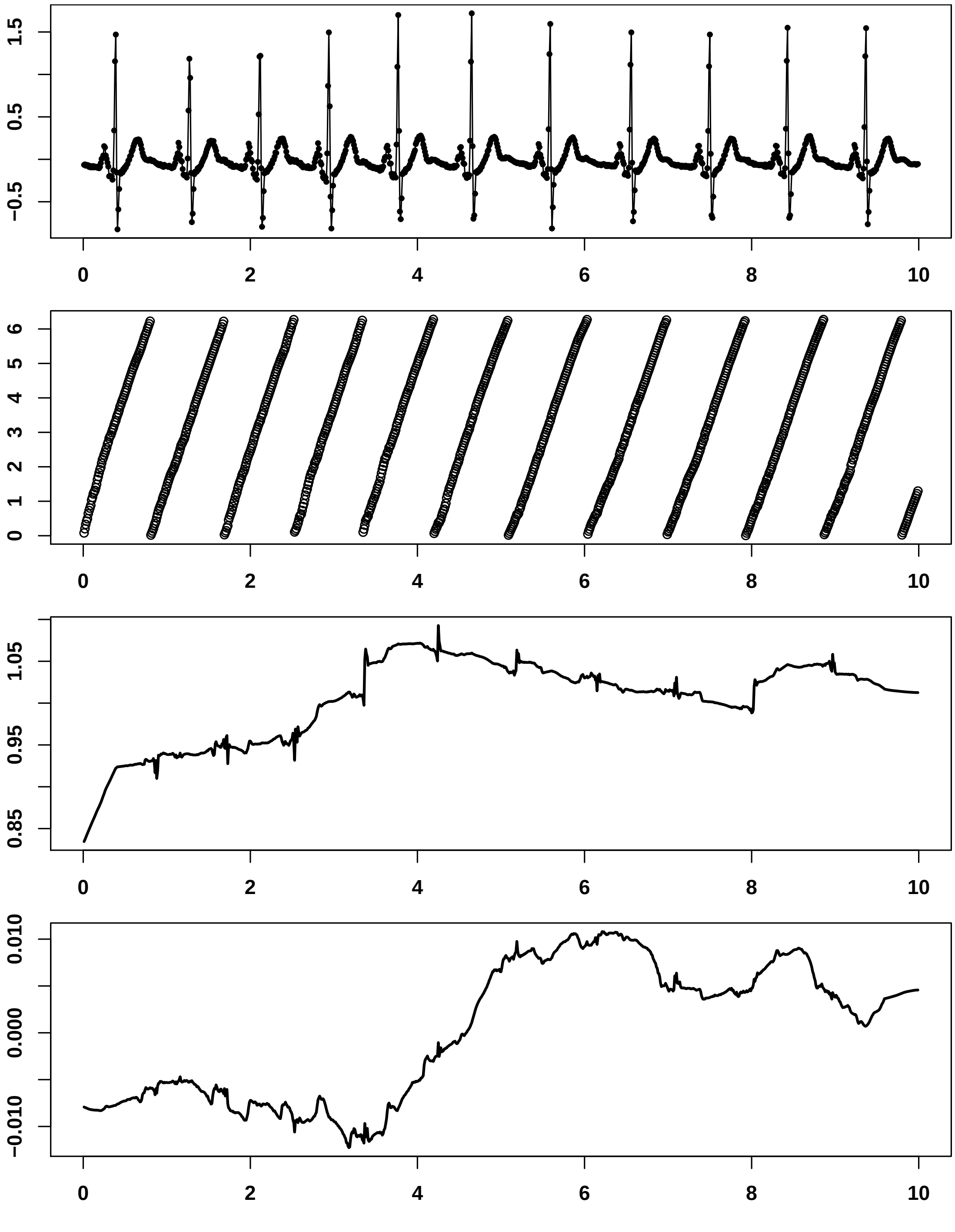

Figure 7.5: Estimation results for the ECG recordings. The plots show (from top to bottom): the ECG data points; the folded phase, the amplitude, and the baseline estimated by the RBPS.

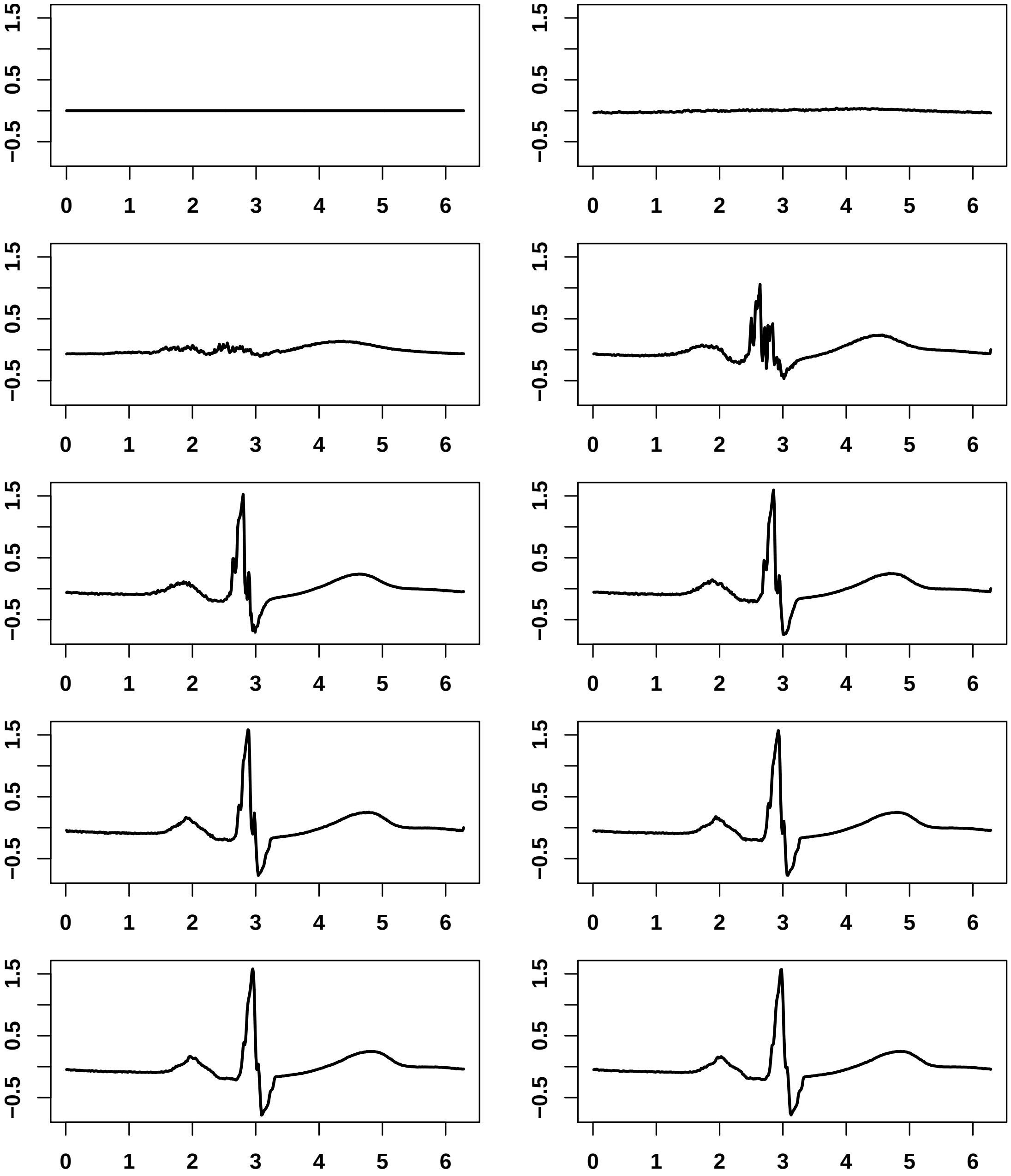

Figure 7.6: The initial fluctuation pattern $\hat{g}^{(0)}$ and the estimated fluctuation patterns $\check{g}^{(m)}$ for the iterations $m = 1, \dots, 9$ of the nonparametric EM algorithm.

7.6.3 Application to Human Electrocardiogram Recordings

Human ECG recordings are characterized by a specific fluctuation pattern, amplitude changes, and baseline shifts. In addition, they are often corrupted by noise. The fluctuation pattern heavily depends on certain characteristics of the specific human being. The baseline shifts are typically caused by respiration or body movements (Clifford, Azuaje, and McSharry 2006). Let's

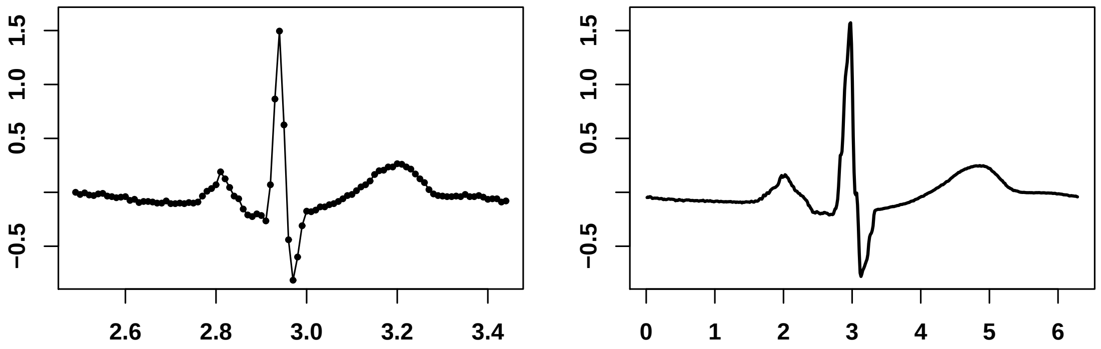

Figure 7.7: Left plot: A fraction of the ECG recordings. Right plot: Estimated fluctuation pattern $\breve{g}^{(9)}$ obtained after nine iterations of the nonparametric EM algorithm.

consider the model

$$Y_t = A_t g(\phi_t) + B_t + \epsilon_t,$$

where, in addition to the amplitude, phase, and baseline, the fluctuation pattern g is unknown.

We use ECG recordings obtained from the PhysioBank database[1]. The data are sampled at a frequency of 0.01 seconds for a duration of 10 seconds (which gives 1000 observations) and they are plotted in the top plot of Figure 7.5.

The RBPS and the EM algorithms are applied to the data in order to obtain estimates for ϕ_t, a_t, b_t, and g. As initial fluctuation pattern we use the trivial choice $\hat{g}^{(0)} \equiv 0$. The only "prior" information used is contained in the initial values for the parameters α and β. We set $\alpha^{(0)} = (1-\beta^{(0)})2\pi/90$ and $\beta^{(0)} = 0.2$ which, as a results of (7.4), implies $\mathbf{E}[\Delta\phi_t] = 2\pi/90$. This is reasonable because the average period of the fluctuation pattern observed in the data is roughly 90 time steps. The estimates for the amplitude, baseline, and phase computed by the RBPS which is applied with $N = 500$ particles and $l = 40$ are given in Figure 7.5. It can be seen that the amplitude changes significantly over time. In contrast, the baseline is almost constant for our data. The estimates of the fluctuation pattern $\breve{g}^{(m)}$ for the iterations $m = 1, \ldots, 9$ are shown in Figure 7.6. Observe how rapidly the estimates of the fluctuation pattern converge. Finally, the estimated fluctuation pattern $\breve{g}^{(9)}$ is compared with one period of the data (Figure 7.7).

It is mentioned that, in practice, our method could be used for denoising of ECG recordings or the detection of anomalies caused by certain diseases (Clifford, Azuaje, and McSharry 2006).

[1] `http://www.physionet.org/physiobank/`

Chapter 8

Software

In this chapter the computer software which was developed in the course of this dissertation is overviewed.

8.1 Overview

The algorithms proposed in this dissertation have been implemented in C++ under Windows using Microsoft Visual Studio 2005. The coding of the algorithms follows the detailed descriptions in the individual chapters. For some methods also R code is provided. The implementations are distributed in different C++ and R packages which are described in Section 8.2. The C++ packages depend on a set of auxiliary classes which are summarized in Section 8.3. The auxiliary classes were compiled into static libraries (`.lib`) which need to be linked at compilation time.

The implementation is strictly object-oriented which provides great flexibility for extending and reusing the software. Whenever possible abstract base classes (interface classes) were defined which give the generic interface for the derived classes. Two examples are given in figures 8.1 and 8.2. The exception handling at run time is done by throwing instance of the class `jcnError`. `jcnError` is derived from the Standard Template Library (STL) class `runtime_error` and it is contained in the auxiliary library `la.lib`. For more details on the implementation we refer to the Doxygen documentation which is available for all C++ source code. The source code and the documentations can be found on the CD accompanying this dissertation.

8.2 Main Software Packages

C++ Packages

`lbfp_demo` This package contains an implementation of the LBFP estimator and a demonstration of its usage. The details of the implementation are given in Section 3.4.2. (Sources: `lbfp_demo.zip`)

`npis` This package demonstrates the use of nonparametric (partial) importance sampling for financial derivative pricing. The option pricing examples as well as the

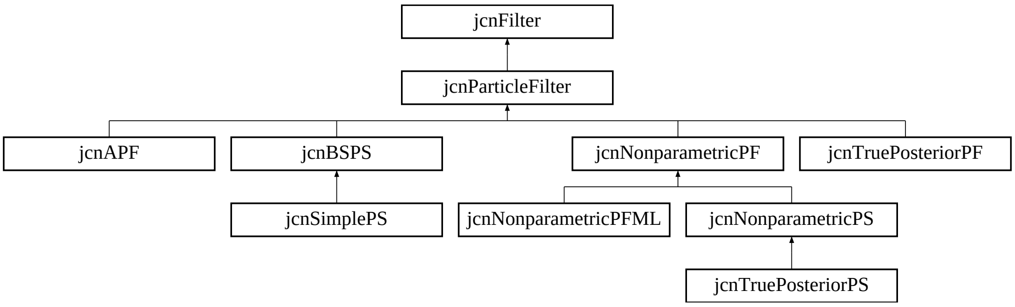

Figure 8.1: Inheritance diagram of the particle filter and smoother classes and the interface class `jcnFilter`.

benchmark Monte Carlo algorithms discussed in Chapter 4 are included. (Sources: `npis.zip`)

`npf` This package contains the source code of the algorithms NPF, NPS, and NPF+ML proposed in Chapter 5. The implementations of some benchmark algorithms are also included. `jcnFilter` is the abstract base class which provides the interface for the filter and smoother classes. All classes for filtering and smoothing have a (non-public) pointer to an object of the type `jcnStateSpaceModel` which specifies (among other things) the observation equation and state equation. The inheritance diagram of the particle filter and smoother classes is shown in Figure 8.1. See the Doxygen documentation for more details. (Sources: `npf.zip`)

`scve_demo` This is an efficient implementation of the on-line estimator for spot cross-volatility (SCVE) developed in Chapter 6. A simple demo is provided which shows how to use it in real applications. See the Doxygen documentation for more details. (Sources: `scve_demo.zip`)

`bpe_demo` This package contains the source code of the Rao-Blackwellized particle smoother and the nonparametric EM algorithm for the estimation of the state-space model for quasi-periodic time series proposed in Chapter 7. The usage is demonstrated through an example with simulated data and the application to the ECG data set. (Sources: `bpe_demp.zip`)

Some of these packages depend on auxiliary libraries which are described in Section 8.3. The details of the dependencies are explained in the readme files of the packages.

R-Packages

`lbfp` This is an R-package which makes the C++ implementation of the LBFP estimator accessible through R (R functions: `dlbfp`, `rlbfp`, `vallbfp`). It can be used to implement LBFP-based algorithms in R. (Sources: `lbfp.tar.gz`, Windows binary `lbfp.zip`)

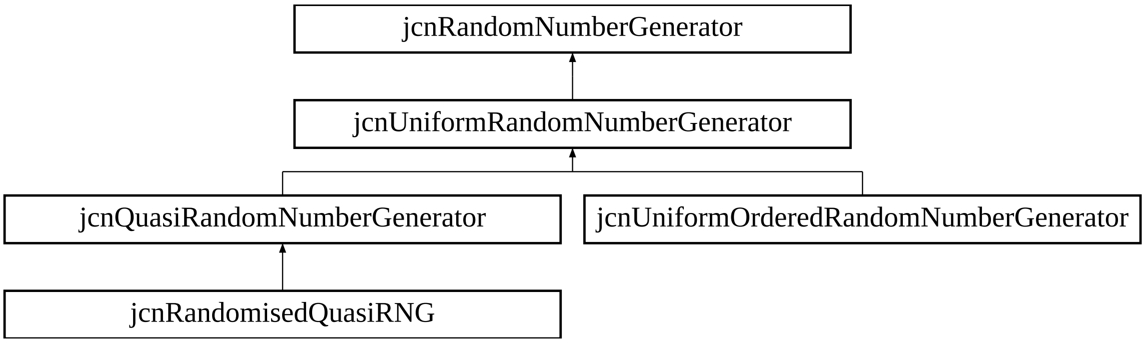
jcnRandomNumberGenerator
jcnUniformRandomNumberGenerator
jcnQuasiRandomNumberGenerator
jcnUniformOrderedRandomNumberGenerator
jcnRandomisedQuasiRNG

Figure 8.2: Inheritance diagram of the uniform random number generators and the interface class jcnRandomNumberGenerator.

`scve` This is an R-package which makes the C++ implementation of the algorithm for on-line spot cross-volatility estimation (SCVE) accessible through R (R function: `tvSpotCrossVolaEst`). In addition, it contains pure R implementations of the algorithms for (univariate) time-constant and time-varying spot volatility estimation (SVE) (R functions: `constSpotVolaEst`, `tvSpotVolaEst`). (Sources: `scve.tar.gz`, Windows binary `scve.zip`)

8.3 Auxiliary Libraries

The auxiliary classes are compiled into three static libraries (`.lib`) to make them usable in different software projects. The sources are available on the accompanying CD.

`la.lib` This library contains vector and matrix classes as well as numerical methods. The implementation of the vector and matrix classes is based on the vector template class `valarrey<double>` which is part of the STL and computationally very efficient. The vector and matrix classes provide overloaded operators including amongst others `operator[]`, `operator=`, `operator+`, `operator*=` for convenient element access and componentwise data manipulations. This allows the use of intuitive statements, for instance $x+ = y - A[1][1]$ with vectors x, y, matrix A, and scalar $A[1][1]$. In addition, many standard linear algebra operations (such as methods for matrix-matrix, matrix-vector multiplication and matrix inversion) and numerical methods are implemented. (Sources: `la.zip`)

`rand.lib` The implementations of random number generators for various distributions (Beta, Cauchy, Exponential, Gamma, multivariate normal, uniform) can be found in this library. It includes implementations of the Mersenne Twister 19937 (Matsumoto and Nishimura 1998) and the Sobol sequence (Sobol 1967) for pseudo- and quasi-random number generation, respectively. For the transformation of uniform random numbers into normal random numbers the Beasley-Springer-Moro approximation (Moro 1995) is used (compare Section 2.6). The random number generator classes which generate non-uniform variates need a uniform random number generator of the type `jcnUniformRandomNumberGenerator` as a source for uniform variates. For this purpose they are endowed with a private pointer to a `jcnUniformRandomNumberGenerator` class. All random number generator classes are inheritance from the interface class `jcnRandomNumberGenerator`. As an example the inheritance diagram for the random number generators which produce uniformly and quasi-uniformly distributed variates are given in Figure 8.2. (Sources: `rand.zip`)

`misc.lib` This library includes miscellaneous classes for plotting, for writing results to `.html` files, and for optimizing the console output. The usage is explained in the Doxygen documentation. (Sources: `misc.zip`)

Conclusions and Prospects

In this project new models and advanced Monte Carlo methods for discrete-time stochastic processes were proposed, investigated, and implemented. The methods developed are inspired by relevant application, however, they are generally applicable. In the following, the main results are briefly summarized and potential future research directions are pointed out.

In Chapter 3, new nonparametric importance sampling algorithms were suggested. The mean square error convergence properties were investigated and asymptotic optimality was shown. In particular, it was established that the NIS algorithm achieves a mean square error rate of $\mathcal{O}(N^{-(d+8)/(d+4)})$ which massively improves the standard Monte Carlo rate $\mathcal{O}(N^{-1})$ in low dimensions. The usefulness of our nonparametric importance sampling methods for practical sample sizes were shown through simulations and an application to a queueing problem. In contrast to previous work on nonparametric importance sampling we favored an LBFP instead of kernel estimators which is computationally much more efficient. It was shown that draws from an LBFP can be generated using the inversion method. Because the inversion method is a monotone transformation, it preserves the structure of the presampled uniformly distributed variates. This offers the opportunity to combine the NIS/NSIS algorithms with other variance reduction techniques such as stratified sampling, moment matching, and quasi-Monte Carlo techniques (Glasserman 2004; Robert and Casella 2004). Additionally, we emphasize that the LBFP estimator is not restricted to usage within nonparametric importance sampling. It is a reasonable alternative to other nonparametric estimators whenever sampling and evaluation are required.

In financial engineering and many other fields high-dimensional integration problems need to be solved. As a result of the curse of dimensionality and increasing computational complexity the direct application of nonparametric importance sampling is intractable for large dimensions. In Chapter 4 an NPIS algorithm was proposed that applies nonparametric importance sampling to a carefully chosen subspace. The mean square error convergence properties were explored. They establish the asymptotic optimality of the approach and suggest that NPIS improves over parametric importance sampling asymptotically. In particular, NPIS is shown to achieve increasing efficiency compared with crude Monte Carlo and parametric importance sampling. Its usefulness for practical sample sizes was verified through different option pricing scenarios. Large variance reduction factors were obtained in certain situations. It was shown that NPIS is advantageous over existing importance sampling methods for problems with low effective dimension, which is often the case in finance. Particularly, situations of rare event dependency or multi-modal optimal proposals are well suited for NPIS. There, existing methods often fail. The combination of NPIS and quasi-Monte Carlo resulted in enormous efficiency gains. In some cases variance reduction factors of the magnitude 10^5 were obtained. It is emphasized that NPIS can be applied without analytical investigation of the payout function. In addition, being generally applicable NPIS is not restricted to a specific kind of diffusion model or payout function. It can be applied to other settings occurring in finance, such as the estimation of option sensitivities or the evaluation of the value-at-risk. Our results suggest that it is generally worthwhile to combine methods from different fields to improve integral approximations. In this work is was shown how nonparametric importance sampling (i.e. an advanced Monte Carlo integration technique) and low-discrepancy sequences (i.e. a numerical integration rule) can be fruitfully combined.

In Chapter 5, it is was shown that nonparametric and quasi-Monte Carlo techniques have great potential to increase the efficiency of sequential Monte Carlo algorithms. In particular, the complexity reduction that can be achieved for smoothing and maximum likelihood algorithms is remarkable. The proposed algorithms are based on nonparametric importance sampling in the marginal space of the state. As a consequence, resampling can be avoided, because the particles' weights are prevented from degeneration. In addition, the developed particle filter and particle smoother allow the direct use of quasi-Monte Carlo sampling which is another advantage over existing methods. Through simulations it was shown that the methods better approximate the target distributions than existing algorithms. The use of quasi-Monte Carlo further improved the results. Until now, nonparametric methods used within particle filters were based on kernel estimators (see, for instance, Hürzeler and Künsch 1998; Musso, Oudjane, and Le Gland 2001). In Musso, Oudjane, and Le Gland (2001), a nonparametric regularization step in discussed. There, samples from nonparametric approximations of the filtering densities are produced to increase the particles' variety. We emphasize, that the nonparametric importance sampling (used in both the NPF and the NPS) not only increases the particles' variety but also serves as a variance reduction technique.

In Chapter 6, a new technique for the on-line estimation of time-varying cross-volatilities (covariance matrices) based on noisy, non-synchronously observed transaction data was presented. An important difference compared with existing methods is that we made a clear distinction between the spot cross-volatility in transaction time and clock time. Our algorithm works directly on the non-synchronous tick-by-tick data avoiding the difficulties associated with data synchronization. It updates the covariance estimate immediately after the occurrence of a new transaction and it is, therefore, as close to the market as possible. The contribution of this work is manifold. First, we proposed a nonlinear market microstructure noise model that well captures the major features (such as the bid-ask bounce, the discreteness of prices, and liquidity constraints) observed in real data. Second, a non-standard state-space model for non-synchronous data was introduced which allows each log-price process to evolve in its individual transaction time. Third, a new particle filter for non-synchronous state-space models was developed which can be used to approximate the filtering distributions of the efficient log-prices. It was shown that the cross-volatilities can be estimated as parameters of the filtering distributions. Forth, we presented a new sequential EM-type algorithm that allows the on-line estimation of (time-varying) covariance matrices. For the univariate case, we additionally proposed an on-line bias correction and a method for adaptive step size selection. Through Monte Carlo simulations and an application to real stock and future data, the usefulness of the algorithm for real-time applications was demonstrated. As an interesting empirical result we obtained that the correlations of high-frequency stock returns vary significantly over the trading day. This is an important result for risk management and trading. It is remarked that the developed sequential EM-type algorithm is not limited to the estimation of covariance matrices but it is a general technique for on-line parameter estimation in general state-space models. Future work might include the improvement of the multivariate method which will, however, be very challenging. In addition, our method could be generalized to more complex (multivariate) models for the efficient log-price processes.

Chapter 7 suggested a new model for stationary time series with a quasi-periodic component. A computationally efficient RBPS algorithm was proposed that allows for simultaneous estimation of the amplitude, the baseline, and the phase. The simulation results confirmed that the RBPS provides precise estimates even in cases of large observation noise which is a distinct advantage compared with existing methods. In contrast to existing methods for phase estimation, our framework models the observation noise explicitly. This is a reason for our method's good performance for noisy signals. We also considered the case when the fluctuation pattern is unknown in addition to the amplitude, baseline, and phase. For this case we developed an original nonparametric EM algorithm. The results for the ECG recordings suggest that this procedure works well in situations which are of practical interest. It is emphasized that the proposed nonparametric EM algorithm is not limited to our specific model but it is a general technique for nonparametric function estimation in general state-space models.

Appendix A

Proofs

A.1 Proof of Theorem 3.1

Proof. We denote q_M^{IS} and $\hat{q}_M^{\mathrm{IS}}$ briefly by q_M and $\hat{q}_M$. Because for $\varphi \geq 0$ we have $q_M = \varphi_M p I_{\varphi_M}^{-1}$, the variance σ_M^2 of $\hat{I}_{\varphi_M}^{\mathrm{NIS}}$ (conditional on $\{\tilde{\mathbf{x}}^1, \ldots, \tilde{\mathbf{x}}^M\}$) is given by

$$(N-M)\sigma_M^2 = I_{\varphi_M}^2 \int \frac{(\hat{q}_M(\mathbf{x}) - q_M(\mathbf{x}))^2}{\hat{q}_M(\mathbf{x})} d\mathbf{x}. \tag{A.1}$$

In order to get rid of $\hat{q}_M(\mathbf{x})$ in the denominator we write

$$\begin{aligned} \frac{N-M}{I_{\varphi_M}^2} \mathbf{E}[\sigma_M^2] &= \mathbf{E}\left[\int \frac{(\hat{q}_M(\mathbf{x}) - q_M(\mathbf{x}))^2}{q_M(\mathbf{x})} d\mathbf{x}\right] - \mathbf{E}\left[\int \frac{(\hat{q}_M(\mathbf{x}) - q_M(\mathbf{x}))^3}{\hat{q}_M(\mathbf{x}) q_M(\mathbf{x})} d\mathbf{x}\right] \\ &= K_M + R_M. \end{aligned}$$

The discrepancy between $\hat{q}_M$ and q_M can be investigated by

$$\begin{aligned} \hat{q}_M(\mathbf{x}) - q_M(\mathbf{x}) &= \frac{\hat{f}_M(\mathbf{x}) - \overline{\omega}_M q_M(\mathbf{x})}{I_{\varphi_M}} + \frac{\delta_M(1 - V_M q_M(\mathbf{x}))}{\overline{\omega}_M + V_M \delta_M} \\ &\quad + \left[\frac{\hat{f}_M(\mathbf{x}) - \overline{\omega}_M q_M(\mathbf{x})}{I_{\varphi_M}}\right] \left(\frac{I_{\varphi_M}}{\overline{\omega}_M + V_M \delta_M} - 1\right) \\ &= W_M(\mathbf{x}) + U_M^1(\mathbf{x}) + U_M^2(\mathbf{x}). \end{aligned} \tag{A.2}$$

It will be established later that $\mathbf{E}[W_M(\mathbf{x})]^2 = O(h^4 + (Mh^d)^{-1})$. Now let's show that

$$\mathbf{E}[U_M^1(\mathbf{x}) + U_M^2(\mathbf{x})]^2$$

is of lower order. Under the assumptions 1 through 3 we yield

$$\begin{aligned} &\mathbf{E}[U_M^1(\mathbf{x}) + U_M^2(\mathbf{x})]^2 \\ &\leq C(V_M\delta_M)^2 + C\left(\mathbf{E}\left[\frac{\hat{f}_M(\mathbf{x}) - \overline{\omega}_M q_M(\mathbf{x})}{I_{\varphi_M}}\right]^4\right)^{1/2} \left(\mathbf{E}\left[\frac{I_{\varphi_M}}{\overline{\omega}_M + V_M\delta_M} - 1\right]^4\right)^{1/2} \\ &\leq C(V_M\delta_M)^2 + \widetilde{C}\left(\frac{1}{Mh^d} + h^4\right)\left(\frac{1}{M^3(V_M\delta_M)^4} + (V_M\delta_M)^2 + \frac{1}{M^2}\right)^{1/2} \end{aligned}$$

with some constants C and $\widetilde{C}$. The last inequality follows from lemmas A.1 and A.2 (see appendices A.2 and A.3). Because, by Assumption 3, $V_M \delta_M = o(h^2)$ and $M^3(V_M\delta_M)^4 \to \infty$, we obtain

$$\mathbf{E}[U_M^1(\mathbf{x}) + U_M^2(\mathbf{x})]^2 = o(\mathbf{E}[W_M(\mathbf{x})]^2).$$

We conclude $K_M \approx \int \mathbf{E}[W_M(\mathbf{x})^2] q_M^{-1}(\mathbf{x}) d\mathbf{x}$.

It is not hard to work out that $\int \mathbf{E}[W_M(\mathbf{x})^2] q_M^{-1}(\mathbf{x}) d\mathbf{x}$ decomposes into an integrated squared bias term L_1 and an integrated variance term L_2:

$$\int \frac{(\mathbf{E}[\hat{f}_M(\mathbf{x}) I_{\varphi_M}^{-1}] - q_M(\mathbf{x}))^2}{q_M(\mathbf{x})} d\mathbf{x} + \int \frac{\mathrm{Var}[\hat{f}_M(\mathbf{x}) I_{\varphi_M}^{-1}]}{q_M(\mathbf{x})} d\mathbf{x} + O(M^{-1}) = L_1 + L_2 + O(M^{-1}).$$

For notational convenience, the following is shown only for $d = 1$. Without loss of generality, we assume $\mathbf{x} \in [-h/2, h/2)$. Then $\hat{f}_M I_{\varphi_M}^{-1}$ simplifies to

$$\frac{\hat{f}_M(\mathbf{x})}{I_{\varphi_M}} = \left(\frac{h/2 - \mathbf{x}}{h}\right) \frac{\hat{f}_0^{\mathrm{UH}}}{I_{\varphi_M}} + \left(\frac{h/2 + \mathbf{x}}{h}\right) \frac{\hat{f}_1^{\mathrm{UH}}}{I_{\varphi_M}} \tag{A.3}$$

where

$$\hat{f}_0^{\mathrm{UH}} = 1/(Mh) \sum_{j=1}^{M} \omega_M^j \mathbf{1}_{[-h,0)}(\tilde{\mathbf{x}}^j)$$

and

$$\hat{f}_1^{\mathrm{UH}} = 1/(Mh) \sum_{j=1}^{M} \omega_M^j \mathbf{1}_{[0,h)}(\tilde{\mathbf{x}}^j)$$

are the heights of the bins $[-h, 0)$ and $[0, h)$, respectively. For the computation of L_1, we need to compare the Taylor expansions of $\mathbf{E}[\hat{f}_M(\mathbf{x}) I_{\varphi_M}^{-1}]$ and q_M, which are given by

$$\begin{aligned} \mathbf{E}[\hat{f}_M(\mathbf{x}) I_{\varphi_M}^{-1}] &= q_M(0) + \mathbf{x} q_M'(0) + h^2 q_M''(0)/6 + O(h^3), \\ q_M(\mathbf{x}) &= q_M(0) + \mathbf{x} q_M'(0) + \mathbf{x}^2 q_M''(0)/2 + O(h^3). \end{aligned}$$

The former follows from (A.3) and from the expansion of the histogram

$$\mathbf{E}[\hat{f}_{0/1}^{\mathrm{UH}} I_{\varphi_M}^{-1}] = q_M(0) \ -/+ \ h q_M'(0)/2 + h^2 q_M''(0)/6 + O(h^3).$$

Thus we obtain

$$\{\mathbf{E}[\hat{f}_M(\mathbf{x}) I_{\varphi_M}^{-1}] - q_M(\mathbf{x})\}^2 \approx (h^2 - 3\mathbf{x}^2)^2 q_M''(0)^2/36. \tag{A.4}$$

Integration over $[-h/2, h/2)$ and using Taylor expansion of $1/q_M(\mathbf{x})$ about 0 leads to

$$\frac{q_M''(0)^2}{36} \int_{-h/2}^{h/2} \frac{(h^2 - 3\mathbf{x}^2)^2}{q_M(\mathbf{x})} d\mathbf{x} = \frac{49}{2880} \frac{q_M''(0)^2}{q_M(0)} h^5 + O(h^6). \tag{A.5}$$

By summing over all bins and applying the standard Riemann approximation, we yield

$$L_1 = \frac{49}{2880} h^4 \int \frac{q_M''(\mathbf{x})^2}{q_M(\mathbf{x})} d\mathbf{x} + O(h^5). \tag{A.6}$$

Next let's derive an approximation to L_2. From (A.3) we have

$$\begin{aligned}\mathrm{Var}[\hat{f}_M(\mathbf{x})I_{\varphi_M}^{-1}] &= \left(\frac{h/2-\mathbf{x}}{h}\right)^2 \mathrm{Var}[\hat{f}_0^{\mathrm{UH}} I_{\varphi_M}^{-1}] + \left(\frac{h/2+\mathbf{x}}{h}\right)^2 \mathrm{Var}[\hat{f}_1^{\mathrm{UH}} I_{\varphi_M}^{-1}] \\ &\quad + \frac{h^2/2-2\mathbf{x}^2}{h^2}\mathrm{Cov}[\hat{f}_0^{\mathrm{UH}} I_{\varphi_M}^{-1}, \hat{f}_1^{\mathrm{UH}} I_{\varphi_M}^{-1}].\end{aligned}$$

In addition, it can be shown that

$$\mathrm{Var}[\hat{f}_i^{\mathrm{UH}} I_{\varphi_M}^{-1}] \approx \frac{q_M(0)^2}{Mhq_0(0)} - \frac{q_M(0)^2}{M}$$

for $i = 0, 1$ and

$$\mathrm{Cov}[\hat{f}_0^{\mathrm{UH}} I_{\varphi_M}^{-1}, \hat{f}_1^{\mathrm{UH}} I_{\varphi_M}^{-1}] \approx -\frac{q_M(0)^2}{M},$$

similarly to Scott (1992, chap. 4). That is, we yield

$$\mathrm{Var}[\hat{f}_M(\mathbf{x})I_{\varphi_M}^{-1}] = \left(\frac{1}{2Mh} + \frac{2\mathbf{x}^2}{Mh^3}\right)\frac{q_M(0)^2}{q_0(0)} + O(M^{-1}).$$

Analogous to (A.5) and (A.6), we then obtain

$$\int_{-h/2}^{h/2} \frac{\mathrm{Var}[\hat{f}_M(\mathbf{x})I_{\varphi_M}^{-1}]}{q_M(\mathbf{x})} d\mathbf{x} = \frac{2q_M(0)}{3Mq_0(0)} + O(h/M)$$

and

$$L_2 = \frac{2}{3Mh}\int \frac{q_M(\mathbf{x})}{q_0(\mathbf{x})} d\mathbf{x} + O(M^{-1}), \tag{A.7}$$

respectively. Very similar computations in the multivariate case yield

$$K_M \approx h^4 H_{M,1} + \frac{2^d}{3^d M h^d} H_{M,2},$$

where

$$H_{M,1} = \frac{49}{2880}\sum_{i=1}^{d}\int \frac{(\partial_i^2 q_M)^2}{q_M} + \frac{1}{64}\sum_{i\neq j}\int \frac{\partial_i^2 q_M \partial_j^2 q_M}{q_M} \quad \text{and} \quad H_{M,2} = \int \frac{q_M}{q_0}.$$

It remains to show that R_M is negligible compared with K_M. To show this we follow the same lines as in Zhang (1996). We consider separately the restriction of R_M on the region A defined through

$$|\hat{q}_M(\mathbf{x}) - q_M(\mathbf{x})| > q_M(\mathbf{x})/2$$

and on its complement A^c. First, the restriction on A is considered. The construction of $\hat{q}_M$ implies $\hat{q}_M \geq \delta_M(\overline{\omega}_M + V_M\delta_M)^{-1} \geq C\delta_M > 0$ and we obtain

$$\begin{aligned}\int \mathbf{E}\left[\frac{(\hat{q}_M(\mathbf{x}) - q_M(\mathbf{x}))^3}{\hat{q}_M(\mathbf{x})q_M(\mathbf{x})}\mathbf{1}_A\right] d\mathbf{x} &\leq \frac{1}{C\delta_M}\int \mathbf{E}\left[\frac{(\hat{q}_M(\mathbf{x}) - q_M(\mathbf{x}))^3}{q_M(\mathbf{x})}\mathbf{1}_A\right] d\mathbf{x} \\ &\leq \frac{\widetilde{C}}{\delta_M}\int \mathbf{E}\left[\frac{(\hat{q}_M(\mathbf{x}) - q_M(\mathbf{x}))^4}{q_M(\mathbf{x})^3}\right] d\mathbf{x} \\ &\leq \frac{\widetilde{C}}{\delta_M c_M^3}\int \mathbf{E}\left[(\hat{q}_M(\mathbf{x}) - q_M(\mathbf{x}))^4\right] d\mathbf{x} \end{aligned} \tag{A.8}$$

with some constants C and $\widetilde{C}$. Second we consider the restriction on A^c. The definition of A implies that $\hat{q}_M(\mathbf{x}) > q_M(\mathbf{x})/2$ on A^c and we obtain

$$\begin{aligned}\int \mathbf{E}\left[\frac{(\hat{q}_M(\mathbf{x}) - q_M(\mathbf{x}))^3}{\hat{q}_M(\mathbf{x}) q_M(\mathbf{x})}\mathbf{1}_{A^c}\right] d\mathbf{x} &\leq C\int \mathbf{E}\left[\frac{(\hat{q}_M(\mathbf{x}) - q_M(\mathbf{x}))^3}{q_M(\mathbf{x})^2}\right] d\mathbf{x} \\ &\leq \frac{C}{c_M^2}\int \mathbf{E}\left[(\hat{q}_M(\mathbf{x}) - q_M(\mathbf{x}))^3\right] d\mathbf{x}. \qquad (A.9)\end{aligned}$$

Now it follows from (A.8) and (A.9) together with Lemma A.2 that, under Assumption 4a, $R_M = o(K_M)$.

The proof is finished by noting that the squared bias term in $\mathbf{E}[\hat{I}_{\varphi_M}^{\text{NIS}} - I_\varphi]^2$ is negligible as a result of Assumption 5a, and that the expressions $I_{\varphi_M}^2$ (in (A.1)), $H_{M,1}$, and $H_{M,2}$ can be substituted by their unrestricted counterparts because their differences are of lower order.

A.2 Lemma A.1

Lemma A.1. *Suppose that the assumptions 1 and 2 (from Chapter 3) hold. Then we have for some integer l*

$$\mathbf{E}\left[\hat{f}_M(\mathbf{x}) - \varpi_M q_M(\mathbf{x})\right]^{2l} = \mathcal{O}\left\{(Mh^d)^{-l} + h^{4l}\right\}.$$

Proof. It is easy to see that

$$\begin{aligned}\mathbf{E}\left[\hat{f}_M(\mathbf{x}) - \varpi_M q_M(\mathbf{x})\right]^{2l} &= \mathbf{E}\Big[\left\{\hat{f}_M(\mathbf{x}) - \mathbf{E}\hat{f}_M(\mathbf{x})\right\} + q_M(\mathbf{x})(I_{\varphi_M} - \varpi_M) \\ &\qquad + \left\{\mathbf{E}\hat{f}_M(\mathbf{x}) - \varphi_M(\mathbf{x})p(\mathbf{x})\right\}\Big]^{2l} \\ &\leq C\left\{\mathbf{E}\left[\hat{f}_M(\mathbf{x}) - \mathbf{E}\hat{f}_M(\mathbf{x})\right]^{2l} + q_M(\mathbf{x})^{2l}\mathbf{E}\left[I_{\varphi_M} - \varpi_M\right]^{2l}\right. \\ &\qquad \left. + \left[\mathbf{E}\hat{f}_M(\mathbf{x}) - \varphi_M(\mathbf{x})p(\mathbf{x})\right]^{2l}\right\}. \qquad (A.10)\end{aligned}$$

The first term in (A.10) is of order $\mathcal{O}\{(Mh^d)^{-l}\}$ as a result of (A.7) and

$$\mathbf{E}\left[\hat{f}_M(\mathbf{x}) - \mathbf{E}\hat{f}_M(\mathbf{x})\right]^{2l} \leq \widetilde{C}\left\{\mathbf{E}\left[\hat{f}_M(\mathbf{x}) - \mathbf{E}\hat{f}_M(\mathbf{x})\right]^2\right\}^l.$$

Because q_M is bounded the second term in (A.10) is $\mathcal{O}(M^{-l})$. The third term is the bias of the estimator $\hat{f}_M(\mathbf{x})$ to the power of $2l$. Thus, from (A.4) we obtain

$$\left[\mathbf{E}\hat{f}_M(\mathbf{x}) - \varphi_M(\mathbf{x})p(\mathbf{x})\right]^{2l} = \mathcal{O}(h^{4l})$$

which completes the proof.

A.3 Lemma A.2

Lemma A.2. *Suppose that the assumptions 1 through 3 (from Chapter 3) hold. For some integer l we have*

$$\mathbf{E}\left[\frac{I_{\varphi_M}}{\overline{\omega}_M + V_M\delta_M} - 1\right]^{2l} = \mathcal{O}\left(\frac{1}{M^{l+1}(V_M\delta_M)^{2l}} + (V_M\delta_M)^2 + \frac{1}{M^l}\right). \tag{A.11}$$

Proof. Analogous to Lemma 1 in Zhang (1996) we consider the expectation in (A.11) restricted on the region A given by

$$|\overline{\omega}_M + V_M\delta_M - I_{\varphi_M}| > I_{\varphi_M}/2$$

and on its complement A^c separately. For A we obtain

$$\begin{aligned}
\mathbf{E}\left[\left(\frac{I_{\varphi_M}}{\overline{\omega}_M + V_M\delta_M} - 1\right)\mathbf{1}_A\right]^{2l} &\le \frac{1}{(V_M\delta_M)^{2l}}\mathbf{E}\left[(I_{\varphi_M} - \overline{\omega}_M - V_M\delta_M)\,\mathbf{1}_A\right]^{2l} \\
&\le \frac{4}{(V_M\delta_M)^{2l} I_{\varphi_M}^2}\mathbf{E}\left[I_{\varphi_M} - \overline{\omega}_M - V_M\delta_M\right]^{2l+2} \\
&\le \frac{C}{(V_M\delta_M)^{2l}}\left\{\mathbf{E}\left[I_{\varphi_M} - \overline{\omega}_M\right]^{2l+2} + (V_M\delta_M)^{2l+2}\right\} \\
&\le \frac{C}{(V_M\delta_M)^{2l}}\left\{M^{-(l+1)} + (V_M\delta_M)^{2l+2}\right\}.
\end{aligned}$$

From the definition of A we yield that $\overline{\omega}_M + V_M\delta_M \ge I_{\varphi_M}/2$ holds on A^c which leads to

$$\begin{aligned}
\mathbf{E}\left[\left(\frac{I_{\varphi_M}}{\overline{\omega}_M + V_M\delta_M} - 1\right)\mathbf{1}_{A^c}\right]^{2l} &\le \left(\frac{2}{I_{\varphi_M}}\right)^{2l}\mathbf{E}\left[I_{\varphi_M} - \overline{\omega}_M - V_M\delta_M\right]^{2l} \\
&\le C\left\{\mathbf{E}\left[I_{\varphi_M} - \overline{\omega}_M\right]^{2l} + (V_M\delta_M)^{2l}\right\} \\
&\le C\left\{M^{-l} + (V_M\delta_M)^{2l}\right\}.
\end{aligned}$$

The lemma follows immediately.

A.4 Proof of Theorem 3.3

Proof. Again q_M is shorthand for q_M^{IS}. Let $f_{\varphi_M} = \left(\frac{\varphi_M p}{I_{\varphi_M}} - \frac{|\varphi_M| p}{\overline{I}_{\varphi_M}}\right)$. Straightforward calculations yield

$$\begin{aligned}
(N-M)\sigma_M^2 &= I_{\varphi_M}^2 \int \left(\frac{\varphi_M p}{I_{\varphi_M}} - \frac{|\varphi_M| p}{\overline{I}_{\varphi_M}} + q_M - \hat{q}_M\right)^2 \hat{q}_M^{-1} \\
&= I_{\varphi_M}^2 \left[\int f_{\varphi_M}^2 \frac{(q_M - \hat{q}_M)}{q_M \hat{q}_M} + 2\int f_{\varphi_M}\frac{(q_M - \hat{q}_M)}{\hat{q}_M} + \int \frac{(q_M - \hat{q}_M)^2}{\hat{q}_M} + \int \frac{f_{\varphi_M}^2}{q_M}\right] \\
&= I_{\varphi_M}^2 \left[T_1 + T_2 + T_3 + T_4\right].
\end{aligned}$$

Term T_4 is independent of the nonparametric estimation and we have $I^2_{\varphi_M} T_4 = \overline{I}^2_{\varphi_M} - I^2_{\varphi_M}$. The expectation of term T_1 can be written as

$$\int f^2_{\varphi_M} \frac{\mathbf{E}[q_M - \hat{q}_M]}{q^2_M} - \int f^2_{\varphi_M} \frac{\mathbf{E}[q_M - \hat{q}_M]^2}{q^3_M} + \int f^2_{\varphi_M} \frac{\mathbf{E}[q_M - \hat{q}_M]^3}{q^3_M \hat{q}_M} = T_{1,1} + T_{1,2} + T_{1,3}.$$

Similar expressions are obtained for quantities T_2 and T_3. We begin with $T_{1,1}$. Analogous to (A.2), we conclude

$$q_M(\mathbf{x}) - \hat{q}_M(\mathbf{x}) \approx -[\hat{f}_M(\mathbf{x}) - \overline{\omega}_M q_M(\mathbf{x})]/\overline{I}_{\varphi_M}.$$

From the proof of Theorem 3.1, we also know that

$$\mathbf{E}[\hat{f}_M(\mathbf{x}) - \overline{\omega}_M q_M(\mathbf{x})/\overline{I}_{\varphi_M}] = \mathbf{E}[\hat{f}_M(\mathbf{x})\overline{I}^{-1}_{\varphi_M}] - q_M(\mathbf{x}) = (h^2 - 3\mathbf{x}^2) q''_M(0)/6 + O(h^3)$$

for $d = 1$ and $\mathbf{x} \in [-h/2, h/2)$. Then we obtain

$$\frac{q''_M(0)}{6} \int_{-h/2}^{h/2} f_{\varphi_M}(\mathbf{x})^2 \frac{h^2 - 3\mathbf{x}^2}{q_M(\mathbf{x})^2} d\mathbf{x} = \frac{h^3}{8} f_{\varphi_M}(0)^2 \frac{q''_M(0)}{q_M(0)^2} + O(h^4)$$

using a Taylor expansion of $f_{\varphi_M}(\mathbf{x})^2/q_M(\mathbf{x})^2$ about 0. Finally, summing over all bins and using the Riemann approximation gives $T_{1,1}$ in the one-dimensional case:

$$-\frac{h^2}{8} \int f_{\varphi_M}(\mathbf{x})^2 \frac{q''_M(\mathbf{x})}{q_M(\mathbf{x})^2} d\mathbf{x} + O(h^3).$$

In the multivariate case we yield

$$T_{1,1} = -\frac{h^2}{8} \int f_{\varphi_M}(\mathbf{x})^2 \frac{\nabla^2 q_M(\mathbf{x})}{q_M(\mathbf{x})^2} d\mathbf{x} + O(h^3).$$

Term $T_{1,2}$ can be treated analogous to $\int \mathbf{E}[W_M(\mathbf{x})^2] q^{-1}_M(\mathbf{x}) d\mathbf{x}$ in the proof of Theorem 3.1. We end up with

$$T_{1,2} = -\frac{2^d}{3^d M h^d} \int \frac{f^2_{\varphi_M}}{q_0 q_M} - \left[\frac{49h^4}{2880} \sum_{i=1}^{d} \int f^2_{\varphi_M} \frac{(\partial^2_i q_M)^2}{q^3_M} + \frac{h^4}{64} \sum_{i \neq j} \int f^2_{\varphi_M} \frac{\partial^2_i q_M \partial^2_j q_M}{q^3_M} \right].$$

Comparing the term in brackets with $T_{1,1}$, we observe that the former is negligible. Furthermore, similarly to R_M in the proof of Theorem 3.1, it follows that $T_{1,3}$ is negligible compared with $T_{1,2}$ provided that Assumption 4b holds.

The calculations for T_2 and T_3 are very similar to those of T_1 and therefore are omitted. Putting all terms together we obtain

$$\begin{aligned}(N - M)\mathbf{E}[\sigma^2_M] \;=\; & I^2_{\varphi_M} \Big\{ \frac{2^d}{3^d M h^d} \left(\int \frac{q_M}{q_0} - 2 \int \frac{f_{\varphi_M}}{q_0} - \int \frac{f^2_{\varphi_M}}{q_0 q_M} \right) \\ & - h^2 \left(\int f^2_{\varphi_M} \frac{\nabla^2 q_M}{8 q^2_M} + \int f_{\varphi} \frac{\nabla^2 q_M}{4 q_M} \right) \Big\} \times \; (1 + o(1)) + (\overline{I}^2_{\varphi_M} - I^2_{\varphi_M}).\end{aligned}$$

We observe that the terms restricted on M can be substituted by their asymptotic limits, which completes the proof.

A.5 Proof of Theorem 3.4

Proof. We denote q_M^{SIS} and $\hat{q}_M^{\mathrm{SIS}}$ briefly by q_M and $\hat{q}_M$. Because the bias of $\hat{I}_{\varphi_M}^{\mathrm{NSIS}}$ is asymptotically negligible, we have

$$\mathbf{E}[\hat{I}_{\varphi_M}^{\mathrm{NSIS}} - I_\varphi]^2 = (N-M)^{-1}\mathbf{E}[\sigma^2_{\mathrm{SIS}}] \times \{1 + o(1)\}.$$

Thus, it suffices to examine $\mathbf{E}[\sigma^2_{\mathrm{SIS}}]$ with σ^2_{SIS} as in (3.4). We obtain, analogous to (A.2)

$$\hat{q}_M(\mathbf{x}) - q_M(\mathbf{x}) = \frac{\hat{f}_M(\mathbf{x}) - \overline{\omega}_M q_M(\mathbf{x})}{\alpha \tilde{I}_{\varphi_M}} + \widetilde{U}_M^1(\mathbf{x}) + \widetilde{U}_M^2(\mathbf{x}).$$

Slightly modified versions of lemmas A.2 and A.3 imply that the remainder term $\widetilde{U}_M^1(\mathbf{x}) + \widetilde{U}_M^2(\mathbf{x})$ is of lower order (compare the proof of Theorem 3.1). The crucial step for proving these modified versions of lemmas A.2 and A.3 is to show that under the assumptions 1 and 2

$$\mathbf{E}[\alpha\tilde{I}_{\varphi_M} - \overline{\omega}_M]^{2l} \le CM^{-l}.$$

This is shown now. We have

$$\begin{aligned} |\alpha\tilde{I}_{\varphi_M} - \overline{\omega}_M| \ &\le\ \left| \alpha\tilde{I}_{\varphi_M} - \frac{1}{M}\sum_{j=1}^{M} |\varphi_M(\tilde{\mathbf{x}}^j) - I_{\varphi_M}| \tilde{p}(\tilde{\mathbf{x}}^j) q_0(\tilde{\mathbf{x}}^j)^{-1} \right| \\ &\quad + \frac{1}{M}\sum_{j=1}^{M} \tilde{p}(\tilde{\mathbf{x}}^j) q_0(\tilde{\mathbf{x}}^j)^{-1} |I_{\varphi_M} - \breve{I}_{\varphi_M}|, \end{aligned}$$

and by applying the Minkowski inequality we obtain

$$\begin{aligned} \left(\mathbf{E}[\alpha\tilde{I}_{\varphi_M} - \overline{\omega}_M]^{2l}\right)^{\frac{1}{2l}} \ &\le\ \left(\mathbf{E}\left[\alpha\tilde{I}_{\varphi_M} - \frac{1}{M}\sum_{j=1}^{M} |\varphi_M(\tilde{\mathbf{x}}^j) - I_{\varphi_M}| \tilde{p}(\tilde{\mathbf{x}}^j) q_0(\tilde{\mathbf{x}}^j)^{-1} \right]^{2l} \right)^{\frac{1}{2l}} \\ &\quad + C\left(\mathbf{E}[I_{\varphi_M} - \breve{I}_{\varphi_M}]^{2l}\right)^{\frac{1}{2l}} \\ &=\ C\left(M^{-1/2} + M^{-1/2}\right). \end{aligned}$$

Hence, we conclude that the remainder term is of lower order. Finally, we need to show that

$$\int \mathbf{E}\left[\left(\frac{\hat{f}_M(\mathbf{x}) - \overline{\omega}_M q_M(\mathbf{x})}{\alpha\tilde{I}_{\varphi_M}} \right)^2 \hat{q}_M(\mathbf{x})^{-1} \right] d\mathbf{x} \approx h^4 H_1 + \frac{2^d}{3^d M h^d} H_2.$$

The main difference to Theorem 3.1 is the dependency of the weights $\widetilde{\omega}_M^j$. Define

$$\breve{\omega}_M^j = |\varphi_M(\tilde{\mathbf{x}}^j) - I_{\varphi_M}| \tilde{p}(\tilde{\mathbf{x}}^j) q_0(\tilde{\mathbf{x}}^j)^{-1},$$

$j = 1, \ldots, M$. As in the proof of Theorem 3.1, let $\breve{f}_{0/1}^{\mathrm{UH}}$ and $\hat{f}_{0/1}^{\mathrm{UH}}$ be unnormalized histogram bins based on the weights $\breve{\omega}_M^j$ and $\widetilde{\omega}_M^j$, respectively. It is not hard to show that

$$\mathbf{E}[\hat{f}_{0/1}^{\mathrm{UH}}(\alpha\tilde{I}_{\varphi_M})^{-1}] = \mathbf{E}[\breve{f}_{0/1}^{\mathrm{UH}}(\alpha\tilde{I}_{\varphi_M})^{-1}] + \mathcal{O}(M^{-1/2})$$

and

$$\mathrm{Var}[\hat{f}_{0/1}^{\mathrm{UH}}(\alpha \tilde{I}_{\varphi_M})^{-1}] = \mathrm{Var}[\check{f}_{0/1}^{\mathrm{UH}}(\alpha \tilde{I}_{\varphi_M})^{-1}] + \mathcal{O}(M^{-1}).$$

The rest of the proof follows analogous to Theorem 3.1, because the weights $\breve{\omega}_M^j$ are independent and the additional $\mathcal{O}(M^{-1/2})$, $\mathcal{O}(M^{-1})$ terms are negligible.

A.6 Derivation of the complexity of the LBFP

Let B_M be the number of bins. It follows that the number of bins in each marginal space is $\mathcal{O}(B_M^{1/d})$. We begin with the analysis of the evaluation of a LBFP. Given location $\mathbf{x}$, we need to find the associated bin midpoints $(t_{k_1}, \dots, t_{k_d})$, which is of order $\mathcal{O}(dB_M^{1/d})$. Then Equation (3.1) can be evaluated, which is $\mathcal{O}(2^d d)$. Now observe $B_M \approx V_M/h^d$ and $h^* = \mathcal{O}(\rho(d)^{1/(d+4)} N^{-1/(d+4)})$ with $\rho(d) = d(2/3)^d$. By assuming that $h = h^*$, we obtain

$$\mathcal{O}(dB_M^{1/d} + 2^d d) \approx \mathcal{O}(\rho(d)^{-1/(d+4)} d N^{1/(d+4)} + 2^d d),$$

neglecting the slowly increasing sequence V_M.

Sampling from a LBFP consists of the three steps described in Section 3.4.2. In Step 1, the marginalized histograms corresponding to the LBFP $\hat{f}(x_{1:i})$, $i = 1, \dots, d-1$, need to be calculated. This can be done recursively in $\mathcal{O}(B_M)$. In the second step, $\hat{F}$ is to be computed at all bin midpoints t_{k_i} using relation (3.5). Thus, it is required to evaluate $\hat{f}(x_{1:i-1}, t_{k_i})$, $i = 1, \dots, d$. This consists of searching the bin midpoints $(t_{k_1}, \dots, t_{k_{i-1}})$ associated with $x_{1:i-1}$ and evaluating equation (3.1) as we discussed earlier. It is sufficient to do the former once. Thus, we end up with $\mathcal{O}(dB_M^{1/d} + 2^d d \times dB_M^{1/d})$, where the latter $dB_M^{1/d}$ is the result of the evaluation of $\hat{F}$ at all t_{k_i} in each marginal dimension. Step 3 has complexity $\mathcal{O}(dB_M^{1/d})$, because in each marginal dimension the bin midpoint t_{k_i} satisfying $y_i \in [\hat{F}(t_{k_i}|x_{1:i-1}), \hat{F}(t_{k_i+1}|x_{1:i-1}))$ must be found. Putting it all together, we yield $\mathcal{O}(B_M + 2^d d^2 B_M^{1/d})$ for generating one sample. As seen earlier, we assume $h = h^*$, substitute $B_M \approx V_M/h^d$, and omit V_M to derive

$$\mathcal{O}(\rho(d)^{-d/(d+4)} N^{d/(d+4)} + 2^d d^2 \rho(d)^{-1/(d+4)} N^{1/(d+4)}).$$

Because Step 1 needs to be carried out only once and because $\rho(d)^{-1/(d+4)}$ is small compared with $2^d d^2$ we obtain approximately $\mathcal{O}(2^d d^2 N^{(d+5)/(d+4)})$ for generating N samples. Finally, we remark that N evaluations are negligible compared with generating N samples.

A.7 Proof of Theorem 4.1

Prerequisites for Theorem 4.1. The following quantities are not required in practical application. However, they are necessary for the proof of Theorem 4.1. Let A_M be an increasing sequence of compact sets defined by $A_M = \{\mathbf{x} \in \mathbb{R}^{|u|} : \check{q}_\varphi^{\mathrm{IS}}(\mathbf{x}) \geq c_M\}$, where $c_M > 0$ and $c_M \to 0$ as M goes to infinity. For any function g, we denote the restriction of g on A_M by g_M.

Furthermore, the volume of A_M is denoted by V_M. The NPIS estimator $\hat{I}^{\text{NPIS}}_{\varphi_M}$ is obtained by substituting $\hat{q}^{\text{IS}}_{\varphi}$ (in the algorithm) for

$$\hat{q}^{\text{IS}}_{M}(\mathbf{x}_u) = \begin{cases} \frac{\hat{f}_M(\mathbf{x}_u)+\delta_M}{\frac{1}{M}\sum_{j=1}^{M}\omega^j_M + V_M\delta_M} & \text{for} \quad \mathbf{x}_u \in A_M, \\ 0 & \text{else.} \end{cases}$$

Assumption 1 $\check{q}^{\text{IS}}_{\varphi}$ has three continuous and square integrable derivatives on its support and it is bounded. In addition, $\int (\nabla^2 \check{q}^{\text{IS}}_{\varphi})^4 (\check{q}^{\text{IS}}_{\varphi})^{-3} < \infty$ where $\nabla^2 \check{q}^{\text{IS}}_{\varphi} = \partial^2 \check{q}^{\text{IS}}_{\varphi}/\partial \mathbf{x}_1^2 + \ldots + \partial^2 \check{q}^{\text{IS}}_{\varphi}/\partial \mathbf{x}_d^2$.

Assumption 2 Trial distribution q_0 is chosen such that $\mathbf{E}[\check{q}^{\text{IS}}_{\varphi} q_0^{-1}]^4$ is finite on supp($\check{q}^{\text{IS}}_{\varphi}$).

Assumption 3 Sample sizes $M, N \to \infty$, bin width h satisfies $h \to 0$ and $Mh^{|u|} \to \infty$. Additionally, we have $\delta_M > 0$, $V_M\delta_M = o(h^2)$ and $M^3(V_M\delta_M)^4 \to \infty$.

Assumption 4 c_M is chosen such that $\frac{h^8+(Mh^{|u|})^{-2}}{\delta_M c_M^3} = o(\frac{h^4+(Mh^{|u|})^{-1}}{c_M})$ and $\frac{h^4+(Mh^{|u|})^{-1}}{c_M} \to 0$.

Assumption 5 The sequence c_M guaranties $(\int \check{q}^{\text{IS}}_{\varphi} \mathbf{1}_{\{\check{q}^{\text{IS}}_{\varphi} < c_M\}})^2 = o(M^{-1}h^4 + (M^2h^{|u|})^{-1})$.

Note that these assumptions are closely related to the assumptions 1 through 3, 4a, and 5a given in Chapter 3.

Proof. Conditional on the samples $\{\tilde{\mathbf{x}}^1, \tilde{\mathbf{x}}^2, \ldots, \tilde{\mathbf{x}}^M\}$, the variance of $\hat{I}^{\text{NPIS}}_{\varphi_M}$ can be written as

$$\begin{aligned} \frac{\sigma^2_{\text{IS}}}{N} &= \frac{I^2_{\varphi}}{N} \int \left\{ \frac{\varphi(\mathbf{x})p(\mathbf{x}_u)}{I_{\varphi}} - \hat{q}^{\text{IS}}_{M}(\mathbf{x}_u) \right\}^2 \frac{p(\mathbf{x}_{-u})}{\hat{q}^{\text{IS}}_{M}(\mathbf{x}_u)} d\mathbf{x} \\ &= \frac{I^2_{\varphi}}{N} \int \left[\frac{\nu(\mathbf{x})^2}{I^2_{\varphi}} + \left\{ \check{q}^{\text{IS}}_{\varphi}(\mathbf{x}_u) - \hat{q}^{\text{IS}}_{\varphi,M}(\mathbf{x}_u) \right\}^2 \right] \frac{p(\mathbf{x}_{-u})}{\hat{q}^{\text{IS}}_{M}(\mathbf{x}_u)} d\mathbf{x} \end{aligned}$$

with $\nu(\mathbf{x}) = \varphi(\mathbf{x})p(\mathbf{x}_u) - \int \varphi(\mathbf{x})p(\mathbf{x})d\mathbf{x}_{-u}$. The right term in brackets (quantifying the nonparametric estimation error) can be treated analogous to the proof of Theorem 3.1 in Section A.1. However, as a result of the integration with respect to $\mathbf{x}_{-u}$, a different variance term is obtained. The optimal bin width is derived through differentiation.

A.8 Proof of Proposition 5.1

Proof. Let the LBFP estimator $\hat{f}_N$ be defined as

$$\hat{f}_N(\mathbf{z}) = \begin{cases} \frac{\tilde{f}(\mathbf{z})+\delta_N}{\frac{1}{N}\sum_{i=1}^{N}\omega^i + V_N\delta_N} & \text{for } \mathbf{z} \in A_N, \\ 0 & \text{else.} \end{cases}$$

Sequence A_N is given by $A_N = \{\mathbf{z} \in \mathbb{R}^d : f(\mathbf{z}) \geq c_N\}$, where $c_N > 0$ and $c_N \to 0$ for $N \to \infty$. V_N is defined as the volume of A_N. We emphasize that the quantities A_N, V_N, c_N, and δ_N are only needed in the proof and can be skipped in practice. Under the assumptions 1 through 3 and provided that $\delta_N > 0$, $V_N\delta_N = o(h^2)$, $N^3(V_N\delta_N)^4 \to \infty$, and

$$\left(\int f(\mathbf{z})\mathbf{1}_{\{f(\mathbf{z})<c_N\}}d\mathbf{z}\right)^2 = o(N^{-1}h^4 + (N^2h^d)^{-1}),$$

it can be shown that

$$\int \mathbf{E}[\hat{f}_N(\mathbf{z}) - f(\mathbf{z})]^2 d\mathbf{z} = \int \{\mathbf{E}[\tilde{f}(\mathbf{z})] - f(\mathbf{z})\}^2 d\mathbf{z} + \int \mathrm{Var}[\tilde{f}(\mathbf{z})]d\mathbf{z} + \mathcal{O}(N^{-1})$$

along the same lines as in the proof of Theorem 3.1 in Section A.1. The following is only shown for $d = 1$. Without loss of generality we assume $\mathbf{z} \in [-h/2, h/2)$. Then, $\tilde{f}$ is given by

$$\tilde{f}(\mathbf{z}) = \left(\frac{h/2 - \mathbf{z}}{h}\right)\tilde{f}_0^{\mathrm{H}} + \left(\frac{h/2 + \mathbf{z}}{h}\right)\tilde{f}_1^{\mathrm{H}},$$

where $\tilde{f}_0^{\mathrm{H}} = 1/(Nh)\sum_{i=1}^N \omega^i \mathbf{1}_{[-h,0)}(\mathbf{z}^i)$ and $\tilde{f}_1^{\mathrm{H}} = 1/(Nh)\sum_{i=1}^N \omega^i \mathbf{1}_{[0,h)}(\mathbf{z}^i)$. Hence,

$$\mathrm{Var}[\tilde{f}(\mathbf{z})] = \left(\frac{h/2 - \mathbf{z}}{h}\right)\mathrm{Var}[\tilde{f}_0^{\mathrm{H}}] + \left(\frac{h/2 + \mathbf{z}}{h}\right)\mathrm{Var}[\tilde{f}_1^{\mathrm{H}}] + \frac{h^2/2 - 2\mathbf{z}^2}{h^2}\mathrm{Cov}[\tilde{f}_0^{\mathrm{H}}, \tilde{f}_1^{\mathrm{H}}]. \tag{A.12}$$

The variance of $\tilde{f}_0^{\mathrm{H}}$ is computed as

$$\begin{aligned}\mathrm{Var}[\tilde{f}_0^{\mathrm{H}}] &= \frac{1}{Nh^2}\left\{\int_{B_0}\int \left(\frac{g(\mathbf{z},\tilde{\mathbf{z}})}{g_0(\mathbf{z},\tilde{\mathbf{z}})}\right)^2 g_0(\mathbf{z},\tilde{\mathbf{z}})d\tilde{\mathbf{z}}d\mathbf{z} - \left(\int_{B_0}\int \frac{g(\mathbf{z},\tilde{\mathbf{z}})}{g_0(\mathbf{z},\tilde{\mathbf{z}})}g_0(\mathbf{z},\tilde{\mathbf{z}})d\tilde{\mathbf{z}}d\mathbf{z}\right)^2\right\} \\ &\approx \frac{1}{Nh}\int \frac{g(0,\tilde{\mathbf{z}})^2}{g_0(0,\tilde{\mathbf{z}})}d\tilde{\mathbf{z}} - \frac{f(0)^2}{N},\end{aligned}$$

where $B_0 = [-h, 0)$. An analogous approximation holds for $\mathrm{Var}[\tilde{f}_1^{\mathrm{H}}]$. For the covariance term we obtain

$$\begin{aligned}\mathrm{Cov}[\tilde{f}_0^{\mathrm{H}}, \tilde{f}_1^{\mathrm{H}}] &= -\frac{1}{Nh^2}\left(\int_{B_0}\int \frac{g(\mathbf{z},\tilde{\mathbf{z}})}{g_0(\mathbf{z},\tilde{\mathbf{z}})}g_0(\mathbf{z},\tilde{\mathbf{z}})d\tilde{\mathbf{z}}d\mathbf{z}\right)\left(\int_{B_1}\int \frac{g(\mathbf{z},\tilde{\mathbf{z}})}{g_0(\mathbf{z},\tilde{\mathbf{z}})}g_0(\mathbf{z},\tilde{\mathbf{z}})d\tilde{\mathbf{z}}d\mathbf{z}\right) \\ &\approx -\frac{f(0)^2}{N}\end{aligned}$$

with $B_1 = [0, h)$. Plugging this into (A.12) yields

$$\mathrm{Var}[\tilde{f}(\mathbf{z})] = \left(\frac{1}{2Nh} + \frac{2\mathbf{z}^2}{Nh^3}\right)\int \frac{g(0,\tilde{\mathbf{z}})^2}{g_0(0,\tilde{\mathbf{z}})}d\tilde{\mathbf{z}} + \mathcal{O}(N^{-1}),$$

and integration over $[-h/2, h/2)$ gives

$$\int_{-h/2}^{h/2} \mathrm{Var}[\tilde{f}(\mathbf{z})]d\mathbf{z} = \frac{2}{3N}\int \frac{g(0,\tilde{\mathbf{z}})^2}{g_0(0,\tilde{\mathbf{z}})}d\tilde{\mathbf{z}} + \mathcal{O}(h/N).$$

Finally, by summing over all bins and applying the standard Riemann approximation one obtains

$$\int \mathrm{Var}[\tilde{f}(\mathbf{z})]d\mathbf{z} = \frac{2}{3Nh}\int\int \frac{g(0,\tilde{\mathbf{z}})^2}{g_0(0,\tilde{\mathbf{z}})}d\tilde{\mathbf{z}} + \mathcal{O}(N^{-1}).$$

Similar computations in the multivariate case establish the result given in the proposition. The proof is finished by noting that H_1 is the standard integrated squared bias of LBFPs.

A.9 Proof of Proposition 6.1

Proof. The likelihood $p(\mathbf{y}_{t_j}|\mathbf{y}_{t_{1:j-1}}, \mathbf{x}_{t_j})$ is equal to one if $x_{t_j,s} \in \log A_{t_j,s}$ for all $s = 1, \dots, S$ and zero otherwise, that is

$$p(\mathbf{y}_{t_j}|\mathbf{y}_{t_{1:j-1}}, \mathbf{x}_{t_j}) = \prod_{s=1}^{S} \mathbf{1}_{\{x_{t_j,s} \,\in\, \log A_{t_j,s}\}}.$$

This and (6.6) recursively imply the uniqueness of the conditional distribution $p(\mathbf{x}_{t_{1:j}}|\mathbf{y}_{t_{1:j}})$. (Note that $p(\mathbf{y}_{t_j}|\mathbf{y}_{t_{1:j-1}})$ does not depend on $\mathbf{x}_{t_{1:j}}$ and is therefore part of the norming constant.) It is easy to verify that the optimal proposal satisfies

$$p(\mathbf{x}_{t_j}|\mathbf{y}_{t_{1:j}}, \mathbf{x}_{t_{j-1}}) \propto p(\mathbf{y}_{t_j}|\mathbf{y}_{t_{1:j-1}}, \mathbf{x}_{t_j})\, p(\mathbf{x}_{t_j}|\mathbf{x}_{t_{j-1}}).$$

Furthermore, the transition prior is given by $p(\mathbf{x}_{t_j}|\mathbf{x}_{t_{j-1}}) = \mathcal{N}(\mathbf{x}_{t_j}|\mathbf{x}_{t_{j-1}}; \Sigma_{t_j})$ leading to the assertion. The expression for the importance weights follows from

$$p(\mathbf{y}_{t_j}|\mathbf{y}_{t_{1:j-1}}, \mathbf{x}^i_{t_{j-1}}) = \int p(\mathbf{y}_{t_j}|\mathbf{y}_{t_{1:j-1}}, \mathbf{x}_{t_j})\, p(\mathbf{x}_{t_j}|\mathbf{x}^i_{t_{j-1}})\, d\mathbf{x}_{t_j} = \int_{\log \mathbf{A}_{t_j}} p(\mathbf{x}_{t_j}|\mathbf{x}^i_{t_{j-1}})\, d\mathbf{x}_{t_j}.$$

A.10 Calculation of the Quasi Mean Squared Error in Section 6.5.2

We now calculate and minimize the mean squared error of

$$\tilde{\Sigma}_{t_i|t_j}(\lambda) := \big(1+\kappa\big)\,\hat{\Sigma}_{t_j} - \kappa\,\hat{\Sigma}^{(1/2)}_{t_j}$$

as an estimator of $\Sigma(t_i)$ with respect to κ. For several reasons the variance of the estimator is very hard to derive (because of the recursive estimation scheme and the nonlinear microstructure noise model). In order not to overstress heuristic considerations we minimize instead the mean squared error of the above estimator in the case where the unknown efficient prices are used instead of the filter particles and call this the quasi mean squared error.

We only give a brief sketch. As in Section 6.5 we only discuss the univariate case. We obtain as in (6.31)

$$\mathbf{E}\,\tilde{\Sigma}_{t_i|t_j}(\lambda) \approx \Sigma(t_i) - \big[(1+\kappa)\big(i-\bar{j}\big) - \kappa\,\big(i-\bar{j}^{\,(1/2)}\big)\big]\,\dot{\Sigma}(\bar{j}) \tag{A.13}$$

and for the variance

$$\begin{aligned} \operatorname{Var}\big(\hat{\Sigma}_{t_j}\big) &\approx \sum_{k=0}^{j-3}\Big[\prod_{\ell=0}^{k-1}\big(1-\lambda_{j-\ell}\big)^2\Big]\lambda_{j-k}^2\, 2\,\Sigma(t_{j-k})^2 + \Big[\prod_{\ell=0}^{j-3}\big(1-\lambda_{j-\ell}\big)^2\Big]\, 2\,\Sigma(t_2)^2 \\ &\approx \left[\sum_{k=0}^{j-3}\Big[\prod_{\ell=0}^{k-1}\big(1-\lambda_{j-\ell}\big)^2\Big]\lambda_{j-k}^2 + \Big[\prod_{\ell=0}^{j-3}\big(1-\lambda_{j-\ell}\big)^2\Big]\right] 2\,\Sigma(t_{\bar{j}})^2. \end{aligned} \tag{A.14}$$

Similarly we obtain

$$\operatorname{Var}\big(\hat{\Sigma}^{(1/2)}_{t_j}\big) \approx \left[\sum_{k=0}^{j-3}\Big[\prod_{\ell=0}^{k-1}\big(1-\frac{\lambda_{j-\ell}}{2}\big)^2\Big]\frac{\lambda_{j-k}^2}{4} + \Big[\prod_{\ell=0}^{j-3}\big(1-\frac{\lambda_{j-\ell}}{2}\big)^2\Big]\right] 2\,\Sigma(t_{\bar{j}})^2$$

and

$$\operatorname{Cov}\big(\hat\Sigma_{t_j},\hat\Sigma^{(1/2)}_{t_j}\big)\approx\left[\sum_{k=0}^{j-3}\Big[\prod_{\ell=0}^{k-1}(1-\lambda_{j-\ell})\big(1-\frac{\lambda_{j-\ell}}{2}\big)\Big]\frac{\lambda^2_{j-k}}{2}+\Big[\prod_{\ell=0}^{j-3}(1-\lambda_{j-\ell})\big(1-\frac{\lambda_{j-\ell}}{2}\big)\Big]\right]2\,\Sigma(t_{\bar j})^2.$$

The terms in the brackets can be calculated by the recursions (6.38) through (6.40). Therefore

$$\begin{aligned}\operatorname{Var}\big(\tilde\Sigma_{t_i|t_j}(\lambda)\big)&\approx\Big[(1+\kappa)^2\,v_{1,j}+\kappa^2 v_{2,j}-2(1+\kappa)\,\kappa\,v_{3,j}\Big]\,2\,\Sigma(t_{\bar j})^2\\&=\Big[v_{1,j}+\kappa\,(2v_{1,j}-2v_{3,j})+\kappa^2\,(v_{1,j}+v_{2,j}-2v_{3,j})\Big]\,2\,\Sigma(t_{\bar j})^2\end{aligned}$$

leading to the mean squared error

$$\begin{aligned}\mathbf{E}\Big(\tilde\Sigma_{t_i|t_j}(\lambda)-\Sigma(t_i)\Big)^2\approx{}&\Big[-(1+\kappa)\big(i-\bar j\big)+\kappa\,\big(i-\bar j^{\,(1/2)}\big)\Big]^2\,\dot\Sigma\big(\bar j\big)^2\\&+\Big[v_{1,j}+\kappa\,(2v_{1,j}-2v_{3,j})+\kappa^2\,(v_{1,j}+v_{2,j}-2v_{3,j})\Big]\,2\,\Sigma(t_{\bar j})^2\,.\end{aligned}$$

Minimization with respect to κ yields with $\dot\Sigma(j)=\Sigma'\big(t_j\big)\,\tau'(j)$

$$\begin{aligned}\kappa_{\min}&=\frac{\big(i-\bar j\big)\big(\bar j-\bar j^{\,(1/2)}\big)\,\dot\Sigma\big(\bar j\big)^2-2\,(v_{1,j}-v_{3,j})\,\Sigma(t_{\bar j})^2}{\big(\bar j-\bar j^{\,(1/2)}\big)^2\,\dot\Sigma\big(\bar j\big)^2+2\,(v_{1,j}+v_{2,j}-2v_{3,j})\,\Sigma(t_{\bar j})^2}\\&=\frac{\big(i-\bar j\big)\big(\bar j-\bar j^{\,(1/2)}\big)\,\big[\frac{\partial}{\partial t}\log\Sigma(t)_{|t=t_{\bar j}}\,\tau'\big(\bar j\big)\big]^2-2\,(v_{1,j}-v_{3,j})}{\big(\bar j-\bar j^{\,(1/2)}\big)^2\,\big[\frac{\partial}{\partial t}\log\Sigma(t)_{|t=t_{\bar j}}\,\tau'\big(\bar j\big)\big]^2+2\,(v_{1,j}+v_{2,j}-2v_{3,j})}\,.\end{aligned}$$

A.11 Reversed Order Initialization

Proof of (6.65). With the definitions given in Section 6.9 we have with

$$\kappa_{2|2}=\frac{2-\bar 2}{\bar 2-\bar 2^{\,(1/2)}}=\frac{\bar 1^{\text{rev}}-1}{\bar 1^{\text{rev}(1/2)}-\bar 1^{\text{rev}}}=\kappa^{\text{rev}}_{1|1}$$

$$\begin{aligned}\tilde\Sigma_{t_2|t_2}&:=\big(1+\kappa_{2|2}\big)\,\hat\Sigma_{t_2}-\kappa_{2|2}\,\hat\Sigma^{(1/2)}_{t_2}\\&=\Big(1+\frac{\bar 1^{\text{rev}}-1}{\bar 1^{\text{rev}(1/2)}-\bar 1^{\text{rev}}}\Big)\,\Big[\Big(1+\frac{2\times\bar 1^{\text{rev}}-2}{\bar 1^{\text{rev}(1/2)}-\bar 1^{\text{rev}}}\Big)\,\hat\Sigma^{\text{rev}}_{t_1}-\frac{2\times\bar 1^{\text{rev}}-2}{\bar 1^{\text{rev}(1/2)}-\bar 1^{\text{rev}}}\,\hat\Sigma^{\text{rev}(1/2)}_{t_1}\Big]\\&\qquad-\frac{\bar 1^{\text{rev}}-1}{\bar 1^{\text{rev}(1/2)}-\bar 1^{\text{rev}}}\,\Big[\Big(1+\frac{\bar 1^{\text{rev}}+\bar 1^{\text{rev}(1/2)}-2}{\bar 1^{\text{rev}(1/2)}-\bar 1^{\text{rev}}}\Big)\,\hat\Sigma^{\text{rev}}_{t_1}-\frac{\bar 1^{\text{rev}}+\bar 1^{\text{rev}(1/2)}-2}{\bar 1^{\text{rev}(1/2)}-\bar 1^{\text{rev}}}\,\hat\Sigma^{\text{rev}(1/2)}_{t_1}\Big]\\&=\Big(\frac{\bar 1^{\text{rev}(1/2)}-1}{\bar 1^{\text{rev}(1/2)}-\bar 1^{\text{rev}}}\Big)\,\Big[\Big(\frac{\bar 1^{\text{rev}}+\bar 1^{\text{rev}(1/2)}-2}{\bar 1^{\text{rev}(1/2)}-\bar 1^{\text{rev}}}\Big)\,\hat\Sigma^{\text{rev}}_{t_1}-\frac{2\times\bar 1^{\text{rev}}-2}{\bar 1^{\text{rev}(1/2)}-\bar 1^{\text{rev}}}\,\hat\Sigma^{\text{rev}(1/2)}_{t_1}\Big]\\&\qquad-\frac{\bar 1^{\text{rev}}-1}{\bar 1^{\text{rev}(1/2)}-\bar 1^{\text{rev}}}\,\Big[\Big(\frac{2\times\bar 1^{\text{rev}(1/2)}-2}{\bar 1^{\text{rev}(1/2)}-\bar 1^{\text{rev}}}\Big)\,\hat\Sigma^{\text{rev}}_{t_1}-\frac{\bar 1^{\text{rev}}+\bar 1^{\text{rev}(1/2)}-2}{\bar 1^{\text{rev}(1/2)}-\bar 1^{\text{rev}}}\,\hat\Sigma^{\text{rev}(1/2)}_{t_1}\Big]\\&=\Big(1+\frac{\bar 1^{\text{rev}}-1}{\bar 1^{\text{rev}(1/2)}-\bar 1^{\text{rev}}}\Big)\,\hat\Sigma^{\text{rev}}_{t_1}-\frac{\bar 1^{\text{rev}}-1}{\bar 1^{\text{rev}(1/2)}-\bar 1^{\text{rev}}}\,\hat\Sigma^{\text{rev}(1/2)}_{t_1}=\tilde\Sigma^{\text{rev}}_{t_1|t_1}.\end{aligned}$$

A.12 Proof of Proposition 7.1

Proof. Under the assumption that g is 2π periodic, it can be seen from (7.7) that

$$\sum_{t=1}^{T} \int\int\int \{y_t - a_t g(\phi_t \mathrm{mod} 2\pi) - b_t\}^2 p_{g^{(m)}}(a_t, b_t, \phi_t | y_{1:T}) da_t db_t d\phi_t$$

needs to be minimized with respect to $g(\phi)$ where $\phi \in [0, 2\pi)$. That is, for fixed $\phi \in [0, 2\pi)$, we need to minimize the quantity

$$\sum_{t=1}^{T} \sum_{\{\phi_t : \phi_t \mathrm{mod} 2\pi = \phi\}} \int\int \{y_t - a_t g(\phi) - b_t\}^2 p_{g^{(m)}}(a_t, b_t, \phi_t | y_{1:T}) da_t db_t.$$

with respect to $g(\phi)$. The minimization leads to

$$\begin{aligned} & g^{(m+1)}(\phi) \\ = & \left[\sum_{t=1}^{T} \sum_{\mathcal{H}_\phi} p_{g^{(m)}}(\phi_t | y_{1:T}) \left\{ y_t \int a_t p_{g^{(m)}}(a_t | \phi_t, y_{1:T}) da_t - \int a_t b_t p_{g^{(m)}}(a_t, b_t | \phi_t, y_{1:T}) da_t db_t \right\} \right] \\ & \times \left[\sum_{t=1}^{T} \sum_{\mathcal{H}_\phi} p_{g^{(m)}}(\phi_t | y_{1:T}) \int a_t^2 p_{g^{(m)}}(a_t | \phi_t, y_{1:T}) da_t \right]^{-1}, \end{aligned} \tag{A.15}$$

where $\mathcal{H}_\phi = \{\phi_t : \phi_t \mathrm{mod} 2\pi = \phi\}$. Based on the output of the RBPS and the approximation of the density $p_{g^{(m)}}(\phi_t | y_{1:T})$ given in (7.8) one yields

$$p_{g^{(m)}}(\phi_t | y_{1:T}) \int a_t b_t p_{g^{(m)}}(a_t, b_t | \phi_t, y_{1:T}) da_t db_t \approx \frac{1}{h_t} \sum_{i=1}^{N} \tilde{\omega}_t^i K\{(\phi_t - \phi_t^i)/h_t\} (\tilde{S}_t^i)_{12}$$

and analogous approximations for the other terms in (A.15). This gives the estimator

$$\hat{g}^{(m+1)}(\phi) = \frac{\sum_{t=1}^{T} \sum_{\mathcal{H}_\phi} \sum_{i=1}^{N} \tilde{\omega}_t^i K\{(\phi_t - \phi_t^i)/h_t\} \{y_t \tilde{a}_t^i - (\tilde{S}_t^i)_{12}\}}{\sum_{t=1}^{T} \sum_{\mathcal{H}_\phi} \sum_{i=1}^{N} \tilde{\omega}_t^i K\{(\phi_t - \phi_t^i)/h_t\} (\tilde{S}_t^i)_{11}}.$$

The estimator given in the proposition is obtained under the assumptions on the kernel and the bandwidths.

A.13 Proof of Proposition 7.2

Proof. By applying Jensen's inequality one yields

$$\begin{aligned}
\log \frac{p_{g^{(m+1)}}(y_{1:T})}{p_{g^{(m)}}(y_{1:T})} &= \log \mathbf{E}_{g^{(m)}} \left[\frac{p_{g^{(m+1)}}(\mathbf{X}_{0:T}, y_{1:T})}{p_{g^{(m)}}(\mathbf{X}_{0:T}, y_{1:T})} \middle| y_{1:T} \right] \\
&\geq \mathbf{E}_{g^{(m)}} \left[\sum_{t=1}^{T} \log \frac{p_{g^{(m+1)}}(y_t|\mathbf{X}_t)}{p_{g^{(m)}}(y_t|\mathbf{X}_t,)} \middle| y_{1:T} \right] \\
&\propto \sum_{t=1}^{T} \mathbf{E}_{g^{(m)}}[\{Y_t - A_t g^{(m)}(\phi_t) - B_t\}^2 | y_{1:T}] \\
&\quad - \sum_{t=1}^{T} \mathbf{E}_{g^{(m)}}[\{Y_t - A_t g^{(m+1)}(\phi_t) - B_t\}^2 | y_{1:T}].
\end{aligned}$$

Because $g^{(m+1)}$ given in (A.15) maximizes (7.7) this establishes $p_{g^{(m+1)}}(y_{1:T}) \geq p_{g^{(m)}}(y_{1:T})$.

References

Adler, R. J. (1990), “An Introduction to Continuity, Extrema, and Related Topics for General Gaussian Processes,” Hayward, California: Institute of Mathematical Statistics.

Aguilar, O. and West, M. (2000), “Bayesian Dynamic Factor Models and Portfolio Allocation,” in *Journal of Business and Economic Statistics*, 18, 338-357.

Aït-Sahalia, Y., Mykland, P.A., and Zhang, L. (2005), “How Often to Sample a Continuous-Time Process in the Presence of Market Microstructure Noise,” in *Review of Financial Studies*, 18, 351-416.

Andersen, T.G., Bollerslev, T., and Meddahi, N. (2006), “Realized Volatility Forecasting and Market Microstructure Noise,” unpublished manuscript.

Ané, T., and Geman, H. (2000), “Order Flow, Transaction Clock, and Normality of Asset Returns,” in *The Journal of Finance*, 55, 2259-2284.

Arulampalam, M.S., Maskell, S., Gordon, N., and Clapp, T. (2002), “A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking,” in *IEEE Transactions on Signal Processing*, 50, 174-188.

Asai, M., McAleer, M., and Yu, J. (2006), “Multivariate Stochastic Volatility: A Review,” in *Econometric Reviews*, 25, 145-175.

Asmussen, S. (2003), *Applied Probability and Queues*, New York: Springer.

Avramidis, A. N. (2002), “Importance Sampling for Multimodal Functions and Application to Pricing Exotic Options,” in *Winter Simulation Conference Proceedings*, 1493-1501.

Ball, C.A. (1988), “Estimation Bias Induced by Discrete Security Prices,” in *The Journal of Finance*, 43, 841-865.

Bandi, F.M., and Russell, J.R. (2006), “Seperating microstructure noise from volatility,” in *Journal of Financial Economics*, 79, 655-692.

— (2008), “Microstructure noise, realized variance, and optimal sampling,” in *Review of Economic Studies*, 75, 339-369.

Barndorff-Nielsen, O.E., Hansen, P.R., Lunde, A., and Shephard, N. (2008a), "Designing Realized Kernels to Measure the Ex-Post Variation of Equity Prices in the Presence of Noise," in *Econometrica*, 76, 1481-1536.

— (2008b), "Multivariate realised kernels: consistent positive semi-definite estimators of the covariation of equity prices with noise and non-synchronous trading," unpublished manuscript.

— (2009), "Realized kernels in practice: trades and quotes," in *Econometrics Journal*, 12, C1-C32.

Blasius, B., Huppert, A., and Stone, L. (1999), "Complex dynamics and phase synchronization in spatially extended ecological systems," in *Nature*, 399, 354-359.

Bos, C.S., Janus, P., and Koopman, S.J. (2009), "Spot Variance Path Estimation and its Application to High Frequency Jump Testing," Discussion Paper TI 2009-110/4, Tinbergen Institute.

Box, G.E.P., and Muller, M.E. (1958), "A note on the generation of random normal deviates," in *Annals of Mathematical Statistics*, 29, 610-611.

Briers, M, Doucet, A., and Maskell, S. (2010), "Smoothing algorithms for state-space models," in *Annals of the Institute of Statistical Mathematics*, 62, 61-89.

Caflisch, R. E., Morokoff, W. J., and Owen, A. B. (1997), "Valuation of mortgage backed securities using Brownian bridges to reduce effective dimension," in *Journal of Computational Finance*, 1, 27-46.

Cappé, O., and Moulines, E. (2009), "On-line expectation-maximization algorithm for latent data models," in *Journal of the Royal Statistical Society, Series B*, 71, 593-613.

Capriotti, L. (2008), "Least Squares Importance Sampling for Monte Carlo Security Pricing," in *Quantitative Finance*, 8, 485-497.

Clifford, G.D., Azuaje, F., and McSharry, P.E. (2006), *Advanced Methods and Tools for ECG Data Analysis*, Norwood: Artech House.

Chopin, N. (2004), "Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference," in *Annals of Statistics*, 32, 2385-2411.

Cox, J., Ingersoll, J. E., and Ross, S. A. (1985), "A Theory of the Term Structure of Interest Rates," in *Econometrica*, 53, 385-407.

Crisan, D. (2001) "Particle filters – A theoretical perspective," in *Sequential Monte Carlo Methods in Practice*, eds. A. Doucet, N. de Freitas, and N. Gordon, New York: Springer, pp. 17-41.

Crisan, D., and Doucet, A. (2002), "A survey of convergence results on particle filtering methods for practitioners," in *IEEE Transactions on Signal Processing*, 50, 736-746.

Christensen, K., Podolskij, M., and Vetter, M. (2009), "Bias-correcting the realised range-based variance in the presence of market microstructure noise," in *Finance and Stochastics*, 13, 239-268.

Dahlhaus, R., and Neddermeyer, J.C. (2009), "Bayesian Phase Estimation for Noisy Quasi-Periodic Time Series," unpublished manuscript.

— (2010a), "Particle Filter-Based On-Line Estimation of Spot Volatility with Nonlinear Market Microstructure Noise Models," unpublished manuscript.

— (2010b), "On-Line Estimation of Spot Cross-Volatility with a State-Space Model for Non-Synchronous Tick-by-Tick Data," unpublished manuscript.

Dahlhaus, R., and Subba Rao, S. (2007), "A recursive online algorithm for the estimation of time-varying ARCH parameters," in *Bernoulli*, 13, 389-422.

de Freitas, N. (2001), "Rao-Blackwellised Particle Filtering for Fault Diagnosis," in *IEEE Aerospace Conference.*

de Freitas, N., Andrieu, C., Højen-Sørensen, P., Niranjan, M., and Gee, A. (2001) "Sequential Monte Carlo Methods for Neural Networks," in *Sequential Monte Carlo Methods in Practice*, eds. A. Doucet, N. de Freitas, and N. Gordon, New York: Springer, pp. 359-379.

Del Moral, P., and Guionnet, A. (1999), " Central limit theorem for nonlinear filtering and interacting particle systems," in *Annals of Applied Probability*, 9, 275-297.

Dempster, A.P., Laird, N.M., and Rubin, D.B. (1977), "Maximum Likelihood from Incomplete Data via the EM Algorithm," in *Journal of the Royal Statistical Society, Series B*, 39, 1-38.

Devroye, L. (1986), *Non-Uniform Random Variate Generation*, New York: Springer.

Douc, R., Cappé, O., and Moulines, E. (2005), "Comparison of resampling schemes for particle filtering," in: *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis*, 64-69.

Doucet, A., de Freitas, N., and Gordon, N. (ed.) (2001), *Sequential Monte Carlo Methods in Practice*, New York: Springer.

Doucet, A., Godsill, S., and Andrieu, C. (2000), "On sequential Monte Carlo sampling methods for Bayesian filtering," in *Statistics and Computing*, 10, 197-208.

Doucet, A., Gordon, N.J., and Krishnamurthy, V. (1999), "Particle Filters for State Estimation of Jump Markov Linear Systems," in *IEEE Transactions on Signal Processing*, 49, 613-624.

Engle, R.F., and Russell, J.R. (1998), "Autoregressive conditional duration: A new model for irregularly spaced transaction data," in *Econometrica*, 66, 1127-1162.

Even-Dar, E., and Mansour, Y. (2003), "Learning Rates for Q-learning," in *Journal of Machine Learning Research*, 5, 1-25.

Fan, J., and Wang, Y. (2008), "Spot volatility estimation for high-frequency data," in *Statistics and Its Interface*, 1, 279-288.

Fearnhead, P. (2005), "Using Random Quasi-Monte-Carlo Within Particle Filters, With Application to Financial Time Series," in *Journal of Computational and Graphical Statistics*, 14, 751-769.

Fearnhead, P., Wyncoll, D., and Tawn, J. (2008), "A Sequential Smoothing Algorithm with Linear Computational Cost," unpublished manuscript.

Foster, D., and Nelson, D. (1996), "Continuous Record Asymptotics for Rolling Sample Estimators," in *Econometrica*, 64, 139-174.

Gabaix, X., Gopikrishnan, P., Plerou, V., and Stanley, H.E. (2003), "A theory of power-law distributions in financial market fluctuations," in *Nature*, 423, 267-270.

Geweke, J. (1989), "Bayesian Inference in Econometric Models using Monte Carlo Integration," in *Econometrica*, 57, 1317-1339.

Gilks, W.R. and Berzuini, C. (2001), "Following a moving target – Monte Carlo inference for dynamic Bayesian models," in *Journal of the Royal Statistical Society, Series B*, 63, 127-146.

Givens, G. H., and Raftery, A. E. (1996), "Local Adaptive Importance Sampling for Multivariate Densities With Strong Nonlinear Relationships," in *Journal of American Statistical Association*, 91, 132-141.

Glasserman, P., and Kou, S.-G. (1995), "Analysis of an Importance Sampling Estimator for Tandem Queues," in *ACM Transactions on Modeling and Computer Simulation*, 5, 22-42.

Glasserman, P., Heidelberger, P., and Shahabuddin, P. (1999), "Asymptotically optimal importance sampling and stratification for pricing path-dependent options," in *Mathematical Finance*, 9, 117-152.

Glasserman, P. (2004), *Monte Carlo Methods in Financial Engineering*, New York: Springer.

Glynn, P. W., and Iglehart, D. L. (1989), "Importance Sampling for Stochastic Simulations," in *Management Science*, 35, 1367-1392.

Godsill, S.J., Doucet, A., and West, M. (2004), "Monte Carlo Smoothing for Nonlinear Time Series," in *Journal of American Statistical Association*, 99, 156-168.

Gordon, N, Salmond, D., and Smith, A. (1993), "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," in *IEE Proceedings-F*, 140, 107-113.

Grossmann, A. , Kronland-Martinet, R., and Morlet, J. (1989), "Reading and Understanding Continuous Wavelet Transforms," in *Wavelets, Time-Frequency Methods and Phase Space*, eds. J.M. Combes, Berlin: Springer.

Guo, D. and Wang, X. (2006), "Quasi-Monte Carlo Filtering in Nonlinear Dynamic Systems," in *IEEE Transactions on Signal Processing*, 54, 2087-2098.

Hannan, E.J. (1973), "The Estimation of Frequency," in *Journal of Applied Probability*, 10, 513-519.

Hansen, P.R., and Lunde, A. (2006), "Realized Variance and Market Microstructure Noise," in *Journal of Business and Economics Statistics*, 24, 127-161.

Hayashi, T., and Yoshida, N. (2005), "On covariance estimation of nonsynchronously observed diffusion processes," in *Bernoulli*, 11, 359-379.

Howison, S., and Lamper, D. (2001), "Trading volume in models of financial derivatives," *Applied Mathematical Finance*, 8, 119-135.

Hürzeler, M., and Künsch, H.R. (1998), "Monte Carlo Approximation for General State-Space Models," in *Journal of Computational and Graphical Statistics*, 7, 175-193.

— (2001), "Approximating and Maximising the Likelihood for a General State-Space Model," in *Sequential Monte Carlo Methods in Practice*, eds. A. Doucet, N. de Freitas, and N. Gordon, New York: Springer, pp. 159-175.

Jäckel, P. (2002), *Monte Carlo methods in finance*, West Sussex: Wiley.

Jacquier, E., Polson, N.G., and Rossi, P.E. (1994), "Bayesian Analysis of Stochastic Volatility Models" (with discussion), in *Journal of Business and Economic Statistics*, 12, 371-417.

Julier, S., and Uhlmann, J.K. (1997), "A New Extension of the Kalman Filter to Nonlinear Systems," in *Proc. of Areosense: 11th Int. Sympos. Aerospace/Defense Sensing, Simulation, and Controls*, Orlando, FL.

Kahn, H. (1950), "Random Sampling (Monte Carlo) Techniques in Neutron Attenuation Problems – II.," in *Nucleonics*, Vol. 6, 60-65.

Kahn, H., and Marshall, A.W. (1953), "Methods of Reducing Sample Size in Monte Carlo Computations," in *Journal of the Operations Research Society of America*, Vol. 1, 263-278.

Kalnina, I., and Linton, O. (2008), "Estimating quadratic variation consistently in the presence of endogenous and diurnal measurement error," in *Journal of Econometrics*, 147, 47-59.

Kalman, R.E. (1960), "A New Approach to Linear Filtering and Prediction Problems," in *Transactions of the American Society of Mechanical Engineers, Journal of Basic Engineering*, 82, 35-45.

Kim, Y. B., Roh, D. S., and Lee, M. Y. (2000), "Nonparametric Adaptive Importance Sampling For Rare Event Simulation," in *Winter Simulation Conference Proceedings*, Vol. 1, 767-772.

Kitagawa, G. (1987), "Non-Gaussian state-space modelling of non-stationary time series," in *Journal of the American Statistical Association*, 82, 1032-1041.

— (1996), "Monte Carlo filter and smoother for non-Gaussian nonlinear state space models," in *Journal of Computational and Graphical Statistics*, 5, 1-25.

Kollman, C., Baggerly, K., Cox, D., and Picard, R. (1999), "Bayesian Inference in Econometric Models using Monte Carlo Integration," in *Annals of Applied Probability*, 9, 391-412.

Kong, A., Liu, J., and Wong, W. (1994), "Sequential imputation and Bayesian missing data problems," in *Journal of American Statistical Association*, 89, 278-288.

Kristensen, D. (2009), "Nonparametric Filtering of the Realised Spot Volatility: A Kernel-based Approach," in *Econometric Theory*, in press.

L'Ecuyer, P. (1994), "Efficiency improvement and variance reduction," in *Winter Simulation Conference Proceedings*, 122-132.

Large, J. (2007), "Estimating Quadratic Variation When Quoted Prices Change By A Constant Increment," unpublished manuscript.

Lazowska, E. D. (1984), *Quantitative System Performance, Computer System Analysis Using Queuing Network Models*, Prentice Hall.

Li, Y., and Mykland, P.A. (2007), "Are volatility estimators robust with respect to modeling assumptions?," in *Bernoulli*, 13, 601-622.

— (2008), " Errors and Volatility Estimation," unpublished manuscript.

Lin, M.T., Zhang, J.L., Cheng, Q., and Chen, R. (2005), "Independent Particle Filters," in *Journal of American Statistical Association*, 100, 1412-1421.

Lloyd, A.L., and May, R.M. (1999), "Synchronicity, chaos and population cycles: spatial coherence in an uncertain world," in *Trends in Ecology & Evolution*, 14, 417-418.

Matoušek, J. (1998), "On the L_2-discrepancy for anchored boxes," in *Journal of Complexity*, 14, 527-556.

Matsumoto, M., and Nishimura, T. (1998), "Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator," in *ACM Transactions on Modeling and Computer Simulations*, 8, 3-30.

Meddahi, N., Renault, E., and Werker, B. (2006), "GARCH and irregularly spaced data," *Economic Letters*, 90, 200-204.

Moro, B. (1995), "The full monte," in *Risk*, 8, 57-58.

Musso, M., Oudjane, N., and Le Gland, F. (2001), "Improving Regularised Particle Filters," in *Sequential Monte Carlo Methods in Practice*, eds. A. Doucet, N. de Freitas and N. Gordon, New York: Springer, pp. 274-271.

Neddermeyer, J.C. (2007), "Sequential Monte Carlo Methods for General State-Space Models," Diplomarbeit (Master thesis), University of Heidelberg.

— (2009), "Computationally Efficient Nonparametric Importance Sampling," in *Journal of the American Statistical Association*, 104, 788-802.

— (2010a), "Nonparametric Partial Importance Sampling for Financial Derivative Pricing," in *Quantitative Finance*, in press.

— (2010b), "Nonparametric Particle Filtering and Smoothing with Quasi-Monte Carlo Sampling," in *Journal of Statistical Computation and Simulation*, in press.

Niederreiter, H. (1992), *Random Number Generation and Quasi-Monte Carlo Methods*, Philadelphia: Society for Industrial and Applied Mathematics.

Ninomiya, S., and Tezuka, S. (1996), "Towards real time pricing of complex financial derivatives," in *Applied Mathematical Finance*, 3, 1-20.

Oh, M. S., and Berger, J. (1992), "Adaptive Importance Sampling in Monte Carlo Integration," in *Journal of Statistical Computation and Simulation*, 41, 143-168.

— (1993), "Integration of Multimodal Functions by Monte Carlo Importance Sampling," in *Journal of American Statistical Association*, 88, 450-456.

Ökten, G., and Eastman, W. (2004), "Randomized Quasi-Monte Carlo methods in pricing securities," in *Journal of Economic Dynamics & Control*, 28, 2399-2426.

Owen, A. B. (1992), "Orthogonal arrays for computer experiments, integration and visualization," in *Statistica Sinica*, 2, 439-452.

— (1995), "Randomly Permuted (t; m; s)-Nets and (t; s)- Sequences," in *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*, eds. H. Niederreiter and P. J.-S. Shiue, New York: Springer, 299-317.

Paskov, S. H., and Traub, J. F. (1995), "Faster valuation of financial derivatives," in *Journal of Portfolio Management*, 22, 113-120.

Pikovsky, A.S., Rosenblum, M.G., Osipov, G.V., and Kurths, J. (1997), "Phase synchronization of chaotic oscillators by external driving," in *Physica D*, 104, 219-238.

Pitt, M.K., and Shephard, N. (1999), "Filtering via simulation: Auxiliary particle filters," in *Journal of American Statistical Association*, 94, 590-599.

Plerou, V., Gopikrishnan, P., Gabaix, X., A Nunes Amaral, L., and Stanley, H.E. (2001), "Price fluctuations, market activity and trading volume," in *Quantitative Finance*, 1, 262-269.

Press, W.H., Teukolsky, S.A., Vetterling, W.T., and Flannery, B.P. (1992), *Numerical Recipes in C (2nd ed.)*, Cambridge: Cambridge University Press.

Raftery, A. E., Givens, G. H., and Zeh, J. E. (1995), "Inference from a Deterministic Population Dynamics Model for Bowhead Whales," in *Journal of American Statistical Association*, 90, 402-430.

Robert, C.P., and Casella, G. (2004), *Monte Carlo Statistical Methods*, New York: Springer.

Robert, C.Y., and Rosenbaum, M. (2008), "Ultra high frequency volatility and co-volatility estimation in a microstructure model with uncertainty zones," unpublished manuscript.

Rosenblum, M.G., Pikovsky, A.S., and Kurths, J. (1996), "Phase Synchronization of Chaotic Oscillators," in *Physical Review Letters*, 76, 1804-1807.

Rubinstein, R. Y. (1981), *Simulation and the Monte Carlo Method*, New York: Wiley.

Sato, M. (2000), "Convergence of on-line EM algorithm," in *Proc. Int. Conf. on Neural Information Processing*, 1, 476-481.

Scott, D. W. (1992), *Multivariate Density Estimation*, New York: Wiley.

Sobol, I. M. (1967), "On the distribution of points in a cube and the approximate evaluation of integrals," in *USSR Journal of Computational Mathematics and Mathematical Physics*, 7, 784-802.

Stadler, J. S., and Roy, S. (1993), "Adaptive Importance Sampling," in *IEEE journal on selected areas in communications*, 11, 309-316.

Su, Y., and Fu, M. C. (2000), "Importance sampling in derivative securities pricing," in *Winter Simulation Conference Proceedings*, 587-596.

— (2002), "Optimal importance sampling in securities pricing," in *Journal of Computational Finance*, 5, 27-50.

Takemura, A. (1993), "Tensor analysis of ANOVA decomposition," in *Journal of American Statistical Association*, 88, 1392-1397.

Taylor, J.W. (2004), "Volatility forecasting with smooth transition exponential smoothing," in *International Journal of Forecasting*, 20, 273-286.

Terrell, G.R. (1983), "The Multilinear Frequency Spline," Technical Report, Rice University, Dept. of Math Sciences.

Terrell, G. R., and Scott, D. W. (1985), "Oversmoothed Nonparametric Density Estimates," in *Journal of American Statistical Association*, 80, 209-214.

Titterington, D.M. (1984), "Recursive Parameter Estimation Using Incomplete Data," in *Journal of the Royal Statistical Society, Series B*, 46, 257-267.

Traub, J. F., and Werschulz, A. G. (1998), *Complexity and Information*, Cambridge: Cambridge University Press.

van der Merwe, R., Doucet, A., de Freitas, N., and Wan, E. (2000), "The Unscented Particle Filter," in *Advances in Neural Information Processing Systems (NIPS13)*, eds. T.G. Dietterich, T.K. Leen, and V. Tresp.

Vazquez-Abad, F., and Dufresne, D. (1998), "Accelerated Simulation for Pricing Asian Options," in *Winter Simulation Conference Proceedings*, 1493-1500.

Voev, V., and Lunde, A. (2007), "Integrated Covariance Estimation using High-Frequency Data in the Presence of Noise," in *Journal of Financial Econometrics*, 5, 68-104.

Wang, X., and Fang, K.-T. (2003), "The effective dimension and quasi-Monte Carlo integration," in *Journal of Complexity*, 19, 101-124.

Wang, S., and Zhao, Y. (2006), "Almost sure convergence of Titterington's recursive estimator for mixture models," in *Statistics & Probability Letters*, 76, 2001-2006.

West, M. (1992), "Modelling with Mixtures," in *Bayesian Statistics 4*, eds. J.M. Bernardo et al., Oxford UK: Oxford University Press, 503-524.

— (1993), "Approximating Posterior Distributions by Mixtures," in *Journal of Royal Statistical Society, Ser. A (General)*, 55, 409-422.

Wu, C.F.J. (1983), "On the Convergence Properties of the EM Algorithm," in *Annals of Statistics*, 11, 95-103.

Zhang, L. (2008), "Estimating Covariation: Epps Effect, Microstructure Noise," unpublished manuscript.

Zhang, P. (1996), "Nonparametric Importance Sampling," in *Journal of American Statistical Association*, 91, 1245-1253.

Zhang, L., Mykland, P.A., and Aït-Sahalia (2005), "A Tale of Two Time Scales: Determining Integrated Volatility with Noisy High-Frequency Data," in *Journal of the American Statistical Association*, 100, 1394-1411.

Zhou, B. (1996), "High-Frequency Data and Volatility in Foreign-Exchange Rates," in *Journal of Business & Economic Statistics*, 14, 45-52.

Zlochin, M., and Baram, Y. (2002), “Efficient nonparametric importance sampling for Bayesian learning,” in Proceedings of the 2002 International Joint Conference on Neural Networks IJCNN’02, 3, 2498-2502.

Zumbach, G, Corsi, F., and Trapletti, A. (2002), “Efficient estimation of volatility using high-frequency data,” Technical Report, Olsen & Associates.