SCUOLA DI DOTTORATO
UNIVERSITÀ DEGLI STUDI DI MILANO-BICOCCA

Department of Statistics and Quantitative Methods (DISMEQ)

PhD in Statistics and Mathematical Finance Cycle: XXXII

Curriculum in Mathematical Finance

Application of Sequential Monte Carlo Methods to Dynamic Asset Pricing Models

Candidate: Gonzato Luca

Registration number: 814542

Tutor: Prof. Silvana Stefani

Supervisor: Prof. Carlo Sgarra

Coordinator: Prof. Giorgio Vittadini

ACADEMIC YEAR: 2019/2020

Application of Sequential Monte Carlo Methods to Dynamic Asset Pricing Models

PhD Thesis

PhD Program in Statistics and Mathematical Finance

XXXII Cycle

Author: Luca Gonzato

Supervisor: Prof. Carlo Sgarra

Department of Statistics and Quantitative Methods

University of Milano-Bicocca

November 7, 2020

Abstract

In this thesis we consider the application of Sequential Monte Carlo (SMC) methods to continuous-time asset pricing models. The first chapter of the thesis gives a self-contained overview on SMC methods. In particular, starting from basic Monte Carlo techniques we move to recent state of the art SMC algorithms.

In the second chapter we review existing methods for the exact simulation of Hawkes processes. From our analysis we infer that the simulation scheme of Dassios and Zhao (2013) outperforms the other algorithms, including the most popular thinning method proposed by Ogata (1981). This chapter serves also as introduction to self-exciting jump processes, which are the subject of Chapter 3.

Hence, in the third chapter we propose a new self-exciting jump diffusion model in order to describe oil price dynamics. We estimate the model by applying a state of the art SMC sampler on both spot and futures data. From the estimation results we find evidence of self-excitation in the oil market, which leads to an improved fit and a better out of sample futures forecasting performance with respect to jump-diffusion models with constant intensity. Furthermore, we compute and discuss two optimal hedging strategies based on futures trading. The optimality of the first hedging strategy proposed is based on the variance minimization, while the second strategy takes into account also the third-order moment contribution in considering the investors attitudes. A comparison between the two strategies in terms of hedging effectiveness is provided.

Finally, in the fourth chapter we consider the estimation of continuous-time Wishart stochastic volatility models by observing portfolios of weighted options as in Orlowski (2019). In this framework we don't know the likelihood in closed-form; then we aim to estimate it using SMC techniques. To this end, we marginalize latent states and perform marginal likelihood estimation by adapting the recently proposed controlled SMC algorithm (Heng *et al.*, 2019). From the numerical experiments we show that the proposed methodology gives much better results with respect to standard filtering techniques. Therefore, the great stability of our SMC method opens the door for effective joint estimation of latent states and unknown parameters in a Bayesian fashion. This last step amounts to design an SMC sampler based on a pseudo-marginal argument and is currently under preparation.

Contents

List of Figures

List of Tables

Introduction

Continuous-time models have established themselves as one of the main mathematical tools to describe financial markets. Starting with the celebrated work by Black and Scholes (1973), a huge literature on financial derivatives pricing and hedging was born. For some recent developments in this area see for instance Andersen *et al.* (2015), Fulop and Li (2019) and Bardgett *et al.* (2019). Another important application of continuous-time models is term structure modelling (see among the others Duffie and Kan (1996), Duffee (2002)). Finally, continuous-time macro-finance models addressing the link between asset prices and economic fluctuations also form an important part of financial economists' toolkit (Wachter, 2013, Brunnermeier and Sannikov, 2016).

The estimation of such models on real data is difficult because latent factors are not observable, precluding closed-form expressions for the likelihood function. Moreover, the observation of financial derivatives introduces a tight link between latent states and parameters, leading to poor performances of standard econometric techniques. Further, the increasing complexity of state of the art stochastic volatility models makes empirical applications even more complicated. Then, in order to answer important questions in risk management and derivatives pricing, we consider in this thesis the application of Sequential Monte Carlo (SMC) methods, which are simulation based algorithms that can handle non-linear and/or non Gaussian models.

An overview of SMC methods, from their basic formulation to current state of the art, is given in Chapter 1. Then, in Chapter 2 we introduce Hawkes processes and compare existing methods for their exact simulation. Despite the much greater popularity of the thinning algorithm proposed by Ogata (1981), we find that the simulation scheme of Dassios and Zhao (2013) is the most efficient for the same level of accuracy.

In Chapter 3 we propose a new self-exciting jump diffusion model for oil price dynamics. Our model specification features stochastic volatility, which can jump simultaneously with prices in a self-exciting fashion, and stochastic convenience yield, described by a Ornstein-Uhlenbeck process. By shutting down different channels we retrieve the most relevant dynamic models proposed in the literature related to oil price dynamics. In the empirical application we consider both spot and futures data; then we adapt the two-stage SMC sampler proposed by Fulop and Li (2019) to get parameters and filtered states. The calibrated model is used to dynamically hedge a position in the spot market by shorting futures contracts. In order to compute the so called optimal hedge ratio we consider two approaches: the first based on variance minimization; the second considering also skewness. The latter hedging strategy exhibits some interesting features and gives to the hedger better results with respect to the most popular minimum variance approach. From the empirical application we infer that our model equipped with the self-exciting component, outperforms the one with constant intensity in terms of hedging effectiveness and realized utility.

Chapter 4 deals with the estimation of continuous-time Wishart option pricing models. Indeed, it is well known that estimating such multidimensional stochastic volatility models is very difficult due to complex option pricing formulas, interconnected latent states and positive definiteness constraints. In particular, a simple Euler approximation between two observations does not guarantee positive definiteness making discretization bias a particularly severe issue (see Kang *et al.*, 2017). Then, We follow the prevailing data augmentation approach (see e.g. the survey in Sørensen (2009), or Durham and Gallant (2002)) filling in latent states at artificial time points between the observations, allowing us to control discretization bias. While most existing work is based on Monte Carlo Markov Chain methods, we propose to employ SMC methods. Here the main difficulty to overcome is to provide an efficient proposal distribution over the path of the unobserved latent states between two observations. Standard filtering techniques, such as the bootstrap particle filter (Gordon *et al.*, 1993) or the locally optimal particle filter (Doucet *et al.*, 2000) do not suffice as they only condition on the past, while here the future observation is also informative on the latent path. Then, we propose to adapt the recent controlled SMC method of Heng *et al.* (2019) to tackle this problem, which allows us a generic approach to propagate information from the future. Numerical experiments on simulated data show that our approach is much more stable than standard filtering techniques. In particular, the noise in the estimated marginal likelihood is much lower;

this is crucial in order to implement effective SMC sampler routines for unknown parameters estimation.

Chapter 2 is a joint work with Riccardo Brignone, Chapter 3 is a joint work with Carlo Sgarra, whereas Chapter 4 is a joint work with Andras Fulop, Jeremy Heng and Fabio Trojani.

Part I

Background

Chapter 1

From Monte Carlo to Sequential Monte Carlo

1.1 Monte Carlo Integration

The use of Monte Carlo (MC) methods is widespread across different disciplines, including for example finance, statistics and engineering. This class of methods allows to tackle challenging integration problems by mean of numerical simulation. At the core of MC methods is the concept of *pseudo-random numbers*, which are used by modern mathematical and statistical software to generate sequence of numbers which are, in some sense, random.

In mathematical finance and financial econometrics it is common to face integration problems where a closed-form solution is not available, or where the dimension of the integral makes classical numerical integrations techniques unfeasible.

For what concerns the field of mathematical finance, MC methods have found a lot of applications in option pricing, where the development of complex models prevents to obtain closed-form solutions for the price of options contracts (see e.g. Glasserman, 2003). From the econometric point of view, MC methods are well suited for Bayesian inference, where we need to estimate complex distributions which are not know explicitly. As the name suggests, Bayesian inference is based on Bayes' theorem, which states that, given two events A, B belonging to a probability space Ω, then

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}, \quad P(B) > 0.$$

In Bayesian econometrics the unknown parameters $\theta \in \Theta$ are not constants as in the frequentist approach, but instead they are random variables, with a *prior distribution* $p(\theta)$, defined on a parameter space Θ. Now, if we denote the *data distribution* (which lives on the space $\mathcal{Y}$) as $p(y \mid \theta)$, we obtain from Bayes' theorem the *posterior distribution* of the parameters:

$$p(\theta \mid y) = \frac{p(y \mid \theta)p(\theta)}{\int_{\mathcal{Y}} p(y \mid \theta)p(\theta)d\theta} = \frac{p(y \mid \theta)p(\theta)}{p(y)}. \tag{1.1}$$

To gain intuition about the usefulness of stochastic simulation, assume we are concerned with the following integration problem:

$$I(X) = \mathbb{E}_f\left[g(X)\right] = \int_{\mathcal{X}} g(x)f(x)dx,$$

where, $\mathcal{X}$ denotes the support of the random variable X. The purpose of MC methods is to approximate the above integral by generating N values $\{X_1, \dots, X_N\}$ from a distribution f and then taking the average:

$$\hat{I}(X) = \frac{1}{N}\sum_{i=1}^{N} g(x_i).$$

The empirical mean will converge almost surely to $I(X)$ by the strong law of large numbers as $N \to \infty$. Associated to this estimate there is also an asymptotic variance:

$$\mathrm{Var}[\hat{I}(X)] = \frac{1}{N^2}\sum_{i=1}^{N}\left(g(x_i) - \hat{I}(X)\right)^2.$$

In addition, we can apply the central limit theorem to define the distribution of the standardized estimate:

$$\frac{\hat{I}(X) - I(X)}{\sqrt{\mathrm{Var}[\hat{I}(X)]}} \sim \mathcal{N}(0, 1).$$

Generally speaking, we can say that MC methods are all based on generation of uniform random numbers, from which it is possible to generate all the other distributions.

In particular, one of the most popular approach is given by the *inverse transform method*, which assumes that we know the inverse cumulative distribution function (CDF) of a random variable. More precisely, this method works as follows:

i) Generate a sequence $U_1, \dots U_N \sim \mathcal{U}(0,1)$,

ii) if we plug the U_i into the inverse CDF, $f^{-1}(\cdot)$, we obtain a sample of N values distributed according to f.

This approach works if the random variable of interest admits the inverse CDF, otherwise we need to consider other methods, e.g. the accept-reject or the importance sampling. The next subsection deals with the latter as it constitutes a fundamental building block of Sequential Monte Carlo (SMC) methods. After that, we describe Markov Chain Monte Carlo (MCMC) methods and finally we deal with SMC.

1.1.1 Importance Sampling

Importance sampling is based on the so called *importance density*, which is an auxiliary distribution denoted by $h(x)$. In this case, we don't know how to sample from the target distribution $f(x)$, so we choose to generate from $h(x)$. Then, in order to correct the gap between the target and the importance density we compute the so called *importance weights*, which are defined by the ratio between the two distributions.
Coming back to our original integral, we can rewrite the integrand as follows:

$$\mathbb{E}_f\left[g(X)\right] = \int_{\mathcal{X}} g(x)f(x)dx = \int_{\mathcal{X}} g(x)\frac{f(x)}{h(x)}h(x)dx = \mathbb{E}_h\left[g(X)\frac{f(X)}{h(X)}\right].$$

Now, we have an expected value with respect to the importance distribution, i.e. we made a change of probability measure. We can define the importance weights as $w(X) = f(X)/h(X)$, such that,

$$\mathbb{E}_f\left[g(X)\right] = \mathbb{E}_f\left[g(X)w(X)\right].$$

At this point we proceed by generating random numbers $X_1, \dots, X_N \sim h$ and then we can approximate the expectation with the empirical mean:

$$\mathbb{E}_f\left[g(X)\right] \approx \frac{1}{N}\sum_{i=1}^{N} g(x_i)w(x_i).$$

The choice of h is almost arbitrary, as long as $\text{supp}(h) \supset \text{supp}(f)$.
This technique can also be used to simulate from complex distributions (Robert and Casella, 2009) and in this case it is called *sampling importance resampling.* Indeed, with importance sampling we obtain a sample $X_1, \ldots, X_N$ from h together with its importance weights $f(X_i)/h(X_i)$. Then, by using multinomial resampling (with replacement) we get a sample almost distributed from f. To be more precise, assume that we can sample with replacement from the weighted set $\{X_1, \ldots, X_N\}$, where each X_i is selected with probability $w(X_i)/N$; then the resulting sample denoted by $X^\star$ would be distributed as

$$P(X^\star \in A) = \sum_{i=1}^{N} P(X^\star \in A \,\text{and}\, X^\star = X_i) = \int_A \frac{f(x)}{h(x)}h(x)dx = \int_A f(x)dx.$$

However, the probabilities $w(X_i)/N$ do not sum up to 1; thus we need to normalize the importance weights

$$\tilde{w}(X_i) = \frac{w(X_i)/N}{\frac{1}{N}\sum_{k=1}^{N} w(X_k)}.$$

Note that the above normalization introduces a bias, which for large sample sizes is negligible (see Robert and Casella, 2009).

1.1.2 Markov Chain Monte Carlo

Up to now we have seen how it is possible to generate i.i.d. samples from a distribution through, for example, importance sampling. On the other hand, we may consider to implement MCMC methods, which instead produces correlated samples. This class of algorithms is based on stochastic processes called Markov chains and allows for more flexible proposal distributions in cases where the importance sampling is difficult to apply. Moreover, as we shall see later, they can be combined with SMC methods to produce more efficient and stable algorithms for statistical inference in complex settings. To present this topic we follow closely Robert and Casella (2004) and Casarin (2004).

Thus, we start from basic definitions and properties of Markov chains, which are useful in order to understand the methodologies we are going to present. A Markov chain is a sequence of random variables indexed by time, where the probability of moving from one state to another is dictated by a so called *transition kernel.*

Definition 1. *A transition kernel is a function K defined on the space $\mathcal{X} \times \mathcal{B}(\mathcal{X})$ such that*

(i) $\forall x \in \mathcal{X}$, $K(x, \cdot)$ *is a probability measure;*

(ii) $\forall A \in \mathcal{B}(\mathcal{X})$, $K(\cdot, A)$ *is measurable.*

The transition kernel gives the conditional probability of transition, which is the probability of the next state being in A given that the current state is x. If the Markov chain is defined on a continuous state space we have that $P(X \in A \mid x) = \int_A K(x, x')dx'$, while in discrete time the transition kernel is a matrix with elements $P_{x,x'} = P(X_{t+1} = x' \mid X_t = x)$. In order to completely describe the law of this stochastic process it is important to specify not only the transition kernel, but also the initial distribution $\mu(X_0)$ for the initial state X_0 at time $t = 0$. A remarkable property of Markov chains is given by the following

Definition 2. *Given a transition kernel $K(\cdot, \cdot)$, a sequence $X_0, \dots, X_n, \dots$ of random variables is a Markov Chain if, for any t, the conditional distribution of X_t given the past values of the chain is the same as the distribution of X_t given x_{t-1}, i.e.*

$$P(X_t \in A \mid x_0, \dots, x_k) = P(X_t \in A \mid x_k) = \int_A K(x_k, dx).$$

In order to implement a valid MCMC algorithm, the transition kernel associated to our Markov chain must satisfy some properties. First of all, we need to ensure the existence of an *invariant distribution* for the Markov chain.

Definition 3. *We say that a Markov chain has σ-finite invariant measure π for the transition kernel K, if*

$$\pi(B) = \int_{\mathcal{X}} K(x, B)\pi(dx), \quad \forall B \in \mathcal{B}(\mathcal{X}).$$

The invariant distribution is also *stationary* if π is a probability measure. To guarantee the convergence of an MCMC algorithm we need the underlying Markov chain to be *irreducible* and *recurrent.*

Definition 4. *A Markov chain is irreducible with respect a measure ϕ, if $\forall A \in \mathcal{B}(\mathcal{X})$ with $\phi(A) > 0$, there exists n such that $K^n(x, A) > 0$, $\forall x \in \mathcal{X}$.*

This property essentially means that the kernel K^1 allows for free moves all over the state-space and that for any X_0, the sequence (X_t) has a positive probability of eventually reaching any region of the state-space.

Definition 5. *A Markov chain (X_t) is recurrent if there exists a measure φ such that (X_t) is irreducible and $\forall A \in \mathcal{B}(\mathcal{X})$ such that $\varphi(A) > 0$, $\mathbb{E}_x[\eta_A] = \infty$ for every $x \in A$.*

In the above definition η_A represents the number of passages of X_t in $A \in \mathcal{B}(\mathcal{X})$. Then, this property ensures that the chain is revisiting any subset of A infinitely often.

Definition 6. *An irreducible Markov chain is aperiodic if $\forall x \in \mathcal{X}$, g.c.d. $\{t > 0, K^n(x, x) > 0\} = 1$.*

In other words, an aperiodic chain is prevented from getting stuck in a cycle of states. In addition, an irreducible and recurrent chain is *positive recurrent* if it has an invariant probability measure. Now, we can state the ergodic theorem, which ensures the uniqueness of the invariant distribution and the *ergodicity* of the chain, i.e. the independence on the starting point of the chain.

Theorem 7. *If the Markov chain, (X_t) is aperiodic and positive recurrent, then its invariant distribution $\pi(\cdot)$ is the unique probability distribution satisfying $\pi(x') = \int_{\mathcal{X}} \pi(x)K^n(x, x')dx$, $\forall x'_i \in \mathcal{X}$ and $\forall n \geq 0$ and the chain is said to be ergodic.*

The ergodicity of a Markov chain is very important from the simulation point of view. Indeed, if a given kernel K produces an ergodic Markov chain with stationary distribution π, generating a chain from this kernel K will eventually produce samples from π. In particular, we can apply the law of large numbers in MCMC setting, i.e. for an integrable function h we can compute

$$\frac{1}{n}\sum_{i=1}^{n} h(X_t) \longrightarrow \mathbb{E}_\pi[h(X)]. \tag{1.2}$$

[1] $K^n(x, A)$ denotes the n-times composition of the transition kernel, that is $K^n(x, A) = \int_{\mathcal{X}} K^{n-1}(x', A)K(x, dx')$.

Finally, it is possible to ensure that π is invariant also trough the time-reversibility of a transition kernel.

Definition 8. *A transition kernel is reversible if*

$$\pi(x)K(x,y) = \pi(y)K(y,x).$$

This condition is known as detailed balance condition.

Metropolis-Hastings

Metropolis-Hastings (MH) is one example of a MCMC algorithm and it will be heavily used in this thesis. This method allows to get samples from the target distribution $f(x)$ using an ergodic Markov chain $(X)_t$ with stationary distribution $f(x)$. In order to implement a valid MH algorithm we need to specify a proposal distribution $q(y \mid x)$, which could feature some property like being symmetric, $q(y \mid x) = q(x \mid y)$, or independent of x. Moreover, the ratio

$$\frac{f(y)}{q(y \mid x)},$$

should be known up to a constant. Then, the algorithmic steps of a MH are given in Algorithm 1.

Algorithm 1 Metropolis-Hastings

```
Given $x^{(t)}$,
  1) Generate $Y_t \sim q(y \mid x^{(t)})$,
  2) Take
     $X^{(t+1)} = \begin{cases} Y_t, & \text{with probability} \quad \rho(x^{(t)}, y_t), \\ x^{(t)}, & \text{with probability} \quad \rho(x^{(t)}, y_t), \end{cases}$
     where,
     $\rho(x,y) = \min\left\{ \frac{f(y)q(x \mid y)}{f(x)q(y \mid x)}, 1 \right\}.$
```

To ensure correct sampling from f we impose some minimal regularity conditions on both f and q. In particular, we require that the support of f is connected and that

$$\operatorname{supp} f(x) \subseteq \bigcup_{x \in \operatorname{supp} f(x)} \operatorname{supp} q(\cdot \mid x).$$

If the support is not connected, there would be some subsets of the support that could never be visited. Now, we can check that f is really the stationary distribution of our MH chain by verifying the detailed balance condition. The transition kernel associated with Algorithm 1 is

$$K(x,y) = \rho(x,y)q(y \mid x) + (1 - r(x))\delta_x(y), \tag{1.3}$$

where, $r(x) = \int \rho(x,y)q(y \mid x)dy$ and δ_x denotes the Dirac mass in x. Then, the first term satisfies

$$\rho(x,y)q(y \mid x)f(x) = \rho(y,x)q(x \mid y)f(y).$$

Indeed,

$$\begin{aligned}
\rho(x,y)q(y \mid x)f(x) &= \min\left\{\frac{f(y)q(x \mid y)}{f(x)q(y \mid x)}, 1\right\} q(y \mid x)f(x) \\
&= \min\left\{f(y)q(x \mid y), q(y \mid x)f(x)\right\} \\
&= \min\left\{1, \frac{q(y \mid x)f(x)}{f(y)q(x \mid y)}\right\} f(y)q(x \mid y) \\
&= \rho(y,x)f(y)q(x \mid y).
\end{aligned}$$

The second term of the transition kernel satisfies

$$(1 - r(x))\delta_x(y)f(x) = (1 - r(y))\delta_y(x)f(y),$$

which is true because both sides of the equation are different from zero only when $x = y$. The choice of proposal distribution q is almost arbitrarily and here we recall the two most popular choices:

- *independent MH*: $q(y \mid x) = q(y)$, which is similar to the accept-reject algorithm, but still producing correlated samples.

- *Random walk MH*: $q(y \mid x) = g(y - x)$, which is equivalent to write

$$Y_t = X_t + \varepsilon,$$

where, for example, $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2)$.

In particular the random walk MH allows to control the acceptance probability by tuning the variance of the proposal distribution. This feature of the algorithm is crucial when we combine SMC techniques with the MH sampler.

Gibbs Sampling

Another important MCMC method is given by the Gibbs sampler, which is useful when the target distribution is defined on high dimension space. In this case, we aim to construct a Markov chain $(X)_t$ with multivariate stationary distribution $f(X)$, where $X = (X_1, \ldots, X_r) \in \mathcal{X}$ is a random vector. The Gibbs sampling applies if we can simulate from the full conditional densities and it is described in Algorithm 2. Under some regularity conditions the Markov chain produced by the algorithm converges to the desired stationary distribution (Robert and Casella, 2004). Furthermore, if it is not possible to sample from some of the conditionals, we can use an MH step, constructing a so called *hybrid sampler*. However, in this thesis, from the MCMC family we are going to consider only the MH algorithm; therefore we refer to Robert and Casella (2004) for additional details on the Gibbs sampler.

Algorithm 2 Gibbs sampler

```
Given $X^{(t)}$, generate $X^{(t+1)}$ as follows:
    1) $X_1^{(t+1)} \sim f(x_1^{(t)} \mid x_2^{(t)}, \ldots, x_r^{(t)})$,
    2) $X_2^{(t+1)} \sim f(x_2^{(t)} \mid x_1^{(t+1)}, x_3^{(t)}, \ldots, x_r^{(t)})$,
    3) $\ldots$
    4) $X_r^{(t+1)} \sim f(x_r^{(t)} \mid x_1^{(t+1)}, x_2^{(t+1)}, \ldots, x_{r-1}^{(t)})$,
```

1.2 Sequential Monte Carlo

SMC methods have been recently proposed as powerful tools for the estimation of non-linear and non Gaussian models (Doucet *et al.*, 2001). In particular, we are going to present this class of methods in the framework of state-space models[2] (SSM). A SSM is defined by a couple of stochastic processes (X_t, Y_t), with $X_t \in \mathcal{X}$ being an hidden Markov state process and $Y_t \in \mathcal{Y}$ the observed time series. More precisely, X_t is defined by an initial distribution μ_θ and a transition kernel f_θ

$$X_0 \sim \mu_\theta(\cdot) \quad \text{and} \quad X_t \mid (X_{t-1} = x) \sim f_\theta(\cdot \mid x). \tag{1.4}$$

The observations are conditionally independent given X_t, and the density of Y_t at time t depends on the latent state at time t:

$$Y_t \mid (X_t = x) \sim g_\theta(\cdot \mid x). \tag{1.5}$$

We denote by $\theta \in \Theta$ the parameters of the model and we assume that μ_θ, f_θ and g_θ are densities with respect to suitable dominating measures, denoted by dx and dy. Then, the couple (1.4)-(1.5) defines a SSM. Now, if we define $x_{1:T} := (x_1, \ldots, x_T)$, $y_{1:T} := (y_1, \ldots, y_T)$ and θ is known, the object of Bayesian inference is the posterior density:

$$p_\theta(x_{1:T} \mid y_{1:T}) \propto p_\theta(x_{1:T}, y_{1:T}), \tag{1.6}$$

where,

$$p_\theta(x_{1:T}, y_{1:T}) = \mu_\theta(x_0) \prod_{t=1}^{T} f_\theta(x_t \mid x_{t-1}) \prod_{t=0}^{T} g_\theta(y_t \mid x_t). \tag{1.7}$$

Instead, if θ is unknown we consider the following posterior density:

$$p(x_{1:T}, \theta \mid y_{1:T}) \propto p_\theta(x_{1:T}, y_{1:T}) p(\theta), \tag{1.8}$$

where, $p(\theta)$ is the prior density assigned to θ. Both (1.6) and (1.8) are intractable for non-linear non Gaussian models, thus one option is to exploit simulation based inference outlined in the following sections.

[2] Although in this thesis we are going to consider only applications to SSM, we point out that SMC techniques are not limited to this class of models.

Before going into technical details, it is important to highlight that SSM are ubiquitous across many fields of science like biology, robotics, navigation, economics and finance. For instance, in financial applications, stochastic volatility models are the most prominent examples of SSM, since volatility is not directly observable.

1.2.1 Particle Filtering

As stated in the previous section the scope of SMC is to estimate both latent states and static parameters. In this section we describe the problem of state filtering, given known parameters, following among the others Doucet and Johansen (2008) and Fulop (2011) (to lighten the notation we omit the dependence on θ). In particular, the filtering problem amounts to the computation of (1.6), which for non-linear and non Gaussian SSMs is not feasible, unless in very specific cases like the linear Gaussian SSM, where the Kalman filter gives the closed-form optimal solution.
State estimation via SMC is also known as *Particle Filtering* (PF), since the general idea is to approximate continuous time distributions with discrete points, called particles. In particular, consider the joint filtering distribution $p(x_{0:t} \mid y_{1:t})$ which, can be written as follows:

$$p(x_{0:t} \mid y_{1:t}) = \frac{p(x_{0:t}, y_{1:t})}{p(y_{1:t})}, \tag{1.9}$$

where, $p(y_{1:t}) = \int p(x_{0:t}, y_{1:t}) dx_{0:t}$ is the likelihood of observed data up to t. For the general dynamic model (1.4)-(1.5), if a new observation is available, we update the filtering distribution (1.9) as follows

$$\begin{aligned} p(x_{0:t+1} \mid y_{1:t+1}) &= \frac{p(x_{0:t+1}, y_{1:t+1})}{p(y_{1:t+1})}, \\ &= \frac{g(y_{t+1} \mid x_{t+1}) f(x_{t+1} \mid x_t) p(x_{0:t} \mid y_{1:t})}{p(y_{1:t})} \frac{p(y_{1:t})}{p(y_{1:t+1})}, \\ &= \frac{g(y_{t+1} \mid x_{t+1}) f(x_{t+1} \mid x_t)}{p(y_{t+1} \mid y_{1:t})} p(x_{0:t} \mid y_{1:t}). \end{aligned}$$

Now, we can integrate over $x_{0:t-1}$ to get the joint distribution of x_t and x_{t+1} given $y_{1:t+1}$,

$$\begin{aligned} p(x_{t:t+1} \mid y_{1:t+1}) &= \frac{g(y_{t+1} \mid x_{t+1}) f(x_{t+1} \mid x_t)}{p(y_{t+1} \mid y_{1:t})} p(x_t \mid y_{1:t}), \\ &\propto g(y_{t+1} \mid x_{t+1}) f(x_{t+1} \mid x_t) p(x_t \mid y_{1:t}). \end{aligned}$$

The (marginal) filtering distribution of x_{t+1} is obtained by integrating out x_t

$$p(x_{t+1} \mid y_{1:t+1}) = \int g(y_{t+1} \mid x_{t+1}) f(x_{t+1} \mid x_t) p(dx_t \mid y_{1:t}). \tag{1.10}$$

However, this recursion is not tractable, then we will consider simulation based schemes to deal with this task. The first attempt to solve the filtering problem is given by the *Sequential Importance Sampling* (SIS) algorithm. As outlined in Section 1.1.1 we choose an importance distribution $q(x_{0:t})$ of the following form

$$q(x_{0:t}) \propto \prod_{k=1}^{t} q(x_k \mid y_t, x_{k-1}) q_0(x_0), \tag{1.11}$$

then, the target can be written as

$$p(x_t \mid y_{1:t}) \propto \prod_{k=1}^{t} g(y_t \mid x_t) f(x_k \mid x_{k-1}) \mu(x_0),$$

in such a way that the importance weights are given by the ratio

$$w(x_{0:t}) = \prod_{k=1}^{t} \frac{g(y_t \mid x_t) f(x_k \mid x_{k-1}) \mu(x_0)}{q(x_k \mid y_t, x_{k-1}) q_0(x_0)}. \tag{1.12}$$

The SIS procedure is presented in Algorithm 3.

Algorithm 3 Sequential Importance Sampling

```
For $i = 1, \dots, M$ particles:
    1) at time $t = 0$,
        – sample $x_0^{(i)} \sim q_0(x_0)$,
        – compute $w_0^{(i)} = \frac{\mu(x_0^{(i)})}{q_0(x_0^{(i)})}$.
    2) For $t > 0$,
        – sample $x_t^{(i)} \sim q(x_t \mid y_t, x_{t-1}^{(i)})$,
        – compute $w_t^{(i)} = w_{t-1}^{(i)} \frac{g(y_t|x_t^{(i)}) f_t(x_t^{(i)}|x_{t-1}^{(i)})}{q_t(x_t^{(i)}|y_t, x_{t-1}^{(i)})}$.
end
```

Unfortunately, after few time steps most of normalized weights will be equal to zero, giving birth to the so called *weight degeneracy*. This happens because the variance of the weights grows with time. In order to fight weight degeneracy Gordon *et al.* (1993) proposed to resample particles using the normalized weights as probabilities. The intuition behind this approach is that unlikely trajectories are eliminated and likely ones are multiplied. This algorithm is called *Sequential Importance Resampling* (SIR) and it is outlined in Algorithm 4, where $\mathcal{R}(M; w)$ means that resampling is performed by drawing M points with replacement from a resampling distribution (e.g. the simplest option is a Multinomial) with (normalized) probabilities w.

Algorithm 4 Sequential Importance Resampling

```
For $i = 1, \dots, M$ particles:
    1) at time $t = 0$,
        – sample $x_0^{(i)} \sim q_0(x_0)$,
        – compute $w_0^{(i)} = \frac{\mu(x_0^{(i)})}{q_0(x_0^{(i)})}$.
    2) For $t > 0$,
        – sample $x_t^{(i)} \sim q_t(x_t \mid y_t, x_{t-1}^{(i)})$,
        – compute $w_t^{(i)} = w_{t-1}^{(i)} \frac{g(y_t|x_t^{(i)}) f_t(x_t^{(i)}|x_{t-1}^{(i)})}{q_t(x_t^{(i)}|y_t, x_{t-1}^{(i)})}$,
        – resample $z(i) \sim \mathcal{R}\left(M; w_t^{(i)} / \sum_{k=1}^{M} w_t^{(k)}\right)$,
        – reset the importance weights $w_t^{(i)}$ to 1,
        – update particles $x_t^{(i)} = x_t^{z_t(i)}$
end
```

In addition, Gordon *et al.* (1993) suggest a simple choice for the proposal distribution q, which is given by the transition density, i.e.

$$q(x_t \mid y_t, x_{t-1}) = f(x_t \mid x_{t-1}). \tag{1.13}$$

In this case the importance weights are equal to the observation density

$$w_t^{(i)} = g(y_t \mid x_t^{(i)}). \tag{1.14}$$

With this choice we get the so called *Bootstrap Filter* (BF). Although it is easy to implement, in most applications the BF will give poor performances, since we are not taking into account the current observation in the sampling step. Hence, if the new observation is very informative, the likelihood will be very peaked, causing the elimination of most particles after the resampling step (this is also known as *sample impoverishment*). In particular, the optimal choice would be the following (Fulop, 2011)

$$q(x_t \mid y_t, x_{t-1}) \propto g(y_t \mid x_t) f(x_t \mid x_{t-1}), \tag{1.15}$$

which, in most cases, is not available. An optimal SIR particle filter is defined exactly as Algorithm 4, where the importance distribution is (1.15).

Nevertheless, to alleviate the issue intrinsic of SIR we may consider the *Auxiliary Particle Filter* (APF) proposed by Pitt and Shepard (1999). In the original algorithm, we start by sampling an auxiliary variable for each particle, according to a distribution which weights each point in terms of its compatibility with the new observation. In other words, we first resample past particles according to the auxiliary weights and then we perform the same steps of Algorithm 4. More precisely, with APF we sample jointly the state and an auxiliary index k from

$$p(x_{t+1}, k \mid y_{1:t+1}) \propto g(y_{t+1} \mid x_{t+1}) f(x_{t+1} \mid x_t^{(k)}) w_t^{(k)}. \tag{1.16}$$

Indeed, if we could draw from this joint density and then discard the index we would have produced a sample from the empirical filtering density

$$\hat{p}(x_{t+1} \mid y_{1:t+1}) \propto g(y_{t+1} \mid x_{t+1}) \sum_{i=1}^{M} f(x_{t+1} \mid x_t^{(i)}) w_t^{(i)}. \tag{1.17}$$

Therefore, the idea is to construct a proposal, $(x_{t+1}^{(i)}, k^{(i)}) \sim q(x_{t+1}, k \mid y_{1:t+1})$, and then assign new weights

$$w_{t+1}^{(i)} = \frac{g(y_{t+1} \mid x_{t+1}^{(i)}) f(x_{t+1} \mid x_t^{k^{(i)}})}{q(x_{t+1}^{(i)}, k^{(i)} \mid y_{1:t+1})}. \tag{1.18}$$

As an example, consider the simple approximation to (1.16) suggested by Pitt and Shepard (1999). In particular, let us define

$$q(x_{t+1}, k \mid y_{1:t+1}) \propto g(y_{t+1} \mid \mu_{t+1}^{(k)}) f(x_{t+1} \mid x_t^{(k)}), \tag{1.19}$$

where, $\mu_{t+1}^{(k)}$ represents a point estimate (for example the mean) associated with the density of $x_{t+1} \mid x_t^{(k)}$. Thus, we can sample from $q(x_{t+1}, k \mid y_{1:t+1})$ by simulating the index from a multinomial distribution with probability $\lambda_k \propto g(y_{t+1} \mid \mu_{t+1}^{(k)})$, which are called first-stage weights. Then, sample from the transition $f(x_{t+1} \mid x_t^{(k)})$ and re-weight according to the second-stage weights

$$w_{t+1}^{(i)} \propto \frac{g(y_{t+1} \mid x_{t+1}^{(i)})}{g(y_{t+1} \mid \mu_{t+1}^{(k)})} = \frac{g(y_{t+1} \mid x_{t+1}^{(i)})}{\lambda_k}. \tag{1.20}$$

The main idea of APF is that the second-stage weights should be much less variable than the corresponding weights from SIR. Although it is not needed (see Doucet and Johansen, 2008), we might further resample from this discrete distribution. In Algorithm 5 we present the pseudo-code related to APF. From Algorithm 5 it is evident that APF can be thought as a standard PF which targets a slightly different distribution using modified weights.

Algorithm 5 Auxiliary Particle Filter

```
For $i = 1, \dots, M$ particles:
    1) at time $t = 0$,
        – sample $x_0^{(i)} \sim q_0(x_0)$,
        – compute $w_0^{(i)} = \frac{\mu(x_0^{(i)})}{q_0(x_0^{(i)})}$.
    2) For $t > 0$,
        – compute a point estimate of $x_t^{(i)}$, e.g. $\mu_t^{(i)} = \mathbb{E}\left[x_t \mid x_{t-1}^{(i)}\right]$
        – compute first-stage weights $\lambda_t^{(i)} \propto g(y_{t+1} \mid \mu_t^{(i)})$,
        – sample indices $k_i$ with probability $\lambda_t^{(i)}$
        – propagate particles $x_t^{(i)} \sim f(x_t \mid x_{t-1}^{k_i})$,
        – compute second-stage weights $w_t^{(i)} \propto \frac{g(y_t|x_t^{(i)})}{g(y_t|\mu_t^{(k_i)})}$.
end
```

1.2.2 Particle MCMC

At this stage we have seen how it is possible to perform latent states estimation in the context of SSMs. Throughout our discussion we made a crucial assumption, i.e. that static parameters of the model are known. Indeed, a huge amount of literature has been devoted to solve this task. For instance, online Bayesian static parameter estimation has been proposed in SSMs using SMC and MCMC moves. Unfortunately, all these methods suffer from the path degeneracy problems (see e.g. Liu and West (2001) and Lopes and Tsay (2011)). From the frequentist point of view it is possible to perform online Maximum Likelihood (ML) static parameter estimation by exploiting smoothing algorithms (see Del Moral *et al.*, 2010) or by the so called *continuous resampling* proposed by Malik and Pitt (2011), which works only for one dimensional state[3]. On the other hand, a full Bayesian approach is possible for batch estimation. Here, standard MCMC methods could fail if the likelihood is not know or if one can only simulate the latent process but does not have access to the transition prior. Indeed, this is the case for most non-linear non Gaussian SSMs. Moreover, it is difficult in some situation to design a good proposal distribution and the efficiency of standard MCMC methods is poor. Then, SMC methods can be combined with MCMC to overcome these limitations. The first step in this direction has been proposed by Andrieu *et al.* (2010). In their seminal paper a PF is nested into an MCMC algorithm giving birth to the so called *Particle Markov Chain Monte Carlo* (PMCMC). The approach of Andrieu *et al.* (2010) is usually referred as pseudo-marginalization. The basic idea is that, for a fixed θ, a PF provides a pointwise estimate of the marginal likelihood $p_\theta(y_{1:T})$ and of the conditional density $p_\theta(x_{1:T} \mid y_{1:T})$. Then, the decomposition (1.8),

$$p(x_{1:T}, \theta \mid y_{1:T}) \propto p_\theta(x_{1:T}, y_{1:T}) p(\theta \mid y_{1:T}),$$

suggests to sample from the marginal distribution of θ using $\hat{p}_\theta(y_{1:T})$ and then draw from the states using $\hat{p}_\theta(x_{1:T} \mid y_{1:T})$. A particularly interesting case for next developments is the Particle Marginal Metropolis-Hastings (PMMH) algorithm, which can be seen as an exact approximation of a marginal MH update targeting directly the marginal density

[3] The simulated likelihood function obtained with a PF is discontinuous because of the resampling step, then its maximization is not feasible. Malik and Pitt (2011) proposed to reorder the particles and perform a piecewise linear approximation of the resulting empirical CDF. By this way we get a continuous simulated likelihood function

$p(\theta \mid y_{1:T})$ of $p(x_{1:T}, \theta \mid y_{1:T})$. We define this algorithm as an ”exact approximation” in the sense that, for any fixed number $M \geq 1$ of particles, their transition kernels leave the target density of interest invariant (see Andrieu *et al.*, 2010). If we want to sample from the posterior of fixed parameters, then the validity of PMCMC follows from the unbiasedness of the likelihood. For what concern the joint posterior $p(x_{1:T}, \theta \mid y_{1:T})$, Andrieu *et al.* (2010) prove the result by extending the state-space to all the random quantities produced by PF plus a particle index. For the MH update the proposal density is of the form

$$q(\theta', x'_{1:T} \mid \theta, x_{1:T}) = q(\theta' \mid \theta) p_{\theta'}(x'_{1:T} \mid y_{1:T}), \tag{1.21}$$

then, the MH acceptance ratio is given by

$$\frac{p(\theta', x'_{1:T} \mid y_{1:T}) q(\theta, x_{1:T} \mid \theta', x'_{1:T})}{p(\theta, x_{1:T} \mid y_{1:T}) q(\theta', x'_{1:T} \mid \theta, x_{1:T})} = \frac{p_{\theta'}(y_{1:T}) p(\theta')}{p_{\theta}(y_{1:T}) p(\theta)} \frac{q(\theta \mid \theta')}{q(\theta' \mid \theta)}. \tag{1.22}$$

The particle approximation to this MH update is given just by plugging the correspondent approximations from the SMC. The resulting PMMH procedure is presented in Algorithm 6.

In practice, the performances of PMMH depend heavily on the MC noise of $\hat{p}_{\theta}(y_{1:T})$. Another critical aspect is intrinsic of MCMC methods, i.e. it is not possible to fully adapt the proposal. To overcome these limitations we are going to present in the next sections more stable algorithms based on the so called SMC sampler, which make use of PMCMC.

Algorithm 6 Particle Marginal Metropolis-Hastings Algorithm

```
1. For $k = 0$
    – Set $\theta(0)$ arbitrarily
    – Run a PF at $\theta(0)$ to estimate $\hat{p}_{\theta(0)}(y_{1:T})$ and $\hat{p}_{\theta(0)}(x_{1:T} \mid y_{1:T})$
    – Extract a sample from
        $$X_{1:T}(0) \sim \hat{p}_{\theta(0)}(x_{1:T} \mid y_{1:T})$$
2. For $k = 1, \dots, n$
    – Sample $\theta' \sim q(\theta' \mid \theta(k-1))$
    – Run a PF at $\theta'$ to estimate $\hat{p}_{\theta'}(y_{1:T})$ and $\hat{p}_{\theta'}(x_{1:T} \mid y_{1:T})$
    – Extract a sample from
        $$X'_{1:T} \sim \hat{p}_{\theta'}(x_{1:T} \mid y_{1:T})$$
    – Compute the acceptance probability
        $$\hat{\rho}(\theta, \theta') = \min\left\{ \frac{\hat{p}_{\theta'}(y_{1:T}) p(\theta')}{\hat{p}_{\theta(k-1)}(y_{1:T}) p(\theta(k-1))} \frac{q(\theta(k-1) \mid \theta')}{q(\theta' \mid \theta(k-1))}, 1 \right\}.$$
    – With probability $\hat{\rho}(\theta, \theta')$ set
        $$\theta(k) = \theta',$$
        $$X_{1:T}(k) = X'_{1:T},$$
        $$\hat{p}_{\theta(k)}(y_{1:T}) = \hat{p}_{\theta'}(y_{1:T}).$$
   end
```

1.3 Sequential Monte Carlo Sampler

In this section we present a recently proposed technology to efficiently handle the estimation of general SSMs. These methods make use of PMCMC, which becomes just a part of the whole mechanism. In particular, we are going to present the data tempered SMC^2 algorithm proposed independently by Chopin *et al.* (2013) and Fulop and Li (2013) and the Two-Stage Density Tempered SMC Sampler developed by Fulop and Li (2019).

To fully understand these algorithms it is necessary to introduce the so called SMC Sampler (SMCS), proposed by Del Moral *et al.* (2006). Assume we want to sample from a sequence of target distributions $\gamma_p(\theta)$, for $p = 1, \ldots, P$ and that (for now) the marginal likelihood $p_\theta(y_{1:T})$ is available in closed-form. In this case, Chopin (2002) proposed a data tempered approach exploiting the following decomposition

$$\gamma_p(\theta) \propto p_\theta(y_{1:p})p(\theta). \tag{1.23}$$

In this framework it is possible to conduct sequential Bayesian inference and each posterior of the sequence is of interest. Another alternative is given by a density tempered approach (Del Moral *et al.*, 2006), where the sequence of posteriors is defined as follows

$$\gamma_p(\theta) \propto p_\theta(y_{1:T})^{\xi_p}p(\theta), \tag{1.24}$$

where, $\xi_0 = 0$ and $\xi_P = 1$. In this scenario, we start from the prior at ξ_0 and evolve gradually to the posterior at ξ_P. We just need a way to sample sequentially from $\gamma_p(\theta)$. To this aim we can exploit SMCS (Del Moral *et al.* (2006)). Hence, if we have a particle system $(\theta^{(i)}, s_{p-1}^{(i)}, i = 1, \ldots, M)$ representing $\gamma_{p-1}(\theta)$, we move to the next posterior $\gamma_p(\theta)$ by attaching some weights

$$s_p^{(i)} = s_{p-1}^{(i)} \frac{\gamma_p(\theta^{(i)})}{\gamma_{p-1}(\theta^{(i)})},$$

in such a way that, the particle set $(\theta^{(i)}, s_p^{(i)}, i = 1, \ldots, M)$ will represents $\gamma_p(\theta)$. At this stage it is essential to avoid particle degeneracy (as in the SIS algorithm), so we trigger the so called Resample-Move step whenever the Effective Sample Size[4] (ESS) is lower than a fixed constant B. More precisely, we first resample particles proportional to their weights $s_p^{(i)}$ to produce an equally weighted set, then using a Markov kernel $K_p(\cdot \mid \theta)$ we

[4] This quantity is defined by $ESS = (\sum_{i=1}^M s_p^{(i)})^2 / \sum_{i=1}^M (s_p^{(i)})^2$.

move particles in order to enrich the support without changing the distribution of the particles (we are going to use always MH kernels). Another relevant aspect of SMCS is that normalizing constants can be obtained as a by-product of the algorithm. Indeed, if we denote the normalizing constant by $Z_p(\theta) = \int_\theta \gamma_p(\theta) d\theta$, the identity

$$\frac{Z_p}{Z_{p-1}} = \int_\theta \frac{\gamma_p(\theta)}{\gamma_{p-1}(\theta)} \frac{\gamma_{p-1}(\theta)}{Z_{p-1}} d\theta, \tag{1.25}$$

suggests the following particle estimation

$$\frac{\widehat{Z_p}}{Z_{p-1}} = \sum_{i=1}^{M} \frac{\gamma_p(\theta^{(i)})}{\gamma_{p-1}(\theta^{(i)})} \pi_{p-1}^{(i)}, \tag{1.26}$$

where, $\pi_{p-1}^{(i)}$ are normalized weights.

Therefore, SMCS has several advantages with respect to standard MCMC methods. First, during the Resample-Move step we don't care about samples being correlated, then it is possible to fully adapt the MH proposal on the cloud of particles to better approximate the target (difficult in standard MCMC setting). Second, with SMCS is like running in parallel M interacting MCMC chains, producing consistent estimates of the target distribution as $M \to \infty$. Third, as a by-prodcut we get estimates of normalizing constants. Fourth, with data tempering (1.23) we can perform sequential inference over θ, while by adopting the density tempering approach (1.24), it is less likely to get trapped in local modes.

Up to now, we have assumed that $p_\theta(y_{1:p})$ is available in closed-form, which is not true for most of SSMs. Hence, it is possible to extend the previous methodology to this case by estimating the likelihood using a PF. Then, the Resample-Move step will not be anymore of MCMC-type, but instead a PMCMC one. We outline in the next subsections two prominent examples of this strategy.

1.3.1 Data Tempered SMC^2

In order to present the SMC^2 we follow closely Fulop and Li (2013), while for theoretical results we refer to Chopin *et al.* (2013). The starting point is given by the following decomposition

$$p(x_{1:T}, \theta \mid y_{1:T}) \propto p_\theta(x_{1:T}, y_{1:T}) p_\theta(y_{1:T}) p(\theta), \tag{1.27}$$

which suggests first to sample sequentially over θ by exploiting the marginal likelihood estimates $\hat{p}_\theta(y_{1:t}) = \prod_{l=1}^t \hat{p}_\theta(y_l \mid y_{1:l-1})$. Second, to sample from the hidden states we use a PF. Despite the estimation error in the likelihood we have the right target for finite M. To obtain this result, we employ a pseudo-marginalization argument as in Andrieu *et al.* (2010). As with PMCMC we extend the state-space by including all the random quantities produced by the PF algorithm. Now, if we denote by u_t the random quantities produce by a PF at step t, then we can write their density at time t as follows

$$\psi_\theta(u_{1:t} \mid y_{1:t}) = \psi_\theta(u_1 \mid y_1) \prod_{l=2}^{t} \psi_\theta(u_l \mid u_{l-1}, y_l), \tag{1.28}$$

and the likelihood of the new observation is

$$\hat{p}_\theta(y_t \mid y_{t-1}) = \hat{p}_\theta(y_t \mid u_t, u_{t-1}). \tag{1.29}$$

Then, we have the following auxiliary density

$$\hat{p}(\theta, u_{1:t} \mid y_{1:t}) \propto p(\theta) \prod_{l=1}^{t} \hat{p}_\theta(y_l \mid u_l, u_{l-1}) \psi_\theta(u_l \mid u_{l-1}, y_l). \tag{1.30}$$

From the unbiasedness property of the likelihood, it follows that $p_\theta(y_{1:t})$ is the marginal distribution of (1.30).

Assume we have a weighted set of particles representing the target distribution (1.30) at time $t-1$: $\{\theta^{(n)}, u_{t-1}^{(n)}, \hat{p}_\theta(y_{1:t-1})^{(n)}, s_{t-1}^{(n)}\}_{n=1}^N$, where $s_{t-1}^{(n)}$ denote the weights. Further we have the following recursive relationship between our target distributions at $t-1$ and t:

$$\tilde{p}(\theta, u_{1:t} \mid y_{1:t}) \propto \hat{p}_\theta(y_t \mid u_t, u_{t-1}) \psi_\theta(u_t \mid u_{t-1}, y_t) \tilde{p}(\theta, u_{1:t-1} \mid y_{1:t-1}). \tag{1.31}$$

In order to implement the recursion (1.31) we perform the following three steps:

1) *Augmentation step.* For each $\theta^{(n)}$, run the PF with M particles on the new observation y_t. This is equivalent to sampling from $\psi_{\theta^{(n)}}(u_t \mid u_{t-1}^{(n)}, y_t)$.

2) *Reweighting.* The incremental weights are equal to $\hat{p}_{\theta^{(n)}}(y_t \mid u_t^{(n)}, u_{t-1}^{(n)})$, leading to new weights

$$s_t^{(n)} = s_{t-1}^{(n)} \hat{p}_{\theta^{(n)}}(y_t \mid u_t^{(n)}, u_{t-1}^{(n)}). \tag{1.32}$$

In addition, we get an estimates of the likelihood of the fixed parameters

$$\hat{p}_\theta(y_{1:t})^{(n)} = \hat{p}_\theta(y_{1:t-1})^{(n)} \hat{p}_{\theta^{(n)}}(y_t \mid u_t^{(n)}, u_{t-1}^{(n)}). \tag{1.33}$$

Therefore, the weighted sample $\{\theta^{(n)}, u_t^{(n)}, \hat{p}_\theta(y_{1:t})^{(n)}, s_t^{(n)}\}_{n=1}^N$ is distributed according to $\tilde{p}(\theta, u_{1:t} \mid y_{1:t})$. The normalized weights are $\pi_t^{(n)} = \frac{s_t^{(n)}}{\sum_{k=1}^N s_t^{(k)}}$, such that $ESS_t = \frac{1}{\sum_{k=1}^N (\pi_t^{(k)})^2}$. The marginal likelihood of the new observation can be computed as

$$p(y_t \mid y_{1:t-1}) = \sum_{n=1}^N \pi_{t-1}^{(n)} s_t^{(n)}. \tag{1.34}$$

3) *Resample-Move.* To deal with sample degeneracy we trigger a Resample-Move step whenever the ESS is below some threshold B_1. Then, particles are resampled proportional to $\pi_t^{(n)}$ to produce an equally weighted sample $\{\theta^{(n)}, u_t^{(n)}, \hat{p}_\theta(y_{1:t})^{(n)}, s_t^{(n)}\}_{n=1}^N$. Next, move particles using a PMMH kernel with stationary distribution $\tilde{p}(\theta, u_{1:t} \mid y_{1:t})$. In particular, the proposal distribution is of the form

$$q(\theta, u_{1:t} \mid \theta') = q_t(\theta \mid \theta') \psi_\theta(u_{1:t}), \tag{1.35}$$

For instance, we could assume a multivariate normal for $q_t(\theta \mid \theta')$ with its mean and covariance matrix fitted to the sample posterior of θ. Instead, proposing from $\psi_\theta(u_{1:t})$ imply to run a PF through the entire dataset at θ. If we denote a new proposed particle by $\{\theta', u_t', \hat{p}_\theta(y_{1:t})'\}$, we have the following acceptance rate

$$\min\left\{ \frac{p(\theta') p_{\theta'}(y_{1:t})}{p(\theta^{(n)}) p_\theta(y_{1:t})^{(n)}} \frac{q_t(\theta^{(n)} \mid \theta')}{q_t(\theta' \mid \theta^{(n)})}, 1 \right\} \tag{1.36}$$

From the previous discussion about SMCS we already know that the Resample-Move step is intended to enrich the particles' population. In the SMC^2 algorithm this is the most computationally expensive part. Fortunately, as t increases we need to do it less and less often. In the original paper, the authors suggest to keep moving while the number of unique particles is below some threshold B_2. In practice, they set $B_1 = B_2 = N/2$.

Now, we have a tool to perform sequential Bayesian inferece in general SSMs. It is also possible to obtain samples over the smoothed path of hidden states by sampling a particle index (as in PMCMC), or in alternative by approximating the following expectation

$$\mathbb{E}\left[f(\theta, x_t \mid y_{1:t})\right] \approx \sum_{n=1}^{N}\sum_{i=1}^{M} \pi_t^{(n)} f(\theta^{(n)}, x_t^{(i,n)}). \tag{1.37}$$

The computational cost of this algorithm is quite big, since we run N PFs, each with M particles. Nevertheless these operations are independent between each other, thus a parallel implementation on GPU or CPU is possible.

1.3.2 Two-Stage Density Tempered SMCS

The last algorithm we are going to present has been recently proposed by Fulop and Li (2019). Their method is intended to provide efficient inference when observations are very informative. In financial applications this happens when we consider financial derivatives which are non-linear functions of the underlying variable. In particular, derivatives are very informative on the hidden states given the parameters and vice-versa, hence introducing strong correlation in standard MCMC settings. Further, general pseudo-marginal routines necessitate low variance estimates of the likelihood to be effective. However, with derivatives this is difficult to achieve, making standard algorithms useless (Fulop and Li, 2019).

To overcome such difficulties, the authors suggest to use common random numbers (CRN) and propose a smooth PF to deal with likelihood estimation. First, CRN introduce positive correlations between two successive likelihood estimators, $p(y \mid \theta)$ and $p(y \mid \theta')$, in order to drastically reduce the variance of the estimator of the likelihood ratio $p(y \mid \theta)/p(y \mid \theta')$. Second, given that observations are informative on latent states, the filtering/smoothing distribution tends to be localized around its posterior mean. Then, a normal approximation to this distribution ensures that the likelihood estimate should be a relatively smooth function of θ. In this way, we bypass the resampling step, which could produce samples completely different from each other, exacerbating the error in the likelihood estimation.

Once we have a sufficiently stable PF for likelihood estimation, the objective is to design an efficient proposal targeting the marginal distribution of parameters. To this end Fulop and Li (2019) propose a two-stage density tempered SMCS, building on the pseudo-marginalized routine of Duan and Fulop (2015). The resulting algorithm is not sequential as the SMC^2, but it provides a direct route from the prior to the posterior using all data in a row. In this framework we proceed in two steps: in the first stage we use a small number of state particles M_1 in order to provide a coarse exploration of the posterior and then in the second stage we increase the number of particles M_2 to correct the error in the likelihood estimation. Usually, the second stage is much faster then the first one leading to an overall advantage with respect to an SMCS which runs throughout with a fixed M_2.

To present the algorithm we start from the following decomposition of the posterior distribution

$$p(\theta \mid y_{1:T}) \propto p(y_{1:T} \mid \theta)p(\theta). \tag{1.38}$$

According to the pseudo-marginal approach of Andrieu *et al.* (2010), we focus on the extended posterior $\tilde{p}(\theta, u_{1:T} \mid y_{1:T})$, which includes all the random quantities u_t produced by a PF at time t. To understand the new algorithm proposed by Fulop and Li (2019), we briefly recall the tempering scheme of Duan and Fulop (2015), which extends the algorithm of Del Moral *et al.* (2006) in order to target $\tilde{p}(\theta, u_{1:T} \mid y_{1:T})$.
The main idea is to begin with a simple posterior distribution and gradually move through a sequence of densities to the ultimate target. This can be achieved by setting a tempering schedule ξ_i, with $i = 1, \ldots, I$, for $\xi_1 = 0$ and $\xi_I = 1$, which allows to move from the extended prior $\pi_1(\theta, u_{1:T})$ to the extended posterior $\pi_I = \tilde{p}(\theta, u_{1:T} \mid y_{1:T})$. In general, we have that

$$\pi_i(\theta, u_{1:T}) = \frac{\gamma_i(\theta, y_{1:T})}{Z_i}, \tag{1.39}$$

$$\gamma_i(\theta, y_{1:T}) = \hat{p}_i(y_{1:T} \mid \theta)^{\xi_i}\tilde{p}_i(u_{1:T} \mid \theta, y_{1:T})p(\theta), \tag{1.40}$$

where, $Z_i = \int \gamma_i(\theta, y_{1:T})d(\theta, u_{1:T})$ is a normalizing constant, $\hat{p}_i(y_{1:T} \mid \theta)^{\xi_i}$ is the (tempered) likelihood estimated by a PF and $\tilde{p}_i(u_{1:T} \mid \theta, y_{1:T})$ is the empirical distribution of the auxiliary variables. In order to pass from $\pi_i(\theta, u_{1:T})$ to $\pi_{i+1}(\theta, u_{1:T})$ we reweight each particle by $\hat{p}_i(y_{1:T} \mid \theta^{(n)})^{\xi_{i+1}-\xi_i}$, for $n = 1, \ldots, N$. The tempering coefficients are chosen in such a way to ensure enough particle diversity[5]. Finally, to avoid the usual sample impoverishment a Resample-Move step is performed.

[5] According to Fulop and Li (2019), this can be done by a grid search, where ESS is evaluated at the grid points of ξ and the one with the ESS closest to a fixed constant is chosen

In contrast with this procedure, Fulop and Li (2019) start by fixing two sets of CRN $u_{1:T}^{M_k}$, where $k = 1, 2$ refers to the first and second stage. These random numbers are going to be used in the filter and denote by $\hat{p}(y_{1:T} \mid \theta, u_{1:T}^{M_k})$ the correspondent estimated likelihood given M_k state particles. First, construct a sequence of I_1 densities between the prior $\pi_{1,1}$ and the posterior π_{1,I_1} using a tempering sequence $\xi_{1,i}$, $i = 1, \ldots, I_1$, for $\xi_{1,1} = 0$ and $\xi_{1,I_1} = 1$, and

$$\pi_{1,i}(\theta) = \frac{\gamma_{1,i}(\theta)}{Z_{1,i}}, \tag{1.41}$$

$$\gamma_{1,i}(\theta) = \hat{p}_{1,i}(y_{1:T} \mid \theta, u_{1:T}^{M_1})^{\xi_{1,i}} p(\theta), \tag{1.42}$$

where, $Z_{1,i} = \int \gamma_{1,i}(\theta) d(\theta)$. Second, construct a sequence of I_2 densities between the final posterior using M_1 state particles, $\pi_{2,1} = \hat{p}(y_{1:T} \mid \theta, u_{1:T}^{M_1})p(\theta)$ and the posterior using M_2 state particles, $\pi_{2,I_2} = \hat{p}(y_{1:T} \mid \theta, u_{1:T}^{M_2})p(\theta)$, using a tempering schedule $\xi_{2,i}$, $i = 1, \ldots, I_2$, for $\xi_{2,1} = 0$ and $\xi_{2,I_2} = 1$, and

$$\pi_{2,i}(\theta) = \frac{\gamma_{2,i}(\theta)}{Z_{2,i}}, \tag{1.43}$$

$$\gamma_{2,i}(\theta) = \hat{p}_{2,i}(y_{1:T} \mid \theta, u_{1:T}^{M_2})^{\xi_{2,i}} \hat{p}_{2,i}(y_{1:T} \mid \theta, u_{1:T}^{M_1})^{1-\xi_{2,i}} p(\theta), \tag{1.44}$$

where, $Z_{2,i} = \int \gamma_{2,i}(\theta) d(\theta)$. Finally, in order to rejuvenate the support of parameter particles, Fulop and Li (2019) suggest to keep moving until the cumulative average acceptance rate across the algorithm achieves a pre-specified constant.

It is important to specify that the use of CRN precludes the identification of this method as a pseudo-marginal one. However, the crucial advantage of using CRN with a smooth likelihood estimator is that the estimation noise in the likelihood does not affect the acceptance rates of the algorithm (Fulop and Li, 2019). Furthermore, by performing an extensive simulation study the authors confirm that the bias coming from this approach is negligible. The detailed description of the algorithmic steps can be found in the Appendix of Fulop and Li (2019).

Part II

Research Papers

Chapter 2

Exact Simulation of Hawkes Processes: a guided tour

In this chapter we review all existing methods for exact simulation of Hawkes processes, which are ubiquitous across many fields of science. We conduct a comprehensive numerical experiment where we compare the performances in terms of accuracy and computational efficiency of various schemes. We find that the algorithms of Ogata and Dassios and Zhao are the overall best performing. Despite the much greater popularity of the former, our numerical and algorithmic analysis show that it is not the most efficient. The exact simulation method proposed by Dassios and Zhao (2013) results the best performing for the same accuracy level.

2.1 Introduction

The introduction of Hawkes processes is due to Hawkes (1971a) and Hawkes (1971b). The main property of these kind of processes is that the occurrence of any event increases the probability of futures events. This is why they are also known as self-exciting point processes. Hawkes processes are ubiquitous accross many fields of science. As first application they have been considered in earthquake modeling, where the aftershocks following a main event are described with an Hawkes process; a notable example is given by Ogata (1998). Moreover, self-exciting features proved to be useful to forecast infection diseases (Meyer and Held, 2014), crimes (Mohler *et al.*, 2011) and terrorist attacks (Porter and White, 2012). The popularity of Hawkes processes have also established in other fields: Xu *et al.* (2014) propose applications in e-marketing, Fox *et al.* (2016) for modeling events and social networks behaviour, Vinogradov *et al.* (2019) in mechanical twinning, Johnson (1996) find applications in neuron activity, Nakagawa *et al.* (2019) for modeling population dynamics.

Nevertheless, an impressive spread of applications of Hawkes processes has recently taken place in economics and finance. Indeed, several econometric studies find evidence of self-exciting effects in financial markets. For example, Filimonov *et al.* (2014) fit a Hawkes process to high frequency data related to many different asset classes. They conclude that the vast majority of price changes are due to self-generated activities rather than novel information.

A first strand of financial applications of Hawkes processes is related to high-frequency trading and limit order book. Some notable examples are given by Chavez-Demoulin and McGill (2012), Bacry *et al.* (2013) and Cartea *et al.* (2014).

For insurance markets, Hainaut (2016, 2017) proposes Hawkes processes to analyse the impact of volatility clustering on the evaluation of equity indexed annuities and the contagion between insurance and financial markets. Further applications in insurance are given by Stabile and Torrisi (2010), Dassios and Zhao (2012), Zhu (2013) and Jang and Dassios (2013). From the point of view of credit risk we mention the paper by Errais *et al.* (2010), where the authors proposed to model credit default events in a portfolio of securities as correlated point processes, where the events dynamics is described by a mutually exciting Hawkes process.

Option pricing models have been proposed in the literature involving Hawkes processes in order to take into account the so-called jump clustering phenomenon, widely observed in financial (and other) markets. Hainaut and Moraux (2018) propose a self-exciting jump diffusion model with constant volatility, while Boswijk *et al.* (2016) introduce a stochastic volatility model with self-exciting jumps. Bernis *et al.* (2020) propose a model where both jumps and volatility evolve according to a Hawkes process. For what concerns the pricing of options with exotic payoff structure, Bernis *et al.* (2018) develop pricing formulas for CLO, Brignone and Sgarra (2020) for Asian options, while Kokholm (2016) consider a multivariate setting and propose formulas for multi-asset options pricing.

Regarding asset pricing, some authors provide outstanding applications of Hawkes-type models in the equity market (e.g. Ait-Sahalia *et al.*, 2015, Maneesoonthorn *et al.*,

2016 and Fulop and Li, 2019) and in the energy markets. In this context, we mention Jiao *et al.* (2019), which propose a model based on branching processes for the power market, Gonzato and Sgarra (2020) that estimate a multi-factor stochastic volatility model on oil data and Eyjolfsson and Tjøshteim (2018) that focus on risk-management applications in electricity market.
The list is certainly non exhaustive, for other applications in economics and other fields we refer to Hawkes (2018).

In this chapter we focus on the simulation of a Hawkes process. This is a very important aspect as simulation plays a fundamental role in many practical applications, like filtering unobserved jumps with self-exciting intensity (e.g. Sequential Monte Carlo methods), bootstrapping methods, scenario simulation for risk-management and option pricing by Monte Carlo. Furthermore, we are going to consider Hawkes processes with exponential kernel, which are by far the most popular and allow to calculate analytically expectations of arbitrary functions of the jump times and to compute efficiently the likelihood function (see Bacry *et al.*, 2015).

Hawkes processes can be simulated exactly, in the sense that no approximations are required at any step of the simulation procedure. The first methodology for exact simulation has been proposed by Ozaki (1979), the author applies inverse transform method by exploiting knowledge of the conditional survival function. Due to the impossibility of inverting analytically the survival function, numerical methods (e.g. root finding algorithms) must be applied for any simulation and any jump time, with the result that the whole simulation procedure is slow. An alternative procedure, named "perfect simulation", still relying on numerical techniques have been proposed by Møller and Rasmussen (2005). Ogata (1981) is the first author proposing a simulation scheme which is both fast, in the sense that it does not require any numerical method at any step, and exact. The proposed "thinning" scheme is widely adopted in the literature, for example it is implemented in the `hawkes` and `YUIMA` packages of the `R` programming language (see Zaatour, 2014 and Iacus and Yoshida, 2018). Alternative methods have been also proposed in the literature which do not require any time consuming numerical technique. Daley and Vere-Jones (2008) propose a modification of the original modified thinning of Ogata (1981), while Dassios and Zhao (2013) propose an "exact simulation" scheme based on a different definition of the Hawkes process (which we outline later). Another relevant

Method	Author	Fast	Literature
Inverse CDF	Ozaki	✗	J96
Thinning	Ogata	✓	DFZ14, BDM12, BCTCML15, CHST18
Perfect	Møller and Rasmussen	✗	FS15, HB14
Exact	Dassios and Zhao	✓	LLO16, BKN19

Table 2.1: Summary of the exact simulation methods considered in the chapter. The scheme proposed by Daley and Vere-Jones (2008) is not included as we didn't find any work where it has been used. The third column indicates the computational performance of a specific method, where by "fast" we mean that it does not require implementation of numerical methods at any step. The fourth column gives a (non exhaustive) list of papers where a specific simulation scheme is applied. The literature references have been synthesized in order to save space. In the following we give all the correspondences. J96: Johnson (1996), DFZ14: Da Fonseca and Zaatour (2014), BDM12: Bacry *et al.* (2012), BCTCML15: Bormetti *et al.* (2015), CHST18: Chen *et al.* (2018), FS15: Filimonov and Sornette (2015), HB14: Hardiman and Bouchaud (2014), LLO16: Lee *et al.* (2016), BKN19: Buccioli *et al.* (2019).

methodology for simulating exactly point processes has been proposed by Giesecke *et al.* (2011), but in the case of Hawkes processes their scheme is equivalent to Ogata (1981) modified thinning (see Dassios and Zhao, 2017). In Table 2.1 we give a brief overview of the exact simulation schemes and provide a (non-exhaustive) list of authors using such methods to carry out their numerical experiments. The thinning method of Ogata (1981) is by far the most popular among authors but both Daley and Vere-Jones (2008) modification and "exact simulation" seem to constitute valid alternatives since still exact and expectantly fast (as not requiring any numerical method for their implementation). Furthermore, despite not exact, Euler scheme can also be used for simulating Hawkes processes, this approach is widely used in practice due to its simplicity (see e.g. Fulop and Li, 2019, Hainaut and Moraux, 2019 or Brignone and Sgarra, 2020).

To the best of our knowledge, no comprehensive studies which compare the different simulation schemes have been proposed in the literature. In this study we aim to fill this gap and give a review of efficient (i.e. non dependent on time consuming numerical methods) simulation schemes and investigate which is the best performing in terms of accuracy and computational speed. Our results show that the Exact simulation scheme proposed by Dassios and Zhao (2013) is faster than the competitors for the same level of accuracy.

The rest of the chapter is organized as follows. In Section 2.2 we define the Hawkes process and its basic properties. In Section 2.3 we give an overview of the simulation methods.

In Section 2.4 we show and discuss numerical performances and finally in Section 2.5 we give some final remarks.

2.2 Definitions and basic properties

2.2.1 Definition based on the conditional intensity

A Hawkes process is a special kind of point process whose conditional intensity depends on the history of the events. First of all, let us define the conditional intensity function:

Definition 9. *Let N_t be a point process and let $\mathcal{F}_t^N$ be the natural filtration generated by N itself. Then, the left continuous process defined by:*

$$\lambda_t := \lambda(t \mid \mathcal{F}_{t-}^N) = \lim_{h \to 0^+} \frac{\mathbb{P}\left[N_{t+h} - N_t > 0 \mid \mathcal{F}_{t-}^N\right]}{h} \tag{2.1}$$

is called the conditional intensity function of the point process.

Therefore, we have the following

Definition 10. *The univariate Hawkes process N with conditional intensity λ_t, can be defined for all $t > 0$ and $h \to 0^+$ as:*

$$\begin{cases} \mathbb{P}\left[N_{t+h} - N_t = 1 \mid \mathcal{F}_{t-}^N\right] & = \lambda_t h + o(h) \\ \mathbb{P}\left[N_{t+h} - N_t > 1 \mid \mathcal{F}_{t-}^N\right] & = o(h) \\ \mathbb{P}\left[N_{t+h} - N_t = 0 \mid \mathcal{F}_{t-}^N\right] & = 1 - \lambda_t h + o(h) \end{cases} \tag{2.2}$$

Furthermore, the dynamics of conditional intensity of a Hawkes process with exponentially decaying function can be represented by the following stochastic differential equation (SDE):

$$d\lambda_t = \beta(a - \lambda_t)dt + adN_t, \tag{2.3}$$

where, β is the *constant rate of decay*, a is the *background intensity*, α is the *magnitude* of self-excited jump and N_t is a univariate Hawkes process.

In order to find the solution of (2.3) is possible to apply Itô formula to $f(t, \lambda_t) = e^{\beta t}\lambda_t$

obtaining

$$\lambda_t = a + (\lambda_0 - a)e^{-\beta t} + \sum_{k=1}^{N_t} \alpha e^{-\beta(T_k - t)}, \quad t \geq 0. \tag{2.4}$$

2.2.2 Definition based on the clusters representation

Nevertheless, there is also a cluster-based definition of Hawkes processes. In particular the events of a Hawkes process can be separated into two types: *immigrants* without extant parents and *offspring* that are produced by existing point events. The following cluster-based definition is adapted from Dassios and Zhao (2013):

Definition 11. *Hawkes process with exponentially decaying intensity is a Poisson cluster process* $\{C(T_i, \alpha)\}_{i=1,2,\ldots}$ *with times* $T_i \in \mathbb{R}^+$ *and constant self-exciting contribution* α*. The number of points in* $(0, t]$ *is defined by* $N_t = N_{C(0,t]}$*; the cluster centers of* C *are the particular points called immigrants, the rest of the points are called offspring, and they have the following structure:*

a) The immigrants $I = \{T_i\}_{i=1,2,\ldots}$ *on* $\mathbb{R}^+$ *are distributed as an inhomogeneous Poisson process of rate* $a + (\lambda_0 - a)e^{-\beta t}$*,* $t \geq 0$*.*

b) Each immigrant T_i *generates one cluster* C_i*, and these clusters are independent.*

c) Each cluster C_i *is a random set formed by points of generations or order* $n = 0, 1, \ldots$ *with the following branching structure:*

- *The immigrant and its jump contribution* (T_i, α) *is said to be of generation 0.*
- *Recursively, given generations* $0, 1, \ldots, n$ *in* C_i*, each* $(T_i, \alpha) \in C_i$ *of generation* n *generates a Poisson process of offspring of generation* $n+1$ *on* (T_j, ∞) *with intensity* $\alpha e^{-\beta(t-T_j)}$*,* $t > T_j$*.*

d) C *consists of the union of all clusters, i.e.* $C = \cup_{i=1,2,\ldots} C_i$*.*

2.2.3 Moments of λ_T and N_T

We give, next, the expressions for the conditional expectation and variance of λ_t and for the conditional expected number of jumps:

$$\mathbb{E}_0[\lambda_T] = \frac{a\beta}{k}\left(\lambda_0 - \frac{a\beta}{k}\right)e^{-kT}, \tag{2.5}$$

$$\mathbb{V}_0[\lambda_T] = \frac{\alpha^2}{k}\left[\left(\frac{a\beta}{2k} - \lambda_0\right)e^{-2kT} + \left(\lambda_0 - \frac{a\beta}{k}\right)e^{-kT} + \frac{a\beta}{2k}\right], \tag{2.6}$$

$$\mathbb{E}_0[N_T] = \frac{a\beta}{k}T + \frac{1}{k}\left(\lambda_0 - \frac{a\beta}{k}\right)(1 - e^{-kT}), \tag{2.7}$$

where, $k = \beta - \alpha > 0$. These analytical expressions will be used, following Dassios and Zhao (2013), to evaluate the correctness and accuracy of the simulations in Section 2.4.

2.3 Simulation schemes

In this section we discuss several approaches that have been proposed in the literature in order to simulate an Hawkes process. First, we describe two procedures which give poor performances in terms of speed and for this they will be excluded from our analysis.

In particular, Ozaki (1979) suggested a methodology based on inverse transform method. To apply this algorithm one must solve numerically a transcendental equation, which makes the simulation procedure inefficient. Another exact scheme (named "perfect" simulation) has been developed by Møller and Rasmussen (2005). Nevertheless, the proposed algorithm is very slow due to the necessity of solving numerically at each iteration an integral equation. Therefore, we exclude from our review these algorithms since it is reasonable to assume that they will be outclassed by alternative exact methods which does not employ numerical methods.

Next, we present some exact simulation methods which do not require any numerical method for their implementation. Furthermore, we also include in our discussion the Euler scheme. Despite affected by time discretization bias, this method is widely used in practice due to its simplicity. Moreover, it is expected to be fast in general, with a lower dependence to the expected number of jumps than the other methods. Together with the description of the algorithms we also provide pseudo codes. These are presented in the easiest possible form in order to allow a straightforward implementation.

2.3.1 Euler scheme

The Euler scheme is ubiquitous in stochastic simulation applications due to its simplicity. Consider a general 1-dimensional Itô process

$$dX(t) = \mu(t, X(t))dt + \sigma(t, X(t))dW(t),$$

where, W_t is a standard Brownian motion. Then, the Euler scheme is used to simulate the exact solution of a discrete time process approximating the system dynamics,

$$X_{t_i} = X_{t_{i-1}} + \mu(t_{i-1}, X_{t_{i-1}})\Delta t + \sigma(t_{i-1}, X_{t_{i-1}})\sqrt{\Delta t}\, z_i,$$

where, Δt is a small enough step size and z_i are iid standard normal random variables. In our framework we can discretize the SDE (2.3) by approximating the jump times N_t with a Bernoulli random variable, as in Fulop and Li (2019). Therefore,

$$\lambda_{t_i} = \lambda_{t_{i-1}} + \beta(a - \lambda_{t_{i-1}})\Delta t + \alpha \Delta N_{t_i}, \tag{2.8}$$

where, $\Delta N_{t_i} = N_{t_i} - N_{t_{i-1}} \sim \text{Ber}(\lambda_{t_{i-1}}\Delta t)$.

Algorithm 7 Euler scheme

```
Input: $\lambda_0$, $a$, $\beta$, $\alpha$, $T$, $n$
Output: $\{\lambda_{t_j}\}_{j=1}^{n}$, $\{N_{t_j}\}_{j=1}^{n}$
Set $\Delta t = \frac{T}{n}$ and $N_0 = 0$
for $j = 1 : n$ do
    Draw $u \sim \mathcal{U}(0, 1)$ % draw from uniform
    if $u < \lambda_{t_{j-1}}\Delta t$ then
        $\Delta N_{t_j} = 1$
        $N_{t_j} = N_{t_{j-1}} + 1$
    else
        $\Delta N_{t_j} = 0$
    end
    Set $\lambda_{t_j} = \lambda_{t_{j-1}} + \beta(a - \lambda_{t_{j-1}})\Delta t + \alpha \Delta N(t_j)$
end
```

2.3.2 Ogata (1981) thinning algorithm

The simulation algorithm for Hawkes processes proposed by Ogata (1981) is probably the most used in the literature. It is based on the Lewis and Shedler (1969) thinning simulation method for non-homogeneous Poisson processes. The idea is to simulate a homogeneous Poisson process and then remove excess points, such that the remaining points satisfy the conditional intensity λ_t. This algorithm requires the conditional intensity to be upper bounded, i.e. there should exists a finite $\bar{\lambda}$ such that $\lambda_t \leq \bar{\lambda}$, $\forall t$.

A generalization of the above algorithm, introduced by Ogata (1981), it is usually called Ogata's modified thinning and is outlined in Algorithm 8. This generalization requires only the local boundedness of conditional intensity. Indeed, if λ_t is a non-increasing function (i.e. $\beta > 0$ in (2.3)) in the interval between two adjacent occurrences, we have that $\lambda_t \leq \lambda_{t_i^+}$ for $t \in (t_i, t_{i+1})$, where t_i^+ is the time just after t_i. Therefore, a local bound $\bar{\lambda}_t$ could be set equal to $\lambda_{t_i^+}$ in the interval (t_i, t_{i+1}) and it has to be updated after each occurrence.

Afterwards, Daley and Vere-Jones (2008) modified Ogata's algorithm by setting $\bar{\lambda}_t = \lambda_t$. In this way we are not interested if t is a point of the process or not; whereas we add a function of time interval of length $L_t = k\lambda_{t^+}$, for arbitrary k (the authors suggest to set $k = 0.5$). Then, $\bar{\lambda}_t$ is updated if a new point of the process occurs or if the time frame L_t has elapsed. In Algorithm 9 we describe how to implement Ogata's thinning algorithm as in Daley and Vere-Jones (2008).

2.3.3 Dassios and Zhao (2013) exact simulation

Dassios and Zhao (2013) proposed an efficient sampling algorithm for Hawkes process that does not rely on the accept-reject method. Moreover, this algorithm can generate jump times with either stationary or non-stationary intensity. The starting point is the cluster-based representation of a Hawkes process as given in Definition 11. To understand how the algorithm works consider the following relation for $t_2 > t_1$

$$\lambda_{t_2} = e^{-\beta(t_2 - t_1)} \lambda_{t_1}.$$

So, given the current intensity λ_{t_1}, the future intensity depends only on the time elapsed since time t_1. Therefore, we can use this property in order to decompose the interarrival

Algorithm 8 Ogata (1981) modified thinning algorithm

```
Input: $\lambda_0$, $a$, $\beta$, $\alpha$, $T$
Output: $N_T$, $\{T_j\}_{j=1}^{N_T}$, $\{\lambda_{T_j}\}_{j=1}^{N_T}$
Set $N_0 = 0$, $\epsilon = 10^{-6}$, $j = 0$, $t = 0$, $T_0 = 0$
while $t < T$ do
    Compute $M = \lambda_{t+\epsilon}$ according to (2.4)
    $u_1 \sim \mathcal{U}(0,1)$, $u_2 \sim \mathcal{U}(0,1)$  % draw from two independent uniforms
    $R = -\frac{\ln(u_1)}{M}$  % transform in an exponential with mean $1/M$
    Compute $H = \lambda_{t+R}/M$ according to (2.4)
    if $u_2 > H$ then
        $t = t + R$
    else
        $j = j + 1$, $t = t + R$ , $T_j = t$, $N_{t_j} = N_{t_{j-1}} + 1$
    end
end
Compute $\{\lambda_{T_j}\}_{j=1}^{N_T}$ using formula (2.4)
```

Algorithm 9 Ogata (1981) modified thinning algorithm as in Daley and Vere-Jones (2008)

```
Input: $\lambda_0$, $a$, $\beta$, $\alpha$, $T$
Output: $N_T$, $\{T_j\}_{j=1}^{N_T}$, $\{\lambda_{T_j}\}_{j=1}^{N_T}$
Set $k = 0.5$, $N_0 = 0$, $\epsilon = 10^{-6}$, $j = 0$, $t = 0$, $T_0 = 0$
while $t < T$ do
    Compute $M = \lambda_t$ according to (2.4)
    $L = k\lambda_{t+\epsilon}$
    $u_1 \sim \mathcal{U}(0,1)$  % draw from uniform
    $R = -\frac{\ln(u_1)}{M}$  % transform in an exponential with mean $1/M$
    if $R > L$ then
        $t = t + L$
    else
        $u_2 \sim \mathcal{U}(0,1)$  % draw from uniform
        Compute $H = \lambda_{t+R}/M$ according to (2.4)
        if $u_2 > H$ then
            $t = t + R$
        else
            $j = j + 1$, $t = t + R$ , $T_j = t$, $N_{t_j} = N_{t_{j-1}} + 1$
        end
    end
end
Compute $\{\lambda_{T_j}\}_{j=1}^{N_T}$ using formula (2.4)
```

times into two independent random variables in the following way: the first random variable S_1 represents the interarrival time of the next event, if it is coming from the background intensity a. The second random variable S_2 represents the interarrival time of the next event if it comes from either the exponential immigrant kernel $(\lambda_0 - a)e^{-\beta\, t}$ or the Hawkes self-exciting kernel from each of the past events $\sum_{j:t_j<t} e^{-\beta(t-t_j)}$. Then, the sampled interarrival time is the minimum of these two cases. Moreover, if the second arrival time is not finite, the next event will be taken from the constant rate. The simulation scheme is outlined in Algorithm 10.

Algorithm 10 Dassios and Zhao (2013) exact simulation

```
Input: $\lambda_0$, $a$, $\beta$, $\alpha$, $T$
Output: $N_T$, $\{T_j\}_{j=1}^{N_T}$, $\{\lambda_{T_j}\}_{j=1}^{N_T}$
Set $N_0 = 0$, $j = 0$, $t = 0$, $T_0 = 0$
while $t < T$ do
    $u_1 \sim \mathcal{U}(0,1)$ and $u_2 \sim \mathcal{U}(0,1)$ % draw from two independent uniforms
    $D = 1 + \beta \frac{\ln u_1}{\lambda_{T_j - a}}$
    $S_1 = -\frac{\ln D}{\beta}$ and $S_2 = -\frac{\ln u_2}{a}$
    if $D > 0$ then
        $S = \min(S_1, S_2)$
    else
        $S = S_2$
    end
    $T_{j+1} = T_j + S$ and $t = T_{j+1}$
    $\lambda_{j+1} = (\lambda_j - a)e^{-\beta(T_{j+1} - T_j)} + a + \alpha$
    $N_{t_{j+1}} = N_{t_j} + 1$
    $j = j + 1$
end
```

2.4 Numerical results

In this section we evaluate the accuracy and efficiency of exact simulation schemes for Hawkes processes. Computations are done using Matlab®(Version R2019b) in Microsoft Windows 10®running on a machine equipped with Intel(R) Core(TM) i7-9750HQ CPU @2.60GHz and 16 GB of RAM.

For the numerical experiment we draw $N_\theta = 10^3$ random model parameters from a uniform distribution in such a way that:

- $\lambda_0 - a \in (0, 2]$ and $a \in [0.1, 6]$
- $\beta - \alpha \in (0, 4]$ and $\alpha \in [0.1, 15]$
- $T \in \{0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5\}$
- $\mathbb{V}_0[\lambda_t] < 10$

With these choices we aim to incorporate standard parameter settings used in the literature, in particular we consider Ait-Sahalia *et al.* (2015) and Boswijk *et al.* (2016) which calibrated the Hawkes model on real financial data. Then, for each parameters combination we simulate the Hawkes process $\tilde{M} = 10^6$ times using the methodologies outlined in Section 2.3 and compute expected value, variance and number of jumps for each parameters combination. Accuracy is then evaluated through a comparison with the analytical formulas in (2.5), (2.6) and (2.7). In other words, for any simulation we record $\{\tilde{\lambda}_T^i\}_{i=1}^{\tilde{M}}$ and the number of jumps $\{\tilde{N}_T^i\}_{i=1}^{\tilde{M}}$, compute sample mean, sample variance and average number of jumps and compare with formulas (2.5), (2.6) and (2.7). Bias is then computed according to

$$\text{bias}_{z,\text{EV}} = \frac{1}{\tilde{M}} \sum_{i=1}^{\tilde{M}} \tilde{\lambda}_T^i - \mathbb{E}_0[\lambda_T], \tag{2.9}$$

$$\text{bias}_{z,\text{Var}} = \frac{1}{\tilde{M}} \sum_{i=1}^{\tilde{M}} \left(\tilde{\lambda}_T^i - \frac{1}{\tilde{M}} \sum_{i=1}^{\tilde{M}} \tilde{\lambda}_T^i \right)^2 - \mathbb{V}_0[\lambda_T], \tag{2.10}$$

$$\text{bias}_{z,\text{N}} = \frac{1}{\tilde{M}} \sum_{i=1}^{\tilde{M}} \tilde{N}_T^i - \mathbb{E}_0[N_T], \tag{2.11}$$

where, $z = 1, \ldots, N_\theta$. These quantities are used to evaluate the accuracy of the various simulation methodologies by taking the sample mean (which gives us the average error) and standard deviation of the biases in (2.9), (2.10) and (2.11). We also split the dataset according to quartiles of $\mathbb{E}_0[N_T]$ to see how average error and standard deviation change according to the expected number of jumps.

For the implementation of the Euler scheme one must also specify the number of time discretization steps (denoted with n in Algorithm 7). As in Broadie and Kaya (2006), we follow the suggestion of Duffie and Glynn (1995) to select $n = \sqrt{\tilde{M}} = 10^3$.

The numerical results are reported in Table 2.2. Few comments are in order. The full sample results show that the exact methods perform equally well on each metric.

	Full Sample					
	Expected Value (2.9)		Variance (2.10)		No. jumps (2.11)	
Method	avg. error	st. deviation	avg. error	st. deviation	avg. error	st. deviation
Euler	-0.0001	0.0018	-0.0168	0.0284	-0.0042	0.0061
Thinning	-1.45e-05	0.0017	0.0003	0.0170	-0.0002	0.0052
D-VJ	7.67e-06	0.0017	-0.0003	0.0181	-2.80e-05	0.0055
Exact	7.92e-05	0.0017	0.0005	0.0170	3.44e-05	0.0053
	I quartile, $\mathrm{E}[N_T] \leq 3.0325$					
	Expected Value (2.9)		Variance (2.10)		No. jumps (2.11)	
Method	avg. error	st. deviation	avg. error	st. deviation	avg. error	st. deviation
Euler	-9.94e-05	0.0015	-0.0117	0.0361	-0.0019	0.0030
Ogata	-6.35e-05	0.0014	0.0009	0.0235	0.0001	0.0023
D-VJ	8.11e-05	0.0017	0.0008	0.0253	0.0001	0.0027
Exact	2.97e-05	0.0014	-0.0007	0.0252	0.0001	0.0024
	II quartile, $3.0325 < \mathrm{E}[N_T] \leq 7.0290$					
	Expected Value (2.9)		Variance (2.10)		No. jumps (2.11)	
Method	avg. error	st. deviation	avg. error	st. deviation	avg. error	st. deviation
Euler	-0.0002	0.0018	-0.0144	0.0244	-0.0032	0.0043
Ogata	8.43e-06	0.0018	0.0012	0.0177	9.63e-05	0.0039
D-VJ	-6.91e-05	0.0015	-0.0004	0.0175	-1.78e-05	0.0041
Exact	0.0001	0.0017	0.0013	0.0165	0.0003	0.0036
	III quartile, $7.0290 < \mathrm{E}[N_T] \leq 14.3775$					
	Expected Value (2.9)		Variance (2.10)		No. jumps (2.11)	
Method	avg. error	st. deviation	avg. error	st. deviation	avg. error	st. deviation
Euler	-9.56e-05	0.0017	-0.0148	0.0211	-0.0050	0.0054
Ogata	-6.22e-05	0.0016	-0.0008	0.0134	9.36e-06	0.0054
D-VJ	-0.0001	0.0018	-0.0018	0.0156	-9.73e-05	0.0054
Exact	2.29e-05	0.0017	0.0006	0.0115	0.0002	0.0052
	IV quartile, $\mathrm{E}[N_T] > 14.3775$					
	Expected Value (2.9)		Variance (2.10)		No. jumps (2.11)	
Method	avg. error	st. deviation	avg. error	st. deviation	avg. error	st. deviation
Euler	-6.90e-06	0.0021	-0.0261	0.0276	-0.0066	0.0089
Ogata	5.89e-05	0.0020	8.54e-05	0.0109	-0.0011	0.0076
DV-J	0.0001	0.0019	-0.0001	0.0112	-0.0001	0.0082
Exact	0.0001	0.0020	0.0011	0.0105	-0.0005	0.0081

Table 2.2: Numerical results from the simulations. For the biases of expected value, variance and expected number of jumps we compute mean (average error) and standard deviation.

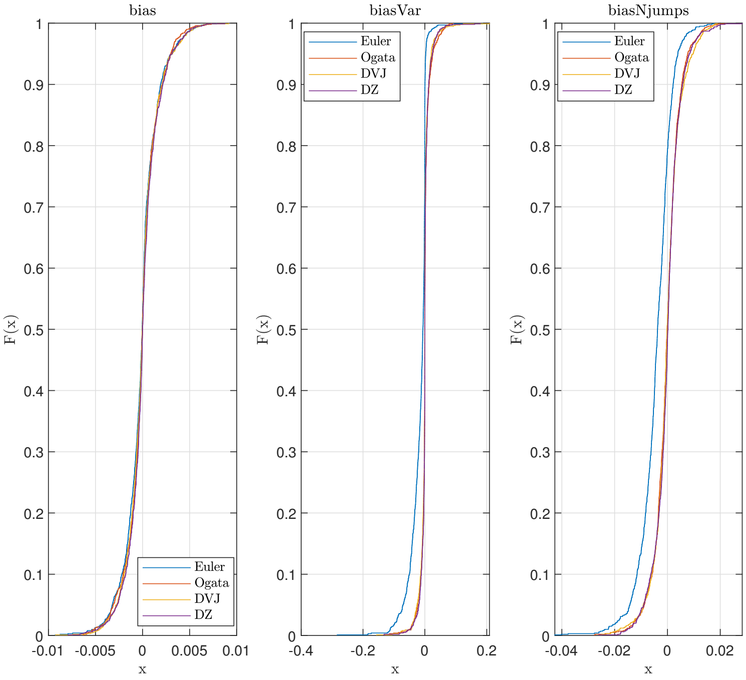

Figure 2.1: Empricial CDF of the 3 statistics: bias_{EV}, bias_{Var} and bias_{N}

The Euler scheme is, as expected, less accurate with a non-negligible average error in computing variance and expected number of jumps. Moreover, the standard deviation of the biases is similar among all the methods, except for the sample standard deviation of the bias of variance of the Euler scheme, which is higher than other methods. Further, we investigate the distribution of the three statistics under consideration by showing their Empirical Cumulative Distribution Function (ECDF) in Figure 2.1. The visual inspection of Figure 2.1 confirm the quantitative results of Table 2.2; i.e. the exact methods seem to be equally unbiased across all dimensions, while the Euler scheme clearly under performs (especially along the variance and expected number of jumps dimensions). An additional statistical evaluation on bias_{EV}, bias_{Var} and bias_{N} has been carried out by performing a pair-wise Kolomogorov-smirnoff test between the algorithms of Ogata and Dassios & Zaho. In particular, we cannot reject the null hypothesis that biases come from the same distributions (p-values are respectively 0.753, 0.7888 and 0.0666).

To evaluate efficiency, we compute the CPU time for running $5 \cdot 10^6$ simulations. We choose such a high number of simulations in order to minimize the risk that the comparison between the different methodologies is influenced by the effective number of jumps over all the simulations. Indeed, with only few simulations, we could observe that a method is faster than another one only because the (random) number of jumps is much smaller and not because the method is effectively faster.

Numerical results are reported in Figure 2.2 (left panel) for different levels of the expected number of jumps (formula 2.7). In the same figure (right panel) we also show the CPU time of the Euler scheme with respect to the final date T. Of course, regarding the exact schemes, we have a positive relationship, i.e. the higher the expected number of jumps, the higher the CPU time. Efficiency of the Euler scheme is not really influenced by the expected number of jumps, while it is sensitive to the length of the time interval considered (or, better, the number of time discretization steps n). For this reason we also report the total average CPU time for each method: the Euler scheme takes 420.47 seconds, Daley and Vere-Jones (2008) takes 28.9281 seconds, Ogata (1981) takes 25.7873 seconds and Dassios and Zhao (2013) takes only 23.0929 seconds, resulting the fastest method. In what follows, we discuss and explain this finding.

The most time consuming steps are: *i*) random sampling from the uniform distribution, *ii*) updating the vectors of jump times and intensity. Indeed, the total number of jumps is unknown at the beginning of each simulation. This fact precludes more efficient vectorizations and pre-allocations. In other words, it is not possible to sample all the needed random quantities in one time at the beginning of the simulation. Contrarily, the Euler scheme does not suffer of this problem as the number of uniforms to be sampled is $n \cdot T$ and does not depend on the effective number of jumps. Despite that, in practical applications n is not a small number since the accuracy of this simulation scheme depends on the number of time discretization points. Then, the greater n, the higher the expected accuracy. For this reason the Euler scheme is slower than the other methods. Therefore, a good environment for its implementation is when the expected number of jumps is very high, but the simulation horizon is small (e.g. $T < 1$).

We compare now the thinning algorithm of Ogata with the adjustment proposed in Daley and Vere-Jones. Let's start by an inspection of Figure 2.2. We note the strange results for the CPU time of the method of Daley and Vere-Jones (2008) for small values of the expected number of jumps. The issue comes exactly from the adjustment they propose. Indeed, a small value for (2.7) is usually due to small values of the intensity parameters. Therefore, in Algorithm 9, we get stuck at the first `if` statement, since the variable L will be always smaller than R, causing a very small update of $t = t +$

L.[1] To the best of our knowledge this unsatisfactory behavior of the algorithm was not previously noted in the literature. However, the proposed modification is expected to give a reasonable compromise between setting the bound M too high, and so generating excessive trial points, and setting it too low, thus requiring too many iterations of steps 5 and 6 in Algorithm 9. In other words, the idea is to avoid running the accept-reject step whenever the possibility of observing a jump is very low, saving the CPU time needed for random sampling. This allows in practice to reduce the number of discarded points, but our numerical results show that this benefit is not commensurate with the extra labour involved. Indeed with respect to Algorithm 8 some extra operations are necessary, i.e. an `if` statement and the computation of L. As a result, Algorithm 9 results slower than Algorithm 8.

Finally, we proceed with a comparison between the exact simulation schemes proposed by Ogata and Dassios and Zhao (2013). Numerical results show that the latter outperforms the former (and, consequently, all the other competitors) in terms of CPU time throughout all the parameter settings considered. This is due to the fact that random sampling is necessary only when the jump effectively occurs, while in thinning algorithms those steps run also when the jump is discarded, resulting in a higher CPU time. Moreover, in Algorithm 8, formula (2.4) must be implemented to compute the upper bound for the intensity M, the acceptance probability H and to update the jump intensity, while, in Algorithm 10, this is evaluated only to update the jump intensity (i.e. only when the jump effectively occurs).

2.5 Concluding Remarks

In this chapter we review the literature related to the simulation of exponentially decaying Hawkes processes, which prove to be very useful across different disciplines. Among the various simulation schemes, we find that, despite rarely used in practise, the exact simulation method proposed by Dassios and Zhao (2013) is the most efficient. This evidence comes from a detailed algorithmic analysis and not only from an observation of the numerical results and is mainly due to the fact that it requires always a smaller or equal amount of random numbers generation than the competitors. This finding raises a

[1] A partial solution to this problem is given by increasing the value of the parameter k in Algorithm 9, which controls the function L.

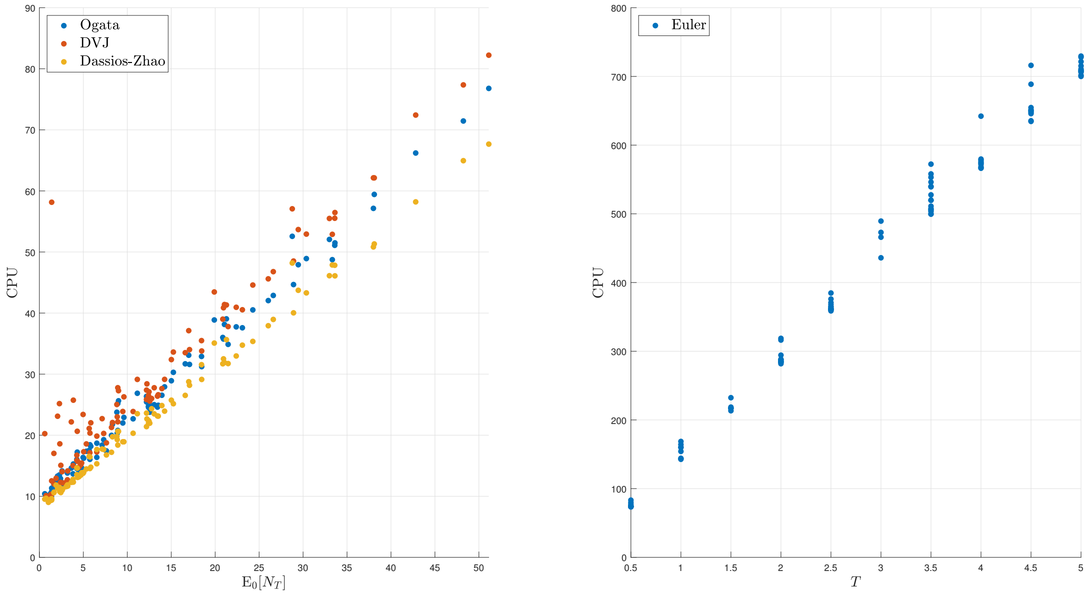

Figure 2.2: CPU time of each simulation scheme with respect to the expected number of jumps (left panel) and the CPU time for Euler scheme with respect to the final date T (right panel).

question about the undisputed greater popularity of Ogata's thinning method. Indeed, for the simulation of exponential Hawkes processes, most authors adopt the latter simulation scheme, which is the standard algorithm in the most popular packages of the computer programming language R.

Moreover, a detailed algorithmic analysis highlights the differences between the two thinning methods considered. Indeed, the modification proposed by Daley and Vere-Jones (2008) should be, theoretically, more efficient with respect to the original algorithm of Ogata (1981). Nevertheless, from our numerical study we observe that this is generally not true. This fact could also explain why the method of Daley and Vere-Jones is scarcely used in practise.

These findings can be useful whenever a high number of Monte Carlo simulations is required in order to compute desired quantities. For instance, we mention the application of Sequential Monte Carlo methods when the jump intensity is not observable. In this framework one needs to sample a high number of particles from the transition density of λ_t at each time step. Another example is certainly the pricing of derivative instruments, where Monte Carlo simulation is often the only possible option due to the mathematical complexity of pricing functions. In both cases, efficiency of the simulation is crucial.

Chapter 3

Self-Exciting Jumps in the Oil Market: Bayesian Estimation and Dynamic Hedging

In this chapter we propose a new self-exciting jump-diffusion model for oil price dynamics based on a Hawkes-type process. In particular, the jump intensity is stochastic and path dependent, implying that the occurrence of a jump will increase the probability of observing a new jump and this feature of the model aims at explaining the jumps clustering effect. Moreover, volatility is described by a stochastic process, which can jump simultaneously with prices. The model specification is completed by a stochastic convenience yield. In order to estimate the model we apply the two-stage Sequential Monte Carlo (SMC) sampler (Fulop and Li, 2019) to both spot and futures quotations. From the estimation results we find evidence of self-excitation in the oil market, which leads to an improved fit and a better out of sample futures forecasting performance with respect to jump-diffusion models with constant intensity. Furthermore, we compute and discuss two optimal hedging strategies based on futures trading. The optimality of the first hedging strategy proposed is based on the variance minimization, while the second strategy takes into account also the third-order moment contribution in considering the investors attitudes. A comparison between the two strategies in terms of hedging effectiveness is provided.

3.1 Introduction

An accurate description of oil price dynamics is crucial for financial applications like risk management, portfolio allocation and derivatives pricing. In addition, the oil market has a strong impact on most aspects of economics in a wider sense; for example, from a macroeconomic perspective oil prices can affect the world GDP growth (Kilian and Figfusson, 2013), the efficiency of oil usage and energy consumption (Wang, 2013) and the term structure of interest rates (Ioannidis and Ka, 2018). This explains the huge amount of literature devoted to provide reliable and accurate methods for oil price dynamics calibration and forecast (Baumeister and Kilian, 2015).

One of the most popular approaches for modeling commodities is represented by factor models, in which continuous-time stochastic differential equations describe the *factors* moving the price dynamics. In particular, among the others we recall the two factor model proposed by Gibson and Schwartz (1990), where the spot price is described by a Geometric Brownian Motion and the convenience yield by an Ornstein-Uhlenbeck process. Later, Ribeiro and Hodges (2004) proposed a multi-factor model, in which the convenience yield is driven by a Cox-Ingersoll-Ross process. This ensures that the convenience yield cannot take negative values and by this way it can avoid arbitrage opportunities. These models account neither for stochastic volatility nor for jumps.

In order to provide a better description of oil prices dynamics, Larsson and Nossman (2011) introduced jumps in both stochastic volatility and returns. They analysed daily spot prices of WTI (West Texas Index) crude oil from 1989 to 2009, and exploited a Markov Chain Monte Carlo method for the model estimation. In contrast with the literature mentioned before, they did not consider mean reversion in the spot price and they only considered a constant convenience yield in the risk-neutral specification of the model. A similar framework was adopted also by Brooks and Prokopczuk (2013).

Another stochastic volatility model with jumps for oil prices was introduced in two recent papers by Fileccia and Sgarra (2015, 2018), where the authors included also a stochastic convenience yield. In their paper the information arising from futures prices is included and the model is estimated under both the historical and the risk-neutral measure via a Particle Markov Chain Monte Carlo method.

In our work we address the problem of describing oil prices dynamics by implementing a different modeling strategy. In particular, we want to investigate if the inclusion of self-exciting effects provides a better understanding of price movements. Self-exciting features have been already systematically investigated in several asset classes. Fulop *et al.* (2015) estimate a model that considers co-jumps in prices and volatility and accounts for jump clustering, on the S&P 500 index data from 1980 to 2012; Ait-Sahalia *et al.* (2015) model financial contagion with mutually exciting jump processes and Maneesoonthorn *et al.* (2016) extend the latter model by introducing self-exciting jumps in volatility in a univariate framework, i.e. no mutual excitation between different assets is considered. Finally, we mention the paper by Fulop and Li (2019), in which the authors propose a non-affine self-exciting jump diffusion model with stochastic volatility together with a new estimation method, the two-stage Sequential Monte Carlo (SMC) sampler. Their methodology is applied on the S&P500 and on the variance swaps observations.

In order to detect self-exciting features in the oil price dynamics we estimate the parameters of a Hawkes-type jump-diffusion model by a particle filtering method. The data set consists of both spot and future quotations of WTI Cushing (Oklahoma) crude oil ranging from January 8, 2008 to December 31, 2018. The particle filtering methodology we apply is based on a hybrid particle filter with a two-stage density tempered SMC method of the same kind of that proposed by Fulop and Li (2019). The model we propose in the present chapter exhibits some similarities with some of the models mentioned above. In particular, it is similar to the affine version of the model considered by Fulop and Li (2019), where in addition we introduce another stochastic differential equation, which describes the evolution of the convenience yield as an OU process. However, an interesting feature of the present work is that we apply this kind of jump diffusion model to WTI crude oil spot and futures quotations. Indeed, there is in the literature some intuition behind the existence of this feature also in commodity markets. For example, Filimonov *et al.* (2014) fit a simple Hawkes process to high frequency data related to many different asset classes, including oil market. Ma *et al.* (2018) consider a plethora of realized range-based volatility models and document an increasing accuracy in futures price volatility forecasting when a Hawkes process is included in the model.

In the present modelling framework jumps with self-exciting features are included in the crude oil spot price dynamics. A remarkable property of the resulting jump-diffusion model is that it is affine, and it allows an explicit computation of the prices of futures contracts as functions of the model parameters, thus providing accurate estimation results. In this chapter we address also the issue of hedging against oil price variations. We provide the computation of an optimal dynamic hedging strategy, where optimality refers to an objective function taking into account not only second order effects (variance), but also skewness. The hedging results highlight the importance of high order effects with respect to the standard approach based only on variance minimization.

The chapter is organized as follows. In Section 3.2 we introduce the Hawkes-type processes and present our self-exciting jump diffusion model. In Section 3.3 we present our estimation method, while in Section 3.4 we describe the data set and the results obtained. In section 3.5 we discuss the optimal hedging strategy based on minimization of variance and skewness. In section 3.6 we provide some concluding remarks.

3.2 The Model

In this section we introduce the most relevant features of Hawkes processes with exponentially decaying intensity. As in Chapter 2, Section 2.2, we can represent the intensity of a Hawkes process by the following stochastic differential equation:

$$d\lambda_t = \beta(\lambda_\infty - \lambda_t)dt + \alpha dN_t, \tag{3.1}$$

where, $\alpha \in \mathbb{R}^+$ is the magnitude of self-excited jump, $\beta \in \mathbb{R}^+$ is the constant rate of decay and $\lambda_\infty \in \mathbb{R}^+$ is the so called background intensity. Then, by applying Itô formula to $f(t, \lambda_t) = e^{\beta t}\lambda_t$, we obtain the solution to the stochastic differential equation above:

$$\begin{aligned} \lambda_t &= \lambda_\infty + (\lambda_0 - \lambda_\infty)e^{-\beta t} + \int_0^t \alpha e^{-\beta(t-s)} dN_s, \\ &= \lambda_\infty + (\lambda_0 - \lambda_\infty)e^{-\beta t} + \sum_{j:t_j<t} \alpha e^{-\beta(t-t_j)}. \end{aligned} \tag{3.2}$$

The solution given by (3.2) allows to simulate the intensity process. Indeed, different simulation methods are proposed in the literature and have been presented in Chapter 2. We can see an example of a simulated path of λ_t in Figure 3.1, where the simulation is performed by applying the algorithm proposed by Dassios and Zhao.

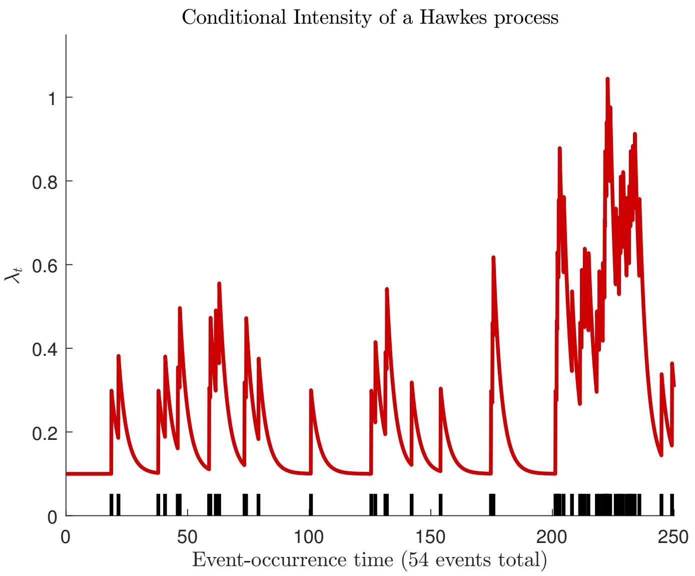

Figure 3.1: Simulated conditional intensity of a Hawkes process on $[0, T]$ with parameters: $T = 250$, $\lambda_0 = \lambda_\infty = 0.1$, $\alpha = 0.2$ and $\beta = 0.3$.

From figure 3.1 we can clearly see the clustering of jumps described by the Hawkes process; for example, in the interval $[200, 240]$ the self-excitation effect looks quite relevant.

An important property of the Hawkes process under consideration is that, although λ_t is clearly non-Markovian, it can be proved that the two-dimensional process (N_t, λ_t) is jointly a Markov and an affine process, and this property improves significantly the analytical tractability of the model. In particular, in Da Fonseca and Zaatour (2014) the infinitesimal generator and the Dynkin's formula can be used in order to find some moments of the process as solutions of ordinary differential equations.

3.2.1 Model Dynamics Under The Historical Measure

Now, we can introduce our dynamic model, which takes into account co-jumps between stock price and volatility, a stochastic convenience yield and self-exciting features. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space with a complete filtration $(\mathcal{F}_t)_{t\geq 0}$, then our model for

$x_t = \ln(S_t/S_0)$ is described by the following system of stochastic differential equations:

$$dx_t = \left(\mu - \frac{1}{2}V_t - \lambda_t \mathbb{E}\left[e^{J_x} - 1\right] - \delta_t\right) dt + \sqrt{V_t} dW_t + dJ_{x,t}, \tag{3.3}$$

$$dV_t = k(\bar{V} - V_t)dt + \sigma_v \sqrt{V_t} dW_{v,t} + dJ_{v,t}, \tag{3.4}$$

$$d\delta_t = \gamma(\bar{\delta} - \delta_t)dt + \sigma_\delta \, dW_{\delta,t}, \tag{3.5}$$

$$d\lambda_t = \beta(\lambda_\infty - \lambda_t)dt + \alpha dN_t, \tag{3.6}$$

From equation (3.3) we see that changes in the log-returns of the underlying are driven by a standard Brownian motion W_t and a compound Poisson process $J_{x,t}$. the number of jumps N_t is an Hawkes process with stochastic intensity λ_t. Furthermore, the amplitude of jumps is described by i.i.d Gaussian random variables with mean μ_J and variance σ_J^2. Equation (3.4) describes the evolution of the volatility by a mean-reverting jump-diffusion process, where $W_{v,t}$ is a standard Brownian motion, possibly correlated with W_t, i.e. $\text{Corr}(dW, dW_v) = \rho_v dt$. This feature of the model is important in order to capture the so called *leverage effect*. Furthermore, recent studies find the presence of co-jumps in prices and volatility, not only on the equity market (Eraker, 2004, Eraker *et al.*, 2003, Fulop and Li, 2019), but also on the commodity market (Larsson and Nossman, 2011, Brooks and Prokopczuk, 2013). Moreover, there is evidence of jump clustering (Ait-Sahalia *et al.*, 2015, Fulop *et al.*, 2015), i.e. an extreme movement tends to be followed by another extreme movement. As a consequence, we introduce jumps in volatility by $J_{v,t}$, which is a compound Poisson process with counting process N_t. To be more precise, returns and volatility jump together with the same self-exciting intensity λ_t and the jump size of volatility follows an exponential distribution with mean μ_v. Finally, there is an additional SDE, which describes the evolution of the convenience yield by a standard Ornstein-Uhlenbeck process as in Gibson and Schwartz (1990); Schwartz (1997); Lai and Mellios (2016); Yan (2002). The convenience yield takes into account both the reduction in cost of acquiring inventory and the value of being able to profit from temporary local shortage of the commodity (Yan, 2002), so it is natural to adopt a stochastic process, which could assume both positive and negative values.

In our empirical application we consider two nested models:

- Model I defined by Equations (3.3)-(3.6),

- Model II with constant jump intensity, i.e. $\beta = \alpha = 0$.

3.2.2 Risk-Neutral Dynamics and Futures Pricing

In order to perform a joint estimation based on both spot and futures data, we need to derive a pricing formula for futures contracts. As usual, we employ a suitable change of measure form the real world probability measure $\mathbb{P}$ to the risk-neutral probability measure $\mathbb{Q}$. To this end we consider the following Radon-Nikodym derivative:

$$\frac{d\mathbb{P}}{d\mathbb{Q}}\Big|_{\mathcal{F}_t} = \exp\left\{-\int_0^t r_s ds - \frac{1}{2}\int_0^t \varphi_x^2(u)du - \int_0^t \varphi_x(u)dW_u - \frac{1}{2}\int_0^t \varphi_\delta^2(u)du - \int_0^t \varphi_\delta(u)dW_u\right\}, \tag{3.7}$$

where r_t denotes the risk-free interest rate. Actually we are pricing only the convenience yield risk, neither volatility nor jump risk. This choice is motivated by the fact the futures price is not a function of spot volatility, jumps or their associated parameters[1] (see Yan, 2002). However, volatility and jumps are important in order to describe appropriately spot prices and for hedging purpose, as we shall see later. Thus, by following the literature (e.g. Pan, 2002, Yu *et al.*, 2011, Fulop and Li, 2019), we leave $\varphi_x(t)$ unspecified and we choose the convenience yield risk premium with the following form:

$$\varphi_\delta(t) = \frac{\varphi_\delta}{\sigma_\delta} \quad \Rightarrow \quad dW_{\delta,t}^{\mathbb{Q}} = dW_{\delta,t}^{\mathbb{P}} + \frac{\varphi_\delta}{\sigma_\delta}dt. \tag{3.8}$$

Thus, under the risk-neutral measure the structure of the model is preserved:

$$dx_t = \left(r - \frac{1}{2}V_t - \lambda_t \mathbb{E}\left[e^{J_x} - 1\right] - \delta_t\right)dt + \sqrt{V_t}dW_t + dJ_{x,t}, \tag{3.9}$$

$$dV_t = k(\bar{V} - V_t)dt + \sigma_v\sqrt{V_t}\,dW_{v,t} + dJ_{v,t}, \tag{3.10}$$

$$d\delta_t = \gamma(\bar{\delta}^{\mathbb{Q}} - \delta_t)dt + \sigma_\delta dW_{\delta,t}^{\mathbb{Q}}, \tag{3.11}$$

$$d\lambda_t = \beta(\lambda_\infty - \lambda_t)dt + \alpha dN_t, \tag{3.12}$$

where, $\bar{\delta}^{\mathbb{Q}} = \bar{\delta} - \varphi_\delta/\gamma$.

The model we are proposing belongs to the class of affine models, which means that

[1] The economic justification of this result relies in the linearity of the futures payoff. Indeed, volatility and jumps affect higher order moments, but not the first one. From the mathematical point of view, the payoff of the contract is just the expected spot price under $\mathbb{Q}$; thus in the spot price dynamics we compensate the drift with the jump compensator $\lambda_t\mu^\star$ and the Itô term $1/2V_t$. As a consequence, the solution of the respective ODEs is equal to zero.

the characteristic function can be computed in closed-form. This feature of affine models is very important in order to price financial derivatives as we shall see below. The payoff of a futures contract $F(t,\tau)$, with time to maturity $\tau = T - t$, is given by the usual relation:

$$F(t,\tau) = \mathbb{E}^{\mathbb{Q}}[S_T \mid \mathcal{F}_t] = \mathbb{E}^{\mathbb{Q}}[e^{x_T} \mid \mathcal{F}_t], \tag{3.13}$$

where $x_T = \ln(S_T)$. We start by considering the moment generating function of x_T:

$$G(w, x_t, V_t, \delta_t, \lambda_t, t, \tau) = \mathbb{E}\left[e^{wx_T} \mid \mathcal{F}_t\right]$$

Now, since under the risk-neutral measure the model structure is the same, we drop the $\mathbb{Q}$ superscript in order to lighten the notation, and by the Feynman-Kac theorem we obtain the following partial differential equation:

$$\begin{aligned}
&- G_\tau + \left(r - \frac{1}{2}V_t - \lambda_t \mathbb{E}\left[e^{J_x} - 1\right] - \delta_t\right) G_x + \frac{1}{2}V_t G_{xx} + k(\bar{V} - V_t)G_v + \\
&+ \frac{1}{2}\sigma_v^2 V_t G_{vv} + \rho_v \sigma_v V_t G_{xv} + \beta(\lambda_\infty - \lambda_t)G_\lambda + \gamma(\bar{\delta} - \delta_t)G_\delta + \frac{1}{2}\sigma_\delta^2 G_{\delta\delta} \\
&+ \lambda_t \int \left[G(w, x_t + J_x, V_t + J_v, \delta_t, \lambda_t + \alpha, t, T) - G(w, x_t, V_t, \delta_t, \lambda_t, t, T)\right] \nu(dJ_x, dJ_v) = 0
\end{aligned} \tag{3.14}$$

with the terminal condition $G_T = \exp(wX_T)$. Now, we guess a solution of the form:

$$G(w, x_t, V_t, \delta_t, \lambda_t, t, \tau) = \exp\left\{wx_t + A(w,\tau) + B(w,\tau)V_t + C(w,\tau)\delta_t + D(w,\tau)\lambda_t\right\} \tag{3.15}$$

subject to $A(0) = 0$, $B(0) = 0$, $C(0) = 0$ and $D(0) = 0$. From the PDE (3.14) we obtain the following system of ODEs:

$$\begin{cases}
\frac{\partial A(w,\tau)}{\partial \tau} = rw + \beta\lambda_\infty D(w,\tau) + k\bar{V}B(w,\tau) + \gamma\bar{\delta}C(w,\tau) + \frac{1}{2}\sigma^2 C^2(w,\tau), \\
\frac{\partial B(w,\tau)}{\partial \tau} = -\frac{1}{2}(w - w^2) - (k - \rho\sigma_v w)B(w,\tau) + \frac{1}{2}\sigma_v^2 B^2(w,\tau), \\
\frac{\partial C(w,\tau)}{\partial \tau} = -w - \gamma C(w,\tau), \\
\frac{\partial D(w,\tau)}{\partial \tau} = -\beta D(w,\tau) + \int \left[e^{wJ_x + B(w,\tau)J_v + D(w,\tau)\alpha} - 1\right] \nu(dJ_x, dJ_v) - \mathbb{E}\left[e^{J_x} - 1\right] w.
\end{cases} \tag{3.16}$$

Now, the futures price is given simply by the moment generating function computed in $w = 1$; then the solution of the second ODE, which is a Riccati equation without the constant term, is $B(\tau) = 0$. Due to this result we also have $D(\tau) = 0$. Hence, as we anticipated before, future prices are not affected by volatility and jumps. Therefore, the solution of the equation for the log-futures is of the form:

$$\ln F(t,\tau) = \ln S_t + A^{\mathbb{Q}}(\tau) + C^{\mathbb{Q}}(\tau)\delta_t, \tag{3.17}$$

where,

$$A^{\mathbb{Q}}(\tau) = r\tau + \frac{\delta^{\mathbb{Q}}\left(-\gamma\tau - e^{-\gamma\tau} + 1\right)}{\gamma} + \frac{\sigma_\delta\left(0.5\gamma\tau - 0.25e^{-2\gamma\tau} + e^{-\gamma\tau} - 0.75\right)}{\gamma^3}, \tag{3.18}$$

$$C^{\mathbb{Q}}(\tau) = \frac{e^{-\gamma\tau} - 1}{\gamma}. \tag{3.19}$$

3.3 Parameters Estimation Method

In order to estimate the parameters by using real data, we cast our model in a state-space form and apply a simple Euler scheme. Denoting by Δt a small time interval, the observation equation for stock prices is given by:

$$\ln S_t = \ln S_{t-1} + \left(\mu - \frac{1}{2}V_{t-1} - \lambda_{t-1}\mathbb{E}\left[e^{J_x} - 1\right] - \delta_{t-1}\right)\Delta t + \sqrt{V_{t-1}\Delta t}\,W_t + J_x\Delta N_t, \tag{3.20}$$

where, $J_x \sim \mathcal{N}(\mu_J, \sigma_J^2)$, $W_t \sim \mathcal{N}(0,1)$ and $\Delta N_t = N_t - N_{t-1} \sim \text{Bernoulli}(\lambda_{t-1}\Delta t)$.

In this work we also consider futures prices with n maturities. In agreement with the literature, since we observe derivatives data with measurement errors (Eraker, 2004, Fulop and Li, 2019, etc.), we have an additional observation equation:

$$\ln F(t,T)^O = \ln F(t,T)^M + \epsilon_t,$$

where, $\ln F(t,T)^O$ is a vector of observed futures prices at time t with maturity T, and $\ln F(t,T)^M$ are the corresponding prices obtained from Equation (3.17). The measurement errors are assumed to follow a multivariate normal distribution, i.e. $\epsilon_t \sim \mathcal{N}(0,\Omega)$, with $\Omega = \Sigma\Sigma^T$ and $\Sigma = \sigma_e I_n$.

Our model features latent states, which cannot be observed. The discretized version of Equations (3.4)-(3.5)-(3.6) is the following:

$$V_t = V_{t-1} + k(\bar{V} - V_{t-1})\Delta t + \sigma_v \sqrt{\Delta t V_{t-1}} Z_t + J_v \Delta N_t \tag{3.21}$$

$$\delta_t = \delta_{t-1} + \gamma(\bar{\delta} - \delta_{t-1})\Delta t + \sigma_\delta \sqrt{\Delta t} W_{\delta,t} \tag{3.22}$$

$$\lambda_t = \lambda_{t-1} + \beta(\lambda_\infty - \lambda_{t-1})\Delta t + \alpha \Delta N_t \tag{3.23}$$

where, for fixed t, Z_t is a standard normal variable correlated with W_t appearing in Equation (3.20), $J_v \sim \text{Exp}(\mu_v)$ and $W_{\delta,t}$ is an independent standard normal variable.
In order to address the estimation of the model, we rely on SMC methods. In particular, we shall consider the two-stage density tempered SMC of Fulop and Li (2019) presented in Chapter 1, Section 1.3. Our estimation approach delivers an approximation of the marginal likelihood,

$$p(y_{1:T}) = \int p(y_{1:T} \mid \Theta) p(\Theta) d\Theta,$$

which can be used to construct Bayes factors for model comparison. More precisely, for two models M_1 and M_2 the Bayes factor is given by the ratio of their marginal likelihoods,

$$BF_{1,2} = \frac{p(y_{1:T} \mid M_1)}{p(y_{1:T} \mid M_2)}.$$

The Bayes factor does not rely on asymptotic distribution theory and provides a simple way to evaluate different models. Besides, it contains a penalty for the introduction of more parameters.

3.4 Empirical Application

3.4.1 Data Description

In this section we present the data we are going to examine in our empirical analysis. In particular, we consider WTI Cushing (Oklahoma) crude oil spot and futures quotations obtained from Bloomberg[2]. In Figure 3.2 we plot a time series of daily log-returns: $Y_t = \log(S_t/S_{t-1})$, ranging from 08/01/2008 to 31/12/2018, thus 2767 spot data are considered. The data cover some crucial historical periods as the global financial crisis,

[2] spot data ticker: `USCRWTIC`, futures data ticker: `CLm`, for $m = \{3, 6, 9, 12, 18, 21, 24\}$.

when oil prices experienced huge fluctuations between 2008 and 2010. Moreover, in the middle of 2014, price started declining due to a significant increase of oil production in USA and to a decrease of demand in the emerging countries. Besides, from 2016 to 2018, complex negotiations with OPEC led to higher variability in oil prices.

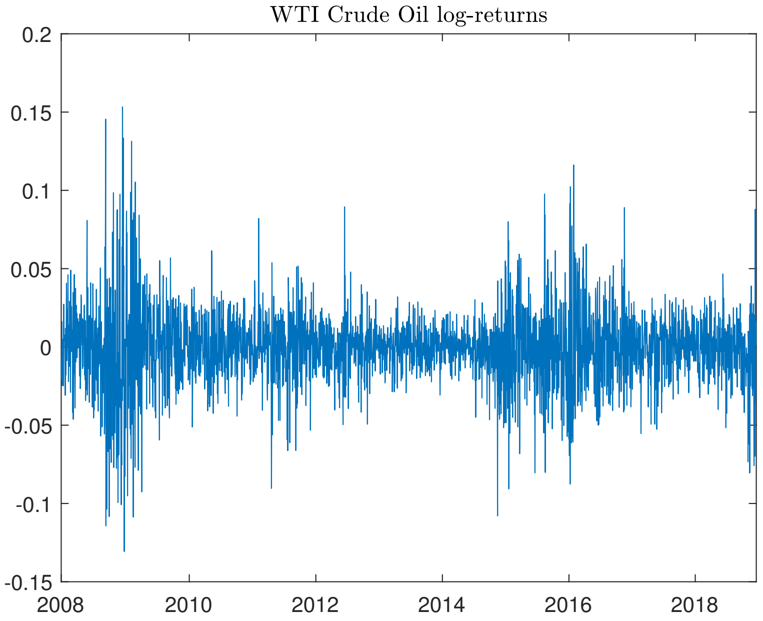

Figure 3.2: WTI crude oil log-returns from 08/01/2008 to 31/12/2018.

First, a preliminary statistical analysis is performed on log-returns. To this end in Table 3.1 we show some descriptive statistics and the result of a Jarque-Bera (JB) normality test. In addition a QQ-plot is provided in Figure 3.3. This simple analysis suggests that observations do not come from a Gaussian distribution.

Statistics	**Log-return**
Observations	2767
Mean	-2.7179e-04
Standard Deviation	0.0245
Skewness	0.1181
Kurtosis	7.3057
Min value	-0.1307
Max value	0.1533
JB test	Rejected

Table 3.1: Descriptive statistics and JB test on WTI crude oil log returns observed with daily frequency over the period 08/01/2008 to 31/12/2018.

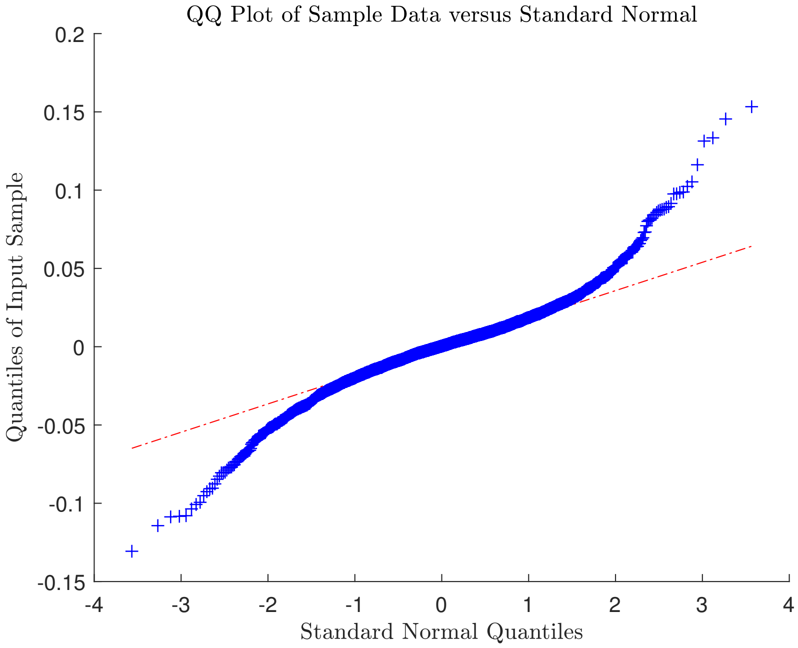

Figure 3.3: Q-Q Plot for daily frequency returns on WTI crude oil.

Moreover we consider futures contracts written on WTI crude oil. In Figure 3.4 we show log-futures prices from 08/01/2008 to 31/12/2018, for maturities τ equal to 3, 12 and 24 months. For estimation purpose, we retain 8 futures contracts with fixed maturities ranging from 3 to 24 months. Contracts with more than 2 years of maturity are less liquid (Lai and Mellios, 2016), therefore they are not considered in our estimation procedure. As we did for spot observations, we conduct a preliminary statistical analysis on the whole futures data in Table 3.2.

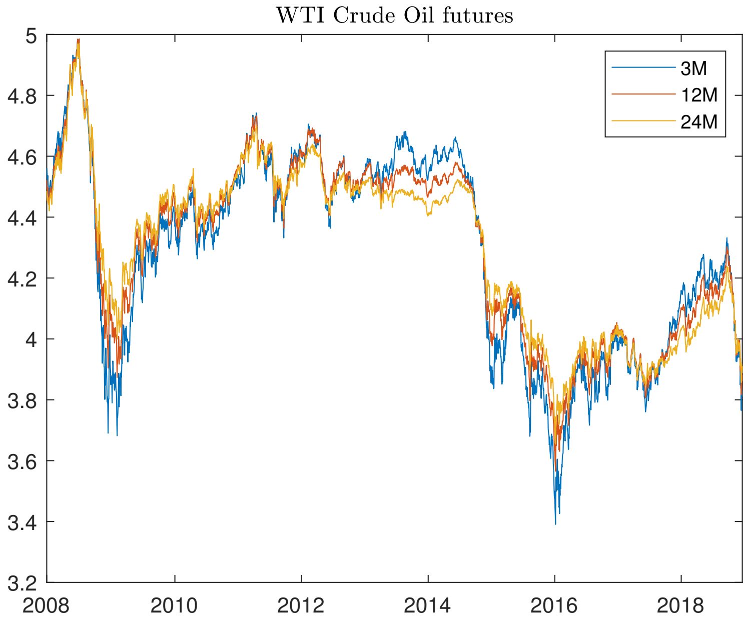

Figure 3.4: WTI futures contracts from 08/01/2008 to 31/12/2018.

Statistics	**CL3**	**CL6**	**CL9**	**CL12**	**CL15**	**CL18**	**CL21**	**CL24**
Mean	4.2842	4.2968	4.3019	4.3041	4.3047	4.3046	4.3043	4.3039
Standard Dev	0.3217	0.3058	0.2956	0.2879	0.2814	0.2760	0.2713	0.2673
Skewness	-0.3142	-0.2751	-0.2547	-0.2422	-0.2337	-0.2277	-0.2224	-0.2201
Kurtosis	2.0059	1.9591	1.9465	1.9358	1.9328	1.9388	1.9475	1.9598
Min value	3.3908	3.4825	3.5293	3.5656	3.5943	3.6168	3.6368	3.6548
Max value	4.9845	4.9894	4.9895	4.9858	4.9822	4.9782	4.9747	4.9716

Table 3.2: Descriptive statistics on WTI crude oil log futures prices observed with daily frequency over the period between 08/01/2008 and 31/12/2018.

3.4.2 Estimation Results

In this subsection we present the results obtained from our estimation procedure. In our empirical application, for the two-stage SMC sampler, we set the number of parameter particles N equal to 1000, the number of state particles at the first stage M_1 equal to 30, and the number of state particles M_2 at the second stage equal to 500. The algorithm is initialized by using the priors in Table 3.3 with $\Sigma = \sigma_e I_n$, where I_n is the unit matrix in n dimensions. The choice of the hyper-parameters of the prior distributions is based on calibration using the training sample approach, which is widely used to calibrate the objective priors (Fulop and Li, 2013; Fulop *et al.*, 2015; Fulop and Li, 2019). It is worth remarking that we find most parameters not sensitive to the selection of the priors. Moreover, we take the random walk proposal, we trigger the resample-move step when the Effective Sample Size reaches $N/2$, and then keep moving until the cumulative average acceptance rate across the population reaches the value 2.

Θ	Dist	Support	(μ_0, σ_0)	Θ	Dist	Support	(μ_0, σ_0)
μ	Normal	$(-\infty, \infty)$	(0.02, 0.15)	φ_δ	Tr. Normal	$(-\infty, \infty)$	(0.5, 0.1)
μ_J	Tr. Normal	$(0, \infty)$	(-0.02, 0.08)	γ	Normal	$(0, \infty)$	(0.9, 0.5)
σ_J	Tr. Normal	$(0, \infty)$	(0.05, 0.1)	$\bar{\delta}$	Tr. Normal	$(0, \infty)$	(0.15, 0.05)
k	Tr. Normal	$(0, \infty)$	(3.0, 4.0)	σ_δ	Tr. Normal	$(0, \infty)$	(0.25, 0.1)
$\bar{V}$	Tr. Normal	$(0, \infty)$	(0.05, 0.06)	α	Tr. Normal	$(0, \infty)$	(2.5, 1.5)
σ_v	Tr. Normal	$(0, \infty)$	(0.25, 0.25)	β	Tr. Normal	$(0, \infty)$	(4.0, 3.0)
ρ_v	Tr. Normal	$[-1, 1]$	(-0.7, 0.5)	λ_∞	Tr. Normal	$(0, \infty)$	(2, 0.8)
μ_v	Tr. Normal	$(0, \infty)$	(0.02, 0.1)				

Table 3.3: Priors specification.

In order to appreciate the efficiency of our estimation method, we can have a look at the acceptance rates related to the moving step in Figure 3.5. The star-line indicates the acceptance rates from the first stage and the circle line those from the second stage. We clearly see that the acceptance rates remain high during both the first and second stage. Furthermore, as proved in Fulop and Li (2019), the number of density-bridging iterations is much smaller in the second stage with respect to the first stage.

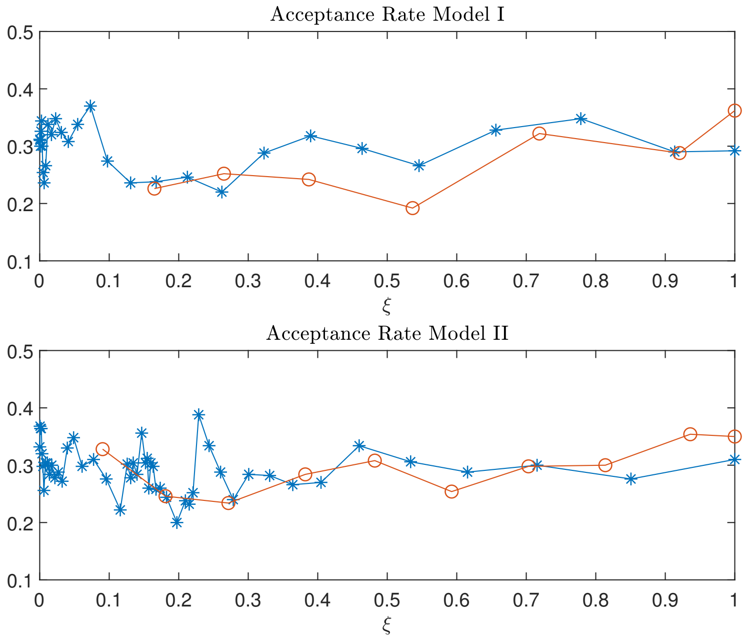

Figure 3.5: The figure plots the last acceptance rate in moving steps at each density-bridging iteration with respect to ξ_i for model I (upper) and model II (bottom). In the algorithm, ξ is automatically selected using a grid search approach. The blue line refers to the first stage, while the orange line refers to the second stage.

Figures 3.6 and 3.7 plot the filtered volatility, convenience yield and jump intensity obtained by running our smooth particle filter (see Appendix A) at the posterior mean. In particular, volatility is quite persistent and in periods when prices fall down we observe a rise in volatility according to the well known leverage effect. The convenience yield is moving in the same direction of oil prices, which is consistent with its definition and provides a clear economic intuition. A large amount of literature devoted to commodities provides a confirmation and an explanation of this behavior, we just mention Alquist *et al.* (2014) and Lautier (2009) among many other contributions on this topics, some based on the Normal Backwardation Theory (Litzenberger and Rabinowitz, 1995), some on the Theory of Storage (Casassus *et al.*, 2005). The intensity process presents a self-exciting behaviour; this will be confirmed later when focusing on the parameters estimates.

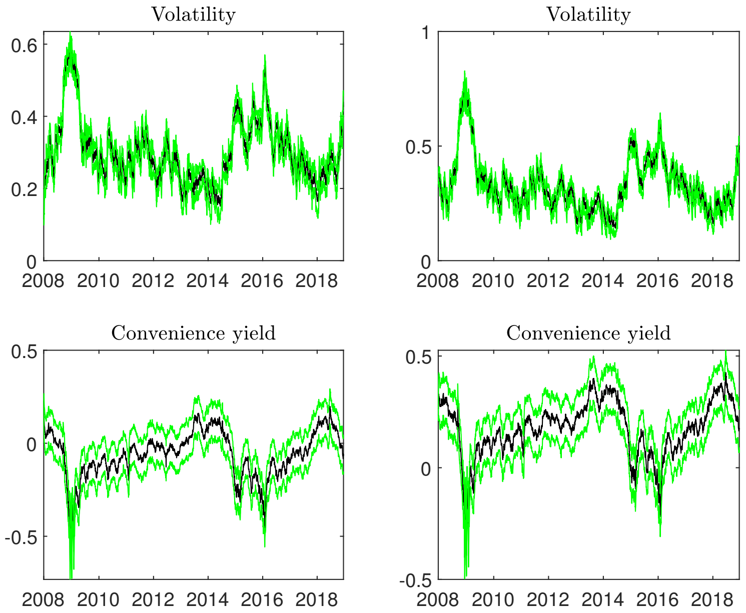

Figure 3.6: The figure presents the filtered volatility and convenience yield from model I (left) and II (right). The posterior mean and $(5, 95)\%$ percentiles are reported at each time point.

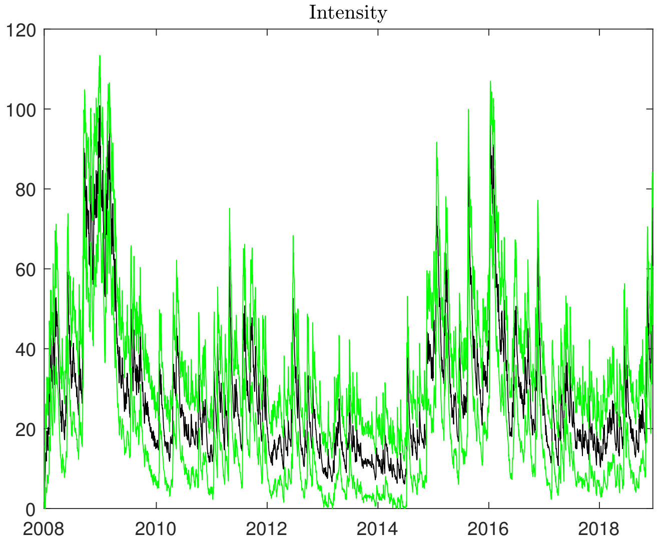

Figure 3.7: The figure presents the filtered jump intensity from model I. The posterior mean and $(5, 95)\%$ percentiles are reported at each time point.

Table 3.4 presents the parameter estimates for the two models obtained with the two-stage SMC sampler. As a first remark, when self-excitation is considered, volatility is less persistent. Indeed, k is equal to 9.2692 (2.1103) in Model I and 2.9157 (0.4418) in Model II. One possible explanation for this phenomenon is that much of the variation is due to the self-exciting jump intensity, which reduces the contribution of volatility. However, the vol-of-vol parameter σ_v is equal to 0.3566 (0.0960) in Model I and 0.8498 (0.0656) in Model II, confirming that volatility dynamics in oil market is clearly stochastic. Second, in line with the previous literature (Larsson and Nossman, 2011; Brooks and Prokopczuk, 2013), we find evidence of volatility jumps. In particular, the estimate for μ_v is 0.0223 (0.0063) in Model I and 0.0311 (0.0089) in Model II. Third, the posterior mean of the parameter controlling the self-exciting effect, α, is equal to 23.4601 (4.1515); then α is well identified and constitutes a key feature of the jump dynamics in the oil market. Fourth, the convenience yield dynamics is pretty much the same within the two models. The parameters are well identified and the convenience yield risk premium is statistically significant, which confirms previous studies in the literature.

Furthermore, it is possible to compare the models by looking at the marginal likelihood and the log-Bayes factor. In both cases we can say that Model I performs better than Model II. For instance, the log-Bayes factor[3] of Model I with respect to Model II is 9.1271, which means that Model I is definitely better than Model II in fitting the data.

[3] For any two given models, M_1 and M_2, if the value of the log-Bayes factor is between 0 and 1.1, M_1 is barely worth mentioning; if it is between 1.1 and 2.3, M_1 is substantially better than M_2; if it is between 2.3 and 3.4, M_1 is strongly better than M_2; if it is between 3.4 and 4.6, M_1 is much strongly better than M_2; and if it is larger than 4.6, M_1 is definitely better than M_2.

	Model I		Model II	
Θ	Mean	Std	Mean	Std
μ	0.1451	0.0944	0.0109	0.0691
μ_J	0.0022	0.0047	0.0118	0.0827
σ_J	0.0408	0.0038	0.1104	0.0537
k	9.2692	2.1103	2.9157	0.4418
$\bar{V}$	0.0131	0.0067	0.1309	0.0134
σ_v	0.3566	0.0960	0.8498	0.0656
ρ_v	-0.7628	0.1645	-0.4621	0.0414
γ	0.6708	0.0041	0.6721	0.6721
$\bar{\delta}$	0.2304	0.0086	0.2639	0.0079
σ_δ	1.7690	0.0183	1.7646	0.0197
β	30.0566	3.7743	(—)	(—)
α	23.4601	4.1515	(—)	(—)
λ_∞	1.3921	0.0079	8.0900	0.0518
φ_δ	0.1272	0.0052	0.1500	0.0048
μ_v	0.0223	0.0063	0.0311	0.0089
MLLH	4.1901e+4		4.1550e+4	

Table 3.4: Parameter estimates for model I and II. For each parameter, the posterior mean and standard deviation are reported. The last row reports the marginal log-likelihood estimated with the smooth particle filter.

Now, we can provide some additional insights regarding the convenience yield dynamics and the forecasting ability of the models in terms of futures pricing. In particular, with this kind of models it is common to observe a lack of fit in the filtered convenience yield process (Carmona and Ludkovski, 2004). To check if our approach could give a reliable estimate of this variable, we follow Carmona and Ludkovski (2004) and compute the implied convenience yield by using the estimated parameters together with the price of traded futures contracts $F(t, T_i)$. Then, we compare the filtered convenience yield from our smooth particle filter with the implied convenience yield computed. The result of this simple computation is shown in Figure 3.8 and confirms that our filtering method is able to well reproduce the convenience yield dynamics implied from data (in particular for longer maturities).

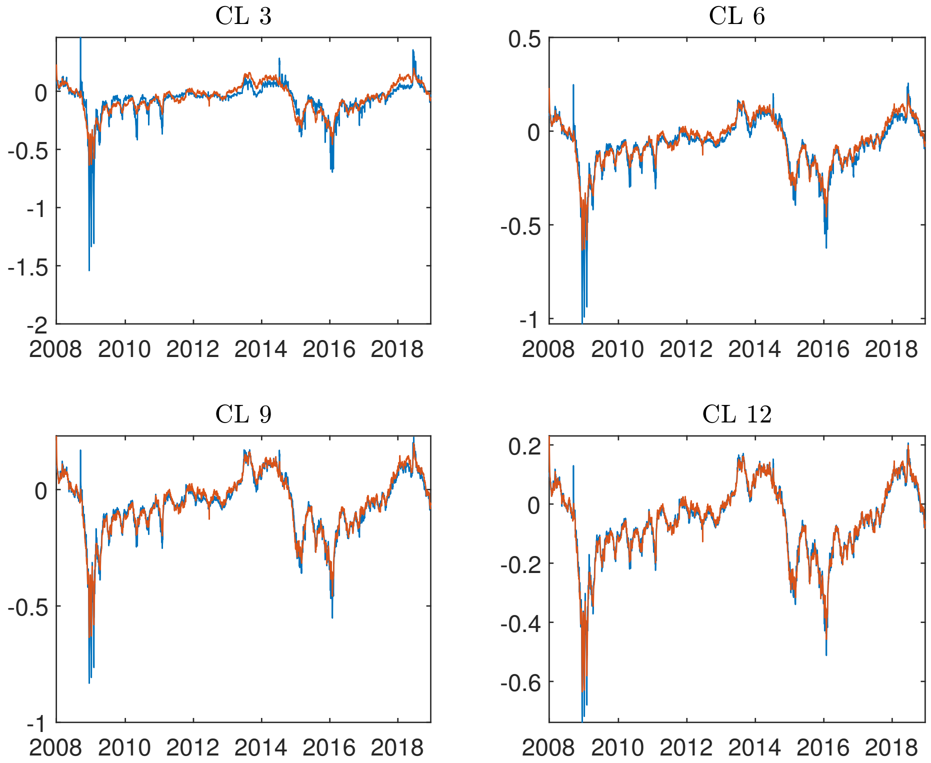

Figure 3.8: The figure compares the filtered convenience yield (orange) against the implied convenience yield (blue) computed using futures contracts with $\tau = \{3, 6, 9, 12\}$ and the posterior mean parameter estimates from Model I.

Second, we check the pricing errors of futures contracts when Model I and Model II are considered. For example, in Figure 3.9, we can see how the models behave in terms of futures pricing out of the sample, i.e. from 02/01/2019 to 17/01/2020 (263 observations). To this end we run our smooth particle filter on the new data by using the parameters obtained during the previous estimation and then we filter out the convenience yield. As we can see the results are good with both models, with better performances for Model I.

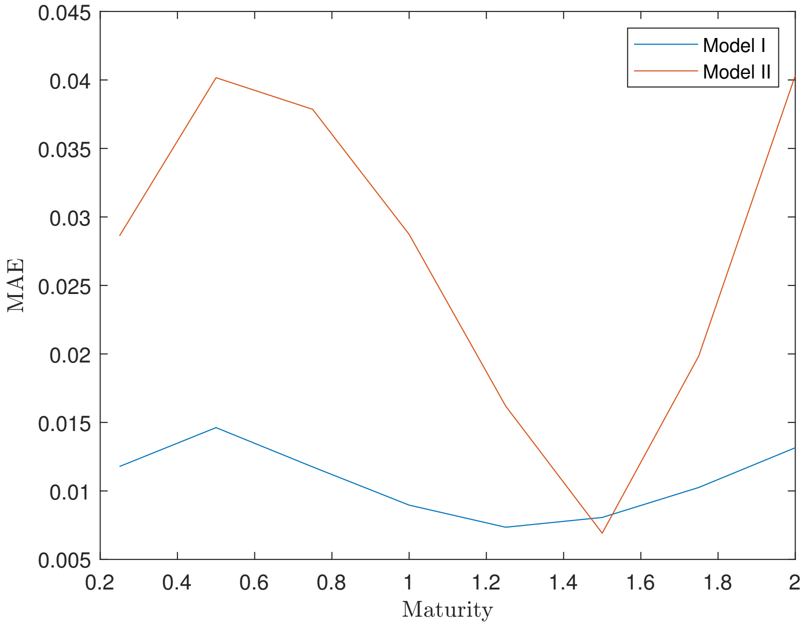

Figure 3.9: The figure presents the out of sample Mean Absolute Error (MAE) in terms of pricing futures contracts with Model I and Model II.

3.5 An Optimal Dynamic Hedging Strategy

Hedging in crude oil market is an important issue not only for producers, but also for energy traders and investors (Billio *et al.*, 2018). Indeed, price fluctuations lead to an increase in volatility, and so the risk coming from investing in the spot market needs to be mitigated. In particular, the natural way to hedge a long (short) position in the spot market is to sell (buy) a certain number of futures contracts. The quantity of futures needed to cover a spot position is called the *hedge ratio*. The determination of the optimal hedge ratio depends on the chosen objective function. In the literature, the hedge ratio is modeled as a time-varying variable (see Kroner and Sultan, 1993; Alizadeh *et al.*, 2008; Chang *et al.*, 2011; Liu *et al.*, 2014; Billio *et al.*, 2018; Batten *et al.*, 2019), which minimizes the variance of the portfolio $\Pi = S - hF$, where h denotes the number of futures. The optimal solution for h according to this criterion is then given by:

$$h^{\star}_{t-1} = \frac{\mathrm{Cov}_t(S, F)}{\mathrm{Var}_t(F)}. \tag{3.24}$$

At this point, it is important to remark that our approach is different from the approach usually considered in the literature. In particular, the most popular choice is given by discrete time models like multivariate GARCH models (Chang *et al.*, 2011; Batten *et al.*, 2019) and Markov switching models (Alizadeh *et al.*, 2008; Billio *et al.*, 2018). On the other hand, continuous-time models have received much less attention and one example of such an approach is given by Liu *et al.* (2014). In their paper, related to industrial metals' market, they specify one dynamics for the spot price and one for the futures and then the estimation is carried out separately. However, our model provides a direct link between spot and futures, meaning that in our estimation framework we cannot avoid no arbitrage issues and we need to resort to a risk-neutral argument. In view of possible extensions of the present estimation method including different derivatives contracts, European options for example, the risk-neutral approach represents the most convenient and natural modelling framework. In the present setting, the risk-neutral approach provides the necessary consistency relations between spot and futures prices.

Now, let us properly specify the hedging portfolio. In particular, the log-spot price is given by:

$$dx_t = \left(r - \frac{1}{2}V_t - \lambda_t \mathbb{E}\left[e^{J_x} - 1\right] - \delta_t\right) dt + \sqrt{V_t} dW_{x,t} + dJ_{x,t}.$$

Moreover the latent states are described by the following equations:

$$\begin{aligned} dV_t &= k(\bar{V} - V_t)dt + \sigma_v \sqrt{V_t}\left(\rho_v dW_{x,t} + \sqrt{1-\rho_v^2} dW_{v,t}\right) + dJ_{v,t}, \\ d\delta_t &= \gamma(\bar{\delta} - \delta_t)dt + \sigma_\delta dW_{\delta,t}, \\ d\lambda_t &= \beta(\lambda_\infty - \lambda_t)dt + \alpha dN_t. \end{aligned}$$

Then, By applying Itô's lemma on the log-futures pricing function $f(X, V, \lambda, \delta, t, T) = x_t + A(\tau) + C(\tau)\delta_t$ we get the dynamics of Y_t:

$$\begin{aligned} dY_t = &\left[r - \frac{1}{2}V_t - \lambda_t \mathbb{E}\left[e^{J_x} - 1\right] - \delta_t - A'(\tau) - C'(\tau)\delta_t + C(\tau)\gamma(\bar{\delta} - \delta_t)\right] dt \\ &+ \sqrt{V_t} dW_{x,t} + C(\tau)\sigma_\delta dW_{\delta,t} + dJ_{x,t} \end{aligned}$$

Then, the Minimum Variance (MV) hedging ratio is given by:

$$h^\star = \frac{V_t + (\mu_J^2 + \sigma_J^2)\lambda_t}{V_t + C^2(\tau)\sigma_\delta^2 + (\mu_J^2 + \sigma_J^2)\lambda_t}. \tag{3.25}$$

Additional details are given in subsection 3.5.2 and in Appendix B, where we also introduce a skewness-type objective function.

3.5.1 Empirical Application

In our application we follow previous studies (e.g. Liu *et al.*, 2014; Billio *et al.*, 2018) and consider as hedging instrument the most liquid futures, i.e. the futures with the shortest maturity, which in our data set are the CL3 contracts. Thus, we compute a daily hedge ratio according to the MV approach. In Figure 3.10 the optimal time varying hedge ratio $h_t^\star$ computed with Model I and Model II is shown. The time varying hedge ratio indicates that the portfolio should be frequently re-balanced as the market conditions change. In particular, during the global financial crisis (2008-2010) we observe a positive jump in the hedge ratio, which means that investors are more cautious and they prefer to hedge more. Then, from 2011 to 2014 the market is less volatile and the hedge ratios are smaller providing to the hedgers a lower exposition on futures. Finally, from 2015-2017 we observe again an increase in the optimal hedging strategy due to another period of market turmoil. Then, we provide some descriptive statistics about the hedge ratios in Table 3.5.

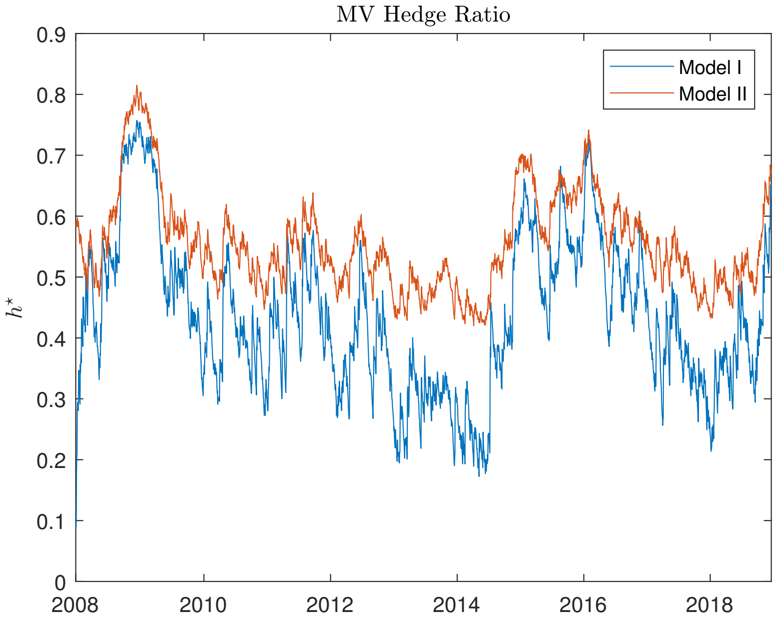

Figure 3.10: The figure presents the optimal MV hedge ratio time series for Model I (blue) and Model II (orange).

	Mean	Std Dev	Skewness	Kurtosis	Min	Max
MV Hedge I	0.4326	0.1250	0.4134	2.7714	0.0851	0.7571
MV Hedge II	0.5552	0.0801	0.9150	3.6083	0.4199	0.8149

Table 3.5: MV Hedge ratio statistics.

In the literature it is common to evaluate a particular hedging strategy in terms of variance reduction and utility improvements with respect to the un-hedged position (Kroner and Sultan, 1993; Alizadeh *et al.*, 2008; Batten *et al.*, 2019). Therefore, we consider the following measure of hedging effectiveness:

$$HE_1 = 1 - \left[\frac{\mathrm{Var}(\Pi_h)}{\mathrm{Var}(\Pi_{un})} \right], \tag{3.26}$$

where the hedging portfolios Π_h are computed by using percentage log-returns of spot and futures and the corresponding optimal hedge ratios. Hence, we are evaluating the variance reduction with respect to the un-hedged portfolio Π_{un}, which is composed only by the spot position. If $HE_1 = 0$ we do not reduce risk at all, whereas $HE_1 = 1$ imply a 100% reduction in the variance. According to our previous findings about the hedge ratio variability, we consider the whole sample and two different sub-periods: 2008-2010 and 2015-2017. In this way we are able to assess the hedging performances related to specific turbulent periods. The results are shown in Table 3.6.

Interval	Model I	Model II
2008-2018	0.6030	0.5554
2008-2010	0.6394	0.6117
2015-2017	0.6568	0.6040

Table 3.6: The table presents the variance reduction with respect to the un-hedged portfolio.

Overall, we obtain clear improvements with respect to the un-hedged position. For instance, during the whole sample, we reduce the portfolio variance by 60.30% with Model I and by 55.54% with model II. Moreover, even during periods of financial turmoil we are able to reduce significantly the variance of the portfolio. This confirms the importance of hedging for producers and investors.

According to Alizadeh *et al.* (2008), hedging effectiveness is more appropriately assessed by considering the economic benefits from hedging using the hedger's utility function. Then, if ξ represents the risk aversion of an investor and R_h is the return on the hedged portfolio, the expected utility function is given by

$$\mathbb{E}[U(R_h)] = \mathbb{E}[R_h] - \xi \text{Var}[R_h]. \tag{3.27}$$

By assuming that expected returns from the hedged portfolio are equal to zero and the degree of risk aversion is 4[4], we compute the realized utility for Model I, Model II and the un-hedged portfolio in Table 3.7.

Interval	Model I	Model II	Un-hedged
2008-2018	-9.5137	-10.6541	-23.9646
2008-2010	-19.3052	-20.7915	-53.5398
2015-2017	-9.3984	-10.8453	-27.3873

Table 3.7: The table presents the realized MV utility for Model I, Model II and the un-hedged portfolio.

Now, some comments are needed. First, by adopting the optimal hedging strategy we obtain substantial utility improvements with respect to the un-hedged position, no matter which sub-period is considered. Second, during the global financial crisis we get the worst results and this is due to the increased portfolio variance. Third, Model I is performing better than Model II, both regarding variance reduction and utility improvement, by confirming the importance of a more elaborate jump structure with respect to the standard Poisson framework with constant intensity.

In the next subsection we are going to explore if the inclusion of high order effects in the objective function could provide some improvements in the present hedging application.

3.5.2 Higher Order Hedging

The evidence of jumps and stochastic volatility in the oil market raises a question about the adequacy of a minimum variance objective function. Indeed, by adopting the MV approach we are not fully exploiting the distributional properties of these models, i.e. we are neglecting higher order effects, which could influence the hedging ratio. For instance,

[4] These assumptions are in line with most empirical studies as Kroner and Sultan (1993); Alizadeh *et al.* (2008).

the relevance of skewness in characterizing risk preferences has been pointed out in a rich amount of literature: we mention the papers by Post *et al.* (2008), Chiu (2010), Dahlquist *et al.* (2017) and Kraus and Litzenberger (1976). Therefore, we consider the minimization of the following objective function:

$$\min_{h}[\mathrm{Var}_t(\Pi_t) - \eta \mathrm{Asy}_t(\Pi_t)], \tag{3.28}$$

where, $\Pi_t = X_t - hY_t$ is the portfolio formed by $X_t = \log(S_t)$ and $Y_t = \log(F_{t,\tau})$; η is a constant risk-aversion parameter and $\mathrm{Asy}[x] = \mathbb{E}[(x - \mathbb{E}[x])^3]$.

In order to compute the optimal hedging ratio we need $\mathrm{Var}_t[\Pi_t]$ and $\mathrm{Asy}_t[\Pi_t]$. Hence, in analogy with Liu *et al.* (2014) we can proceed by computing the instantaneous conditional moments. The detailed derivation of the hedging ratio is given in Appendix B. Once we have obtained the expression for the hedging portfolio we take the first derivative with respect to h and set it equal to zero. In the end we get two solutions for $h^\star$:

$$\begin{aligned} h^\star = \pm \frac{1}{2\left(3\eta\lambda_t\mu_J^3 + 9\eta\lambda_t\mu_J\sigma_J^2\right)} \Big(&\left(2C^2\sigma_\delta^2 - 6\eta\lambda_t\mu_J^3 - 18\eta\lambda\mu_J\sigma_J^2 + 2\lambda_t\mu_J^2 + 2\lambda_t\sigma_J^2 + 2V\right)^2 \\ &- 4\left(3\eta\lambda_t\mu_J^3 + 9\eta\lambda_t\mu_J\sigma_J^2\right)\left(3\eta\lambda_t\mu_J^3 + 9\eta\lambda_t\mu_J\sigma^2 - 2\lambda_t\mu_J^2 - 2\lambda_t\sigma_J^2 - 2V_t\right) - 2C^2\sigma_\delta^2 + 6\eta\lambda_t\mu_J^3 \\ &+ 18\eta\lambda_t\mu_J\sigma_J^2 - 2\lambda_t\mu_J^2 - 2\lambda_t\sigma_J^2 - 2V_t\Big)^{1/2}. \end{aligned} \tag{3.29}$$

Since crude oil spot and futures are positively correlated, from now on, we shall consider only the positive solution of (3.29). As in the previous section, we compute daily hedge ratios according to our optimal strategy. Nevertheless, in Figure 3.11 we observe a surprising result: the MV and the Asy-Variance hedging strategies give substantially the same hedge ratio. This mean that the contribution given by the third-order moment is negligible in computing the optimal hedging strategy.

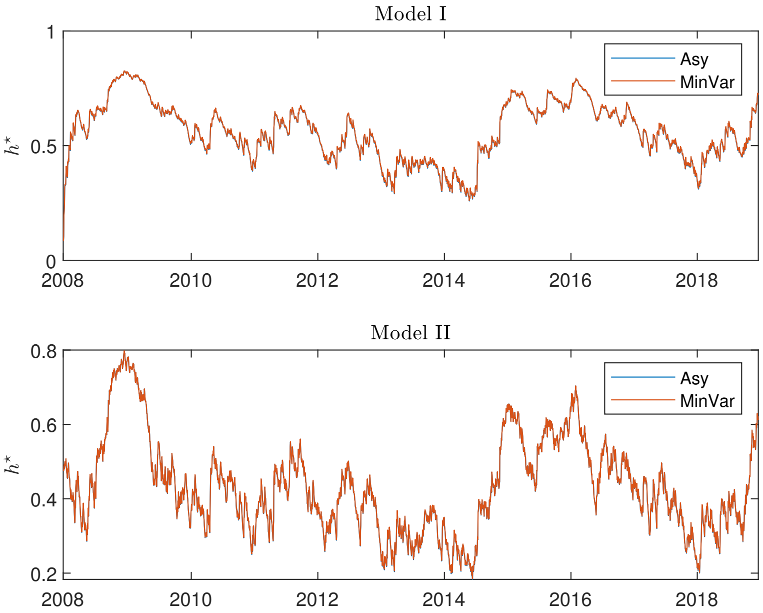

Figure 3.11: The figure presents the optimal MV and Asy-Variance hedge ratios time series for Model I (top) and Model II (bottom).

Given this result, we include the skewness instead of the third order central moment; then the optimization problem reads as follows

$$\min_h [\mathrm{Var}_t(\Pi_t) - \eta \mathrm{Skew}_t(\Pi_t)], \tag{3.30}$$

where,

$$\mathrm{Skew}(x) = \frac{\mathbb{E}[(x - \mathbb{E}[x])^3]}{\mathrm{Var}(x)^{3/2}}.$$

In this way we are not able to get a closed-form solution, so a numerical minimization method is adopted. Hence, the hedge ratio obtained numerically is compared to the MV hedging ratio in Figure 3.12, from which it is clear that the contribution of skewness is not negligible. This intuition will be formally tested empirically in what follows, where we denote the Variance-Skewness hedging by VS.

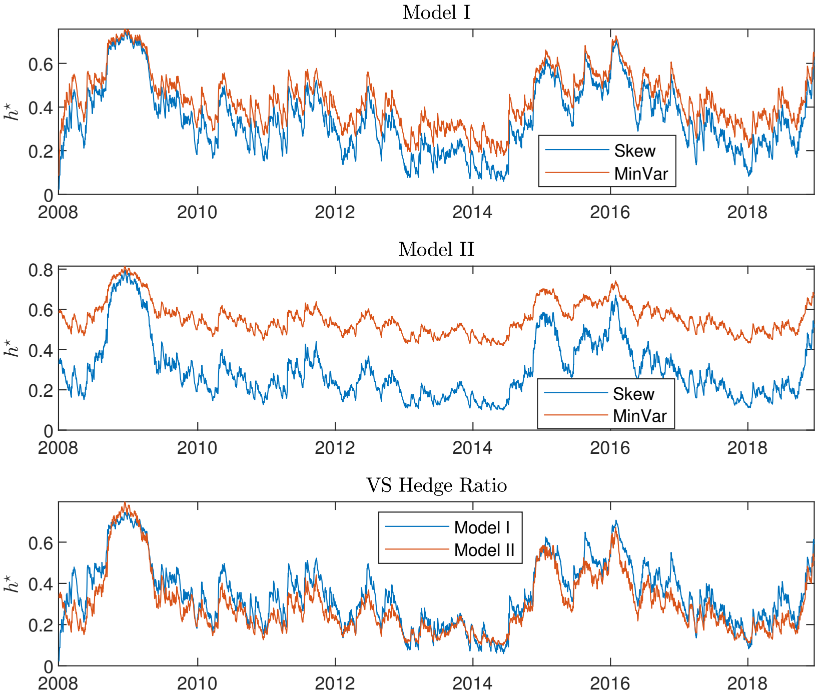

Figure 3.12: The figure presents the optimal MV and VS hedge ratios time series for Model I in the top panel and Model II in the middle panel. The bottom panel plots the VS hedge ratios for Model I (blue) and Model II (orange).

Now, in order to investigate the performance of this new strategy we need to consider a metric which is coherent within the optimizations goals. Then, we resort again to the realized utility:

$$\mathbb{E}[U(R_h)] = -\text{Var}[R_h] + \eta \text{Skew}[R_h], \tag{3.31}$$

where, η is the risk aversion parameter related to skewness and it is set equal to 4. The result of this computation is given in Table 3.8.

Interval	Model I	Model II	Un-hedged
2008-2018	-1.0730	-1.5485	-5.5189
2008-2010	-1.9980	-2.5387	-12.6846
2015-2017	-1.9193	-2.2930	-5.7923

Table 3.8: The table presents the realized VS utility for Model I, Model II and the un-hedged portfolio.

From the results we infer that the realized utility is giving good performances. In particular, Model I is performing better than Model II in every scenario. Moreover, both models implied utilities achieve better results with respect to the un-hedged position. The explanation stems from the fact that positive skewness is increasing the utility of the investor.

In order to further investigate this phenomenon we test the MV and VS hedging strategies out of sample, i.e. we consider 263 observations from 02/01/2019 to 17/01/2020. Hence, we filter out the state variables by running our smooth particle filter by using the parameters estimated in the sample. Then, we compute the realized utility for each strategy. The results are shown in Table 3.9. By this way we can appreciate the performance of each strategy with respect to the un-hedged position, by matching their respective optimization goals.

Utility	Model I	Model II	Un-hedhged
MV	-6.0292	-4.0241	-17.9647
VS	-0.9270	-1.1730	-3.9147

Table 3.9: The table presents the out of sample MV and VS realized utility for Model I and Model II. The realized utility of the un-hedged portfolio is also reported.

The results indicate that both strategies give better results with respect to the un-hedged portfolio. In particular, we observe that Model I performs better than Model II when the VS utility is considered.

Finally, as a robustness check we estimate again the models on a small sub-period and test the out-of-sample realized utility as before. To perform this task we estimate Model I and Model II using 200 days from 2008-2009; then we form the hedging portfolios by considering the MV and the VS strategies and we calculate their associated realized utility out-of-sample on additional 100 days. From the results in Table 3.10 we infer that Model I performs better with respect to Model II, no matter which utility is considered. As a general remark we confirm the well-known intuition that hedging during turbulent period is very important, as we can see from the performance of the un-hedged portfolios.

Utility	Model I	Model II	Un-hedhged
MV	-49.3415	-52.9350	-142.1638
VS	-10.2988	-10.6385	-33.9919

Table 3.10: The table presents the out of sample MV and VS realized utility for Model I and Model II during the global financial crisis. The realized utility of the un-hedged portfolio is also reported.

By resuming the previous results, it seems that the inclusion of high order effects lead to substantial improvements in the present hedging application with respect to the usual MV approach. Moreover, the model with self-exciting jump intensity outperforms the model with constant jump intensity.

3.6 Concluding Remarks

In this chapter we propose a jump-diffusion model for oil price dynamics including self-exciting effects. Our model includes a stochastic dynamics for both the volatility and the yield coefficient. We provide an estimation method of Bayesian type based on a suitable adaptation of a SMC method proposed by Fulop and Li (2019). Our results show that the introduction of a jump intensity of Hawkes type improves the forecasting ability of this model with respect to a model without self-exciting effects. Moreover its affine feature allows to compute explicitly futures contracts prices by providing a setting for accurate parameters estimation. Finally we compute an optimal hedging strategy based on futures trading. This optimal hedging strategy is obtained not only by minimizing variance, but by taking into account higher order moments, i.e. skewness. This optimal hedging strategy exhibits some interesting features and gives to the hedger better results with respect to the most popular MV approach. Furthermore, from the empirical application we infer that Model I, equipped with the self-exciting component, outperforms Model II in terms of hedging effectiveness and realized utility.

Chapter 4

Bayesian Filtering of Wishart Option Pricing Models

In this chapter we focus on the estimation of Wishart-type stochastic volatility models by exploiting state of the art SMC methods.

4.1 Introduction

Good models of time-varying covariance matrices are of central importance in finance for derivative pricing, portfolio allocation and modelling economic and financial uncertainty in general. A particularly interesting class of models are continuous-time Wishart models (see Da Fonseca *et al.*, 2008 and Gruber *et al.*, 2020) allowing for flexible forms of variance and covariance dynamics while guaranteeing positive semi-definiteness. However, economic and financial data is typically observed in discrete time, making the econometric analysis of continuous-time models difficult. The common practice in the literature is to take a simple Euler discretization to the continuous system and directly estimate the resulting discrete time state-space model (see for instance Eraker *et al.* (2003) for MCMC based approaches and Fulop *et al.* (2015) for SMC based estimation). However such methods face two issues: first, it is hard to control the discretization bias due to non-linearity and the time elapsed between observations; second, there is no generic method that provides efficient likelihood estimators necessary for likelihood based inference.

In the framework of Wishart stochastic volatility models a simple Euler approximation between two observations does not guarantee positive definiteness making discretization bias a particularly severe issue (see Kang *et al.*, 2017). Then, We follow the prevailing data augmentation approach (see e.g. the survey in Sørensen (2009), or Durham and Gallant (2002)) filling in latent states at artificial time points between the observations, allowing us to control discretization bias. While most existing work is based on MCMC methods, we propose to employ SMC methods. Here the main difficulty to overcome is to provide an efficient proposal distribution over the path of the unobserved latent states between two observations. Standard filtering techniques, such as the bootstrap particle filter (Gordon *et al.*, 1993) or the locally optimal particle filter (Doucet *et al.*, 2000) do not suffice as they only condition on the past, while here the future observation is also informative on the latent path. Then, we propose to adapt the recent controlled SMC method of Heng *et al.* (2019) to tackle this problem, which allows us a generic approach to propagate information from the future.

The chapter is organized as follows. In Section 4.2 we introduce the Wishart stochastic volatility model and its associated Moment Generating Function (MGF). Then, in Section 4.3 we cast the model in a State-Space form by exploiting options portfolios as measurement equation. In Section 4.4 we propose a controlled SMC approach for state filtering and likelihood estimation. Finally, we provide some numerical results on simulated data in Section 4.5, while Section 4.6 gives some concluding remarks.

4.2 The Model

Assume the price process under the risk-neutral probability is given by:

$$\frac{dS_t}{S_t} = rdt + Tr[\sqrt{V_t} dZ_t], \tag{4.1}$$

where, $V_t \in \mathcal{S} \subset \mathbb{R}^{n \times n}$ is a symmetric positive definite matrix process described by the following SDE,

$$dV_t = (\beta Q^T Q + MV_t + V_t M^T)dt + \sqrt{V_t} dB_t Q + Q^T dB_t^T \sqrt{V_t}, \tag{4.2}$$

where, $\beta \in \mathbb{R}$, $Q, M \in \mathbb{R}^{n\times n}$ and B_t is a $n \times n$ standard Brownian motion. Moreover,

$$Z_t = B_t R + W_t \sqrt{I_n - RR^T}, \tag{4.3}$$

where, W_t is another $n \times n$ Brownian motion independent from B_t. Then, the MGF of $Y_t = \ln S_t$ is given by

$$\Psi(\tau;\gamma) = \mathbb{E}_t[e^{\gamma Y_T}] = \exp\left(\gamma Y_t + Tr[A(\gamma,\tau)V_t] + B(\gamma,\tau)\right), \tag{4.4}$$

where, $\gamma \in \mathbb{R}$, $A(\gamma,\tau) \in \mathbb{R}^{n\times n}$ and $B(\gamma,\tau) \in \mathbb{R}$. The solution is given by the following system of ODEs:

$$\begin{cases} \frac{\partial}{\partial\tau}A(\gamma,\tau) = 2A(\gamma,\tau)Q^TQA(\gamma,\tau) + A(\gamma,\tau)(M + \gamma Q^TR^T) + (M^T + \gamma RQ)A(\gamma,\tau) + \frac{\gamma}{2}(\gamma-1)I_n, \\ \frac{\partial}{\partial\tau}B(\gamma,\tau) = \beta Tr[A(\gamma,\tau)Q^TQ] + \gamma r, \end{cases} \tag{4.5}$$

with initial conditions $A(0) = 0$ and $B(0) = 0$. It is possible to find a closed-form solution by linearization method (Da Fonseca *et al.*, 2008):

$$A(\gamma,\tau) = C_{22}(\tau)^{-1}C_{21}(\tau), \tag{4.6}$$

where,

$$\begin{bmatrix} C_{11}(\tau) & C_{12}(\tau) \\ C_{21}(\tau) & C_{22}(\tau) \end{bmatrix} = \operatorname{expm}\left(\tau \begin{bmatrix} M + \gamma Q^TR & -2Q^TQ \\ \frac{\gamma(\gamma-1)}{2}I_n & -(M^T + \gamma R^TQ) \end{bmatrix}\right). \tag{4.7}$$

The other coefficient is given by

$$B(\gamma,\tau) = \tau\gamma r - \frac{\beta}{2}Tr\left[\ln C_{22}(\tau) - \tau C_{22}(\tau)\right], \tag{4.8}$$

where, the logarithm is computed in the matrix sense.

4.3 State-Space Model

In this section we construct a SSM where latent states are described by a Wishart process. In particular, we consider as measurement equation portfolios of weighted options as detailed in Orlowski (2019). This kind of contracts summarize information contained in the risk-neutral density and are linked to the state variables by a linear relation. This means that we can avoid to compute option prices, which are complex non-linear functions of state variables and parameters. Moreover, in state of the art affine models, option prices require the the numerical inversion of the characteristic function of log-returns, making the estimation routine unfeasible. In Appendix A we present this idea which builds on Bakshi and Madan (2000) and Feunou and Okou (2018). Then, by exploiting such contracts we define

$$y(t_p) = B'(\gamma,\tau) + Tr\left[A'(\gamma,\tau)V(t_p)\right] + u(t_p), \tag{4.9}$$

where, $p = 1,\dots,P$ and $B'(\gamma,\tau) \in \mathbb{R}$, $A'(\gamma,\tau) \in \mathbb{R}^{n\times n}$ are computable from the MGF slope of log-returns. For example, a realistic case considered in Orlowski (2019) is given by $\gamma = [0,\ 0.5,\ 1]$ and $\tau = [1/12,\ 6/12]$, then we can write

$$y_k(t_p) = B_k'(\gamma,\tau) + Tr\left[A_k'(\gamma,\tau)V(t_p)\right] + u_k(t_p),$$

where, $k = 1,\dots,6$ and $u_k(t_p) \sim \mathcal{N}(0,\sigma_k^2 I_{n_k})$. Therefore, $y_k(t_p) \in \mathbb{R}^{n_k}$, with $n_k = n_\gamma \times n_\tau$. The latent state is described by a continuous time Wishart process $V_t \in \mathcal{S} \subset \mathbb{R}^{n\times n}$ with dynamics given by Equation (4.2). We can approximate (4.2) using a discrete time truncated Euler scheme (see Kang *et al.*, 2017):

$$V_t = \left(V_{t-1} + (\beta Q^TQ + MV_{t-1} + V_{t-1}M^T)\Delta t + \sqrt{\Delta t}\sqrt{V_{t-1}}B_tQ + Q^TB_t^T\sqrt{V_{t-1}}\sqrt{\Delta t}\right)^+. \tag{4.10}$$

where, A^+ denotes the positive part of a symmetric matrix A: we set $A^+ = P\text{diag}(\lambda_1^+,\dots,\lambda_n^+)P^T$, where P is the matrix formed by eigenvectors. To make V_t positive semi-definite, we take the positive part at each time grid using the above decomposition.

4.3.1 Conditional moments

In view of filtering we need expressions for the conditional mean and conditional variance of V_t given V_{t-1}. To this end we use the `vec` operator and consider the formulae from

Gruber *et al.* (2020):

$$\mu(\hat{V}_{t-1}) = \beta\bar{\mu} + \Phi V_{t-1}\Phi^T, \tag{4.11}$$

$$\Sigma(\hat{V}_{t-1}) = (I_n + \mathcal{K}_{n_v})\left(\Phi V_{t-1}\Phi^T \otimes \bar{\mu} + \beta\bar{\mu} \otimes \bar{\mu} + \bar{\mu} \otimes \Phi V_{t-1}\Phi^T\right) \tag{4.12}$$

with,

$$\bar{\mu} = -\frac{1}{2}C_{12}C_{11}^T, \quad \Phi = \text{expm}(\Delta t M),$$
$$C = \text{expm}\left(\Delta t \begin{bmatrix} M & -2Q^TQ \\ 0 & -M^T \end{bmatrix}\right) = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$$

where, $C_{11}, C_{12}, C_{21}, C_{22}$ are $n \times n$ matrices and $n_v = n^2$, such that $\mathcal{K}_{n_v}$ is a $n_v \times n_v$ commutation matrix. Then, if we define $\hat{V}_t = \text{vec}(V_t)$ and $\hat{\mu}(\hat{V}_{t-1}) = \text{vec}(\mu(\hat{V}_{t-1}))$, we get

$$f(\hat{V}_{t-1}, d\hat{V}_t) = \mathcal{N}\left(\hat{\mu}(\hat{V}_{t-1}), \Sigma(\hat{V}_{t-1})\right) d\hat{V}_t. \tag{4.13}$$

Therefore, the vectorized hidden state $\hat{V}_t$ is a $n_v \times 1$ vector.[1]

4.3.2 Higher frequency state-space model

In order to better approximate (4.2) we can sample additional L points between each value. In this case we write the transition kernel (4.13) as follows,

$$f_{p,l}(\hat{V}(t_{p,l-1}), d\hat{V}(t_{p,l})) = \mathcal{N}\left(\hat{V}(t_{p,l});\ \hat{\mu}(\hat{V}(t_{p,l-1})),\ \Sigma(\hat{V}(t_{p,l-1}))\right) d\hat{V}(t_{p,l}) \tag{4.14}$$

for $l = 1, \ldots, L$, $p = 1, \ldots, P$ and $t_{p-1} = t_{p,0} < t_{p,1} < \ldots < t_{p,L} = t_p$ are intermediate points on the interval $[t_{p-1}, t_p]$. For simplicity consider $t_{p,l} = t_{p-1} + l \times \Delta p$, with $\Delta p = (t_p - t_{p-1})/L$.

In this case the marginal likelihood is given by

$$P(y_k(t_1), \ldots, y_k(t_p)) = \int \nu(d\hat{V}(0)) \prod_{p=1}^{P}\prod_{l=1}^{L} f_{p,l}(\hat{V}(t_{p,l-1}), d\hat{V}(t_{p,l})) \prod_{p=1}^{P}\prod_{l=1}^{L} g_{p,l}(\hat{V}(t_{p,l})), \tag{4.15}$$

[1] For the implementation, since V_t is symmetric, we are going to consider only its upper diagonal, which means that $\hat{V}_t$ is a $n_z \times 1$ vector, with $n_z = n(n+1)/2$.

where,

$$g_{p,l}(\hat{V}_{p,l}) = \begin{cases} g(\hat{V}(t_p), y_k(t_p)), & \text{if} \quad l = L, \\ 1, & \text{if} \quad l \neq L, \end{cases} \tag{4.16}$$

for $p = 1, \ldots, P$. By this way we ensure that at intermediate filling points the observation density is one. The latent state is also defined by an initial distribution

$$\nu(d\hat{V}(0)) = \mathcal{N}(\hat{\mu}(\hat{V}(0)), \Sigma(\hat{V}(0)))d\hat{V}(0), \tag{4.17}$$

where, we initialize V_0 to be the steady state V_∞, which can be computed by solving the Lyapunov equation $MV_\infty + V_\infty M^T = Q^T Q$.

4.4 Particle Filtering

We pursue the controlled SMC approach for state filtering and likelihood approximation (Heng *et al.*, 2019). Given a sequence of functions $\psi_0 : \mathbb{R}^{n_v} \to \mathbb{R}_+$ and $\psi_{p,l} : \mathbb{R}^{n_v} \to \mathbb{R}_+$ for $p = 1, \ldots, P$ and $l = 1, \ldots, L$ we define the twisted initial distribution and the twisted Markov transition kernel,

$$\nu^\psi(d\hat{V}(0)) = \frac{\nu(d\hat{V}(0))\psi_0(\hat{V}(0))}{\nu(\psi_0)}, \quad f^\psi_{p,l}(\hat{V}(t_{p,l-1}), d\hat{V}(t_{p,l})) = \frac{f(\hat{V}(t_{p,l-1}), d\hat{V}(t_{p,l}))\psi_{p,l}(\hat{V}(t_{p,l}))}{f(\psi_{p,l})(\hat{V}(t_{p,l-1}))}.$$

Notice that $\nu(\psi_0)$ is the expectation of ψ_0 with respect to ν, while $f(\psi_{p,l})(\hat{V}(t_{p,l-1}))$ is the conditional expectation of $\psi_{p,l}$ with respect to (4.14).

In order to approximate (4.15), we define the twisted potentials,

$$\begin{aligned} g^\psi_0(\hat{V}(0)) &= \frac{\nu(\psi_0)f(\psi_{0,1})(\hat{V}(0))}{\psi_0(\hat{V}(0))}, \\ g^\psi_{p,l}(\hat{V}(t_{p,l-1}), \hat{V}(t_{p,l})) &= \frac{g(\hat{V}(t_{p,l-1}), \hat{V}(t_{p,l}), y_k(t_{p,l}))f(\psi_{p,l+1})(\hat{V}(t_{p,l}))}{\psi_{p,l}(\hat{V}(t_{p,l-1}), \hat{V}(t_{p,l}))}, \; p = 1, \ldots, P, \; l = 1, \ldots, L-1 \\ g^\psi_{P,L}(\hat{V}(t_{P,L-1}), \hat{V}(t_{P,L})) &= \frac{g(\hat{V}(t_{P,L-1}), \hat{V}(t_{P,L}), y_k(t_{P,L}))}{\psi_{P,L}(\hat{V}(t_{P,L-1}), \hat{V}(t_{P,L}))}. \end{aligned}$$

Therefore the marginal likelihood can be written as follows

$$P(y_k(t_1), \ldots, y_k(t_p)) = \int \nu^\psi(d\hat{V}(0)) \prod_{p=1}^{P}\prod_{l=1}^{L} f^\psi_{p,l}(\hat{V}(t_{p,l-1}), d\hat{V}(t_{p,l})) \prod_{p=1}^{P}\prod_{l=1}^{L} g^\psi_{p,l}(\hat{V}(t_{p,l})),$$

The optimal policy $\psi^\star$ is given by the backward recursion

$$\begin{aligned}\psi^\star_{P,L}(\hat{V}(t_{P,L-1}),\hat{V}(t_{P,L})) &= g(\hat{V}(t_{P,L-1}),\hat{V}(t_{P,L}),y_k(t_{P,L})),\\ \psi^\star_{p,l}(\hat{V}(t_{p,l-1}),\hat{V}(t_{p,l})) &= g(\hat{V}(t_{p,l-1}),\hat{V}(t_{p,l}),y_k(t_{p,l}))f(\psi_{p,l+1})(\hat{V}(t_{p,l})),\\ \psi^\star_0(\hat{V}(0)) &= f(\psi_{0,1})(\hat{V}(0)).\end{aligned} \tag{4.18}$$

In our setting the optimal policy cannot be computed analytically. Then, in the following sub-section we consider a suitable approximation scheme.

4.4.1 Function classes

We consider

$$\begin{aligned}\psi_{p,l}(\hat{V}(t_{p,l-1}),\hat{V}(t_{p,l})) = &\exp\Big\{ -\hat{V}(t_{p,l})^T A_{p,l}\hat{V}(t_{p,l}) - \hat{V}(t_{p,l})^T b_{p,l}\\ &- \hat{V}(t_{p,l})^T c_{p,l}(\hat{V}(t_{p,l-1})) - d_{p,l}(\hat{V}(t_{p,l-1}))\Big\}\end{aligned} \tag{4.19}$$

for some $A_{p,l} \in \mathbb{R}^{n_v\times n_v}$, $b_{p,l} \in \mathbb{R}^{n_v}$, $c_{p,l}(\hat{V}(t_{p,l-1})) = C_{p,l}\hat{V}(t_{p,l-1})$, with $C_{p,l} \in \mathbb{R}^{n_v\times n_v}$, $d_{p,l}(\hat{V}(t_{p,l-1})) = \hat{V}(t_{p,l-1})^T D_{p,l}\hat{V}(t_{p,l-1}) + \hat{V}(t_{p,l-1})^T e_{p,l} + f_{p,l}$, with $D_{p,l} \in \mathbb{R}^{n_v\times n_v}$, $e_{p,l} \in \mathbb{R}^{n_v}$ and $f_{p,l} \in \mathbb{R}$.

Under this setting we derive the twisted objects. In the Appendix B we report the full calculations. For the initial point we consider

$$\psi_0(\hat{V}(0)) = \Big\{-\hat{V}(0)^T A_0\hat{V}(0) - \hat{V}(0)^T b_0 - f_0\Big\} \tag{4.20}$$

for some $A_0 \in \mathbb{R}^{n_v\times n_v}$, $b_0 \in \mathbb{R}^{n_v}$ and $c_0 \in \mathbb{R}$. Then,

$$\begin{aligned}\nu(\psi_0) = &\det(\Sigma(\hat{V}(0)))^{-\frac{1}{2}}\det(K_0)^{\frac{1}{2}}\exp\left\{-\frac{1}{2}\hat{\mu}^T(\hat{V}(0))\Sigma(\hat{V}(0))^{-1} - f_0\right\}\\ &\times\exp\left\{\frac{1}{2}\left(\Sigma(\hat{V}(0))^{-1}\hat{\mu}(\hat{V}(0)) - b_0\right)^T K_0\left(\Sigma(\hat{V}(0))^{-1}\hat{\mu}(\hat{V}(0)) - b_0\right)\right\},\end{aligned} \tag{4.21}$$

where, $K_0 := (\Sigma(\hat{V}(0))^{-1} + 2A_0)^{-1}$. Given the above solution for the expectation we can compute the twisted initial kernel

$$\nu^\psi(d\hat{V}(0)) = \mathcal{N}\left(\hat{V}(0);\, K_0(\Sigma(\hat{V}(0))^{-1}\mu(\hat{V}(0)) - b_0), K_0\right)d\hat{V}(0). \tag{4.22}$$

For the twisted Markov transition kernels we consider (4.19) and begin with the following expectation

$$\begin{aligned} f_{p,l}(\psi_{p,l})(\hat{V}(t_{p,l-1}), d\hat{V}(t_{p,l})) = {} & \det(\Sigma(\hat{V}(t_{p,l-1})))^{-\frac{1}{2}} \det(K_{p,l})^{\frac{1}{2}} \\ & \times \exp\left\{-\frac{1}{2}\hat{\mu}^T(\hat{V}(t_{p,l-1}))\Sigma(\hat{V}(t_{p,l-1}))^{-1}\hat{\mu}(\hat{V}(t_{p,l-1})) - d_{p,l}(\hat{V}(t_{p,l-1}))\right\} \\ & \times \exp\left\{\frac{1}{2}\left[\Sigma(\hat{V}(t_{p,l-1}))^{-1}\hat{\mu}(\hat{V}(t_{p,l-1})) - (b_{p,l} + c_{p,l}(\hat{V}(t_{p,l-1})))\right]^T\right. \\ & \left. K_{p,l}\left[\Sigma(\hat{V}(t_{p,l-1}))^{-1}\hat{\mu}(\hat{V}(t_{p,l-1})) - (b_{p,l} + c_{p,l}(\hat{V}(t_{p,l-1})))\right]\right\}, \end{aligned} \tag{4.23}$$

where, $K_{p,l} := (\Sigma(\hat{V}(t_{p,l-1}))^{-1} + 2A_{p,l})^{-1}$. Given the solution for the expectation we can compute the twisted Markov transition kernel

$$\begin{aligned} & f^{\psi}_{p,l}(\hat{V}(t_{p,l-1}), d\hat{V}(t_{p,l})) = \\ & \mathcal{N}\left(\hat{V}(t_{p,l});\ K_{p,l}(\Sigma(\hat{V}(t_{p,l-1}))^{-1}\hat{\mu}(\hat{V}(t_{p,l-1})) - (b_{p,l} + c_{p,l}(\hat{V}(t_{p,l-1})))), K_{p,l}\right) d\hat{V}(t_{p,l}). \end{aligned} \tag{4.24}$$

4.4.2 Tempered smoothing distribution

To approximate the optimal policy (4.18) we use regression methods. In order to perform this task we need good samples, which could be difficult to get in some scenarios. Indeed, this is due to the discrepancy between the observations and the parameter set. To alleviate this issue we will consider intermediate smoothing distributions defined by

$$p(d\hat{V}(t_{0:P}) \mid y_k(t_{1:P}), \lambda) = p(y_k(t_{1:P}) \mid \lambda)^{-1}\nu(d\hat{V}(0))\prod_{p=1}^{P}\prod_{l=1}^{L} f_{p,l}(\hat{V}(t_{p,l-1}), d\hat{V}(t_{p,l}))\prod_{p=1}^{P}\prod_{l=1}^{L} g_{p,l}(\hat{V}(t_{p,l}))^{\lambda}, \tag{4.25}$$

where, $\lambda \in [0, 1]$ and

$$p(y_k(t_{1:P}) \mid \lambda) = \int \nu(d\hat{V}(0))\prod_{p=1}^{P}\prod_{l=1}^{L} f_{p,l}(\hat{V}(t_{p,l-1}), d\hat{V}(t_{p,l}))\prod_{p=1}^{P}\prod_{l=1}^{L} g_{p,l}(\hat{V}(t_{p,l}))^{\lambda} d\hat{V}(t_{0:P}),$$

$$g_{p,l}(\hat{V}_{p,l}) = \begin{cases} g(\hat{V}(t_p), y_k(t_p)), & \text{if} \quad l = L \\ 1, & \text{if} \quad l \neq L \end{cases}$$

Note that $\lambda = 0$ corresponds to the latent dynamics and $\lambda = 1$ recovers the smoothing distribution of interest. To approximate (4.25) for each $\lambda \in [0, 1]$, we construct a proposal distribution

$$q^{\psi}(d\hat{V}(t_{0:P})) = \nu^{\psi}(d\hat{V}(0)) \prod_{p=1}^{P} \prod_{l=1}^{L} f^{\psi}_{p,l}(\hat{V}(t_{p,l-1}), d\hat{V}(t_{p,l})) \tag{4.26}$$

defined by a policy ψ which depends on coefficients $\beta = (A_{p,l}, b_{p,l}, C_{p,l}, D_{p,l}, e_{p,l}, f_{p,l})$, for $p = 0, \ldots, P$ and $l = 1, \ldots, L$. To construct a SMC method to target (4.25) using (4.26) we define

$$\begin{aligned} g^{\psi}_0(\hat{V}(0)) &= \frac{\nu(\psi_0) f(\psi_{0,1})(\hat{V}(0))}{\psi_0(\hat{V}(0))}, \\ g^{\lambda,\psi}_{p,l}(\hat{V}(t_{p,l-1}), \hat{V}(t_{p,l})) &= \frac{g(\hat{V}(t_{p,l-1}), \hat{V}(t_{p,l}), y_k(t_{p,l}))^{\lambda} f(\psi_{p,l+1})(\hat{V}(t_{p,l}))}{\psi_{p,l}(\hat{V}(t_{p,l-1}), \hat{V}(t_{p,l}))}, \, p = 1, \ldots, P, \, l = 1, \ldots, L-1 \\ g^{\lambda,\psi}_{P,L}(\hat{V}(t_{P,L-1}), \hat{V}(t_{P,L})) &= \frac{g(\hat{V}(t_{P,L-1}), \hat{V}(t_{P,L}), y_k(t_{P,L}))^{\lambda}}{\psi_{P,L}(\hat{V}(t_{P,L-1}), \hat{V}(t_{P,L}))}. \end{aligned}$$

Thus, we represent the intermediate smoothing distribution as

$$p(d\hat{V}(t_{0:P}) \mid y_k(t_{1:P}), \lambda) = g^{\psi}_0(\hat{V}(0)) \prod_{p=1}^{P} \prod_{l=1}^{L} g^{\lambda,\psi}_{p,l}(\hat{V}(t_{p,l-1}), \hat{V}(t_{p,l})) q^{\psi}(d\hat{V}(t_{0:P}))$$

In Algorithm 11 we present the resulting SMC.

Now, we outline how it is possible to update the tempering parameter λ. Let $0 = \lambda_0 < \lambda_1 < \ldots < \lambda_{I_1} = 1$. We initialize our SMC procedure at $\lambda_0 = 0$ and with $\psi(0)$ defined by the coefficients $\beta^{(0)}$ equal to zeros. Assume, for $i = 1, \ldots, I_1$, to have a policy $\psi(i-1)$ defined by

$$\beta^{(i-1)} = (A^{(i-1)}_{p,l}, b^{(i-1)}_{p,l}, f^{(i-1)}_{p,l})^{P,L}_{p,l=1,1}$$

such that the proposal $q^{\psi(i-1)}(d\hat{V}(t_{0:P}))$ is an approximation of $p(d\hat{V}(t_{0:P}) \mid y_k(t_{1:P}), \lambda_i)$. To construct an approximation $q^{\psi(i)}(d\hat{V}(t_{0:P}))$ of $p(d\hat{V}(t_{0:P}) \mid y_k(t_{1:P}), \lambda_i)$, we modify $\psi(i-1)$ using a policy $\phi(i)$ defined by

$$\tilde{\beta}^{(i)} = (\tilde{A}^{(i)}_{p,l}, \tilde{b}^{(i)}_{p,l}, \tilde{f}^{(i)}_{p,l})^{P,L}_{p,l=1,1},$$

with the update $\psi^{(i)}_{p,l} = \psi^{(i-1)}_{p,l} \cdot \phi^{(i)}_{p,l}$, or equivalently,

$$\beta^{(i)}_{p,l} = \beta^{(i-1)}_{p,l} + \tilde{\beta}^{(i)}_{p,l}.$$

Algorithm 11 Sequential Monte Carlo

```
Input: number of particles $N$, number of intermediate latent state points $L$, policy $\psi$ defined by coefficients $\beta$ and tempering level $\lambda$.

1. For time $t_{p,l} = t_0 = 0$ and for $n = 1, \dots, N$
    (a) set $K_0 := (\Sigma(\hat{V}(0))^{-1} + 2A_0)^{-1}$ and $K_{0,1} := (\Sigma(\hat{V}(0))^{-1} + 2A_{0,1})^{-1}$;
    (b) sample $\hat{V}(0)^n \sim \mathcal{N}\left(K_0(\Sigma(\hat{V}(0))^{-1}\mu(\hat{V}(0)) - b_0), K_0\right)$ ;
    (c) compute expectation
        $\nu(\psi_0) = \det(\Sigma(\hat{V}(0)^n))^{-\frac{1}{2}} \det(K_0)^{\frac{1}{2}} \exp\left\{-\frac{1}{2}\hat{\mu}^T(\hat{V}(0)^n)\Sigma(\hat{V}(0)^n)^{-1}\hat{\mu}(\hat{V}(0)^n) - f_0\right\}$
        $\times \exp\left\{\frac{1}{2}\left(\Sigma(\hat{V}(0)^n)^{-1}\hat{\mu}(\hat{V}(0)^n) - b_0\right)^T K_0 \left(\Sigma(\hat{V}(0)^n)^{-1}\hat{\mu}(\hat{V}(0)^n) - b_0\right)\right\}$;
    (d) compute conditional expectations
        $f(\psi_{0,1})(\hat{V}(0)^n) = \det(\Sigma(\hat{V}(0)^n))^{-\frac{1}{2}} \det(K_{0,1})^{\frac{1}{2}} \exp\left\{-\frac{1}{2}\hat{\mu}^T(\hat{V}(0)^n)\Sigma(\hat{V}(0)^n)^{-1}\hat{\mu}(\hat{V}(0)^n) - d_{0,1}(\hat{V}(0)^n)\right\}$
        $\times \exp\left\{\frac{1}{2}\left[\Sigma(\hat{V}(0)^n)^{-1}\hat{\mu}(\hat{V}(0)^n) - (b_{0,1} + c_{0,1}(\hat{V}(0)^n))\right]^T\right.$
        $\left. K_{0,1}\left[\Sigma(\hat{V}(0)^n)^{-1}\hat{\mu}(\hat{V}(0)^n) - (b_{0,1} + c_{0,1}(\hat{V}(0)^n))\right]\right\}$;
    (e) evaluate policy $\psi_0(\hat{V}(0)^n) = \exp\{-(\hat{V}(0)^n)^T A_0 \hat{V}(0)^n - (\hat{V}(0)^n)^T b_0 - f_0\}$;
    (f) compute weights $g_0^{\psi}(\hat{V}(0)^n) = \nu(\psi_0) f(\psi_{0,1})(\hat{V}(0)^n)/\psi_0(\hat{V}(0)^n)$;
    (g) normalized weights $W_0^n = g_0^{\psi}(\hat{V}(0)^n)/\sum_{m=1}^N g_0^{\psi}(\hat{V}(0)^m)$;
    (h) compute effective sample size $ESS_0 = \{\sum_{n=1}^N (W_0^n)^2\}^{-1}$;
    (i) sample ancestors $A_0^n \sim \mathcal{R}(W_0^1, \dots, W_0^N)$;
2. For time $t_{p,l}$, with $p = 1, \dots, P$ and $l = 1, \dots, L$ and for $n = 1, \dots, N$
    (a) sample
        $\hat{V}(t_{p,l})^n \sim \mathcal{N}\left(K_{p,l}(\Sigma(\hat{V}(t_{p,l-1})^{A_{p,l-1}^n})^{-1}\hat{\mu}(\hat{V}(t_{p,l-1})^{A_{p,l-1}^n}) - (b_{p,l} + c_{p,l}(\hat{V}(t_{p,l-1})^{A_{p,l-1}^n}))), K_{p,l}\right)$;
    (b) if $t_{p,l} < P$ compute $K_{p+1,l+1} := (\Sigma(\hat{V}(t_{p+1,l}))^{-1} + 2A_{p+1,l+1})^{-1}$ and the conditional expectations $f(\psi_{p,l+1})(\hat{V}(t_{p,l})^n)$ using (4.23), else set $f(\psi_{P,L+1})(\hat{V}(t_{P,L})^n) = 1$;
    (c) evaluate policy $\psi_{p,l}(\hat{V}(t_{p,l-1})^n, \hat{V}(t_{p,l})^n)$ using (4.19);
    (d) compute weights
        $g_{p,l}^{\lambda,\psi}(\hat{V}(t_{p,l-1})^n, \hat{V}(t_{p,l})^n) = \frac{g(\hat{V}(t_{p,l-1})^n, \hat{V}(t_{p,l})^n, y_k(t_{p,l}))^{\lambda} f(\psi_{p,l+1})(\hat{V}(t_{p,l})^n)}{\psi_{p,l}(\hat{V}(t_{p,l-1})^n, \hat{V}(t_{p,l})^n)}$;
    (e) normalize weights $W_{p,l}^n = g_{p,l}^{\lambda,\psi}(\hat{V}(t_{p,l-1})^n, \hat{V}(t_{p,l})^n)/\sum_{m=1}^N g_{p,l}^{\lambda,\psi}(\hat{V}(t_{p,l-1})^m, \hat{V}(t_{p,l})^m)$;
    (f) compute effective sample size $ESS_{p,l} = \{\sum_{n=1}^N (W_{p,l}^n)^2\}^{-1}$;
    (g) sample ancestors $A_{p,l}^n \sim \mathcal{R}(W_{p,l}^1, \dots, W_{p,l}^N)$;
3. Compute likelihood estimator
        $\hat{p}(y_k(t_{0:P}) \mid \lambda) = \left\{N^{-1}\sum_{n=1}^N g_0^{\psi}(\hat{V}(0)^n)\right\} \prod_{p=1}^P \prod_{l=1}^L \left\{N^{-1}\sum_{n=1}^N g_{p,l}^{\lambda,\psi}(\hat{V}(t_{p,l-1})^n, \hat{V}(t_{p,l})^n)\right\}$

Output: samples $\hat{V}(t_{p,l})_{p=0}^P$ and likelihood estimator $\hat{p}(y_k(t_{0:P}) \mid \lambda)$.
```

We now consider several representation of (4.25). Note that

$$\begin{aligned}
p(d\hat{V}(t_{0:P}) \mid y_k(t_{1:P}), \lambda_i) &= p(y_k(t_{1:P}) \mid \lambda_i)^{-1} q^{\psi(0)}(d\hat{V}(t_{0:P}) \prod_{p=1}^{P} \prod_{l=1}^{L} g(\hat{V}(t_{p,l}), y_k(t_{p,l}))^{\lambda_i} \\
&= p(y_k(t_{1:P}) \mid \lambda_i)^{-1} q^{\psi(i-1)}(d\hat{V}(t_{0:P}) g_0^{\psi(i-1)}(\hat{V}(0)) \prod_{p=1}^{P} \prod_{l=1}^{L} g_{p,l}^{\lambda_i,\psi(i-1)}(\hat{V}(t_{p,l})) \\
&= p(y_k(t_{1:P}) \mid \lambda_i)^{-1} q^{\psi(i)}(d\hat{V}(t_{0:P}) g_0^{\psi(i-1)\cdot\phi(i)}(\hat{V}(0)) \prod_{p=1}^{P} \prod_{l=1}^{L} g_{p,l}^{\lambda_i,\psi(i-1)\cdot\phi(i)}(\hat{V}(t_{p,l})),
\end{aligned}$$

where,

$$\begin{aligned}
g_0^{\lambda_i,\psi(i-1)\cdot\phi(i)}(\hat{V}(0)) &= \frac{\nu(\psi_0^{(i-1)} \cdot \phi_0^{(i)}) f(\psi_{0,1}^{(i-1)} \cdot \phi_{0,1}^{(i)})(\hat{V}(0))}{\psi_0^{(i-1)}(\hat{V}(0))\phi_0^{(i)}(\hat{V}(0))}, \\
g_{p,l}^{\lambda_i,\psi(i-1)\cdot\phi(i)}(\hat{V}(t_{p,l-1}), \hat{V}(t_{p,l})) &= \frac{g(\hat{V}(t_{p,l-1}), \hat{V}(t_{p,l}), y_k(t_{p,l}))^{\lambda_i} f(\psi_{p,l+1}^{(i-1)} \phi_{p,l+1}^{(i)})(\hat{V}(t_{p,l}))}{\psi_{p,l}(\hat{V}(t_{p,l-1}^{(i-1)}), \hat{V}(t_{p,l}))\phi_{p,l}(\hat{V}(t_{p,l-1}^{(i)}), \hat{V}(t_{p,l}))}, \\
g_{P,L}^{\lambda_i,\psi(i-1)\cdot\phi(i)}(\hat{V}(t_{P,L-1}), \hat{V}(t_{P,L})) &= \frac{g(\hat{V}(t_{P,L-1}), \hat{V}(t_{P,L}), y_k(t_{P,L}))^{\lambda_i}}{\psi_{P,L}^{(i-1)}(\hat{V}(t_{P,L-1}), \hat{V}(t_{P,L}))\phi_{P,L}^{(i)}(\hat{V}(t_{P,L-1}), \hat{V}(t_{P,L}))}.
\end{aligned} \tag{4.27}$$

The optimal choice of $\phi^{\star}(i)$ is defined by the recursion

$$\begin{aligned}
\phi_{P,L}^{\star}(\hat{V}(t_{P,L-1}), \hat{V}(t_{P,L})) &= \frac{g(\hat{V}(t_{P,L-1}), \hat{V}(t_{P,L}), y_k(t_{P,L}))^{\lambda_i}}{\psi_{P,L}^{(i-1)}(\hat{V}(t_{P,L-1}), \hat{V}(t_{P,L}))}, \\
\phi_{p,l}^{\star}(\hat{V}(t_{p,l-1}), \hat{V}(t_{p,l})) &= \frac{g(\hat{V}(t_{p,l-1}), \hat{V}(t_{p,l}), y_k(t_{p,l}))^{\lambda_i} f(\psi_{p,l+1}^{(i-1)} \phi_{p,l+1}^{\star})(\hat{V}(t_{p,l}))}{\psi_{p,l}^{(i-1)}(\hat{V}(t_{p,l-1}), \hat{V}(t_{p,l}))}, \\
\phi_0^{\star}(\hat{V}(0)) &= \frac{\nu(\psi_0^{(i-1)} \cdot \phi_0^{(i)}) f(\psi_{0,1}^{(i-1)} \cdot \phi_{0,1}^{\star})(\hat{V}(0))}{\psi_0^{(i-1)}(\hat{V}(0))}.
\end{aligned} \tag{4.28}$$

Hence, the optimal policy at λ_i is given by $\psi^{\star}(i) = \psi(i-1) \cdot \phi^{\star}(i)$. Nevertheless, since the backward recursion (4.28) is intractable, we approximate it using regression.

4.4.3 Approximate dynamic programming

To better understand how the approximated backward recursion works we simplify the notation of (4.28) as follows

$$\begin{aligned}\phi^\star_{P,L}(\hat V(t_{P,L-1}),\hat V(t_{P,L})) &= \frac{g(\hat V(t_{P,L-1}),\hat V(t_{P,L}),y_k(t_{P,L}))^\lambda}{\psi_{P,L}(\hat V(t_{P,L-1}),\hat V(t_{P,L}))},\\ \phi^\star_{p,l}(\hat V(t_{p,l-1}),\hat V(t_{p,l})) &= \frac{g(\hat V(t_{p,l-1}),\hat V(t_{p,l}),y_k(t_{p,l}))^\lambda f(\psi_{p,l+1}\phi^\star_{p,l+1})(\hat V(t_{p,l}))}{\psi_{p,l}(\hat V(t_{p,l-1}),\hat V(t_{p,l}))},\\ \phi^\star_0(\hat V(0)) &= f(\psi_{0,1}\cdot\phi^\star_{0,1})(\hat V(0))/\psi_0(\hat V(0)),\end{aligned} \tag{4.29}$$

where, ψ is the current policy defined by β and $\phi^\star$ is the optimal refinement to target $p(d\hat V(t_{0:P})\mid y_k(t_{1:P}),\lambda)$.

At time P, we have

$$\phi^\star_{P,L}(\hat V(t_{P,L-1}),\hat V(t_{P,L})) = \frac{g(\hat V(t_{P,L-1}),\hat V(t_{P,L}),y_k(t_{P,L}))^\lambda}{\psi_{P,L}(\hat V(t_{P,L-1}),\hat V(t_{P,L}))}$$

where,

$$\begin{aligned}\psi_{P,L}(\hat V(t_{P,L-1}),\hat V(t_{P,L})) = \exp\Big\{ &-\hat V(t_{P,L})^T A_{P,L}\hat V(t_{P,L}) - \hat V(t_{P,L})^T b_{P,L}\\ &-\hat V(t_{P,L})^T c_{P,L}(\hat V(t_{P,L-1})) - d_{P,L}(\hat V(t_{P,L-1}))\Big\}\end{aligned}$$

and we are looking for a function

$$\begin{aligned}\phi_{P,L}(\hat V(t_{P,L-1}),\hat V(t_{P,L})) = \exp\Big\{ &-\hat V(t_{P,L})^T \tilde A_{P,L}\hat V(t_{P,L}) - \hat V(t_{P,L})^T \tilde b_{P,L}\\ &-\hat V(t_{P,L})^T \tilde c_{P,L}(\hat V(t_{P,L-1})) - \tilde d_{P,L}(\hat V(t_{P,L-1}))\Big\}\end{aligned}$$

defined by the coefficients $\tilde\beta_{P,L}$ to approximate $g(\hat V(t_{P,L-1}),\hat V(t_{P,L}),y_k(t_{P,L}))^\lambda/\psi_{P,L}(\hat V(t_{P,L-1}),\hat V(t_{P,L}))$. Therefore, in logarithmic scale, this amounts to solve the following least squares problem

$$\begin{aligned}\arg\min\mathbb{E}[(\log\phi_{P,L}(\hat V(t_{P,L-1}),\hat V(t_{P,L})) - \lambda\log g(\hat V(t_{P,L-1}),\hat V(t_{P,L}),y_k(t_{P,L}))\\ +\log\psi_{P,L}(\hat V(t_{P,L-1}),\hat V(t_{P,L})))^2].\end{aligned}$$

Hence, given a set of particles $(\hat{V}^{(n)}(t_{P,L}), \hat{V}^{(n)}(t_{P,L-1}))_{n=1}^{N}$, we solve

$$\begin{aligned}\tilde{\beta}_{P,L} = \arg\min \sum_{n=1}^{N} (\log \phi_{P,L}(\hat{V}^{(n)}(t_{P,L-1}), \hat{V}^{(n)}(t_{P,L})) - \lambda \log g(\hat{V}^{(n)}(t_{P,L-1}), \hat{V}^{(n)}(t_{P,L}), y_k(t_{P,L}))\\ + \log \psi_{P,L}(\hat{V}^{(n)}(t_{P,L-1}), \hat{V}^{(n)}(t_{P,L})))^2.\end{aligned}$$

Then, we get the refined policy $\psi_{P,L} \cdot \phi_{P,L}$ by performing the update $\beta_{P,L} + \tilde{\beta}_{P,L}$. For time $(P, L-1)$, we have

$$\phi^{\star}_{P,L-1}(\hat{V}(t_{P,L-2}), \hat{V}(t_{P,L-1})) = \frac{g(\hat{V}(t_{P,L-2}), \hat{V}(t_{P,L-1}), y_k(t_{P,L-1}))^{\lambda} f(\psi_{P,L}\phi^{\star}_{P,L})(\hat{V}(t_{P,L-1}))}{\psi_{P,L-1}(\hat{V}(t_{P,L-2}), \hat{V}(t_{P,L-1}))}$$

where,

$$\begin{aligned}\psi_{P,L-1}(\hat{V}(t_{P,L-2}), \hat{V}(t_{P,L-1})) = \exp\Big\{ &- \hat{V}(t_{P,L-1})^T A_{P,L-1} \hat{V}(t_{P,L-1}) - \hat{V}(t_{P,L-1})^T b_{P,L-1}\\ &- \hat{V}(t_{P,L-1})^T c_{P,L-1}(\hat{V}(t_{P,L-2})) - d_{P,L-1}(\hat{V}(t_{P,L-2}))\Big\}\end{aligned}$$

and we are looking for a function

$$\begin{aligned}\phi_{P,L-1}(\hat{V}(t_{P,L-2}), \hat{V}(t_{P,L-1})) = \exp\Big\{ &- \hat{V}(t_{P,L-1})^T \tilde{A}_{P,L-1} \hat{V}(t_{P,L-1}) - \hat{V}(t_{P,L-1})^T \tilde{b}_{P,L-1}\\ &- \hat{V}(t_{P,L-1})^T \tilde{c}_{P,L-1}(\hat{V}(t_{P,L-2})) - \tilde{d}_{P,L-1}(\hat{V}(t_{P,L-2}))\Big\}\end{aligned}$$

defined by the coefficients $\tilde{\beta}_{P,L-1}$ to approximate

$$\frac{g(\hat{V}(t_{P,L-2}), \hat{V}(t_{P,L-1}), y_k(t_{P,L-1}))^{\lambda} f(\psi_{P,L}\phi_{P,L})(\hat{V}(t_{P,L-1}))}{\psi_{P,L-1}(\hat{V}(t_{P,L-2}), \hat{V}(t_{P,L-1}))},$$

where we substitute our approximation $\phi_{P,L} \approx \phi^{\star}_{P,L}$. In logarithmic scale we define the following least squares problem

$$\begin{aligned}\arg\min \mathbb{E}[(\log \phi_{P,L-1}(\hat{V}(t_{P,L-2}), \hat{V}(t_{P,L-1})) - \lambda \log g(\hat{V}(t_{P,L-2}), \hat{V}(t_{P,L-1}), y_k(t_{P,L-1}))\\ - \log f(\psi_{P,L}\phi_{P,L})(\hat{V}(t_{P,L-1})) + \log \psi_{P,L-1}(\hat{V}(t_{P,L-2}), \hat{V}(t_{P,L-1})))^2].\end{aligned}$$

Then, if we have samples $(\hat{V}^{(n)}(t_{P,L-1}), \hat{V}^{(n)}(t_{P,L-2}))_{n=1}^{N}$, we solve

$$\begin{aligned}&\tilde{\beta}_{P,L-1} = \\ &\arg\min \sum_{n=1}^{N} (\log \phi_{P,L-1}(\hat{V}^{(n)}(t_{P,L-2}), \hat{V}^{(n)}(t_{P,L-1})) - \lambda \log g(\hat{V}^{(n)}(t_{P,L-2}), \hat{V}^{(n)}(t_{P,L-1}), y_k(t_{P,L-1})) \\ &\qquad - \log f(\psi_{P,L}\phi_{P,L})(\hat{V}^{(n)}(t_{P,L-1})) + \log \psi_{P,L-1}(\hat{V}^{(n)}(t_{P,L-2}), \hat{V}^{(n)}(t_{P,L-1})))^2.\end{aligned}$$

Then, we get the refined policy $\psi_{P,L-1} \cdot \phi_{P,L-1}$ by performing the update $\beta_{P,L-1} + \tilde{\beta}_{P,L-1}$. We proceed in this way until time $t = 0$ when we seek to approximate $\phi_0^\star(\hat{V}(0)) = f(\psi_{0,1} \cdot \phi_{0,1}^\star)(\hat{V}(0))/\psi_0(\hat{V}(0))$ where

$$\psi_0(\hat{V}(0)) = \left\{ -\hat{V}(0)^T A_0 \hat{V}(0) - \hat{V}(0)^T b_0 - f_0 \right\}.$$

Then, we seek for a function

$$\phi_0(\hat{V}(0)) = \left\{ -\hat{V}(0)^T \tilde{A}_0 \hat{V}(0) - \hat{V}(0)^T \tilde{b}_0 - \tilde{f}_0 \right\}.$$

defined by coefficients $\tilde{\beta}_0$ to approximate $f(\psi_{0,1} \cdot \phi_{0,1})(\hat{V}(0))/\psi_0(\hat{V}(0))$, where we substitute our approximation $\phi_0 \approx \phi_0^\star$. In logarithmic scale we define the following least squares problem

$$\arg\min \mathbb{E}[(\log \phi_0(\hat{V}(0)) - \log f(\psi_{0,1}\phi_{0,1})(\hat{V}(0)) + \log \psi_0(\hat{V}(0)))^2].$$

Then, if we have samples $\hat{V}^{(n)}(0)$ we solve

$$\tilde{\beta}_0 = \arg\min \sum_{i=1}^{N} (\log \phi_0(\hat{V}^{(n)}(0)) - \log f(\psi_{0,1}\phi_{0,1})(\hat{V}^{(n)}(0)) + \log \psi_0(\hat{V}^{(n)}(0)))^2.$$

The procedure we outlined defines an approximate dynamic programming (ADP) method and it is presented in Algorithm 12. In summary, the resulting controlled SMC algorithm is presented in Algorithm 13.

Algorithm 12 Approximate Dynamic Programming

```
Input: samples $\hat{V}^{(n)}(t_{p,l})$, policy $\psi$ defined by coefficients $\beta$ and tempering level $\lambda$.

  1. For time $p = P, \dots, 1$ and intermediate points $l = 1, \dots, L$
      (a) Solve the least squares problem
          $\tilde{\beta}_{p,l} = \arg\min \sum_{n=1}^{N} (\log \phi_{p,l}(\hat{V}^{(n)}(t_{p,l-1}), \hat{V}^{(n)}(t_{p,l})) - \lambda \log g(\hat{V}^{(n)}(t_{p,l-1}), \hat{V}^{(n)}(t_{p,l}), y_k(t_{p,l})) + \log \psi_{p,l}(\hat{V}^{(n)}(t_{p,l-1}), \hat{V}^{(n)}(t_{p,l})))^2.$
          to obtain coefficients defining $\phi_{p,l}$.
      (b) update the policy by performing $\tilde{\beta}_{p,l} + \beta_{p,l}$ to obtain $\psi_{p,l} \cdot \phi_{p,l}$.
  2. For time $p = 0$
      (a) Solve the least squares problem
          $\tilde{\beta}_0 = \arg\min \sum_{i=1}^{N} (\log \phi_0(\hat{V}^{(n)}(0)) - \log f(\psi_{0,1}\phi_{0,1})(\hat{V}^{(n)}(0)) + \log \psi_0(\hat{V}^{(n)}(0)))^2.$
          to obtain coefficients defining $\phi_0$.
      (b) update the policy by performing $\tilde{\beta}_0 + \beta_0$ to obtain $\psi_0 \cdot \phi_0$.

Output: refined policy $\psi \cdot \phi$ defined by coefficients $\beta + \tilde{\beta}$.
```

Algorithm 13 Controlled Sequential Monte Carlo (for a given tempering level λ)

```
Input: number of particles $N$, number of tempering steps $I_1$ and schedule $\lambda_i$, for $i = 1, \dots, I_1$, number of policy iterations $I_2$.

Initialize by sampling $N$ trajectories from $p(d\hat{V}_{0:P})$;

For $i = 1, \dots, I_1$
  1. set $\lambda^* = \min(\lambda_i, \lambda)$;
     For $j = 1, \dots, I_2$
      2a. run ADP at $\lambda^*$ to learn refined policy $\psi(\lambda^*)$ using previous samples from SMC;
      2b. run SMC targeting $p(d\hat{V}_{0:P} \mid y_{1:P}, \lambda^*)$, with proposal $q^{\psi}(d\hat{V}_{0:P})$ defined by $\psi = \psi(\lambda^*)$;
  3. terminate if $\lambda^* = \lambda$.

Output: policy $\psi$ and SMC output.
```

4.5 Numerical experiments

In this section we perform a numerical study on the proposed algorithm. To this end we simulate 1 year of weekly data using the parameters in Table 4.1, which are those estimated by Gruber *et al.* (2020). Further, we choose $\tau = \{1M, 6M, 12M, 18M\}$, $\gamma = \{0, 0.5, 1\}$ and $\sigma_e = 1\%$.

β	M_{11}	M_{21}	M_{22}	Q_{11}	Q_{12}	Q_{22}	R_{11}	R_{12}	R_{22}
1.0012	-0.0079	1.0265	-2.6808	0.0698	-0.077	0.2924	-0.297	-0.8708	-0.4057

Table 4.1: Model parameters used in the numerical study.

First of all we test the quality of the controlled SMC by inspecting one run of the algorithm with $L = 3$ intermediate artificial points. The figures below have been produced by performing $I_1 = 5$ tempering steps, where for each one we have $I_2 = 2$ controlled SMC iterations. The number of particles is $N = 512$.

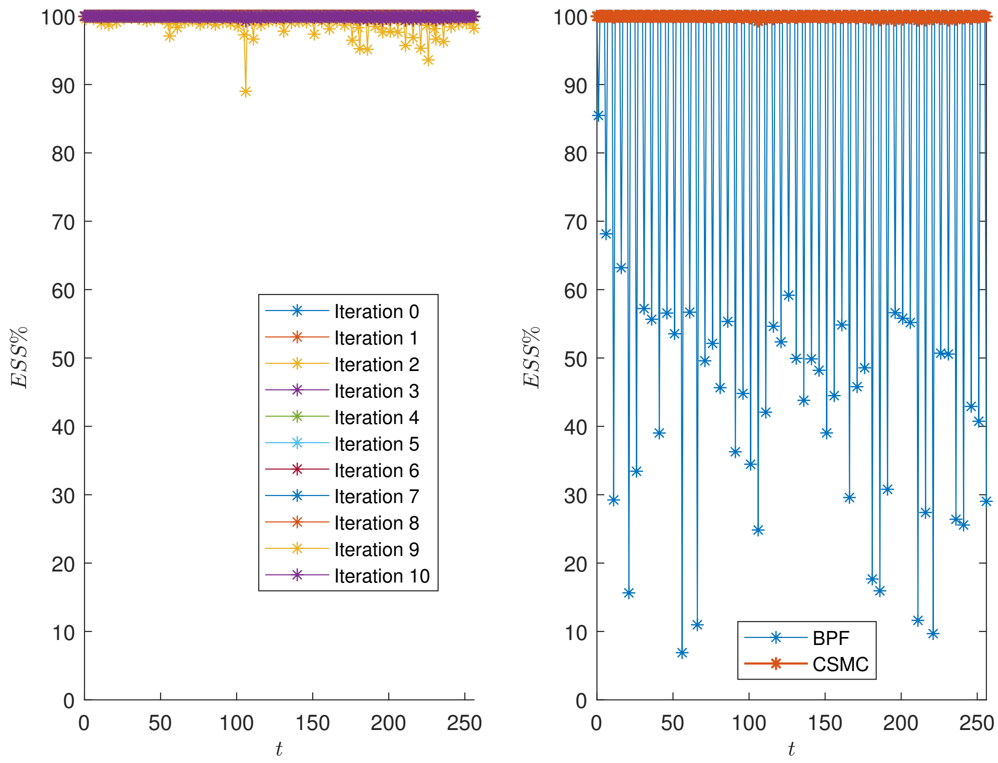

Figure 4.1: Effective Sample Size from controlled SMC across iterations (left panel). On the right panel we compare the ESS from BPF with respect to the last iteration of our controlled SMC.

From Figure 4.1 it is evident the benefit from both tempering and ADP. Indeed, the ESS across time indicates that the particles' population is not deteriorating at all. A comparison with the Bootstrap Particle Filter (BPF), see Algorithm 4 in Section 1.2.1, further support this evidence.

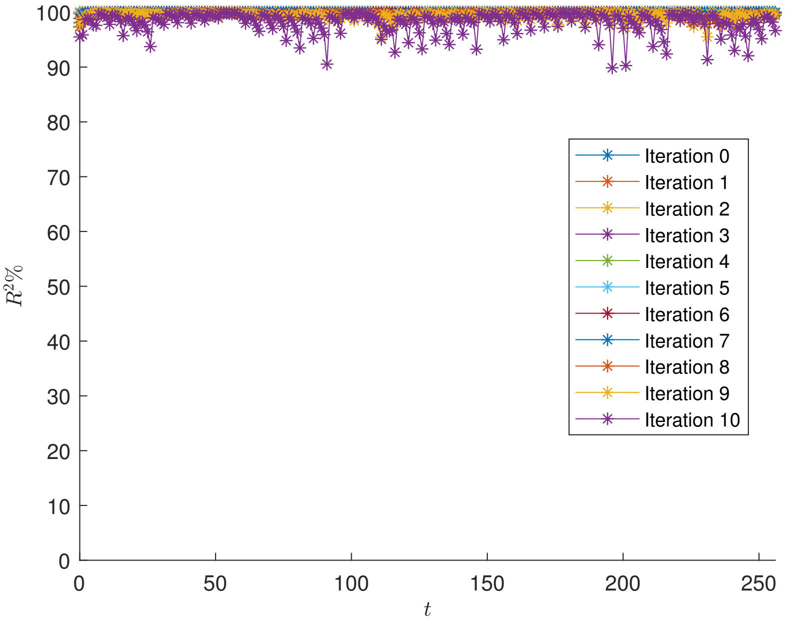

Figure 4.2: R^2 from ADP regression.

The R^2 from ADP regression shows very good performance of the algorithm in learning the policy function ψ, which means that we are including information from the future in a very efficient way. To have an intuition about the evolution of the policy function, we can have a look at Figures 4.4-4.5-4.6, where some elements of ψ are plotted. Indeed, from those graphs we infer that the marginal utility one could get from running more than one controlled SMC iterations is very small in terms of policy fitting.

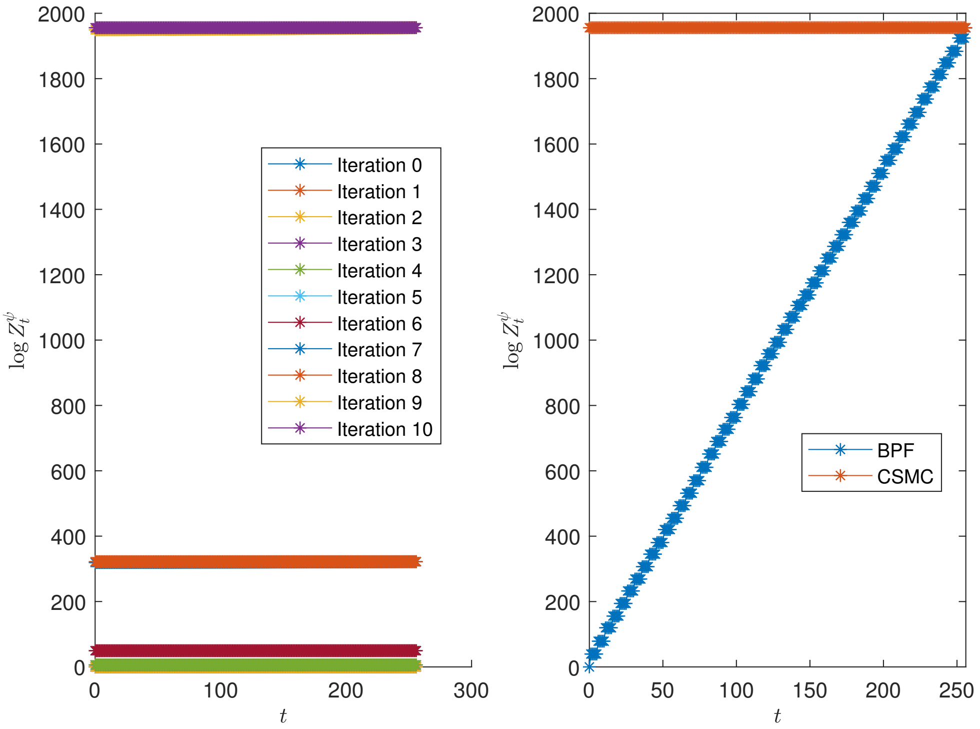

Figure 4.3: Log normalizing constant estimation from controlled SMC across iterations (left panel). On the right panel we compare the Log normalizing constant from BPF with respect to the last iteration of our controlled SMC.

The most important issue we are concerned with is the noise in the marginal likelihood estimation. Indeed, it is well known that the efficiency of SMCS-type routines for parameter estimation depends heavily on how stable is the estimated likelihood from a PF (see Heng *et al.*, 2019). In Figure 4.3 we observe that the log-normalizing constant is almost constant across time for increasing tempering levels, which is not true for standard BPF. As we shall see below this will lead to a huge improvement in terms of likelihood's MC variance with respect to a BPF algorithm.

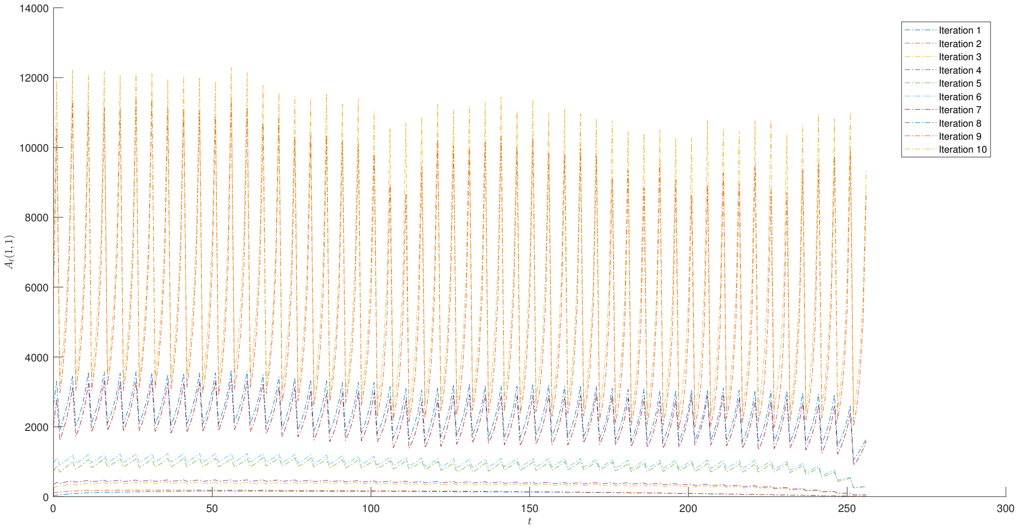

Figure 4.4: Element $A(1,1)$ of refined policy.

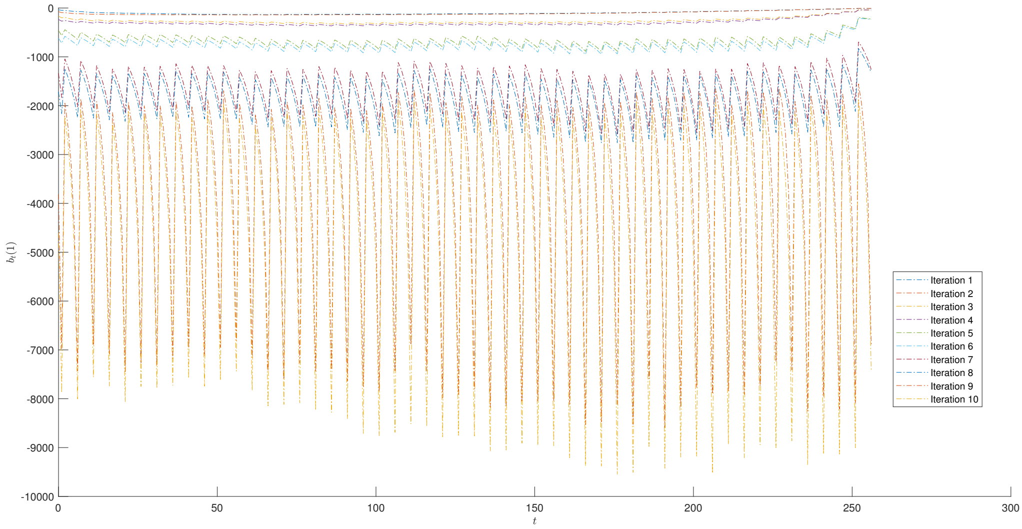

Figure 4.5: Element $b(1)$ of refined policy.

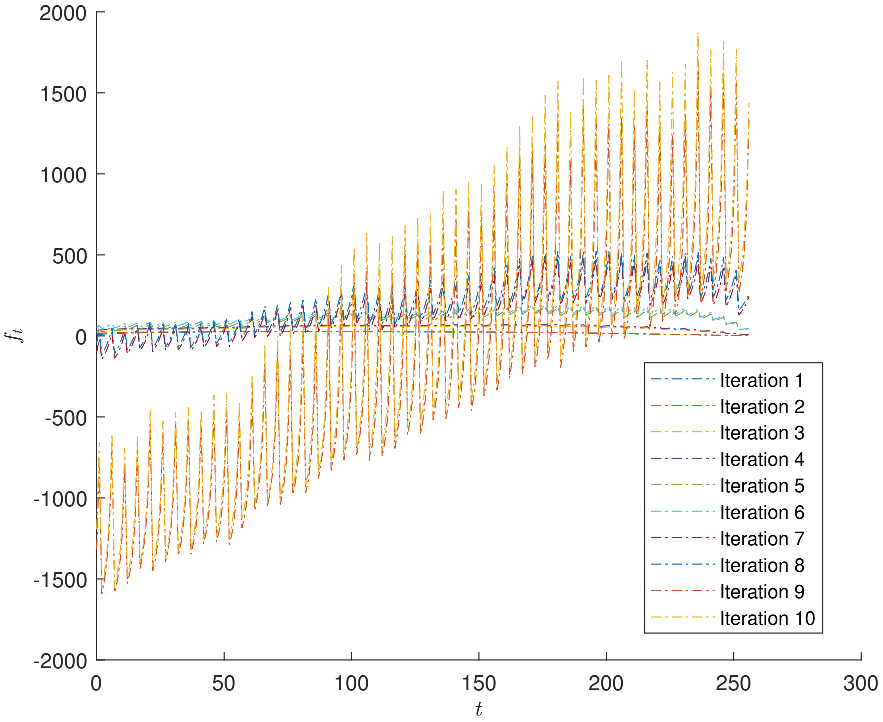

Figure 4.6: Element f of refined policy.

Now, we check how the filter behaves in term of likelihood estimation. Hence, we perform 50 independent runs and we compare the estimated marginal likelihood from a standard BPF with respect to the controlled SMC, for different levels of measurement noise, i.e. $\{5\%, 10\%, 15\%\}$. From the box-plots we clearly see the great improvement offered by our SMC algorithm.

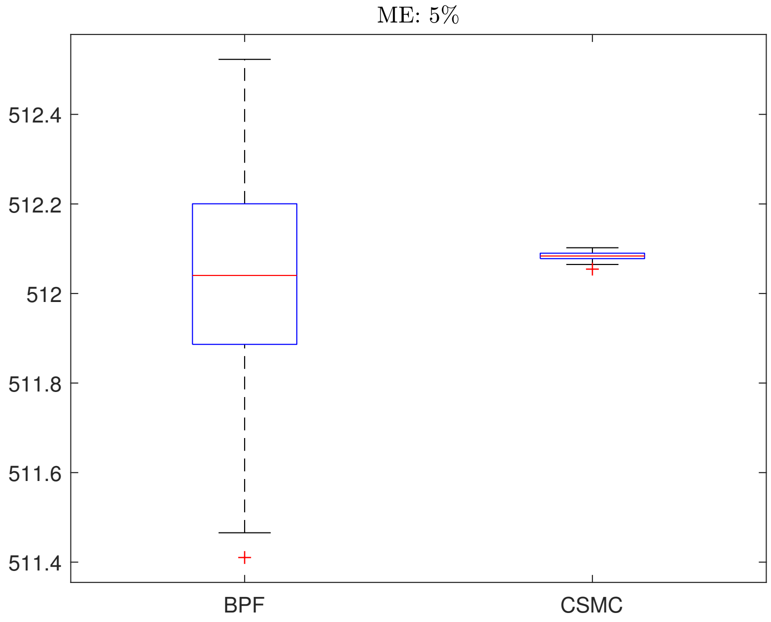

Figure 4.7: Log-marginal likelihood estimates obtained with 50 independent repetitions of BPF and controlled SMC, when measurement standard deviation is equal to 5%.

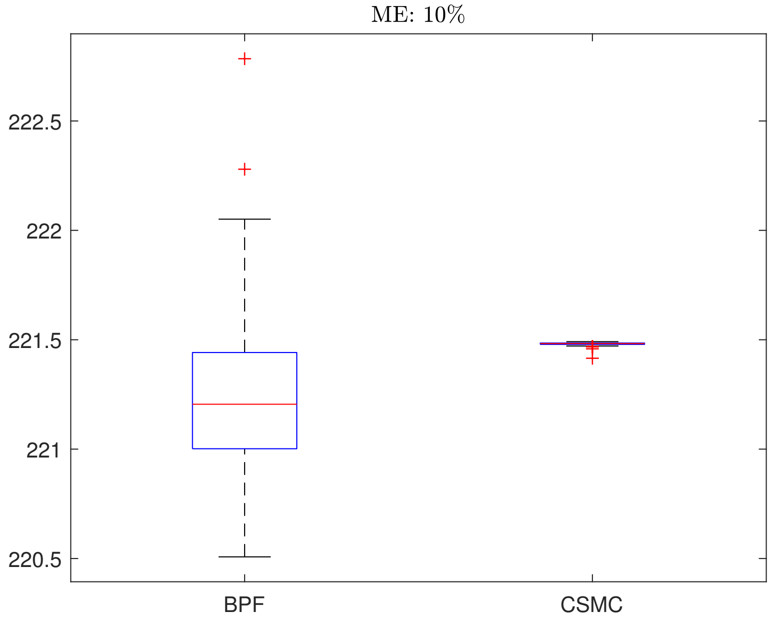

Figure 4.8: Log-marginal likelihood estimates obtained with 50 independent repetitions of BPF and controlled SMC, when measurement standard deviation is equal to 10%.

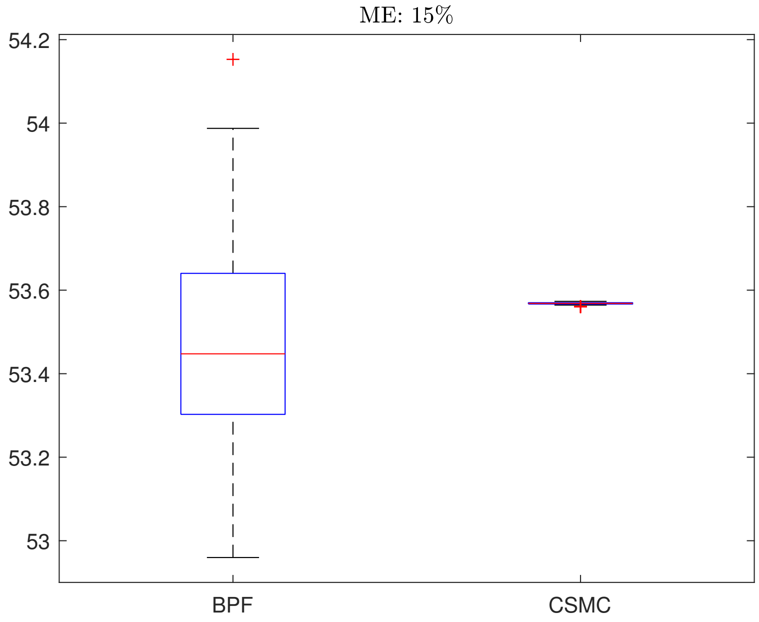

Figure 4.9: Log-marginal likelihood estimates obtained with 50 independent repetitions of BPF and controlled SMC, when measurement standard deviation is equal to 15%.

4.6 Concluding Remarks

In this chapter we propose an efficient SMC algorithm in order to perform marginal likelihood estimation in a Wishart-type stochastic volatility model. Our state-space model considers, as measurement equation, portfolios of weighted options as in Orlowski (2019). Numerical results show very good performances of the proposed controlled SMC algorithm. In particular, with respect to standard filtering techniques, we observe better results in terms of ESS and MC noise in marginal likelihood estimation. The latter property is fundamental in order to perform reliable parameters estimation in a pseudo-marginal context.

Appendix A

Appendix Chapter 3

A A Hybrid Particle Filter

Our estimation strategy is based on Bayesian inference. Hence, if we denote the set of model parameters as Θ and all observations and latent states as $y_{1:T} = \{\ln S_t, \ln F(t,T)^O\}_{t=1}^T$ and $x_{1:T} = \{V_t, \delta_t, \lambda_t\}_{t=1}^T$ respectively, we can define the joint posterior distribution as

$$p(\Theta, x_{1:T} \mid y_{1:T}) = p(x_{1:T} \mid \Theta, y_{1:T}) p(\Theta \mid y_{1:T}) \tag{A.1}$$

The state-space model we are concerned with is clearly non-linear and non Gaussian, thus standard Kalman filtering techniques cannot be applied. Therefore, we rely on the application of particle filtering methods, which are simulation based methods able to take into account the complexity of our model. The general idea is to approximate continuous time distributions with discrete points, called particles.
Given a set of particles $\{x_{t-1}^{(i)}\}_{i=1}^M$ representing the filtering distribution $p(X_{t-1} \mid y_{1:t-1})$ at time $t-1$, we can decompose the filtering distribution at time t as follows:

$$p(x_t \mid y_{1:t}) \propto \int p(y_t \mid x_t) p(x_t \mid x_{t-1}) p(x_{t-1} \mid y_{1:t-1}) dx_{t-1}. \tag{A.2}$$

Now, by importance sampling we can sample from a proposal density $q(x_t \mid x_{t-1})$ and then attach importance weights w_t to account for the difference between the target and the proposal:

$$w_t^{(i)} = \frac{p(y_t \mid x_t^{(i)})p(x_t^{(i)} \mid x_{t-1}^{(i)})}{q(x_t^{(i)} \mid x_{t-1}^{(i)})}, \tag{A.3}$$

and denote with $W_t^{(i)} = w_t^{(i)} / \sum_{j=1}^{M} w_t^{(j)}$ the normalized weights. This algorithm is known as sequential importance sampling and suffers from the so called *particle degeneracy.* To solve this problem we resample the particles proportional to $W_t^{(i)}$, obtaining an equally weighted sample that can be used to approximate the filtering distribution:

$$\hat{p}(x_t \mid y_{1:t}) = \frac{1}{M}\sum_{i=1}^{M} \delta_{x_t^{(i)}}(\tilde{x}_t^{(i)}), \tag{A.4}$$

where, $\tilde{x}_t^{(i)}$ are the resampled particles. This is known in the literature as sequential importance resampling (SIR). As a byproduct, the particle filter algorithm delivers an unbiased estimate of the marginal likelihood:

$$\hat{p}(y_{1:t} \mid \Theta) = \prod_{l=2}^{T} \hat{p}(y_l \mid y_{1:l-1}, \Theta)\hat{p}(y_1 \mid \Theta), \tag{A.5}$$

where,

$$\hat{p}(y_l \mid y_{1:l-1}, \Theta) = \frac{1}{M}\sum_{i=1}^{M} w_l^{(i)}. \tag{A.6}$$

The most common particle filter is the bootstrap filter presented in Gordon *et al.* (1993), where the proposal density is simply the state transition law, i.e. $q(x_t \mid x_{t-1}) = p(x_t \mid x_{t-1})$, which does not take into account the new observation y_t, leading to poor performances if the observation is highly informative (e.g. in the case of a jump). In our setting we are not able to derive the optimal proposal, but we can design a hybrid smooth SIR, which is a relatively efficient particle filter with respect to outliers. In particular, to robustify the filter against outliers, we propose jump times ΔN_t from a Bernoulli with probability 0.5. Furthermore (as in Fulop and Li, 2019) instead of resampling, we fit a multivariate normal distribution on the smoothing distribution and sample from this normal using the inverse CDF method. The detailed algorithm is outlined below.

At time t we have from time $t-1$ a weighted sample of M particles representing the filtering distribution: $x_{t-1}^{(i)} = \{V_{t-1}^{(i)}, \lambda_{t-1}^{(i)}, \delta_{t-1}^{(i)}\}_{i=1}^{M}$.

Hence, for each particle, the PF works as follows:

- **Step 1 (Sample $\Delta N_t^{(i)}$ from proposal):** $\Delta N_t^{(i)} \sim Ber(0.5)$. Then, sample return jumps $J_x^{(i)} \sim \mathcal{N}(\mu_J, \sigma_J^2)$, variance jumps $J_v^{(i)} \sim Exp(\mu_v)$ and the latent states from their transitions:

$$\begin{aligned}
\lambda_t^{(i)} &= \lambda_{t-1}^{(i)} + \beta(\lambda_\infty - \lambda_{t-1}^{(i)})\Delta t + \alpha \Delta N_t^{(i)} \\
V_t^{(i)} &= V_{t-1}^{(i)} + k(\bar{V} - V_{t-1}^{(i)})\Delta t + \sigma_v \sqrt{\Delta t V_{t-1}^{(i)}} \left(\rho_v Z_t^{(i)} + \sqrt{1-\rho_v^2} W_{v,t}^{(i)}\right) + J_v^{(i)} \Delta N_t^{(i)} \\
\delta_t^{(i)} &= \delta_{t-1}^{(i)} + \gamma(\bar{\delta} - \delta_{t-1}^{(i)})\Delta t + \sigma_\delta \sqrt{\Delta t} W_{\delta,t}^{(i)}
\end{aligned}$$

where, $W_{v,t}^{(i)}$ and $W_{\delta,t}^{(i)}$ are independent normals and

$$Z_t^{(i)} = \frac{\ln S_t - \ln S_{t-1} - \mu_x \Delta t - J_x^{(i)} \Delta N_t^{(i)}}{V_{t-1}^{(i)} \Delta t},$$

with, $\mu_x = \mu - \frac{1}{2} V_{t-1} - \lambda_{t-1} \mathbb{E}\left[e^{J_x} - 1\right] - \delta_{t-1}$.

- **Step 2 (Reweight):** Compute log-weights according to

$$\log p(\ln S_t \mid \ln S_{t-1}, x_{t-1}^{(i)}, \Delta N_t) + \log p(\ln F_{t,T} \mid \ln S_t, \delta_{t-1}^{(i)}) + \log \pi_t \to \log w_t^{(i)}$$

where,

$$\begin{aligned}
&\text{if} \quad \Delta N_t^{(i)} = 0 \quad \Rightarrow \quad \log \pi_t = -\lambda_t^{(i)} \Delta t - \log(0.5) \\
&\text{if} \quad \Delta N_t^{(i)} = 1 \quad \Rightarrow \quad \log \pi_t = -\log(1 - \exp(\lambda_t^{(i)} \Delta t)) - \log(0.5)
\end{aligned}$$

- **Step 3 (Smooth approximation):** Generate from $x_{t-1|t} = \{V_{t-1|t}, \delta_{t-1|t}, \lambda_{t-1|t}\}$ fitting a multivariate normal. As suggested by Fulop and Li (2019) we add a moment-matching step.

B Optimal Hedging Solution

In this appendix we provide detailed calculations of the optimal hedge ratio. Let us start from the portfolio variance:

$$\frac{1}{dt} \text{Var}_t[d\Pi_t] = \frac{1}{dt} \left(\text{Var}_t[dX_t] + h^2 \text{Var}_t[dY_t] - 2h \text{Cov}_t[dX_t, dY_t] \right).$$

consider the first term:

$$\begin{aligned}\frac{1}{dt}\mathrm{Var}_t[dX_t] =&\frac{1}{dt}\mathrm{Var}_t\left[\left(r-\frac{1}{2}V_t-\mu^\star\lambda_t-\delta_t\right)dt+\sqrt{V_t}dW_{x,t}+dJ_{x,t}\right],\\ =&V_t+\frac{1}{dt}\mathrm{Var}_t\left[J_x dN_t\right],\end{aligned}$$

where, $\mu^\star = \mathbb{E}[e^{J_x}-1]$ and

$$\begin{aligned}\mathrm{Var}_t\left[J_x dN_t\right] =&\mathbb{E}[J_x^2 dN_t^2]-\left(\mathbb{E}[J_x dN_t]\right)^2=\mathbb{E}[J_x^2]\mathbb{E}[dN_t]-\left(\mathbb{E}[J_x]\mathbb{E}[dN_t]\right)^2,\\ &=(\mu_J^2+\sigma_J^2)\lambda_t dt-\mu_J\lambda_t^2 dt^2=(\mu_J^2+\sigma_J^2)\lambda_t dt.\end{aligned}$$

Therefore we have

$$\frac{1}{dt}\mathrm{Var}_t[dX_t]=V_t+(\mu_J^2+\sigma_J^2)\lambda_t.$$

The next object is the instantaneous variance of log-futures:

$$\begin{aligned}\frac{1}{dt}\mathrm{Var}_t[dY_t] =&\frac{1}{dt}\mathrm{Var}_t\left[\sqrt{V_t}dW_{x,t}+C(\tau)\sigma_\delta dW_{\delta,t}+J_x dN_t\right],\\ &=V_t+C^2(\tau)\sigma_\delta^2+(\mu_J^2+\sigma_J^2)\lambda_t.\end{aligned}$$

The covariance between X and Y is given by

$$\begin{aligned}\frac{1}{dt}\mathrm{Cov}_t[dX_t,dY_t]&=\frac{1}{dt}\mathrm{Cov}_t\Big[\sqrt{V_t}dW_{x,t}+J_x dN_t,\sqrt{V_t}dW_{x,t}+C(\tau)\sigma_\delta dW_{\delta,t}+J_x dN_t\Big],\\ &=V_t+(\mu_J^2+\sigma_J^2)\lambda_t.\end{aligned}$$

Then we obtained the conditional variance of the portfolio:

$$\frac{1}{dt}\mathrm{Var}_t[d\Pi_t]=V_t+(\mu_J^2+\sigma_J^2)\lambda_t+h^2\Big[V_t+C^2(\tau)\sigma_\delta^2+(\mu_J^2+\sigma_J^2)\lambda_t\Big]-2h\Big[V_t+(\mu_J^2+\sigma_J^2)\lambda_t\Big].$$

Now we consider the (non-standardized) third moment:

$$\frac{1}{dt}\mathbb{E}_t\left[(d\Pi-\mathbb{E}(d\Pi))^3\right]=\frac{1}{dt}\mathbb{E}_t\left[d\Pi^3-(\mathbb{E}(d\Pi))^3-3d\Pi^2\mathbb{E}(d\Pi)+3d\Pi(\mathbb{E}(d\Pi))^2\right].$$

The first element can be computed as follows

$$\begin{aligned}\frac{1}{dt}\mathbb{E}_t\left[d\Pi^3\right] &= \frac{1}{dt}\mathbb{E}_t\left[dX^3 - h^3dY^3 - 3hdX^2dY + 3h^2dXdY^2\right], \\ &= (\mu_J^3 + 3\mu_J\sigma_J^2)\lambda_t - h^3(\mu_J^3 + 3\mu_J\sigma_J^2)\lambda_t - 3h(\mu_J^3 + 3\mu_J\sigma_J^2)\lambda_t + 3h^2(\mu_J^3 + 3\mu_J\sigma_J^2)\lambda_t.\end{aligned}$$

Now define,

$$\begin{aligned}\mu_X &= r - \frac{1}{2}V_t - \mu^\star\lambda_t - \delta_t, \\ \mu_Y &= r - \frac{1}{2}V_t - \lambda_t\mu^\star - \delta_t - A'(\tau) - C'(\tau)\delta_t + C(\tau)\gamma(\bar{\delta} - \delta_t).\end{aligned}$$

Then we compute,

$$\begin{aligned}\mathbb{E}_t\left[d\Pi\right] &= \mathbb{E}_t\left[dX\right] - h\mathbb{E}_t\left[dY\right], \\ &= \mu_X dt + \mu_J\lambda_t dt - h\left(\mu_Y dt + \mu_J\lambda_t dt\right).\end{aligned}$$

given this we have that $(\mathbb{E}_t\left[d\Pi\right])^3 = (\mathbb{E}_t\left[d\Pi\right])^2 = \mathbb{E}_t\left[d\Pi^2\mathbb{E}_t\left[d\Pi\right]\right] = 0$. Therefore,

$$\begin{aligned}&\frac{1}{dt}\mathbb{E}_t\left[(d\Pi - \mathbb{E}(d\Pi))^3\right] = \frac{1}{dt}\mathbb{E}_t\left[d\Pi^3\right], \\ &= (\mu_J^3 + 3\mu_J\sigma_J^2)\lambda_t - h^3(\mu_J^3 + 3\mu_J\sigma_J^2)\lambda_t - 3h(\mu_J^3 + 3\mu_J\sigma_J^2)\lambda_t + 3h^2(\mu_J^3 + 3\mu_J\sigma_J^2)\lambda_t.\end{aligned}$$

Now, we take the derivative of the objective function with respect tot h,

$$\begin{aligned}\mathrm{Var}_t(\Pi_t) - \eta\mathrm{Asy}_t(\Pi_t) &= V_t + (\mu_J^2 + \sigma_J^2)\lambda_t + h^2\left[V_t + C^2(\tau)\sigma_\delta^2 + (\mu_J^2 + \sigma_J^2)\lambda_t\right] - 2h\left[V_t + (\mu_J^2 + \sigma_J^2)\lambda_t\right] \\ &\quad - \eta\left[(\mu_J^3 + 3\mu_J\sigma_J^2)\lambda_t - h^3(\mu_J^3 + 3\mu_J\sigma_J^2)\lambda_t - 3h(\mu_J^3 + 3\mu_J\sigma_J^2)\lambda_t + 3h^2(\mu_J^3 + 3\mu_J\sigma_J^2)\right.\end{aligned}$$

If we set this quantity equal to zero and solve for h we get (3.29).

Appendix B

Appendix Chapter 4

A Options Portfolios

In this Appendix we briefly recall the idea of options portfolios as in Bakshi and Madan (2000), Feunou and Okou (2018) and Orlowski (2019).

The starting point is the concept of risk-neutral moments replication using suitable portfolios of options. Consider any twice-continuously differentiable payoff, which can be spanned (Bakshi and Madan, 2000) as follows

$$\mathcal{G}(S) = \mathcal{G}(\bar{S}) + (S - \bar{S})\mathcal{G}_S(\bar{S}) + \int_{\bar{S}}^{\infty} \mathcal{G}_{SS}(X)(S-X)^+ dX + \int_{0}^{\bar{S}} \mathcal{G}_{SS}(X)(X-S)^+ dX, \quad \text{(B.1)}$$

where, X is the strike price and $\mathcal{G}_S(\cdot)$ and $\mathcal{G}_{SS}(\cdot)$ denote the first and second derivatives with respect to S. Equation (B.1) represents a position in the slope and the curvature of the payoff function. Then, in order to price a contingent claim we take expectation of (B.1)

$$\begin{aligned}\mathbb{E}_t^{\mathbb{Q}}[e^{-r\tau}\mathcal{G}(S)] = e^{-r\tau}(\mathcal{G}(\bar{S}) - \bar{S}\mathcal{G}(\bar{S})) + \mathcal{G}(\bar{S})S_t + \int_{\bar{S}}^{\infty} \mathcal{G}_{SS}(X)C(t,\tau;X)dX \\ + \int_{0}^{\bar{S}} \mathcal{G}_{SS}(X)P(t,\tau;X)dX,\end{aligned} \quad \text{(B.2)}$$

where, $\tau = T - t$. If we denote log-returns on $S = e^y$ between t and $t+\tau$ by $r_{t,\tau} = y_{t+\tau} - y_t$, we can construct risk-neutral moments with a payoff function $G(S) = r_{t,\tau}^n$ describing power contracts[1]. Hence, cumulants are linked to the expected payoff of power contracts by $\chi_n = \mathbb{E}_t^{\mathbb{Q}}[r_{t,\tau}^n]$, such that under $\mathbb{Q}$

$$\begin{cases} \mathrm{CUM}_{t,\tau}^2 = \chi_2 - \chi_1^2, \\ \mathrm{CUM}_{t,\tau}^3 = \chi_3 - 3\chi_2\chi_1 + 2\chi_1^3, \\ \mathrm{CUM}_{t,\tau}^4 = \chi_4 - 4\chi_3\chi_1 - 3\chi_2^2 + 12\chi_2\chi_1^2 - 6\chi_1^4. \end{cases} \tag{B.3}$$

Thus, it is possible to construct risk-neutral moments from the data using weighted portfolios of OTM options given by Equation (B.2). On the other hand, it is possible to derive model-implied cumulants by taking the $n-th$ derivative of the CGF of log-returns, which is available in closed-form for affine models.

Indeed, if we denote by F_t the factors on which the distribution of y depends, the conditional characteristic function of log-returns in the affine setting is given as follows

$$\psi(\gamma; y_t, F_t, T) = \exp\left(\gamma y_t + A(\gamma,\tau) + B(\gamma,\tau)F_t\right), \tag{B.4}$$

where, $A(\cdot)$ and $B(\cdot)$ are solutions to a system of ODEs. Then, in order to retrieve risk-neutral cumulants we can consider the following result

$$\mathrm{CUM}_{t,\tau}^n = \frac{\partial^n \ln\psi(\gamma;T)}{(\partial\gamma)^n}\bigg|_{\gamma=0} = \frac{\partial^n A(\gamma,\tau)}{(\partial\gamma)^n}\bigg|_{\gamma=0} + \frac{\partial^n B(\gamma,\tau)}{(\partial\gamma)^n}\bigg|_{\gamma=0} F_t. \tag{B.5}$$

Given Equation (B.5), Feunou and Okou (2018) apply an approximated Kalman Filtering technique in order to estimate the model proposed by Andersen *et al.* (2015) using risk-neutral cumulants.

A slightly different approach is considered by Orlowski (2019), who propose to summarize information from options prices by calculating the CGF slope (so only the first derivative is needed) at $\gamma = \{0, 0.5, 1\}$. These specific points replicate the variance swap contract ($\gamma = 0$), the Hellinger swap contract ($\gamma = 0.5$) and the Gamma swap contract ($\gamma = 1$). In particular, we are going to consider the approach proposed by Orlowski (2019). Indeed, in some cases (e.g. the Wishart model), we are not able to derive ana-

[1] According to Feunou and Okou (2018) risk-neutral variance, skewness, and kurtosis summarize most of the information embedded in option prices. Then, the most interesting cases are given by $n = 2, 3, 4$.

lytically the derivatives of the CGF, then a numerical differentiation must be performed. This step could be problematic for higher order derivatives as considered by Feunou and Okou (2018), while it is very efficient in the framework proposed by Orlowski (2019).

A.1 Simulation

Now, we check that we get correct results by comparing the closed-form solutions with MC simulations. To this end we need first to simulate a Wishart process and second to simulate log-returns in order to compute the MGF slope. For the first task we have already seen in Section 4.3 how to discretize the Wishart model. For the price process we proceed as in Kang *et al.* (2017).

If we define $Y_t = \ln S_t$ and integrate the related SDE we get a useful form of the log-price discretization

$$Y_{t_i} = Y_{t_{i-1}} + r(t_i - t_{i-1}) - \frac{1}{2}\int_{t_{i-1}}^{t_i} Tr(V_s)ds + Tr\left[\int_{t_{i-1}}^{t_i} \sqrt{V_s}dB_sR + \int_{t_{i-1}}^{t_i} \sqrt{V_s}dW_s\sqrt{I_n - RR^T}\right]$$

Indeed, we can use the following first order approximation:

$$Y_{t_i} = Y_{t_{i-1}} + \left(r - \frac{1}{2}Tr\left[V_{t_{i-1}}\right]\right)\Delta t + Tr\left[\sqrt{V_{t_{i-1}}}\left(\Delta B_{t_i}R + \Delta Z_{t_i}\sqrt{I_n - RR^T}\right)\right].$$

Now, we are interested in the comparison with the first derivative of the Cumulative Generating Function (CGF) of log-returns with respect to γ evaluated in $\{0, 1/2, 1\}$. Indeed, if we denote by K_t the CGF, in general affine models it is possible to express the slope of K_t as follows

$$K_t'(\gamma, \tau) = A'(\gamma, \tau) + B'(\gamma, \tau)F_t, \tag{B.6}$$

where, the coefficients A' and B' are given as solution to the usual ODEs system arising in affine models and F_t is a latent factor.

In particular, it is possible to provide a finite difference approximation for the coefficients $A'(\gamma, \tau)$ and $B'(\gamma, \tau)$ as follows:

$$A'(\gamma, \tau) \approx \frac{A(\gamma + \Delta\gamma, \tau) - A(\gamma, \tau)}{\Delta\gamma}, \tag{B.7}$$

$$B'(\gamma, \tau) \approx \frac{B(\gamma + \Delta\gamma, \tau) - B(\gamma, \tau)}{\Delta\gamma}, \tag{B.8}$$

where, $A(\cdot,\tau)$, $B(\cdot,\tau)$ are obtained from the solutions (4.6) and (4.8). Then, for the Wishart model we have

$$K_t'(\gamma,\tau) \approx B'(\gamma,\tau) + Tr[A'(\gamma,\tau)V_t]. \tag{B.9}$$

To check the correctness of this formula we implement a Monte Carlo evaluation of the CGF slope. The Monte Carlo counterpart of (B.6) is given by Orlowski (2019). If we denote the CGF by

$$K_t(\gamma,\tau) = \ln \mathbb{E}\left[\left(\frac{Y_{t+\tau}}{Y_t}\right)^\gamma\right],$$

we can write the CGF slope as follows

$$K_t'(\gamma,\tau) = \frac{\mathbb{E}\left[\left(\frac{Y_{t+\tau}}{Y_t}\right)^\gamma \ln \frac{Y_{t+\tau}}{Y_t}\right]}{e^{K_t(\gamma,\tau)}},$$

the latter can be evaluated by simulations and compared to (B.9).

For the numerical experiment we take the parameters from Gruber *et al.* (2020) as in Table 4.1. Further, we set the time step $\Delta t = 1/1000$ and the number of simulations $N_s = 10^5$. In Figure B.1 we show the results for $\tau = 1M$ and for $\gamma \in [0,1]$.

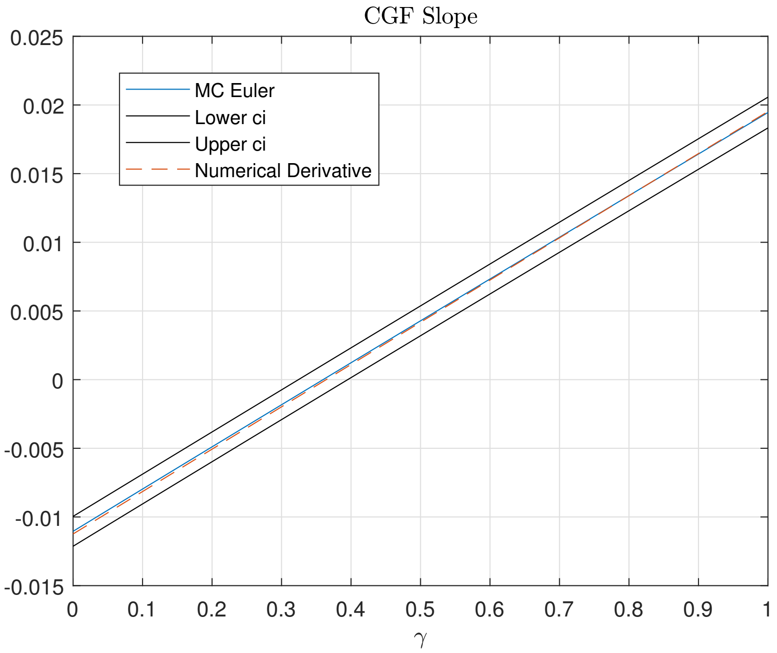

Figure B.1: MC evaluation of CGF slope with $\tau = 1$ months.

B Twisted kernels

Here we provide detailed calculations of the twisted kernels. To lighten the notation we omit the intermediate filling points. For example, we use V_t in place of $V(t_{p,l})$.

Initial twisted kernel

At time $t = 0$ the initial state is described by (4.17)

$$\nu(d\hat{V}_0) = \mathcal{N}(\hat{\mu}_0, \Sigma_0)d\hat{V}_0,$$

and the twisting function is given by (4.20)

$$\psi_0(\hat{V}_0) = \left\{-\hat{V}_0^T A_0 \hat{V}_0 - \hat{V}_0^T b_0 - f_0\right\}.$$

Then, we want to compute $\nu^{\psi}(d\hat{V}_0) = \frac{\nu(d\hat{V}_0)\psi_0(\hat{V}_0)}{\nu(\psi_0)}$, which requires the knowledge of $\nu(\psi_0)$. Therefore, we have that

$$\begin{aligned}\nu(\psi_0) &= \int \psi_0(\hat{V}_0)\nu(d\hat{V}_0)\\ &= \int \frac{1}{2\pi^{n_v/2}\det(\Sigma_0)^{1/2}} \exp \underbrace{\left\{-\hat{V}_0^T A_0 \hat{V}_0 - \hat{V}_0^T b_0 - f_0 - \frac{1}{2}(\hat{V}_0 - \hat{\mu}_0)^T \Sigma_0^{-1}(\hat{V}_0 - \hat{\mu}_0)\right\}}_{A} d\hat{V}_0\end{aligned}$$

Now, we focus on the term inside the exponent,

$$\begin{aligned}A &= -\hat{V}_0^T A_0 \hat{V}_0 - \hat{V}_0^T b_0 - f_0 - \frac{1}{2}\left(\hat{V}_0 - \hat{\mu}_0\right)^T \Sigma_0^{-1}\left(\hat{V}_0 - \hat{\mu}_0\right)\\ &= -\hat{V}_0^T A_0 \hat{V}_0 - \hat{V}_0^T b_0 - f_0 - \frac{1}{2}\hat{V}_0^T \Sigma_0^{-1}\hat{V}_0 + \frac{1}{2}\hat{V}_0^T \Sigma_0^{-1}\hat{\mu}_0 + \frac{1}{2}\hat{\mu}_0^T \Sigma_0^{-1}\hat{V}_0 - \frac{1}{2}\hat{\mu}_0^T \Sigma_0^{-1}\hat{\mu}_0\\ &= -\frac{1}{2}\hat{V}_0^T(\Sigma_0^{-1} + 2A_0)\hat{V}_0 - \hat{V}_0^T b_0 - f_0 + \frac{1}{2}\hat{V}_0^T \Sigma_0^{-1}\hat{\mu}_0 + \frac{1}{2}\hat{\mu}_0^T \Sigma_0^{-1}\hat{V}_0 - \frac{1}{2}\hat{\mu}_0^T \Sigma_0^{-1}\hat{\mu}_0\end{aligned}$$

we can define $K_0 := (\Sigma_0^{-1} + 2A_0)^{-1}$ and move back to the integral in order to bring outside the terms which do not depend on $\hat{V}_0$,

$$\begin{aligned}\nu(\psi_0) =& \det(\Sigma_0)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\hat{\mu}_0^T \Sigma_0^{-1}\hat{\mu}_0 - f_0\right\}\\ &\times \int (2\pi)^{-\frac{n_v}{2}} \exp\left\{-\frac{1}{2}\left(\hat{V}_0^T K_0^{-1}\hat{V}_0 + 2\hat{V}_0^T b_0 - \hat{V}_0^T \Sigma_0^{-1}\hat{\mu}_0 - \hat{\mu}_0^T \Sigma_0^{-1}\hat{V}_0\right)\right\} d\hat{V}_0\end{aligned}$$

We can simplify this expression by completing the quadratic form inside the integral at the exponent, which we call E. To facilitate the calculations we define $\Omega = K_0^{-1}$ and $\Gamma = \Sigma_0^{-1}$. Thus,

$$\begin{aligned} E &= \hat{V}_0^T \Omega \hat{V}_0 + \hat{V}_0^T b_0 + \hat{V}_0^T b_0 - \hat{V}_0^T \Gamma \hat{\mu}_0 - \hat{\mu}_0^T \Gamma \hat{V}_0 \\ &= \hat{V}_0^T \Omega \hat{V}_0 + b_0^T \hat{V}_0 + \hat{V}_0^T b_0 - \hat{V}_0^T \Gamma \hat{\mu}_0 - \hat{\mu}_0^T \Gamma \hat{V}_0 \\ &= \hat{V}_0^T \Omega \hat{V}_0 - (\Gamma \hat{\mu}_0 - b_0)^T \hat{V}_0 - \hat{V}_0^T (\Gamma \hat{\mu}_0 - b_0), \end{aligned}$$

we obtain the last equality by putting together $\hat{V}_0^T b_0$ and $\hat{V}_0^T \Gamma \hat{\mu}_0$, which gives $\hat{V}_0^T(\Gamma\hat{\mu}_0 - b_0)$, and $b_0^T \hat{V}_0$ and $\hat{\mu}_0^T \Gamma \hat{V}_0$, which gives $(\hat{\mu}_0^T \Gamma - b_0^T)\hat{V}_0$. Since $\Gamma^T = \Gamma$, the latter is equivalent to $(\hat{V}_0^T(\Gamma\hat{\mu}_0 - b_0))^T$.

Now, we define $B = (\Gamma\hat{\mu}_0 - b_0)$ and to complete the square add and subtract $B^T \Omega^{-1} B$. Moreover, we multiply by $\Omega^{-1}\Omega = I$ all terms except the first one. Then,

$$\begin{aligned} E &= \hat{V}_0^T \Omega \hat{V}_0 - B^T \hat{V}_0 - \hat{V}_0^T B + B^T \Omega^{-1} B - B^T \Omega^{-1} B \\ &= \hat{V}_0^T \Omega \hat{V}_0 - B^T \Omega^{-1} \Omega \hat{V}_0 - B^T \Omega^{-1} \Omega \hat{V}_0 + B^T \Omega^{-1} \Omega \Omega^{-1} B - B^T \Omega^{-1} \Omega \Omega^{-1} B, \end{aligned}$$

define $\omega = \Omega^{-1} B$, then

$$\begin{aligned} E &= \overbrace{\hat{V}_0^T \Omega \hat{V}_0 - \omega^T \Omega \hat{V}_0 - \omega^T \Omega \hat{V}_0 + \omega^T \Omega \omega}^{\text{square}} - \omega^T \Omega \omega \\ &= (\hat{V}_0 - \omega)^T \Omega (\hat{V}_0 - \omega) - \omega^T \Omega \omega \\ &= (\hat{V}_0 - \Omega^{-1} B)^T \Omega (\hat{V}_0 - \Omega^{-1} B) - B^T \Omega^{-1} \Omega \Omega^{-1} B \\ &= (\hat{V}_0 - K_0(\Sigma_0^{-1}\hat{\mu}_0 - b_0))^T K_0^{-1} (\hat{V}_0 - K_0(\Sigma_0^{-1}\hat{\mu}_0 - b_0)) - (\Sigma_0^{-1}\hat{\mu}_0 - b_0)^T K_0 (\Sigma_0^{-1}\hat{\mu}_0 - b_0). \end{aligned}$$

Moving back to the integral we now have,

$$\begin{aligned}\nu(\psi_0) =& \det(\Sigma_0)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\hat{\mu}_0^T\Sigma_0^{-1}\hat{\mu}_0 - f_0\right\} \\ &\times \int (2\pi)^{-\frac{n_v}{2}} \exp\left\{-\frac{1}{2}(\hat{V}_0 - K_0(\Sigma_0^{-1}\hat{\mu}_0 - b_0))^T K_0^{-1}(\hat{V}_0 - K_0(\Sigma_0^{-1}\hat{\mu}_0 - b_0))\right\} \\ &\times \exp\left\{\frac{1}{2}(\Sigma_0^{-1}\hat{\mu}_0 - b_0)^T K_0(\Sigma_0^{-1}\hat{\mu}_0 - b_0)\right\} d\hat{V}_0 \\ &= \det(\Sigma_0)^{-\frac{1}{2}}\det(K_0)^{\frac{1}{2}} \exp\left\{-\frac{1}{2}\hat{\mu}_0^T\Sigma_0^{-1}\hat{\mu}_0 - f_0\right\} \exp\left\{\frac{1}{2}(\Sigma_0^{-1}\hat{\mu}_0 - b_0)^T K_0(\Sigma_0^{-1}\hat{\mu}_0 - b_0)\right\} \\ &\times \underbrace{\int (2\pi)^{-\frac{n_v}{2}}\det(K_0)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\hat{V}_0 - K_0(\Sigma_0^{-1}\hat{\mu}_0 - b_0))^T K_0^{-1}(\hat{V}_0 - K_0(\Sigma_0^{-1}\hat{\mu}_0 - b_0))\right\} d\hat{V}_0}_{=1}\end{aligned}$$

Therefore, we obtained (4.21),

$$\nu(\psi_0) = \det(\Sigma_0)^{-\frac{1}{2}} \det(K_0)^{\frac{1}{2}} \exp\left\{\frac{1}{2}\left(\Sigma_0^{-1}\hat{\mu}_0 - b_0\right)^T K_0 \left(\Sigma_0^{-1}\hat{\mu}_0 - b_0\right)\right\} \exp\left\{-\frac{1}{2}\hat{\mu}_0^T\Sigma_0^{-1} - f_0\right\}$$

Now, we are ready to derive the twisted initial distribution

$$\begin{aligned}\nu^{\psi}(d\hat{V}_0) &= \frac{\nu(d\hat{V}_0)\psi_0(\hat{V}_0)}{\nu(\psi_0)} \\ &= \frac{(2\pi)^{-\frac{n_v}{2}} \det(\Sigma_0)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\hat{V}_0 - \hat{\mu}_0)^T\Sigma_0^{-1}(\hat{V}_0 - \hat{\mu}_0)\right\} \exp\left\{-\hat{V}_0^T A_0\hat{V}_0 - \hat{V}_0^T b_0 - f_0\right\}}{\det(\Sigma_0)^{-\frac{1}{2}} \det(K_0)^{\frac{1}{2}} \exp\left\{\frac{1}{2}\left(\Sigma_0^{-1}\hat{\mu}_0 - b_0\right)^T K_0 \left(\Sigma_0^{-1}\hat{\mu}_0 - b_0\right)\right\} \exp\left\{-\frac{1}{2}\hat{\mu}_0^T\Sigma_0^{-1} - f_0\right\}} \\ &= (2\pi)^{-\frac{n_v}{2}} \det(K_0)^{-\frac{1}{2}} \exp\Big\{ -\frac{1}{2}\hat{V}_0^T\Sigma_0^{-1}\hat{V}_0 + \frac{1}{2}\hat{V}_0^T\Sigma_0^{-1}\hat{\mu}_0 + \frac{1}{2}\hat{\mu}_0^T\Sigma_0^{-1}\hat{V}_0 - \frac{1}{2}\hat{\mu}_0^T\Sigma_0^{-1}\hat{\mu}_0 \\ &- \hat{V}_0^T A_0\hat{V}_0 - \hat{V}_0^T b_0 - f_0 + \frac{1}{2}\hat{\mu}_0\Sigma_0^{-1}\hat{\mu}_0 + f_0 - \frac{1}{2}\left(\Sigma_0^{-1}\hat{\mu}_0 - b_0\right)^T K_0 \left(\Sigma_0^{-1}\hat{\mu}_0 - b_0\right) \Big\}\end{aligned}$$

We focus on the term inside the exponent, which we denote by E,

$$\begin{aligned}E &= -\frac{1}{2}\hat{V}_0^T(\Sigma_0^{-1} + 2A_0)\hat{V}_0 - \hat{V}_0^T b_0 + \frac{1}{2}\hat{V}_0^T\Sigma_0^{-1}\hat{\mu}_0 + \frac{1}{2}\hat{\mu}_0\Sigma_0^{-1}\hat{V}_0 - \frac{1}{2}\left(\Sigma_0^{-1}\hat{\mu}_0 - b_0\right)^T K_0 \left(\Sigma_0^{-1}\hat{\mu}_0 - b_0\right) \\ &= -\frac{1}{2}\left[\hat{V}_0^T K_0^{-1}\hat{V}_0 + 2\hat{V}_0^T b_0 - \hat{V}_0^T\Sigma_0^{-1}\hat{\mu}_0 - \hat{V}_0^T\Sigma_0^{-1}\hat{\mu}_0 + \left(\Sigma_0^{-1}\hat{\mu}_0 - b_0\right)^T K_0 \left(\Sigma_0^{-1}\hat{\mu}_0 - b_0\right)\right],\end{aligned}$$

where, $K_0 := (\Sigma_0^{-1} + 2A_0)^{-1}$. Now, we proceed in a similar way as before. Indeed, we can write

$$\begin{aligned}E &= -\frac{1}{2}\left[\hat{V}_0^T K_0^{-1}\hat{V}_0 + \hat{V}_0^T b_0 + b_0^T \hat{V}_0 - \hat{V}_0^T \Sigma_0^{-1}\hat{\mu}_0 - \hat{\mu}_0^T \Sigma_0^{-1}\hat{V}_0 + \left(\Sigma_0^{-1}\hat{\mu}_0 - b_0\right)^T K_0 \left(\Sigma_0^{-1}\hat{\mu}_0 - b_0\right)\right] \\ &= -\frac{1}{2}\left[\hat{V}_0^T K_0^{-1}\hat{V}_0 - (\Sigma_0^{-1}\hat{\mu} - b_0)^T \hat{V}_0 - \hat{V}_0^T(\Sigma_0^{-1}\hat{\mu} - b_0) + \left(\Sigma_0^{-1}\hat{\mu}_0 - b_0\right)^T K_0 \left(\Sigma_0^{-1}\hat{\mu}_0 - b_0\right)\right].\end{aligned}$$

At this point it is useful to define $\Omega = K_0^{-1}$ and $B = (\Sigma_0^{-1}\hat{\mu} - b_0)$. Thus,

$$\begin{aligned}E &= -\frac{1}{2}\left[\hat{V}_0^T \Omega \hat{V}_0 - B^T \hat{V}_0 - \hat{V}_0^T B + B^T \Omega^{-1} B\right] \\ &= -\frac{1}{2}\left[\hat{V}_0^T \Omega \hat{V}_0 - B^T \Omega^{-1}\Omega \hat{V}_0 - \hat{V}_0^T \Omega^{-1}\Omega B + B^T \Omega^{-1}\Omega\Omega^{-1} B\right],\end{aligned}$$

if we define $\omega = \Omega^{-1}B$ it is easy to recognize the square term

$$\begin{aligned}E &= -\frac{1}{2}\left[\hat{V}_0^T \Omega \hat{V}_0 - \omega^T \Omega \hat{V}_0 - \omega^T \Omega \hat{V}_0 + \omega^T \Omega \omega\right] \\ &= -\frac{1}{2}\left[(\hat{V}_0 - \omega)^T \Omega (\hat{V}_0 - \omega)\right] \\ &= -\frac{1}{2}\left[(\hat{V}_0 - \Omega^{-1}B)^T \Omega (\hat{V}_0 - \Omega^{-1}B)\right] \\ &= -\frac{1}{2}\left[(\hat{V}_0 - K_0(\Sigma_0^{-1}\hat{\mu} - b_0))^T K_0^{-1}(\hat{V}_0 - K_0(\Sigma_0^{-1}\hat{\mu} - b_0))\right].\end{aligned}$$

Therefore, we derived Equation (4.22)

$$\begin{aligned}\nu^{\psi}(d\hat{V}_0) &= (2\pi)^{-\frac{n_v}{2}} \det(K_0)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\hat{V}_0 - K_0(\Sigma_0^{-1}\hat{\mu} - b_0))^T K_0^{-1}(\hat{V}_0 - K_0(\Sigma_0^{-1}\hat{\mu} - b_0))\right\} \\ &= \mathcal{N}\left(\hat{V}_0;\ K_0(\Sigma_0^{-1}\hat{\mu} - b_0), K_0\right) d\hat{V}_0.\end{aligned}$$

B.1 Twisted Markov transition kernels

For $t > 0$ we consider twisting functions (4.19) and Markov kernel (4.14) and begin with the following expectation

$$
\begin{aligned}
f(\psi_t)(\hat{V}_{t-1}) &= \int \psi_t(\hat{V}_t, \hat{V}_{t-1}) f_t(V_{t-1}, d\hat{V}_t) \\
&= \int (2\pi)^{-\frac{n_v}{2}} \det(\Sigma_t)^{-\frac{1}{2}} \exp\left\{-\hat{V}_t^T A_t \hat{V}_t - \hat{V}_t^T b_t - \hat{V}_t^T c_t(\hat{V}_{t-1}) - d_t(\hat{V}_{t-1})\right\} \\
&\times \exp\left\{-\frac{1}{2}(\hat{V}_t - \hat{\mu}_t)^T \Sigma_t^{-1} (\hat{V}_t - \hat{\mu}_t)\right\} d\hat{V}_t \\
&= \int (2\pi)^{-\frac{n_v}{2}} \det(\Sigma_t)^{-\frac{1}{2}} \exp\Bigg\{ - \hat{V}_t^T A_t \hat{V}_t - \hat{V}_t^T b_t - \hat{V}_t^T c_t(\hat{V}_{t-1}) - d_t(\hat{V}_{t-1}) \\
&- \frac{1}{2}(\hat{V}_t - \hat{\mu}_t)^T \Sigma_t^{-1} (\hat{V}_t - \hat{\mu}_t) \Bigg\} d\hat{V}_t.
\end{aligned}
$$

For ease of notation we suppress the dependence of c_t and d_t on $\hat{V}_{t-1}$ and denote by A the term inside the exponent. Then,

$$
\begin{aligned}
A &= -\frac{1}{2}\hat{V}_t^T(\Sigma_t^{-1} + 2A_t)\hat{V}_t - \hat{V}_t^T b_t - \hat{V}_t^T c_t - d_t + \frac{1}{2}\hat{V}_t^T \Sigma_t^{-1} \hat{V}_t + \frac{1}{2}\hat{\mu}_t^T \Sigma_t^{-1} \hat{V}_t - \frac{1}{2}\hat{\mu}_t^T \Sigma_t^{-1} \hat{\mu}_t \\
&= -\frac{1}{2} \underbrace{\left(\hat{V}_t^T K_t^{-1} \hat{V}_t + 2\hat{V}_t^t (b_t + c_t) - \hat{V}_t^T \Sigma_t^{-1} \hat{\mu}_t - \hat{\mu}_t^T \Sigma_t^{-1} \hat{V}_t\right)}_{E} - d_t - \frac{1}{2}\hat{\mu}_t^T \Sigma_t^{-1} \hat{\mu}_t,
\end{aligned}
$$

where, $K_t := (\Sigma_t^{-1} + 2A_t)^{-1}$. Now, let us work on the term denoted by E,

$$
\begin{aligned}
E &= \hat{V}_t^T K_t^{-1} \hat{V}_t + \hat{V}_t^t (b_t + c_t) + (b_t + c_t)^T \hat{V}_t - \hat{V}_t^T \Sigma_t^{-1} \hat{\mu}_t - \hat{\mu}_t^T \Sigma_t^{-1} \hat{V}_t \\
&= \hat{V}_t^T K_t^{-1} \hat{V}_t - \hat{V}_t^T \left(\Sigma_t^{-1}\hat{\mu}_t - (b_t + c_t)\right) - \left(\Sigma_t^{-1}\hat{\mu}_t - (b_t + c_t)\right)^T \hat{V}_t,
\end{aligned}
$$

at this stage it is useful to define $\Omega = K_t^{-1}$ and $B = \left(\Sigma_t^{-1}\hat{\mu}_t - (b_t + c_t)\right)$. Then, in order to complete the square we add and subtract $B^T\Omega^{-1}B$ and multiply each element, except the first, by $\Omega^{-1}\Omega$ or $\Omega\Omega^{-1}$,

$$
\begin{aligned}
E &= \hat{V}_t^T \Omega \hat{V}_t - \hat{V}_t^T B - B^T \hat{V}_t + B^T \Omega^{-1} B - B^T \Omega^{-1} B \\
&= \hat{V}_t^T \Omega \hat{V}_t - B^T \Omega^{-1} \Omega \hat{V}_t - B^T \Omega^{-1} \Omega \hat{V}_t + B^T \Omega^{-1} \Omega \Omega^{-1} B - B^T \Omega^{-1} \Omega \Omega^{-1} B
\end{aligned}
$$

if we call $\omega = \Omega^{-1}B$ we easily recognize the square

$$\begin{aligned}
E &= \hat{V}_t^T\Omega\hat{V}_t - \omega^T\Omega\hat{V}_t - \omega^T\Omega\hat{V}_t + \omega^T\Omega\omega - \omega^T\Omega\omega \\
&= (\hat{V}_t - \omega)^T\Omega(\hat{V}_t - \omega) - \omega^T\Omega\omega \\
&= (\hat{V}_t - \Omega^{-1}B)^T\Omega(\hat{V}_t - \Omega^{-1}B) - B^T\Omega^{-1}\Omega\Omega^{-1}B \\
&= \left(\hat{V}_t - K_t\left(\Sigma_t^{-1}\hat{\mu}_t - (b_t + c_t)\right)\right)^T K_t^{-1}\left(\hat{V}_t - K_t\left(\Sigma_t^{-1}\hat{\mu}_t - (b_t + c_t)\right)\right) \\
&- \left(\Sigma_t^{-1}\hat{\mu}_t - (b_t + c_t)\right)^T K_t\left(\Sigma_t^{-1}\hat{\mu}_t - (b_t + c_t)\right).
\end{aligned}$$

Now, we move back to the integral,

$$\begin{aligned}
f(\psi_t)(\hat{V}_{t-1}) &= \det(\Sigma_t)^{-\frac{1}{2}}\det(K_t)^{\frac{1}{2}}\exp\left\{-d_t - \frac{1}{2}\hat{\mu}_t^T\Sigma_t^{-1}\hat{\mu}_t\right\} \\
&\times\exp\left\{\frac{1}{2}\left[\Sigma_t^{-1}\hat{\mu}_t - (b_t + c_t)\right]^T K_t\left[\Sigma_t^{-1}\hat{\mu}_t - (b_t + c_t)\right]\right\}\int(2\pi)^{-\frac{nv}{2}}\det(K_t)^{-\frac{1}{2}} \\
&\times\exp\left\{-\frac{1}{2}\left[\hat{V}_t - K_t\left(\Sigma_t^{-1}\hat{\mu}_t - (b_t + c_t)\right)\right]^T K_t^{-1}\left[\hat{V}_t - K_t\left(\Sigma_t^{-1}\hat{\mu}_t - (b_t + c_t)\right)\right]\right\}d\hat{V}_t \\
&= \det(\Sigma_t)^{-\frac{1}{2}}\det(K_t)^{\frac{1}{2}}\exp\left\{-d_t - \frac{1}{2}\hat{\mu}_t^T\Sigma_t^{-1}\hat{\mu}_t\right\} \\
&\times\exp\left\{\frac{1}{2}\left[\Sigma_t^{-1}\hat{\mu}_t - (b_t + c_t)\right]^T K_t\left[\Sigma_t^{-1}\hat{\mu}_t - (b_t + c_t)\right]\right\},
\end{aligned}$$

which is expression (4.23). Now, we are ready to derive the twisted Markov transition kernel

$$\begin{aligned}
f_t^{\psi}(\hat{V}_{t-1}, d\hat{V}_t) &= \frac{f(\hat{V}_{t-1}, d\hat{V}_t)\psi_t(\hat{V}_t)}{f(\psi_t)(\hat{V}_{t-1})} \\
&= \frac{(2\pi)^{-\frac{nv}{2}}\det(\Sigma_t)^{-\frac{1}{2}}\exp\left\{-\frac{1}{2}(\hat{V}_t - \hat{\mu}_t)^T\Sigma_t^{-1}(\hat{V}_t - \hat{\mu}_t) - \hat{V}_t^T A_t\hat{V}_t - \hat{V}_t^T b_t - \hat{V}_t^T c_t(\hat{V}_{t-1}) - d_t(\hat{V}_{t-1})\right\}}{\det(\Sigma_t)^{-\frac{1}{2}}\det(K_t)^{\frac{1}{2}}\exp\left\{-d_t - \frac{1}{2}\hat{\mu}_t^T\Sigma_t^{-1}\hat{\mu}_t + \frac{1}{2}\left[\Sigma_t^{-1}\hat{\mu}_t - (b_t + c_t)\right]^T K_t\left[\Sigma_t^{-1}\hat{\mu}_t - (b_t + c_t)\right]\right\}} \\
&= (2\pi)^{-\frac{nv}{2}}\det(K_t)^{-\frac{1}{2}}\exp\left\{ - \frac{1}{2}(\hat{V}_t - \hat{\mu}_t)^T\Sigma_t^{-1}(\hat{V}_t - \hat{\mu}_t) - \hat{V}_t^T A_t\hat{V}_t - \hat{V}_t^T b_t \right. \\
&\left. - \hat{V}_t^T c_t(\hat{V}_{t-1}) - d_t(\hat{V}_{t-1}) + d_t + \frac{1}{2}\hat{\mu}_t^T\Sigma_t^{-1}\hat{\mu}_t - \frac{1}{2}\left[\Sigma_t^{-1}\hat{\mu}_t - (b_t + c_t)\right]^T K_t\left[\Sigma_t^{-1}\hat{\mu}_t - (b_t + c_t)\right]\right\}
\end{aligned}$$

We define with E the term inside the exponent,

$$\begin{aligned}
E = & \frac{1}{2}\hat{V}_t^T(\Sigma_t^{-1} + 2A_t)\hat{V}_t - \hat{V}_t^T(b_t + c_t) + \frac{1}{2}\hat{V}_t^T\Sigma_t^{-1}\hat{\mu}_t + \frac{1}{2}\hat{\mu}_t^T\Sigma_t^{-1}\hat{V}_t \\
& - \frac{1}{2}\left[\Sigma_t^{-1}\hat{\mu}_t - (b_t + c_t)\right]^T K_t \left[\Sigma_t^{-1}\hat{\mu}_t - (b_t + c_t)\right] \\
= & -\frac{1}{2}\Big\{\hat{V}_t^T K_t^{-1}\hat{V}_t + 2\hat{V}_t^T(b_t + c_t) - \hat{V}_t^T\Sigma_t^{-1}\hat{\mu}_t - \hat{\mu}_t^T\Sigma_t^{-1}\hat{V}_t \\
& + \left[\Sigma_t^{-1}\hat{\mu}_t - (b_t + c_t)\right]^T K_t \left[\Sigma_t^{-1}\hat{\mu}_t - (b_t + c_t)\right]\Big\} \\
= & -\frac{1}{2}\Big\{\hat{V}_t^T K_t^{-1}\hat{V}_t - \hat{V}_t^T(\Sigma_t^{-1}\hat{\mu}_t - (b_t + c_t)) - (\Sigma_t^{-1}\hat{\mu}_t - (b_t + c_t))^T\hat{V}_t \\
& + \left[\Sigma_t^{-1}\hat{\mu}_t - (b_t + c_t)\right]^T K_t \left[\Sigma_t^{-1}\hat{\mu}_t - (b_t + c_t)\right]\Big\}
\end{aligned}$$

Now, define $\Omega = K_t^{-1} := (\Sigma_t + 2A_t)$, $B = (\Sigma_t^{-1}\hat{\mu}_t - (b_t + c_t))$ and $\omega = \Omega^{-1}B$. Then,

$$\begin{aligned}
E & = -\frac{1}{2}\left[\hat{V}_t^T\Omega\hat{V}_t - \hat{V}_t^T B - B^T\hat{V}_t + B^T\Omega^{-1}B\right] \\
& = -\frac{1}{2}\left[\hat{V}_t^T\Omega\hat{V}_t - B^T\Omega^{-1}\Omega\hat{V}_t - B^T\Omega^{-1}\Omega\hat{V}_t + B^T\Omega^{-1}\Omega\Omega^{-1}B\right] \\
& = -\frac{1}{2}\left[\hat{V}_t^T\Omega\hat{V}_t - \omega^T\Omega\hat{V}_t - \omega^T\Omega\hat{V}_t + \omega^T\Omega\omega\right] \\
& = -\frac{1}{2}\left[(\hat{V}_t - \omega)^T\Omega(\hat{V}_t - \omega)\right] \\
& = -\frac{1}{2}\left[(\hat{V}_t - \Omega^{-1}B)^T\Omega(\hat{V}_t - \Omega^{-1}B)\right] \\
& = -\frac{1}{2}\left[(\hat{V}_t - K_t(\Sigma_t^{-1}\hat{\mu}_t - (b_t + c_t)))^T K_t^{-1}(\hat{V}_t - K_t(\Sigma_t^{-1}\hat{\mu}_t - (b_t + c_t)))\right]
\end{aligned}$$

Thus, we get the desired kernel (4.24),

$$\begin{aligned}
f_t^{\psi}(\hat{V}_{t-1}, d\hat{V}_t) = & (2\pi)^{-\frac{nv}{2}}\det(K_t)^{-\frac{1}{2}}\exp\Bigg\{-\frac{1}{2}\Big[(\hat{V}_t - K_t(\Sigma_t^{-1}\hat{\mu}_t - (b_t + c_t)))^T K_t^{-1} \\
& (\hat{V}_t - K_t(\Sigma_t^{-1}\hat{\mu}_t - (b_t + c_t)))\Big]\Bigg\} \\
= & \mathcal{N}\left(\hat{V}_t;\, K_t(\Sigma_t^{-1}\hat{\mu}_t - (b_t + c_t)), K_t\right) d\hat{V}_t.
\end{aligned}$$

Acknowledgments

Along the road I encountered many great people, who were able to testify that following our deepest passion is still meaningful, even in this nihilist time. Therefore, I desire to express my sincere gratitude to them for having inspired me.
First of all, I had the pleasure to be supervised by Prof. Carlo Sgarra. Dear Carlo, you have been not only a supervisor with great mathematical skills, but also an example of moral integrity, honesty and kindness. In my darkest hour I found again the desire to pursue my δαίμων (daìmon) mainly because of you. Thank you.
Next, I would like to thank Prof. Andras Fulop for having been the turning point of my PhD. Your endless knowledge in any field of finance and econometrics is accompanied by great passion and emotional intelligence. I am not able to quantify how much you inspired me. Thank you so much.
My PhD experience has been possible because of the support of the DiSMEQ department and especially I need to thank Prof. Fabio Bellini and Prof. Emanuela Rosazza for being always on my side.

Then, I would like to thank all the colleagues who shared with me this journey. In particular, the first year of the PhD program will remain unforgettable because I met super nice people. It was a pleasure to share, infinitely often, tears and blood with Francesco, Luigi, Andrea, Nikos and Mattia. Also, I had a lot of funny moments with Riccardo C., Alexis, Gabriele, Maria, Anna, Andrea and Paolo. A special mention goes to Riccardo. Your friendship has been essential during these years. Doing research with a true friend is priceless. Thank you.
Furthermore, I was lucky enough to meet very nice people during my stay at ESSEC. A special thank goes to Bastien, Nima, Aisha and Marcel. We had countably infinite moments of (Illy) coffee, fun, philosophy and evolutionary biology (Nima, you should

really open a YouTube channel). You are awesome friends and I hope one day we will meet again in the same department.

To conclude I desire to thank my longtime friends who gave me their unconditional support and to my family for being always my safe harbor. A special mention goes to my (not anymore) little sister Chiara. If I had to describe what you mean for me, it would be something like this: "the familiar sound of the lovely laugh from the love of my life".

Bibliography

Ait-Sahalia, Y., Chaco-Diaz, S. and Laeven, R. (2015) Modelling financial contagion using mutually exciting jump processes. *Journal of Financial Economics*, **117**, 585–606.

Alizadeh, A. H., Nomikos, N. K. and Pouliasis, P. K. (2008) A Markov regime switching approach for hedging energy commodities. *Journal of Banking and Finance*, **32**, 1970—-1983.

Alquist, R., Bauer, H. and Diez de los Rios, A. (2014) What does the convenience yield curve tell us about the crude oil market? *Working Paper-Bank of Canada*, **2014-42**.

Andersen, T. G., Fusari, N. and Todorov, V. (2015) The risk premia embedded in index options. *Journal of Financial Economics*, **117**, 558–584.

Andrieu, C., Doucet, A. and Holenstein, R. (2010) Particle Markov chain Monte Carlo. *Journal of the Royal Statistical Society. Series B*, **72**, 269–342.

Bacry, E., Dayri, K. and Muzy, J. (2012) Non-parametric kernel estimation for symmetric Hawkes processes: Application to high frequency financial data. *The European Physical Journal B*, **85**.

Bacry, E., Delattre, S., Hoffman, M. and Muzy, J. (2013) Modelling microstructure noise by mutually exciting point processes. *Quantitative Finance*, **13**, 65–77.

Bacry, E., Mastromatteo, I. and Muzy, J. F. (2015) Hawkes processes in finance. *Market Microstructure and Liquidity*, **1**, 193–198.

Bakshi, G. and Madan, D. (2000) Spanning and derivative-security valuation. *Journal of Financial Economics*, **55**, 205–238.

Bardgett, C., Gourier, E. and M., L. (2019) Inferring volatility dynamics and risk premia from the S&P 500 and VIX markets. *Journal of Financial Economics*, **131**, 593–618.

Batten, J. A., Kinateder, H., Szilagyi, P. G. and F., W. N. (2019) Hedging stocks with oil. *Energy Economics (forthcoming).*

Baumeister, C. and Kilian, L. (2015) Forecasting the real price of oil in a changing world: A forecast combination approach. *Journal of Business and Economic Statistics*, **33**, 338–351.

Bernis, G., Brignone, R., Scotti, S. and Sgarra, C. (2020) A Gamma Ornstein-Uhlenbeck model driven by a Hawkes process. *Working paper, available on `https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3370304`*, 1612–1639.

Bernis, G., Salhi, K. and Scotti, S. (2018) Sensitivity analysis for marked Hawkes processes: application to CLO pricing. *Mathematics and Financial Economics*, **12**, 541–559.

Billio, M., Casarin, R. and Osuntuyi, A. (2018) Markov switching GARCH models for bayesian hedging on energy futures markets. *Energy Economics*, **70**, 545–562.

Black, F. and Scholes, M. (1973) The pricing of options and corporate liabilities. *Journal of Political Economy*, **81**, 637–654.

Bormetti, G., Calcagnile, L., Treccani, M., Corsi, F., Marmi, S. and Lillo, F. (2015) Modelling systemic price cojumps with Hawkes factor models. *Quantitative Finance*, **15**, 1137–1156.

Boswijk, P., Laeven, R. A. and Lalu, A. (2016) Asset returns with self-exciting jumps: Option pricing and estimation with a continuum of moments. *Working Paper available on `https://pdfs.semanticscholar.org/7a56/2f0438baa25e5990418c33af0b99cd62c21c.pdf`.*

Brignone, R. and Sgarra, C. (2020) Asian options pricing in Hawkes-type jump-diffusion models. *Annals of Finance*, **16**, 101–119.

Broadie, M. and Kaya, O. (2006) Exact simulation of stochastic volatility and other affine jump diffusion processes. *Operations Research*, **54**, 217–231.

Brooks, C. and Prokopczuk, M. (2013) The dynamics of commodity prices. *Quantitative Finance*, **13**, 527–542.

Brunnermeier, M. K. and Sannikov, Y. (2016) Macro, money, and finance: A continuous-time approach. *Handbook of Macroeconomics*, **2**, 1497–1545.

Buccioli, A., Kokholm, T., Nicolosi, M. and Trinh, M. (2019) Expected shortfall and portfolio management in contagious markets. *Journal of Banking and Finance*, **102**, 100–115.

Carmona, R. and Ludkovski, M. (2004) Spot convenience yield models for the energy markets. *Contemporary Mathematics*, **351**, 65–80.

Cartea, Á., Jaimungal, S. and Ricci, J. (2014) Buy low, sell high: A high frequency trading perspective. *SIAM Journal of Financial Mathematics*, **5**, 415–444.

Casarin, R. (2004) *Simulation Methods for Nonlinear and Non-Gaussian Models in Finance.* PhD thesis, Universitá Ca' Foscari Venezia.

Casassus, M., Collin-Dufresne, R. and Osuntuyi, A. (2005) Stochastic convenience yield implied for commodity futures markets and interest rates. *The Journal of Finance*, **60**, 2283–2331.

Chang, C. L., McAleer, M. and Tansuchat, R. (2011) Crude oil hedging strategies using dynamic multivariate GARCH. *Energy Economics*, **33**, 912–923.

Chavez-Demoulin, V. and McGill, J. A. (2012) High-frequency financial data modeling using Hawkes processes. *Journal of Banking and Finance*, **36**, 3415–3426.

Chen, J., Hawkes, A. G., Scalas, E. and Trinh, M. (2018) Performance of information criteria for selection of Hawkes process models of financial data. *Quantitative Finance*, **18**, 225–235.

Chiu, W. H. (2010) Skewness preference, risk taking and expected utility maximization. *The Geneva Risk and Insurance Review*, **35**, 108–129.

Chopin, N. (2002) A sequential particle filter for static models. *Biometrika*, **89**, 539–551.

Chopin, N., Jacob, P. E. and Papaspiliopoulos, O. (2013) SMC^2: an efficient algorithm for sequential analysis of state-space models. *Journal of the royal statistical society. Series B.*, **75**, 397–426.

Da Fonseca, J., Grasselli, M. and Tebaldi, C. (2008) A multifactor volatility Heston model. *Quantitative Finance*, **8**, 591–604.

Da Fonseca, J. and Zaatour, R. (2014) Hawkes process: Fast calibration, application to trade clustering and diffusive limit. *Journal of Futures Markets*, **34.6**, 548–579.

Dahlquist, M., Farago, A. and Tédongap, R. (2017) Asymmetries and portfolio choice. *The Review of Financial Studies*, **30**, 667–702.

Daley, D. J. and Vere-Jones, D. (2008) *An Introduction to the Theory of Point Processes. Volume I: Elementary Theory and Methods.* Springer.

Dassios, A. and Zhao, H. (2012) Ruin by dynamic contagion claims. *Insurance: Mathematics and Economics*, **51**, 93–106.

Dassios, A. and Zhao, H. (2013) Exact simulation of Hawkes process with exponentially decaying intensity. *Electronic Communications in Probability*, **18**, 1–13.

Dassios, A. and Zhao, H. (2017) Efficient simulation of clustering jumps with CIR intensity. *Operations Research*, **65**, 1494–1515.

Del Moral, P., Doucet, A. and Jasra, A. (2006) Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B*, **68**, 411–436.

Del Moral, P., Doucet, A. and Sing, S. S. (2010) Forward smoothing using sequential monte carlo. *Technical report Cambridge University.*

Doucet, A., De Freitas, N. and Gordon, N. (2001) *Sequential Monte Carlo methods in practice.* Springer-Verlag.

Doucet, A., Godsill, S. and Andrieu, C. (2000) On sequential monte carlo sampling methods for bayesian filtering. *Statistics and computing*, **10**, 197–208.

Doucet, A. and Johansen, A. M. (2008) A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of Nonlinear Filtering*, **12**, 656–704.

Duan, J. C. and Fulop, A. (2015) Density-tempered marginalized sequential monte carlo samplers. *Journal of Business and Economic Statistics*, **33**, 192–202.

Duffee, G. R. (2002) Term premia and interest rate forecasts in affine models. *The Journal of Finance*, **57**, 405–443.

Duffie, D. and Glynn, P. (1995) Efficient Monte Carlo estimation of security prices. *Annals of Applied Probability*, **4**, 897–9058.

Duffie, D. and Kan, R. (1996) A yield-factor model of interest rates. *Mathematical Finance*, **6**, 379–406.

Durham, G. B. and Gallant, A. R. (2002) Numerical techniques for maximum likelihood estimation of continuous-time diffusion processes. *Journal of Business & Economic Statistics*, **20**, 297–338.

Eraker, B. (2004) Do stock prices and volatility jump? reconciling evidence from spot and option prices. *Journal of Finance*, **59**, 1367–1403.

Eraker, B., Johannes, M. and Polson, N. (2003) The impact of jumps in volatility and returns. *The Journal of Finance*, **58**, 1269–1300.

Errais, E., Giesecke, K. and Goldberg, L. (2010) Affine point processes and portfolio credit risk. *SIAM Journal on Financial Mathematics*, **1**, 642–665.

Eyjolfsson, H. and Tjøshteim, D. (2018) Self-exciting jump processes with applications to energy markets. *Ann. Inst. Stat. Math.*, **70**, 373–393.

Feunou, B. and Okou, C. (2018) Risk-neutral moment-based estimation of affine option pricing models. *Journal of Applied Econometrics*, **33**, 1007–1025.

Fileccia, G. and Sgarra, C. (2015) Historical and risk-neutral estimation in a two factor stochastic volatility model for oil market. *International Journal of Computational Economics and Econometrics*, **5**, 451–479.

Fileccia, G. and Sgarra, C. (2018) A particle filtering approach to oil futures price calibration and forecasting. *Journal of Commodity Markets*, **9**, 21–34.

Filimonov, V., Bicchetti, D., Maystre, N. and Sornette, D. (2014) Quantification of the high level of endogeneity and of structural regime shifts in commodity markets. *Journal of International Money and Finance*, **42**, 174–192.

Filimonov, V. and Sornette, D. (2015) Apparent criticality and calibration issues in the Hawkes self-excited point process model: application to high-frequency financial data. *Quantitative Finance*, **15**, 1293–1314.

Fox, E., Short, M. B., Schoenberg, F. P., Coronges, K. D. and Bertozzi, A. L. (2016) Modeling E-mail networks and inferring leadership using self-exciting point processes. *Journal of the American Statistical Association*, **111**, 564–584.

Fulop, A. (2011) Filtering methods. *Handbook of Computational Finance.*

Fulop, A. and Li, J. (2013) Efficient learning via simulation: a marginalized resample-move approach. *Journal of Econometrics*, **176**, 146–161.

Fulop, A. and Li, J. (2019) Bayesian estimation of dynamic asset pricing models with informative observations. *Journal of Econometrics*, **209**, 114–138.

Fulop, A., Li, J. and Yu, J. (2015) Self-exciting jumps, learning, and asset pricing implications. *The Review of Financial Studies*, **28**, 876–912.

Gibson, R. and Schwartz, E. S. (1990) Stochastic convenience yield and the pricing of oil contingent claims. *Journal of Finance*, **45**, 959–976.

Giesecke, K., Kakavand, H. and Mousavi, M. (2011) Exact simulation of point processes with stochastic intensities. *Operations Research*, **59**, 1233–1245.

Glasserman, P. (2003) *Monte Carlo Methods in Financial Engineering.* Stochastic Modelling and Applied Probability. Springer.

Gonzato, L. and Sgarra, C. (2020) Self-exciting jumps in the oil market: Bayesian estimation and dynamic hedging. *Working paper, available at SSRN: https://ssrn.com/abstract=3564136.*

Gordon, N. J., Salmond, D. J. and Smith, A. F. M. (1993) A novel approach to nonlinear and non gaussian bayesian state estimation. *IEE Proceedings. Part F: Radar and Sonar Navigation*, **140**, 107–113.

Gruber, P., Tebaldi, C. and Trojani, F. (2020) The price of the smile and variance risk premia. *forthcoming Management Science.*

Hainaut, D. (2016) Impact of volatility clustering on equity indexed annuities. *Insurance: Mathematics and Economics*, **71**, 367–381.

Hainaut, D. (2017) Contagion modeling between the financial and insurance markets with time changed processes. *Insurance: Mathematics and Economics*, **74**, 63–77.

Hainaut, D. and Moraux, F. (2018) Hedging of options in the presence of jump clustering. *Journal of Computational Finance*, **22**, 1–35.

Hainaut, D. and Moraux, F. (2019) A switching self-exciting jump diffusion process for stock prices. *Annals of Finance*, **15**, 267–306.

Hardiman, S. and Bouchaud, J. (2014) Branching-ratio approximation for the self-exciting Hawkes process. *Physical review. E, Statistical, Nonlinear, and Soft Matter Physics*, **90**.

Hawkes, A. G. (1971a) Point spectra of some mutually exciting point processes. *Journal of the Royal Statistical Society. Series B (Methodological)*, 438–443.

Hawkes, A. G. (1971b) Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, **58**, 83–90.

Hawkes, A. G. (2018) Hawkes processes and their applications to finance: a review. *Quantitative Finance*, **18**, 193–198.

Heng, J., Bishop, A. N., Deligiannidis, G. and Doucet, A. (2019) Controlled sequential monte carlo. *to appear in Annals of Statistics.*

Iacus, S. M. and Yoshida, N. (2018) Simulation and inference for stochastic processes with YUIMA. a comprehensive R framework for SDEs and other stochastic processes. *Springer.*

Ioannidis, C. and Ka, K. (2018) The impact of oil price shocks on the term structure of interest rates. *Energy Economics*, **72**, 601–620.

Jang, J. and Dassios, A. (2013) A bivariate shot noise self-exciting process for insurance. *Insurance: Mathematics and Economics*, **53**, 524–532.

Jiao, Y., Ma, C., Scotti, S. and Sgarra, C. (2019) A branching process approach to power markets. *Energy Economics*, **79**, 144–156.

Johnson, D. H. (1996) Point process models of single-neuron discharges. *Journal of Computational Neuroscience*, **3**, 275–299.

Kang, C., Kang, W. and Lee, J. M. (2017) Exact simulation of the Wishart multidimensional stochastic volatility model. *Operations Research*, **65**, 1190–1206.

Kilian, L. and Figfusson, R. (2013) Do oil prices help forecast U.S. real GDP? the role of nonlinearities and asymmetries. *Journal of Business and Economic Statistics*, **31**, 78–93.

Kokholm, T. (2016) Pricing and hedging of derivatives in contagious markets. *Journal of Banking and Finance*, **66**, 19–34.

Kraus, A. and Litzenberger, R. (1976) Skewness preferences and valuation of risk assets. *Journal of Finance*, **31**, 1085–1100.

Kroner, K. F. and Sultan, J. (1993) Time-varying distributions and dynamic hedging with foreign currency futures. *Journal of Financial and Quantitative Analysis*, **28**, 531–551.

Lai, A. N. and Mellios, C. (2016) Valuation of commodity derivatives with an unobservable convenience yield. *Computers and Operations Research*, **66**, 402–414.

Larsson, K. and Nossman, N. (2011) Jumps and stochastic volatility in oil prices: Time series evidence. *Energy Economics*, **33**, 504–514.

Lautier, D. (2009) Convenience yield and commodity markets. *Bankers, Markets and Investors*, **102**, 59–66.

Lee, Y., Lim, K. and Ong, C. (2016) Hawkes processes with stochastic excitations. *Proceedings of The 33rd International Conference on Machine Learning (ICML), JMLR*, 79–88.

Lewis, P. A. and Shedler, G. S. (1969) Simulation of nonhonmogeneous poisson processes by thinning. *Naval Research Logistic Quarterly*, **26.3**, 403–413.

Litzenberger, R. and Rabinowitz, N. (1995) Backwadation in oil futures market: Theory and empirical evidence. *Journal of Finance*, **50**, 1517–1545.

Liu, J. and West, M. (2001) Combined parameter and state estimation in simulation-based filtering. *in Simulation-Based Filtering. In: Doucet A., de Freitas N., Gordon N. (eds) Sequential Monte Carlo Methods in Practice*, 197–223.

Liu, Q., Chng, T. and Xu, D. (2014) Hedging industrial metals with stochastic volatility models. *Journal of Futures Markets*, **34**, 704–730.

Lopes, H. F. and Tsay, R. S. (2011) Particle filters and bayesian inference in financial econometrics. *Journal of forecasting*, **30**, 168–209.

Ma, F., Zhang, Y., Huang, D. and Lai, X. (2018) Forecasting oil futures price volatility: New evidence from realized range-based volatility. *Energy Economics*, **75**, 400–409.

Malik, S. and Pitt, M. K. (2011) Particle filters for continous likelihood evaluation and maximisation. *Journal of Econometrics*, **165**, 190–209.

Maneesoonthorn, W., Forbes, C. S. and Martin, G. M. (2016) Inference on self-exciting jumps in prices and volatility using high frequency measures. *Journal of Applied Econometrics*, **32**, 504–532.

Meyer, S. and Held, L. (2014) Power-law models for infectious disease spread. *The Annals of Applied Statistics*, **8**, 1612–1639.

Mohler, G., Short, M. B., Brantigham, P. J., Schoenberg, F. P. and Tita, G. E. (2011) Self-exciting point process modeling of crime. *Journal of the American Statistical Association*, **106**, 100–108.

Møller, J. and Rasmussen, J. G. (2005) Perfect simulation of Hawkes processes. *Advances in Applied Probability*, **37**, 629–646.

Nakagawa, T., Subbey, S. and Solvang, H. K. (2019) Integrating Hawkes process and bio mass models to capture impulsive population dynamics. *Dynamics of Continuous, Discrete and Impulsive Systems Series B: Applications & Algorithms*, **26**, 153–170.

Ogata, Y. (1981) On the Lewis' simulation method for point processes. *IEEE Transactions on Information Theory*, **27.1**, 23–31.

Ogata, Y. (1998) Space-time point-process models for earthquake occurrences. *Annals of the Institute of Statistical Mathematics*, **50**, 379–402.

Orlowski, P. (2019) Informative option portfolios in unscented kalman filter design for affine jump diffusion models. *working paper.*

Ozaki, T. (1979) Maximum likelihood estimation of Hawkes self-exciting point processes. *Annals of the Institute of Statistical Mathematics*, **31**, 145–155.

Pan, J. (2002) The jump-risk premia implicit in option: Evidence from an integrated time series study. *Journal of Financial Economics*, **63**, 3–50.

Pitt, M. K. and Shepard, N. (1999) Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Society*, **94**, 590–599.

Porter, M. D. and White, G. (2012) Self-exciting hurdle models for terrorist activity. *Annals of Applied Statistics*, **6**, 106–124.

Post, T., Vliet, P. and Levy, H. (2008) Risk aversion and skewness preference. *Journal of Banking and Finance*, **32**, 1178–1187.

Ribeiro, D. and Hodges, S. D. (2004) A two factor model for commodity prices and futures valuation. *EFMA 2004 Basel Meetings Paper. Available at SSRN: https://ssrn.com/abstract=498802 or http://dx.doi.org/10.2139/ssrn.498802.*

Robert, C. P. and Casella, G. (2004) *Monte Carlo Statistical Methods.* Springer, 2nd edition.

Robert, C. P. and Casella, G. (2009) *Introducing Monte Carlo Methods with R.* Springer.

Schwartz, E. S. (1997) The stochastic behavior of commodity prices: Implications for valuation and hedging. *Journal of Finance*, **52**, 923–973.

Sørensen, M. (2009) Parametric inference for discretely sampled stochastic differential equations. In *Handbook of financial time series*, 531–553. Springer.

Stabile, G. and Torrisi, G. (2010) Risk processes with non-stationary Hawkes claims arrivals. *Methodology and Computing in Applied Probability*, **12**, 415–429.

Vinogradov, A., Agletdinov, E. and Merson, D. (2019) Mechanical twinning is a correlated dynamic process. *Scientific Reports*, **9**.

Wachter, J. A. (2013) Can time-varying risk of rare disasters explain aggregate stock market volatility? *The Journal of Finance*, **68**, 987–1035.

Wang, Y. (2013) Oil price effects on personal consumption expenditures. *Energy Economics*, **36**, 198–204.

Xu, L., Duan, J. A. and Whinston, A. (2014) Path to purchase: A mutually exciting point process model for online advertising and conversion. *Management Science*, **60**, 1392–1412.

Yan, X. (2002) Valuation of commodity derivatives in a new multi-factor model. *Review of Derivative Research*, **5**, 251–271.

Yu, C. L., Li, H. and Wells, M. T. (2011) MCMC estimation of Levy jump models using stock and option prices. *Mathematical Finance*, **21**, 383–422.

Zaatour, R. (2014) Hawkes process simulation and calibration toolkit. URL `https://CRAN.R-project.org/package=hawkes`.

Zhu, L. (2013) Ruin probabilities for risk processes with non-stationary arrivals and subexponential claims. *Insurance: Mathematics and Economics*, **53**, 544–550.