

Hawkes Process-Driven Models for Limit Order Book Dynamics



Candidate Number: 1037839

University of Oxford

A thesis submitted in partial fulfilment of the
Master of Science in Mathematical and Computational Finance

Trinity 2020

Acknowledgements

I would like to thank my supervisor for his time and his guidance throughout this dissertation.

I would also like to express my sincerest gratitude to my family, for their unwavering support.

Abstract

In this dissertation, we investigate the use of Hawkes processes to model limit order book dynamics. We give empirical evidence for the applicability of this counting process to model order arrivals in a limit order book. We then use a scaling limit theorem and the framework developed by Hambly et al. in [18] to obtain a multivariate SDE model for the profile of the entire limit order book. To demonstrate the effectiveness of this model, we calibrate the SDE parameters to a real data set and simulate the order book dynamics. We compare our results to those obtained using the SPDE model given in [18] and we find that, while our model does a reasonable job of simulating the price process, it overestimates the volatility of the order book profile. Finally, we perform our analysis on a day of unusually high volatility and we assess the robustness of our assumptions and methodologies.

Contents

1	Introduction	1
2	Hawkes Processes	3
2.1	Definitions and Preliminary Results	3
2.2	Parameter Estimation	6
2.3	A Scaling Limit	8
3	Limit Order Books	10
3.1	Preliminaries	10
3.2	Limit Order Book Dynamics in a Static Setting	11
3.2.1	The Microscopic Order Book Model	11
3.2.2	Scaling Limit of the Static Microscopic Model	13
3.3	Limit Order Book Dynamics in a Dynamic Setting	14
3.3.1	The Microscopic Model	15
3.3.2	The Scaling Limit in a Dynamic Setting	16
4	Numerical Investigation	19
4.1	Data Set Description	19
4.2	Modelling event arrival times using Hawkes Processes	20
4.3	Parameter Estimation	21
4.4	Simulating the Mesoscopic Model	24
4.4.1	The Model	24
4.4.2	Numerical Setup	25
4.4.3	Parameter Estimation	27
4.4.4	Results	30
4.4.5	A More Volatile Data Set	32
5	Conclusion	35
5.1	Conclusion	35
5.2	Further Research	35

A	37
A.1 Proofs of Theorems in Section 2.3	37
A.2 Mesoscopic Model in Four Dimensions	41
B	44
B.1 Goodness of Fit of Estimated Hawkes Processes	44
B.2 Simulation Results	44
B.3 Estimated Parameters and Simulation Results for March Data Set	46
B.3.1 Estimated Drift and Covariance Parameters	46
B.3.2 Simulation Results	46
C Code	49
C.1 Maximum Likelihood Estimation	49
Bibliography	63

Section 1

Introduction

Electronic, order-driven markets are becoming evermore popular and with limit order books at their core, a rich literature focused on building robust models for the limit order book (LOB) has been developed. The potential advantages of developing a meaningful model of the dynamics of LOBs are extremely broad: from gaining clearer insight into the role of supply and demand in price dynamics, to informing the design of electronic trading algorithms and optimal execution strategies. Abergal et al. give a comprehensive overview of the theoretical and empirical literature available [3].

LOB models typically come from one of two independent schools of thought. The first is based on a *perfect-rationality* approach. Pioneered by economists, this approach assumes all market participants employ optimal strategies when placing limit orders. A review of models in this framework can be found in [32]. The second approach, initiated by physicists and mathematicians, is the *zero-intelligence* approach, where the order book is treated as a purely stochastic system, and no strategic order placement is taken into account. While there is no clear consensus on which is the best approach, there has been some attempt to bridge the gap between them, for an example, see Section 3.2 in [24]. This dissertation falls into the framework of the zero-intelligence approach.

Finding a large-scale diffusive limit of the LOB process has been given a lot attention in recent literature and several recent papers have developed stochastic partial differential equation models to describe the evolution of the LOB or its price dynamics ([11], [24]), [18], [13]). Much of the literature is concerned with modelling the first queue of the LOB, as this is where the majority of the activity occurs. The rest of the book is disregarded and when the queue is depleted, its new value is taken from some stationary distribution. The question of modelling the entire profile of the order book is, however, also worth exploring, and there have been several efforts to answer this question in recent years. In [8] and [25], functional central limit theorems for both sides of the book are developed and used to obtain ODE-PDE models for the price-volume process. Similar approaches are also taken in [13] and [28] to obtain SPDE models for the order book profiles. Hambly et al. start from the framework of understanding the LOB as a multi-class queuing system (as in [1], [10], [24]) and find the scaled limit of this system in time and space to arrive at another SPDE model for the dynamics of the book.

There is significant empirical evidence that the rate of order arrivals depends on the state of the book at the time of the arrival, and many of the models discussed above include this dependence in the dynamics of order arrivals. There is, however, also empirical evidence of clustering of order arrivals and interdependence between the arrival rates of orders across different types and queues (see, for example, [23] and [24]), and these phenomena are often not taken into account.

Hawkes processes, first introduced in [19], provide a natural and tractable extension of the Poisson process that enable us to model the clustering and cross-dependencies of events. Their application in limit order book modelling has been studied extensively, see [5] or [21] for a review. Another topic of interest in the last decade has been the diffusion limit of Hawkes processes (see [2], [4], and [27]), and Hawkes processes with state-dependent intensities have also been the topic of several recent papers ([33], [31] and [26]). In [3], Abergel et al. combine the ergodic theory of Markov processes with Martingale convergence theorems to derive a large scale limit of the price process when driven by an exponential, state-dependent Hawkes processes. In [26], Hawkes random measures are used to obtain an SDE-ODE model for the limit order book. Other than this, little has been done in the way of finding an SPDE model for the limit order book when the underlying order arrivals are driven by Hawkes processes. The aim of this dissertation is to investigate the possibility of extending the SPDE model in [18] to include event arrival rates driven by Hawkes processes.

The organisation of this dissertation is as follows: in Section 2, we introduce Hawkes processes and develop the theory necessary to investigate Hawkes processes as the driving processes of LOB events. In Section 3, we define limit order books in more detail and we extend the work done in [18] to include clustering and cross-dependencies of order book events. Finally, in Section 4, we give empirical justification for our use of the Hawkes process. In this section, we will also implement our model numerically using data from LOBSTER, and we will demonstrate its ability to reproduce reasonable price series and order book profiles.

Section 2

Hawkes Processes

In this chapter, we will state some basic definitions and present some of the main results about Hawkes processes that will be used in what follows. Throughout this dissertation we will work on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ satisfying the usual conditions.

2.1 Definitions and Preliminary Results

We begin by giving some basic definitions about *point processes*, referring the reader to [15] for more detail.

Definition 2.1.1. A point process is a stochastic process $(T_k)_{k \in \mathbb{N}}$ taking values in \mathbb{R}_+ such that for every $k \in \mathbb{N}$, $T_k \leq T_{k+1}$.

Definition 2.1.2. Given a point process $(T_k)_{k \in \mathbb{N}}$, the process defined by

$$N(t) = \sum_{k \in \mathbb{N}} \mathbb{1}_{\{T_k \leq t\}}$$

is the counting process associated with $(T_k)_{k \in \mathbb{N}}$.

An important characteristic of the counting process is its so-called *conditional intensity*:

Definition 2.1.3. The conditional intensity of a counting process $N(t)$ is defined to be

$$\lambda(t) := \lim_{\Delta \rightarrow 0} \mathbb{E}[N(t + \Delta) - N(t) | \mathcal{F}_t],$$

where the filtration \mathcal{F}_t is the filtration generated by the counting process, i.e. $\mathcal{F}_t := \sigma(N(s) : 0 \leq s \leq t)$.

Another important quantity in the study of point processes is the point-wise integral of the conditional intensity of the process, $\Lambda(t)$, i.e.

$$\Lambda(t) = \int_0^t \lambda(s) ds. \tag{2.1}$$

We call this value the *compensator* of the process, and we have the following useful result about compensators (Lemma 7.2.V in [15]):

Theorem 2.1.1. Suppose the \mathcal{F}_t -adapted counting process $(N(t))_{t \in [0, \infty)}$ has left-continuous intensity $\lambda(t)$. Then, the process $M(t) = N(t) - \Lambda(t)$ is an \mathcal{F} -martingale, i.e. for every $0 \leq s \leq t$,

$$\mathbb{E}[M(t)|\mathcal{F}_s] = M(s).$$

We also have several notions of regularity for counting processes, namely *stationarity*, *simplicity* and *orderliness*. A counting process N is (strictly) stationary if $N(t+s) - N(s)$ has the same distribution for all $s \geq 0$. N is simple when

$$\mathbb{P}(N(\{t\}) = 0 \text{ or } 1 \text{ for all } t) = 1, \quad (2.2)$$

and it is orderly when it is stationary and

$$\mathbb{P}(N(t+h) - N(t) \geq 2) = o(h) \text{ as } h \rightarrow 0. \quad (2.3)$$

One of the most commonly encountered counting processes is a *Poisson process*, which is defined via its intensity as follows:

Definition 2.1.4. Let N be a counting process and consider a sequence $(s_i, t_i]$, of half-open intervals with $0 \leq s_i < t_i \leq s_{i+1}$, $i \in \{1, \dots, k\}$. N is called a Poisson process with intensity $\lambda > 0$ if

$$\mathbb{P}[N(t_i) - N(s_i) = n_i, i \in \{1, \dots, k\}] = \prod_{i=1}^k \frac{[\lambda(s_i - t_i)]^{n_i}}{n_i!} e^{-\lambda(s_i - t_i)}.$$

It follows from this definition that a Poisson process satisfies the following two properties:

- (i) The number of events in an interval $[s, t)$, $0 \leq s < t$, follows a Poisson distribution with mean $\lambda(t - s)$,
- (ii) the process has *independent increments*, i.e. the number of arrivals in disjoint intervals are independent.

Remark 2.1.2. The Poisson process is also often defined as a counting process that satisfies properties (i) and (ii) above.

It is natural to extend the Poisson process to allow its intensity to depend on time or to be random. These extensions are called *non-homogeneous Poisson* and *Cox* processes, respectively.

An important property of the Poisson process is that it is *memoryless*, i.e. the distribution of future arrival times depends only on the information available at the current time, and not on what has happened in the past. Because of this, the Poisson process fails to capture the clustering (self-exciting) and interdependent (cross-exciting) behaviour of order book events. To address this, we will consider a natural and tractable extension of the Poisson process: the *Hawkes process*.

The Hawkes process was first defined in [19] as a counting process whose intensity function is a linear regression on the past of the process, i.e.

$$\lambda(t) = \mu + \int_0^t \phi(t-s) dN_s,$$

where $\mu \in \mathbb{R}_+$ is the exogenous intensity of the process and $\phi(t) : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ is the kernel function of the process.

In one dimension, this process is self-exciting, i.e. the arrival of an event increases the likelihood of observing events in the near future. It is also useful to consider the case when there is more than one type of event, and there is mutual excitement between the different events. For d such events, we define a d -dimensional Hawkes process:

Definition 2.1.5. A d -dimensional counting process $N = (N_1, \dots, N_d)$ is a multivariate Hawkes process when

$$\lambda_i(t) = \mu_i + \sum_{j=1}^d \int_0^t \phi_{ij}(t-s) dN_{j,s}, \quad (2.4)$$

where $\mu_i \in \mathbb{R}_+$ are called the exogenous intensities and the functions ϕ_{ij} are called the (exciting) kernel functions of the process and satisfy:

- (i) $\phi_{ij}(t) \geq 0$ for all $t \in \mathbb{R}$ and $i, j \in \{1, \dots, d\}$,
- (ii) $\phi_{ij}(t) = 0$ for all $t < 0$ and $i, j \in \{1, \dots, d\}$, and
- (iii) ϕ_{ij} is locally integrable for all $i, j \in \{1, \dots, d\}$, i.e. $\phi_{ij} \in L_{1,loc}(\Omega)$.

As in [5], we present two alternative notations of (2.4). The first is the more compact *convolutional* notation:

$$\boldsymbol{\lambda}(t) = \boldsymbol{\mu} + \boldsymbol{\Phi} * d\mathbf{N}_t, \quad (2.5)$$

where $\boldsymbol{\lambda} = (\lambda_i)_{i \in \{1, \dots, d\}}$, $\boldsymbol{\mu} = (\mu_i)_{i \in \{1, \dots, d\}}$, $\boldsymbol{\Phi} = (\phi_{ij})_{i, j \in \{1, \dots, d\}}$, and $*$ corresponds to matrix multiplication in which ordinary products are replaced by convolutions. Secondly, we can consider (2.4) in terms of event times: if we take $t_{j,r}$ to be the time of the j^{th} event of type r , equation (2.4) becomes

$$\lambda_i(t) = \mu_i + \sum_{j=1}^d \sum_{t_{j,r} \leq t} \phi_{ij}(t - t_{j,r}).$$

In order to ensure stationarity of the processes defined in Definition 2.1.5, we have the following well-known stability condition (Proposition 2.1 in [6]):

Proposition 2.1.3. Let \mathbf{K} be the matrix defined by

$$\mathbf{K} := \int_0^\infty \boldsymbol{\Phi}(t) dt = \left(\int_0^\infty \phi_{ij}(t) dt \right)_{1 \leq i, j \leq d}. \quad (2.6)$$

If \mathbf{K} has a spectral radius strictly smaller than one, then N has stationary increments.

When this condition is satisfied, we have another useful result (Proposition 2.2 in [6]):

Proposition 2.1.4. The average intensity vector $\Lambda = (\Lambda_1, \dots, \Lambda_d) := (\mathbb{E}[\lambda_1(t)], \dots, \mathbb{E}[\lambda_d(t)])$ of N satisfies

$$\Lambda = (\mathbf{Id} - \mathbf{K})^{-1} \boldsymbol{\mu},$$

where $\boldsymbol{\mu} = (\mu_1, \dots, \mu_d)$.

An alternative description of the Hawkes process as a cluster process was offered in [20]: the arrival times of the process are interpreted as the arrival of migrating individuals and the birth of new individuals of type $i \in \{1, \dots, d\}$ in a population. Migrants of type i arrive according to a Poisson process with intensity μ_i . Each individual can give birth to children of all other types. Individuals of type j who were born (or migrated into the population) at time t give birth to children of type i according to a non-homogeneous Poisson distribution with rate $\phi_{ij}(\cdot - t)$. This representation of the Hawkes process allows for the following interpretation of its parameters:

- (i) $\int_0^\infty \phi_{ij}(t) dt$ is the average number of children of type i that an individual of type j will give birth to, and
- (ii) $\frac{\Lambda_i}{\Lambda_j} \int_0^\infty \phi_{ij}(t) dt$ is the proportion of individuals of type i whose parent is of type j .

In this dissertation, we will focus on the Hawkes process in its simplest form, i.e. with kernel functions of the form

$$\phi_{ij}(t) = \alpha_{ij} e^{-\beta_{ij} t}.$$

In this case, we call $N(t)$ an *exponential* Hawkes process and the process $(N(t), \lambda(t))$ is a Markov process (see Proposition 2 in [5]). We will also follow [33] and extend this process by allowing the exogenous intensity to depend on a càdlàg state process $q(t)$, so we consider a counting process $N = (N^1, \dots, N^d)$ with conditional intensity

$$\lambda_i(t) = \mu_i(q(t^-)) + \sum_{j=1}^d \int_0^t \alpha_{ij} e^{-\beta_{ij}(t-s)} dN_{j,s}. \quad (2.7)$$

2.2 Parameter Estimation

When using a Hawkes process to model event times, we will estimate parameters using maximum likelihood estimation (MLE). The log-likelihood function $L(\theta)$ of a Hawkes process is given by the following theorem (Theorem 7.3.III in [15]):

Theorem 2.2.1. Let $t = (t_k, e_k)_{k \in \{1, \dots, n\}}$, $n \in \mathbb{N}$, be a realisation of a d -dimensional Hawkes process $N = (N^1, \dots, N^d)$ on the interval $[0, T]$, for some $T > 0$. Here $t_1 < t_2 < \dots < t_n$ and $e_k \in \{1, \dots, d\}$, for $k \in \{1, \dots, n\}$. Then the log-likelihood function of the Hawkes process is given by

$$L(t|\theta) = \sum_{i=1}^d L_i(t|\theta), \text{ where } L_i(t|\theta) = \int_0^T \log(\lambda^i(t, \theta | \mathcal{F}_t)) dN_t^i - \int_0^T \lambda^i(t, \theta | \mathcal{F}_t) dt. \quad (2.8)$$

Here θ represents the set of parameters and λ_i is the intensity function of the i^{th} component of N .

In the case of state-dependent exponential Hawkes processes with conditional intensity given by (2.7), the likelihood function reduces to

$$\begin{aligned}
L(t|\theta) &= \sum_{i=1}^d \int_0^T \log \left(\mu_i(q(t^-)) + \sum_{j=1}^d \int_0^t \alpha_{ij} e^{-\beta_{ij}(t-s)} dN_{j,s} \right) dN_t^i \\
&\quad - \sum_{i=1}^d \int_0^T \left(\mu_i(q(t^-)) + \sum_{j=1}^d \int_0^t \alpha_{ij} e^{-\beta_{ij}(t-s)} dN_{j,s} \right) dt \\
&= \sum_{i=1}^d \sum_{k=1}^{N^i} \log \left(\mu_i(q(t_k^{i,-})) + \sum_{j=1}^d \sum_{t_l^j < t_k^i} \alpha_{ij} e^{-\beta_{ij}(t_k^i - t_l^j)} \right) \\
&\quad - \sum_{i=1}^d \int_0^T \left(\mu_i(q(t^-)) + \sum_{j=1}^d \sum_{t_l^j < t} \alpha_{ij} e^{-\beta_{ij}(t - t_l^j)} \right) dt.
\end{aligned} \tag{2.9}$$

Here, N^i denotes the total number of events of type i that occur in the interval $[0, T]$ and t_k^i denotes the k^{th} event of type i . We will optimise this function to obtain the maximum likelihood estimates of the parameters. To do this, we will also need to calculate the gradients of this function with respect to each of these parameters:

$$\frac{\partial L}{\partial \mu_i(q)} = \sum_{k=1}^{N^i} \frac{\mathbb{1}_{\{q(t_k^{i,-}) = q\}}}{\mu_i(q(t_k^{i,-})) + \sum_{j=1}^d \sum_{t_l^j < t_k^i} \alpha_{ij} e^{-\beta_{ij}(t_k^i - t_l^j)}} - \int_0^T \mathbb{1}_{\{q(t_k^{i,-}) = q\}} dt, \tag{2.10}$$

$$\frac{\partial L}{\partial \alpha_{ij}} = \sum_{k=1}^{N^i} \frac{\sum_{t_l^j < t_k^i} e^{-\beta_{ij}(t_k^i - t_l^j)}}{\mu_i(q(t_k^{i,-})) + \sum_{j=1}^d \sum_{t_l^j < t_k^i} \alpha_{ij} e^{-\beta_{ij}(t_k^i - t_l^j)}} - \int_0^T \sum_{t_l^j < t} e^{-\beta_{ij}(t - t_l^j)} dt, \text{ and} \tag{2.11}$$

$$\frac{\partial L}{\partial \beta_{ij}} = \sum_{k=1}^{N^i} \frac{-\alpha_{ij}(t_k^i - t_l^j) \sum_{t_l^j < t_k^i} e^{-\beta_{ij}(t_k^i - t_l^j)}}{\mu_i(q(t_k^{i,-})) + \sum_{j=1}^d \sum_{t_l^j < t_k^i} \alpha_{ij} e^{-\beta_{ij}(t_k^i - t_l^j)}} + \int_0^T \sum_{t_l^j < t} \alpha_{ij}(t - t_l^j) e^{-\beta_{ij}(t - t_l^j)} dt. \tag{2.12}$$

The following theorem (Theorem 7.4.VI in [15]) forms the basis of one of our assessment of how well the estimated model fits the data:

Theorem 2.2.2. *Let $(t_i)_{i \in \mathbb{N}}$ be an unbounded, increasing sequence of time points in \mathbb{R}_+ . Let N be a simple point process with compensator $\Lambda(t)$ (as defined in (2.1)) such that $\Lambda(t) \rightarrow \infty$ as $t \rightarrow \infty$, almost surely. Then, with probability one, the transformed sequence $(\tau_i = \Lambda(t_i))$ is a realization of a unit-rate Poisson process if, and only if, the original sequence (t_i) is a realization from the point process defined by $\Lambda(t)$.*

In particular, for a d -dimensional exponential Hawkes process with state-dependent exogenous intensities, we have the following corollary (Proposition 3.2.2 in [9]).

Corollary 2.2.3. Let (N^1, \dots, N^d) be a d -dimensional Hawkes process with intensities λ_i defined by (2.7) for each $i \in \{1, \dots, d\}$. Let t_k^i denote the arrival time of the k^{th} event of type i . Then the sequences $(\tau_k^i)_{k \in \mathbb{N}}$ defined by

$$\begin{aligned} \tau_k^i &= \Lambda(t_k^i) - \Lambda(t_{k-1}^i) \\ &= \int_{t_{k-1}^i}^{t_k^i} \lambda_i(s) ds \\ &= \int_{t_{k-1}^i}^{t_k^i} \left(\mu_i(q(s^-)) + \sum_{j=1}^d \sum_{t_l^j < s} \alpha_{ij} e^{-\beta_{ij}(s-t_l^j)} \right) ds \end{aligned} \tag{2.13}$$

are independent sequences of i.i.d. exponential random variables with mean 1.

2.3 A Scaling Limit

In Section 3.2.2, we will consider a scaling limit of a limit order book model driven by a state-dependent exponential Hawkes process. We will make use of the technical results presented in this section.

We consider a d -dimensional process $N = (N_t^1, \dots, N_t^d)_{t \geq 0}$ with intensity defined by (2.7). We extend the results of [4] to include the state dependence of the exogenous drift in (2.7). For completeness, the proofs of these theorems are included in Appendix A.1, but only minor modifications are needed to include the state-dependent drift. This is because the state dependence is included only in the exogenous intensity and not in the kernel functions. A similar result can also be found in [3].

Let

$$\hat{\boldsymbol{\mu}} = [\hat{\mu}_1, \dots, \hat{\mu}_d]^T := \mathbb{E}(\boldsymbol{\mu}) = [\mathbb{E}(\mu_1), \dots, \mathbb{E}(\mu_d)]^T. \tag{2.14}$$

We let the matrices $\boldsymbol{\Phi}$ and \mathbf{K} be as defined in (2.5) and (2.6), respectively. In what follows, we will also assume that the conditions in Proposition 2.1.3 are satisfied, i.e. that the spectral radius of \mathbf{K} is less than one.

Under these conditions, we have the following Law of Large Numbers, a slight adaptation of Theorem 1 in [4]:

Theorem 2.3.1. *In the setting described above, $N_t \in L^2(\mathbb{P})$ for all $t \geq 0$ and*

$$\sup_{t \in [0,1]} \|n^{-1} N_{nt} - t(\mathbf{Id} - \mathbf{K})^{-1} \hat{\boldsymbol{\mu}}\| \rightarrow 0 \text{ as } n \rightarrow \infty$$

almost surely and in $L^2(\mathbb{P})$.

We also have the following Functional Central Limit Theorem (Corollary 1 in [4]):

Theorem 2.3.2. *The re-scaled process*

$$\sqrt{n} \left(\frac{1}{n} N_{nt} - t(\mathbf{Id} - \mathbf{K})^{-1} \hat{\boldsymbol{\mu}} \right), \quad t \in [0, 1]$$

converges in law for the Skorokhod topology to

$$(\mathbf{Id} - \mathbf{K})^{-1} \Sigma^{\frac{1}{2}} W_t, t \in [0, 1],$$

as $n \rightarrow \infty$, where $(W_t)_{t \in [0,1]} = (W_t^1, \dots, W_t^d)_{t \in [0,1]}$ is a standard d -dimensional Brownian motion and Σ is the diagonal matrix defined by

$$\Sigma_{ii} = ((\mathbf{Id} - \mathbf{K})^{-1} \hat{\boldsymbol{\mu}})_i. \tag{2.15}$$

Section 3

Limit Order Books

In this section, we will introduce limit order books and formulate our model for the dynamics of the limit order book.

3.1 Preliminaries

We begin by introducing limit order books and presenting the mathematical framework that we will use to describe the limit order book. For more details, we refer the reader to [17].

Limit order books are the mechanism for matching buyers and sellers in an *order-driven* market. In a quote-driven market, market makers publish prices at which they are willing to buy or sell a traded asset. These are the only prices available to traders who wish to buy or sell the asset, so one can think of a market maker as a liquidity provider. The market maker makes a profit by charging a premium for this provision. In an order driven market, on the other hand, each trader is able to post *limit orders*. This increase in flexibility for the trader means that order-driven markets are becoming increasingly popular.

Definition 3.1.1. A buy (resp. sell) limit order submitted at time t with price p and size ω is a commitment to sell (resp. buy) up to ω units of a traded asset at a price no less (resp. no more) than p .

For a given limit order book, the units of price and order size are known as the *tick* and *lot size*, respectively. We denote the tick size by θ and the lot size by σ , and together we call these the *resolution parameters* of the book. All limit orders must have a size that is an integer multiple of the σ and a price specified to the accuracy of θ .

When a buy (resp. sell) limit order is posted, the book's *trade matching algorithm* determines whether there is a pre-existing sell (resp. buy) limit order already on the book to which it can be matched. If so, the orders are matched. Orders that are immediately matched to a limit order are known as *market orders*. If the order cannot be matched, it becomes *active* and is recorded in the limit order book until it is matched or *cancelled*. The instruction to cancel an existing limit order is called a *cancellation order*. This leads us to our formal definition of the limit order book:

Definition 3.1.2. The limit order book $\mathcal{L}(t)$ is the collection of all active limit orders in a market at time t .

To fully characterise the state of the limit order book at time t , we also need to specify the *best bid* and *best ask* prices. The best bid (resp. best ask) price, $b(t)$ (resp. $a(t)$), is the highest (resp. lowest) price at which there is an active buy (resp. sell) limit order at time t . The average of these two prices is known as the *mid*, $m(t)$, and the difference between them is called the *bid/ask spread*, denoted $s(t)$. Note that the mid is not necessarily a multiple of the tick size, θ , but the spread is.

Typically, we partition the limit order book into the set of active buy orders, $\mathcal{B}(t)$, and active sell orders, $\mathcal{S}(t)$. Throughout, we will also refer to $\mathcal{B}(t)$ and $\mathcal{S}(t)$ as the bid and ask sides of the book, respectively. Together with the discretisation achieved by imposing a tick size, this leads to a natural interpretation of a limit order book as a set of queues, each consisting of active buy or sell limit orders at a given price. The queues can be characterised by the distance of the given price from the best bid/ask price (depending on the side of the book). We define the i^{th} queue on the bid (resp. ask) side of the book at time t to be the set of active buy (resp. sell) limit orders with price $p = b(t) - i\theta$ (resp. $p = a(t) + i\theta$). We will denote the number of active buy (resp. sell) limit orders in the i^{th} queue on the bid (resp. ask) side of the book at time t by $X_{i,t}^b$ (resp. $X_{i,t}^a$).

In a limit order book, the price evolution depends on the way orders are matched. Consider a buy (resp. sell) limit order x placed immediately after time t with price p and size ω . The arrival of this order can affect the state of the limit order book in several ways:

- (i) If $p \leq b(t)$ (resp. $p \geq a(t)$), then x is a limit order that becomes active on arrival and does not change the price.
- (ii) If $p > b(t)$ (resp. $p < a(t)$), then x is a limit order that becomes active on arrival. The price change caused by this order is $b(t^+) = p$ (resp. $a(t^+) = p$).
- (iii) If $p \geq a(t)$ (resp. $p \leq b(t)$), then x is a market order and is executed immediately upon arrival. Whenever such an order arrives, it is matched to the price of an active order, even if this is not the same as the price of the incoming order. Furthermore, whether this order induces a price change depends on the size of the order: the bid (resp. ask) price upon arrival of a sell (resp. buy) market order is $b(t^+) = \max(p, i^*\theta)$, where

$$i^* = \arg \max_i \sum_{k=0}^i X_{k,t}^b > \omega$$

$$\text{(resp. } a(t^+) = \max(p, i^*\theta), \text{ where } i^* = \arg \max_i \sum_{k=0}^i X_{k,t}^a > \omega).$$

3.2 Limit Order Book Dynamics in a Static Setting

3.2.1 The Microscopic Order Book Model

We begin by specifying the dynamics of the *microscopic* (discrete volume and price) order book model in a static setting, i.e. we give the discrete dynamics of the order book model between price changes,

as was done in [18] and [24]. We take $m \in \mathbb{R}$ to be the fixed mid and we adopt the convention that the spread is constantly equal to two ticks. As in [18], we work on a relative price grid given by $\{0, 1, \dots, N\}$, for some $N \in \mathbb{N}$, where the i^{th} point on the grid refers to the point i ticks away from the mid, m . Note the difference between this and the definition of the i^{th} queue in the previous section. We change this definition without losing the convenience of the definition in the previous section as we are now assuming the spread is always two ticks. For each $n \in \mathbb{N}$, we consider two $(N - 1)$ -dimensional processes, $Z_n^b = (Z_n^{b,1}, \dots, Z_n^{b,N-1})$ and $Z_n^a = (Z_n^{a,1}, \dots, Z_n^{a,N-1})$, taking values in \mathbb{Z}_+^{N-1} , where $Z_n^{b,i}$ (resp. $Z_n^{a,i}$) represents the number of outstanding buy (resp. sell) limit orders at price $m - i$ (resp. $m + i$). We assume that order and cancellation sizes are always one. At a given price point on the bid (resp. ask) side of the book, the number of outstanding buy (resp. sell) limit orders can increase with the arrival of a buy (resp. sell) limit order and decrease with the arrival of a sell (resp. buy) market order or the cancellation of an existing buy (resp. sell) limit order.

In this model, we will extend the model in [18] by assuming that the limit orders and cancellation/market orders arrive according to a $2(N - 1)$ -dimensional, state-dependent exponential Hawkes process, instead of independent, state-dependent Poisson processes. On the bid side of the book, we will consider the process

$$N_{b,n}(t) = (N_{b,n,i}^1(t), N_{b,n,i}^2(t), \dots, N_{b,n,N-1}^1(t), N_{b,n,N-1}^2(t)),$$

where $N_{b,n,j}^1(t)$, and $N_{b,n,j}^2(t)$ count the arrival of limit orders and cancellation/market orders at price $m - j$, respectively. For a fixed price level, the process will be both self-exciting and cross-exciting. Between queues, we allow cancellation orders to excite limit orders in the adjacent queue closer to the mid. For example, on the bid side of the book, the arrival of a cancellation order at level $m - (i + 1)$ will excite the arrival of limit orders at level $m - i$. The inter-queue interaction on the ask side of the book is defined analogously. The kernel functions will be zero for all other interactions. This is intended to capture the effects of traders re-positioning their orders, i.e. cancelling an order in one queue and quickly placing another order in the adjacent queue *closer* to the mid. We assume all other cross-exciting effects between the adjacent queues are zero. In this setup, the dynamics of the bid side of the book can be described as follows:

For $i \in \{1, 2, \dots, N - 1\}$, let $e_i = (a_1, a_2, \dots, a_{N-1})$ be such that $a_j = 0$ if $j \neq i$ and $a_i = 1$, and define $e_0 = e_N = 0$. For each i , let $N_{b,n,i}^1$ and $N_{b,n,i}^2$ denote the arrival of limit orders and market/cancellation orders at the i^{th} queue, respectively. Let $N_{b,n,i}^3$ and $N_{b,n,i}^4$ denote the arrival of limit orders and market/cancellation orders at the $(i + 1)^{th}$ queue, respectively. Note that this means that $N_{b,n,i}^1 = N_{b,n,i+1}^3$, but we adopt this notation for clarity below. Using our notation from Section 2.1, let $\alpha_{b,n,i}^{ij} e^{-\beta_{b,n,i}^{ij}(t)}$ denote the kernel function responsible for the exciting effect that the process $N_{b,n,i}^j$ has on $N_{b,n,i}^i$. Then, for $i \in \{1, 2, \dots, N - 1\}$,

(i) $Z_n^b \rightarrow Z_n^b + e_i$ according to the Hawkes process $N_{b,n,i}^1(t)$ with intensity

$$\lambda_{b,n,i}^1(t) = \mu_{b,n,i}^1(Z_n^{b,i}) + \int_0^t \alpha_{b,n,i}^{11} e^{-\beta_{b,n,i}^{11}(t-s)} dN_{b,n,i}^1(s) + \int_0^t \alpha_{b,n,i}^{12} e^{-\beta_{b,n,i}^{12}(t-s)} dN_{b,n,i}^2(s) + int_0^t \alpha_{b,n,i}^{14} e^{-\beta_{b,n,i}^{14}(t-s)} dN_{b,n,i}^4(s),$$

(ii) $Z_n^b \rightarrow Z_n^b - e_i$ according to the Hawkes process $N_{b,n,i}^2(t)$ with intensity

$$\lambda_{b,n,i}^2(t) = \left(\mu_{b,n,i}^2(Z_n^{b,i}) + \int_0^t \alpha_{b,n,i}^{21} e^{-\beta_{b,n,i}^{21}(t-s)} dN_{b,n,i}^1(s) + \int_0^t \alpha_{b,n,i}^{22} e^{-\beta_{b,n,i}^{22}(t-s)} dN_{b,n,i}^2(s) \right) \mathbb{1}_{\{Z_n^{b,i} > 0\}},$$

The dynamics for the ask side of the book are defined analogously.

3.2.2 Scaling Limit of the Static Microscopic Model

As in [18], we accelerate time by a factor of n and divide volumes by \sqrt{n} to consider the limits of the $\frac{1}{\sqrt{n}}\mathbb{N}^{N-1}$ -valued processes

$$\tilde{Z}_n^b(t) = \frac{Z_n^b(nt)}{\sqrt{n}} \text{ and } \tilde{Z}_n^a(t) = \frac{Z_n^a(nt)}{\sqrt{n}}.$$

The limits of these processes will define our *mesoscopic* (continuous volume and discrete price) order book model and will take values in $[0, \infty)^{N-1}$. In order to obtain convergence, we need to make the following assumptions:

(i) For $k \in \{a, b\}$, $n \in \mathbb{N}$ and $i \in \{1, \dots, N-1\}$, there exists a constant $C_{i,k}$ such that

$$\sigma_{1,i,k,n}^2 - \sigma_{2,i,k,n}^2 = \frac{C_{i,k}}{\sqrt{n}}, \quad (3.1)$$

where expressions for $\sigma_{1,i,k,n}$ and $\sigma_{2,i,k,n}$ are given in the proof of Theorem 3.2.1.

(ii) For $k \in \{a, b\}$, there exists a \mathbb{R}_+ valued random variable $X^k(0)$ such that $\tilde{Z}_n^k(0)$ converges to $X^k(0)$ in law as $n \rightarrow \infty$.

Theorem 3.2.1. *The $\frac{1}{\sqrt{n}}\mathbb{N}^{N-1} \times \frac{1}{\sqrt{n}}\mathbb{N}^{N-1}$ valued process $(\tilde{Z}_n^b, \tilde{Z}_n^a)$ converges weakly in $\mathcal{M}(\mathbb{D}([0, \infty); \mathbb{R}^{N-1}) \times \mathbb{D}([0, \infty); \mathbb{R}^{N-1}))$ to the unique $[0, \infty)^{N-1} \times [0, \infty)^{N-1}$ -valued strong Markov diffusion process (X^b, X^a) which satisfies the following system of reflected SDEs:*

$$dX_t^{b,i} = C_{i,b} dt + \zeta^{b,i} dW_t^{b,i} + d\eta_t^{b,i},$$

$$dX_t^{a,i} = C_{i,a} dt + \zeta^{a,i} dW_t^{a,i} + d\eta_t^{a,i},$$

for $i \in \{1, \dots, N-1\}$ with the pinning conditions $X^{k,0} = X^{k,N} = 0$, where $W^{k,i}$, $k \in \{a, b\}$, are independent Brownian motions and for $k \in \{a, b\}$ fixed, $\langle W^{k,i}, W^{k,j} \rangle_t = \rho_{k,j} t$ for each $i, j \in \{1, \dots, N-1\}$. The $\eta^{k,i}$ are reflection measures which maintain the positivity of $X^{k,i}$.

Proof. We begin by considering the general M -dimensional Hawkes process $N = (N^1, N^2, \dots, N^M)$, where M is a positive, even integer, and for each $i \in \{1, \dots, M\}$, N^i has conditional intensity

$$\lambda^i(t) = \mu_i(q(t^-)) + \int_0^t \sum_{j=1}^M \alpha^{ij} e^{-\beta^{ij}(t-s)} dN_j(s).$$

Assume that, for $i \in \{1, 3, \dots, M-3\}$, $\alpha_{ij} = 0$ for all $j \notin \{i, i+1, i+3\}$, and for $i \in \{2, 4, \dots, M\}$, $\alpha_{ij} = 0$ whenever $j \notin \{i-1, i\}$. Let Z be the $\frac{M}{2}$ -dimensional process defined by

$$Z_j(t) = N_{2j-1}(t) - N_{2j}(t), \text{ for } j \in \left\{1, \dots, \frac{M}{2}\right\},$$

and let \mathbf{K} , $\hat{\boldsymbol{\mu}}$, and $\boldsymbol{\Sigma}$ be defined as in equations (2.6), (2.14), and (2.15), respectively, i.e.

$$\begin{aligned} \mathbf{K} &:= \int_0^\infty \boldsymbol{\Phi}(t) dt = \left(\int_0^\infty \phi_{ij}(t) dt \right)_{1 \leq i, j \leq M}, \\ \hat{\boldsymbol{\mu}} &= [\hat{\mu}_1, \dots, \hat{\mu}_d]^T := \mathbb{E}(\boldsymbol{\mu}) = [\mathbb{E}(\mu_1), \dots, \mathbb{E}(\mu_M)]^T, \text{ and} \\ \boldsymbol{\Sigma}_{ii} &= ((\mathbf{Id} - \mathbf{K})^{-1} \hat{\boldsymbol{\mu}})_i. \end{aligned}$$

From Theorem 2.3.2, we know that the process $\frac{1}{\sqrt{n}} X_{tn} := \sqrt{n} \left(\frac{1}{n} N_{nt} - t(\mathbf{Id} - \mathbf{K})^{-1} \hat{\boldsymbol{\mu}} \right)$ converges in law to $Y_t := (\mathbf{Id} - \mathbf{K})^{-1} \boldsymbol{\Sigma}^{\frac{1}{2}} W_t$, where $W_t = (W_t^1, \dots, W_t^M)$ is a standard M -dimensional Brownian motion. From this and Condition (3.1), it follows that the $\frac{M}{2}$ -dimensional process $\left(\frac{1}{\sqrt{n}} Z_{tn}^1 - \sqrt{nt}(\sigma_1^2 - \sigma_2^2), \frac{1}{\sqrt{n}} Z_{tn}^2 - \sqrt{nt}(\sigma_3^2 - \sigma_4^2), \dots, \frac{1}{\sqrt{n}} Z_{tn}^{\frac{M}{2}} - \sqrt{nt}(\sigma_{M-1}^2 - \sigma_M^2) \right)$ converges in distribution to $(\zeta_1 \tilde{W}_t^1, \dots, \zeta_{\frac{M}{2}} \tilde{W}_t^{\frac{M}{2}})$, where $\langle \tilde{W}^i, \tilde{W}^j \rangle_t = \rho_{jk} t$, $j, k \in \{1, \dots, \frac{M}{2}\}$. Expressions for the σ_j 's, ζ_j 's, and ρ_{jk} 's in the four-dimensional case can be found in Appendix A.2.

Taking $M = 2(N-1)$ and N^j and N^{j+1} to represent the arrival of limit and market/cancellation orders at the $\left(\frac{j+1}{2}\right)^{th}$ queue of the bid/ask side of the limit order book, respectively, we have convergence of $\tilde{Z}_n^b(t)$ (resp. $\tilde{Z}_n^a(t)$) to $X^b(t)$ (resp. X^a). The reflection measure is a result of the indicator function in the arrival rate of market/cancellation orders which ensures that whenever the volume at a given queue is zero, the next order to arrive will be a limit order. \square

Remark 3.2.2 (The Macroscopic Model in a Static Setting). At this point in [18], the tick size is also made to tend to zero and an SPDE model is derived. Given the inherent dependence structure in the mesoscopic model proposed in Theorem 3.2.1, the theory used in [18] is not applicable in this case, and we do not find the macroscopic limit in this case.

3.3 Limit Order Book Dynamics in a Dynamic Setting

In this section, we present the mechanism for incorporating price movements into the models in Section 3.2. This section follows the methodology in Section 2.3 of [18] closely. Note that the same idea is used in [24] to include price dynamics in the model.

In general, a price change can be triggered by one of the following three events:

- (i) The placement of a limit order inside the bid-ask spread. This will cause the price to decrease/increase if it is a sell/buy limit order placed below/above the best ask/bid price.
- (ii) The cancellation of the last remaining limit order in the first queue on either the bid/ask side of the book. The cancellation of such a sell/buy limit order will cause the price to increase/decrease.
- (iii) The execution of the last remaining limit order at either the best bid/ask price. This corresponds to the placement of a sell/buy market order at the best bid/ask price that consumes all of the available liquidity at that price and has the same effect on the price as the cancellation of the limit order.

3.3.1 The Microscopic Model

We assume that price changes occur according to independent, positive exponential rates, and we allow these rates to depend on the current profile of the book and the mid. In [18], an example is made of the case where these rates depend on the volume imbalance on either side of the book. In this case, relatively more bid limit orders will encourage a price increase, and relatively more sell limit orders will encourage a decrease in price.

For each $n \in \mathbb{N}$, $m \in \mathbb{R}$, we denote the rate of upward and downward price changes by $\theta_{u,m}^n$ and $\theta_{d,m}^n$, respectively, where $\theta_{k,m}^n : \mathbb{N}^{N-1} \times \mathbb{N}^{N-1} \rightarrow \mathbb{R}_+$, $k \in \{u, d\}$. We assume that the price changes are of a fixed, positive amount which we denote by ϵ . We also define the functions $R^n : \mathbb{N}^{N-1} \times \mathbb{N}^{N-1} \times \{u, d\} \rightarrow \mathcal{M}(\mathbb{N}^{N-1} \times \mathbb{N}^{N-1})$. These determine the distribution of the profiles of the bid and ask side of the book following a price change. The functions R will be allowed to depend on the profile of the book at the time of the price change and the direction of the price change.

To construct the dynamic process, we need to introduce the two sequences $(Y_{n,u}^i)_{i \in \mathbb{N}}$ and $(Y_{n,d}^i)_{i \in \mathbb{N}}$ of i.i.d. exponential random variables with mean one. These sequences are independent of one another and of the counting processes that drive the arrival of orders. Now, let $Z_{n,1}^b$ and $Z_{n,1}^a$ be processes evolving according to the dynamics of the n^{th} static microscopic order book model given in Section 3.2.1. Let $Z_{n,1}^b(0)$ and $Z_{n,1}^a(0)$ be the initial profiles of these processes, respectively, and let m_n^1 be the *fixed* initial mid for this model. Consider the stopping times $\tau_{n,u}^i$ and $\tau_{n,d}^i$ defined by

$$\tau_{n,u}^1 := \inf \left\{ t \geq 0 \mid \int_0^t \theta_{u,m_n^1}^n(Z_{n,1}^b(s), Z_{n,1}^a(s)) ds \geq Y_{n,u}^1 \right\}, \text{ and}$$

$$\tau_{n,d}^1 := \inf \left\{ t \geq 0 \mid \int_0^t \theta_{d,m_n^1}^n(Z_{n,1}^b(s), Z_{n,1}^a(s)) ds \geq Y_{n,d}^1 \right\}.$$

For $k \in \{u, d\}$, $\tau_{n,k}^i$ can be interpreted as an exponential waiting time with arrival rate at time t determined by $\theta_{k,m}^n(Z_{n,1}^b(t), Z_{n,1}^a(t))$. Let $\tau_n^1 := \tau_{n,d}^1 \wedge \tau_{n,u}^1$. Assume a price change is triggered when the first of the exponential waiting times is met, i.e. at time τ_n^1 . If $\tau_n^1 = \tau_{n,u}^1$, then the price change is positive and we set $m_n^2 := m_n^1 + \epsilon$. Similarly, when $\tau_n^1 = \tau_{n,d}^1$, the price change is negative and $m_n^2 := m_n^1 - \epsilon$. We then define $(Z_{n,2}^b, Z_{n,2}^a)$ to be the process evolving according to the dynamics of the n^{th} static microscopic order book model with fixed mid m_n^2 . The initial profile of this model is determined by the law

$R^n(Z_{n,1}^b(\tau_n^1), Z_{n,1}^a(\tau_n^1), u)$ if the price change at time τ_n^1 was positive and $R^n(Z_{n,1}^b(\tau_n^1), Z_{n,1}^a(\tau_n^1), d)$ if the price change at time τ_n^1 was negative. We assume that, given $Z_{n,1}^b(\tau_n^1), Z_{n,1}^a(\tau_n^1)$, the direction of the price change at time τ_n^1 and the mid m_n^2 , the processes $Z_{n,2}^b$ and $Z_{n,2}^a$ are independent of the past of the order book. To determine the dynamics of the order book after the M^{th} price change, we proceed inductively as follows:

Given $(Z_{n,i}^b)_{i \in \{1, \dots, M\}}, (Z_{n,i}^a)_{i \in \{1, \dots, M\}}, (\tau_n^i)_{i \in \{1, \dots, M-1\}}$, and $(m_n^i)_{i \in \{1, \dots, M\}}$, we define

$$\begin{aligned} \tau_{n,u}^M &:= \inf \left\{ t \geq 0 \mid \int_0^t \theta_{u, m_n^M}^n(Z_{n,M}^b(s), Z_{n,M}^a(s)) ds \geq Y_{n,u}^M \right\}, \\ \tau_{n,d}^M &:= \inf \left\{ t \geq 0 \mid \int_0^t \theta_{d, m_n^M}^n(Z_{n,M}^b(s), Z_{n,M}^a(s)) ds \geq Y_{n,d}^M \right\}, \text{ and} \end{aligned}$$

$\tau_n^M = \tau_{n,u}^M \wedge \tau_{n,d}^M$. If $\tau_n^M = \tau_{n,u}^M$, let $m_n^{M+1} = m_n^M + \epsilon$ and if $\tau_n^M = \tau_{n,d}^M$, let $m_n^{M+1} = m_n^M - \epsilon$. The dynamics of the order book after the M^{th} price change are then given by $(Z_{n,M+1}^b, Z_{n,M+1}^a)$, which evolve according to our n^{th} microscopic model with mid m_n^{M+1} . If the M^{th} price change was positive, the initial profile of this process has law $R^n(Z_{n,M}^b(\tau_n^M), Z_{n,M}^a(\tau_n^M), u)$, and if it was negative, the initial profile has law $R^n(Z_{n,M}^b(\tau_n^M), Z_{n,M}^a(\tau_n^M), d)$. Again, given $Z_{n,M}^b(\tau_n^M), Z_{n,M}^a(\tau_n^M)$, the direction of the price change at time τ_n^M , and the mid m_n^{M+1} , the processes $Z_{n,M+1}^b$ and $Z_{n,M+1}^a$ are independent of the past of the order book.

Putting all of these processes together, we get that our n^{th} dynamic microscopic order book model can be described by the $\mathbb{N}^{N-1} \times \mathbb{N}^{N-1} \times \mathbb{R}$ valued process $(\hat{Z}_n^b(t), \hat{Z}_n^a(t), m_n(t))$. Here,

$$\hat{Z}_n^b(t) = \sum_{i \in \mathbb{N}} Z_{n,i}^b \left(t - \sum_{j=1}^{i-1} \tau_n^j \right) \mathbb{1}_{\left\{ \sum_{j=1}^{i-1} \tau_n^j \leq t < \sum_{j=1}^i \tau_n^j \right\}},$$

and

$$\hat{Z}_n^a(t) = \sum_{i \in \mathbb{N}} Z_{n,i}^a \left(t - \sum_{j=1}^{i-1} \tau_n^j \right) \mathbb{1}_{\left\{ \sum_{j=1}^{i-1} \tau_n^j \leq t < \sum_{j=1}^i \tau_n^j \right\}}$$

describe the evolution of the bid and ask sides of the book through time, respectively, and

$$m_n(t) = \sum_{i \in \mathbb{N}} m_n^i \mathbb{1}_{\left\{ \sum_{j=1}^{i-1} \tau_n^j \leq t < \sum_{j=1}^i \tau_n^j \right\}}$$

describes the evolution of the mid through time.

3.3.2 The Scaling Limit in a Dynamic Setting

In this section, we define the dynamic analogy to our mesoscopic model in Section 3.2.2 and we show that our dynamic microscopic model converges to it. The dynamic mesoscopic model is defined in almost the same way as the dynamic microscopic model. Upward and downward price movements occur according to positive exponential rates $\theta_{u,m}$ and $\theta_{d,m}$, respectively. In this case, for $k \in \{u, d\}$, the functions $\theta_{k,m} : (\mathbb{R}_+)^{N-1} \times (\mathbb{R}_+)^{N-1} \rightarrow \mathbb{R}_+$, and they are allowed to depend on the profile of the book

and the current mid. Furthermore, we assume that these functions are continuous, uniformly bounded over m , and that there is some $c > 0$ such that $\theta_{k,m} \geq c$. The dynamics of the order book between price changes are determined by the static mesoscopic model in Section 3.2.1. In this case, the law of the order book profile after a price change is determined by the continuous function $R : (\mathbb{R}_+)^{N-1} \times (\mathbb{R}_+)^{N-1} \times \{u, d\} \rightarrow \mathcal{M}((\mathbb{R}_+)^{N-1} \times (\mathbb{R}_+)^{N-1})$, and we equip the space $\mathcal{M}((\mathbb{R}_+)^{N-1} \times (\mathbb{R}_+)^{N-1})$ with the topology of weak convergence.

Let $k \in \{u, d\}$ and define $P_n : \mathbb{N}^{N-1} \times \mathbb{N}^{N-1} \rightarrow (\mathbb{R}_+)^{N-1} \times (\mathbb{R}_+)^{N-1}$ by

$$\left((v_1^1, \dots, v_1^{N-1}), (v_2^1, \dots, v_2^{N-1}) \right) \mapsto \left(\left(\frac{v_1^1}{\sqrt{n}}, \dots, \frac{v_1^{N-1}}{\sqrt{n}} \right), \left(\frac{v_2^1}{\sqrt{n}}, \dots, \frac{v_2^{N-1}}{\sqrt{n}} \right) \right).$$

We assume that the functions defined above can be approximated by their microscopic counterparts in the following ways:

- (i) For $v_1, v_2 \in \mathbb{N}^{N-1}$, and $u_1, u_2 \in (\mathbb{R}_+)^{N-1} \times (\mathbb{R}_+)^{N-1}$,

$$\left| n\theta_{k,m}^n(v_1, v_2) - \theta_{k,m} \right| \leq r \left(\|P_n((v_1, v_2) - (u_1, u_2))\| \right),$$

where $\lim_{x \rightarrow 0} r(x) = 0$.

- (ii) If (v_1^n, v_2^n) is a sequence in $\mathbb{N}^{N-1} \times \mathbb{N}^{N-1}$ such that $P_n((v_1^n, v_2^n)) \rightarrow (u_1, u_2)$, then

$$R^n(v_1^n, v_2^n, k) \circ P_n^{-1} \rightarrow R(u_1, u_2, k)$$

in law in $\mathcal{M}((\mathbb{R}_+)^{N-1} \times (\mathbb{R}_+)^{N-1})$.

Now, define the processes X_i^b, X_i^a , the i.i.d sequences (Y_u^i) and (Y_d^i) , the stopping times $\tau_u^i, \tau_d^i, \tau^i$ and the price sequence m^i as we did in Section 3.3.1. Note that, in this case, our sequences of exponential random variables do not depend on n . We then have the dynamic mesoscopic model $(\hat{X}^b(t), \hat{X}^a(t), m(t))$, where

$$\begin{aligned} \hat{X}^b(t) &= \sum_{i \in \mathbb{N}} X_i^b \left(t - \sum_{j=1}^{i-1} \tau_n^j \right) \mathbb{1}_{\left\{ \sum_{j=1}^{i-1} \tau_n^j \leq t < \sum_{j=1}^i \tau_n^j \right\}}, \\ \hat{X}^a(t) &= \sum_{i \in \mathbb{N}} X_i^a \left(t - \sum_{j=1}^{i-1} \tau_n^j \right) \mathbb{1}_{\left\{ \sum_{j=1}^{i-1} \tau_n^j \leq t < \sum_{j=1}^i \tau_n^j \right\}}, \end{aligned}$$

and

$$m(t) = \sum_{i \in \mathbb{N}} m^i \mathbb{1}_{\left\{ \sum_{j=1}^{i-1} \tau_n^j \leq t < \sum_{j=1}^i \tau_n^j \right\}}.$$

We can then use Theorem 2.4 in [18] to obtain convergence of our re-scaled dynamic microscopic model to this mesoscopic model:

Theorem 3.3.1. *Suppose that*

$$\left(\frac{Z_{n,1}^b(0)}{\sqrt{n}}, \frac{Z_{n,1}^a(0)}{\sqrt{n}} \right) \rightarrow \left(X_1^b(0), X_1^a(0) \right)$$

weakly in $\mathcal{M}((\mathbb{R}_+)^{N-1} \times (\mathbb{R}_+)^{N-1})$. For each $n \in \mathbb{N}$, let $(\hat{Z}_n^b(t), \hat{Z}_n^a(t), m_n(t))$ be a dynamic microscopic model with initial data

$$\left(\frac{Z_{n,1}^b(0)}{\sqrt{n}}, \frac{Z_{n,1}^a(0)}{\sqrt{n}}, m^1 \right),$$

and let $(\hat{X}^b(t), \hat{X}^a(t), m(t))$ be the dynamic mesoscopic model with initial data $(X_1^b(0), X_1^a(0), m^1)$.

Then

$$\left(\frac{\hat{Z}_{n,1}^b(nt)}{\sqrt{n}}, \frac{\hat{Z}_{n,1}^a(nt)}{\sqrt{n}}, m_n(nt) \right) \rightarrow \left(\hat{X}_1^b(t), \hat{X}_1^a(t), m(t) \right)$$

weakly in $\mathcal{M}(\mathbb{D}([0, \infty), \mathbb{R}^{N-1}) \times \mathbb{D}([0, \infty), \mathbb{R}^{N-1}) \times \mathbb{D}([0, \infty), \mathbb{R}))$.

Section 4

Numerical Investigation

4.1 Data Set Description

We will use data from LOBSTER (Limit Order Book System, The Efficient Reconstructor), a database tool developed by frishedaten UG [29]. LOBSTER reconstructs limit order book data for all NASDAQ traded stocks from NASDAQ's Historical TotalView-ITCH files from June 27 2007 until the day before yesterday. The reconstructed data consists of two files for each stock: the *message* file and the *order book* file. The order book file contains snapshots of the state of the order book up to a chosen number of occupied levels, and the message file contains information about the type of event that changes the state of the order book at a given time within the specified price range: it records the placement and cancellation of limit orders, the execution of both *hidden* and *visible* limit orders, and any trading halts and cross trades that occur throughout the day. Note that the execution of a buy (resp. sell) limit order corresponds to a sell (resp. buy) market order. All events are timestamped according to seconds after midnight, with the available decimal precision ranging from milliseconds to nanoseconds depending on the requested period. The k^{th} row of the message book corresponds to the event that causes the limit order book to change from the state recorded in the $(k - 1)^{th}$ row of the order book file to the state recorded in the k^{th} row.

We will consider the data from the SPDR Trust Series I limit order book up to the 50 best occupied levels on each side of the book, between 11:00:00.000 and 12:00:00.000 EST, on 25 September 2019. The tick size for this data set is 1 cent and the time stamps have nanosecond decimal precision.

This data set does not contain any events corresponding to trading halts or cross trades, and we choose to omit the execution of hidden limit orders, as these have no impact on the visible limit order book and only account for a very small proportion, 0.1923%, of the order book events.

There are also several instances in which multiple order book events share the same timestamp. This happens in two cases: in the first case, multiple limit order executions in the same queue on the same side of the book share a timestamp. We interpret this as a single market order fulfilling multiple limit orders, and we reduce this to one limit order execution at the given price level and time stamp, with

order size equal to the sum of the sizes of the orders sharing the timestamp. The second case is the quick succession of the cancellation of limit orders at one price level and the placement/execution of limit orders of the same size at an adjacent level or execution of limit orders at the same price level. As this is an effect we would like to capture in our modelling, we edit the timestamps of these orders by spreading them out evenly across their common nanosecond.

4.2 Modelling event arrival times using Hawkes Processes

The use of the Hawkes process to model the arrival time of limit order book events has been widely explored in the literature. In this section, we will give some empirical evidence for the use of Hawkes processes to model the arrival of events that increase liquidity (limit orders) and events that decrease liquidity (market/cancellation orders).

Figure 4.1 illustrates the clustering effect observed in orders of the same type. The red vertical lines in Figure 4.1(a) correspond to the arrival times of market/cancellation orders placed at the best bid price, where the blue vertical lines correspond to the arrival times of a simulated homogeneous Poisson process. The rate of the Poisson process is chosen so that the expected number of Poisson arrivals is equal to the number of order book events of the appropriate type. The Poisson observations are generated using the `tick` python package [7]. The red and blue step functions correspond to the number of market/cancellation and Poisson events per second, respectively. Figure 4.1(b) can be understood analogously, replacing market/cancellation orders at the best bid price with limit orders at the best ask price. It is clear in both cases that the Poisson arrivals are more evenly spread out across the interval and that there is much less fluctuation in the number of arrivals per second. This clustering is captured by the *self-excitation* component of Hawkes processes.

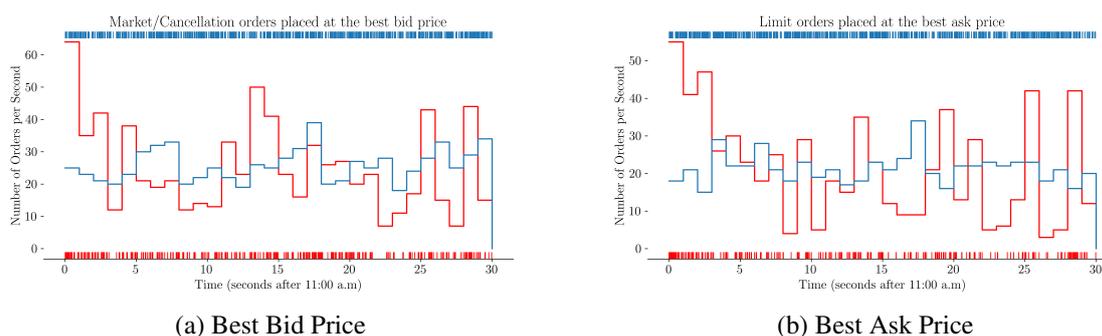


Figure 4.1: Arrival times of orders and number of orders per second.

In Section 3.2.1, we hypothesised that, over and above the self-excitation observed in Figure 4.1, the arrival of limit orders at a given price level depends on the arrival of market/cancellation orders at the same price level and one tick further away from the best bid/ask price. Furthermore, we hypothesised that the arrival of a market/cancellation order at a given price level is excited by the arrival of

both market/cancellation orders and limit orders at that level. In order to establish if there is cross-excitation between order book events of two different types (or across two different levels), we consider the empirical distributions of:

- (i) the inter-arrival times of all events of the two different types, and
- (ii) the time between the arrival of an event of the first type and the subsequent event of the second type.

Figure 4.3(a) (resp. 4.2(b)) compares these empirical distributions for limit orders at the best bid and market/cancellation orders at the best bid (resp. market/cancellation orders one tick away from the best bid). On the other hand, Figure 4.2(a) (resp. 4.3(b)) compares the empirical distributions for market/cancellation orders at the best bid and limit orders at the best bid (resp. market/cancellation orders one tick away from the best bid). In both Figures 4.2 and 4.3, the blue line is the empirical distribution of the inter-arrival times of all events under consideration and the orange line is the empirical distribution of the time between the arrival of an event of the first type and the subsequent event of the other type. The significant increase in the empirical probability of short time intervals in Figures 4.2(a), 4.2(b), 4.3(a), as well as the lack of such an increase in Figure 4.3(b) shows that, in the case of the best bid price level on 25 September 2019, these hypotheses are empirically justified. Similar observations can be made for the ask side of the book and for deeper levels of the order book.

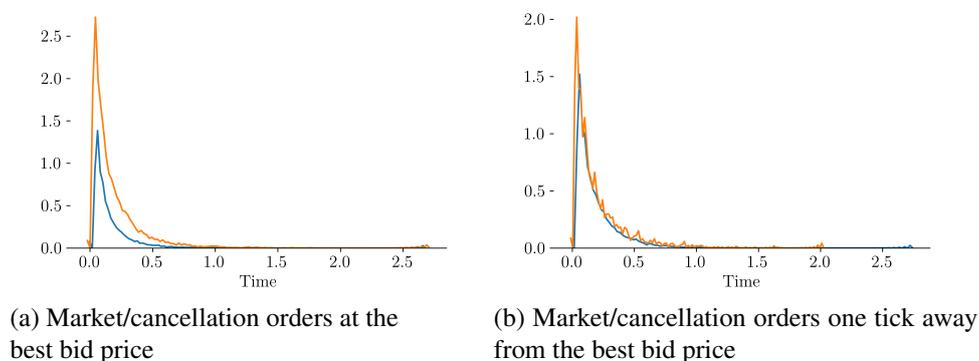


Figure 4.2: Comparison of empirical distributions for limit orders at the best bid price.

4.3 Parameter Estimation

Based on the empirical observations in Section 4.2, we fit the Hawkes processes described in Section 3.2.1 to our data set.

To demonstrate the validity of the assumptions made in Section 3.2.1, we will describe the estimation procedure and present the results for the first three levels of the book. We consider the arrival times of

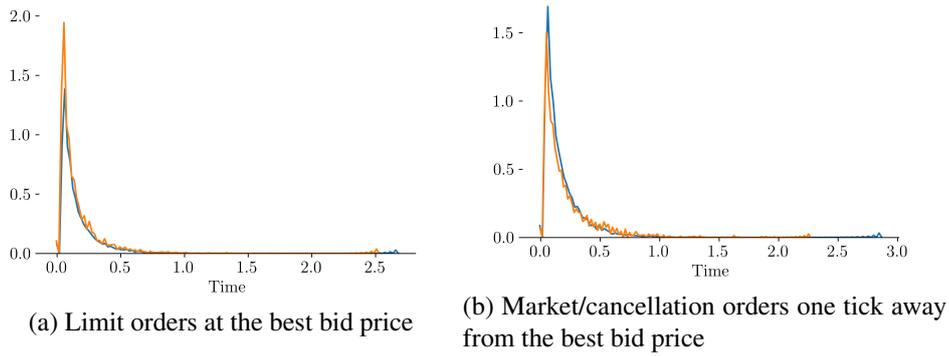


Figure 4.3: Comparison of empirical distributions for market/cancellation orders at the best bid.

limit orders and their cancellations and executions at the first three levels on each side of the book. For each event time, we determine the volume at the given level of the book the instant before the event takes place. In order to reduce the number of parameters in our estimation, we then approximate these volumes by

$$q(t^-) = \left\lceil \frac{u^i(t^-)}{100} \right\rceil, \quad i \in \{0, 1, 2\},$$

where $u^0(t)$, $u^1(t)$, and $u^2(t)$ are the time t volumes at the best bid/ask, one tick away from the best bid/ask and two ticks away from the best bid/ask, respectively. We then use the maximum likelihood estimation procedure described in Section 2.2 to fit a state-dependent Hawkes process to the data. This estimation procedure is similar to that used in [33], however, in this case, the parameters β_{ij} are also estimated and not given to the model as hyper-parameters. For comparison, we will fit several counting processes to our data:

- (i) independent, state-dependent Poisson processes, which we will fit using the methodology described in [24],
- (ii) a ‘rich’ six-dimensional Hawkes process in which none of the parameters are forced to be zero, and
- (iii) the six-dimensional Hawkes process that we chose as our microscopic model in Section 3.2.

In Tables 4.1 and 4.2, we present the log-likelihood, $\log L$, for the three models considered for the first three queues on the bid and ask sides of the book, respectively. For each of the three models, we also calculate the values of the *Akaike Information Criterion* (AIC)

$$AIC = 2k - 2 \log L$$

and the *Schwartz Information Criterion* (BIC)

$$BIC = k \log(n) - 2 \log L,$$

	$\log L$	AIC	BIC
Model (i)	30055.78	-58119.56	-48112.03
Model (ii)	115950.19	-229764.38	-219033.41
Model (iii)	72332.42	-142612.84	-132303.87

Table 4.1: Log-likelihood, AIC and BIC for each of the three models for the first three levels on the bid side of the book.

	$\log L$	AIC	BIC
Model (i)	-44310.75	89869.50	96130.39
Model (ii)	167463.30	-333534.6	-326551.30
Model (iii)	123904.24	-246500.49	-239938.60

Table 4.2: Log-likelihood, AIC and BIC for each of the three models for the first three levels on the ask side of the book.

where k is the number of parameters in the model and n is size the of the sample.

According to these criteria, the best model is that which minimizes their respective values. It is clear that, in both cases, Model (iii) provides a better fit than the state-dependent Poisson process according to both the AIC and the BIC. It also has a higher log-likelihood in both cases. While Model (iii) does not outperform the more complicated Hawkes process model, even when including a penalty for complexity, due to its simplicity and improvement on the state-dependent Poisson process model, we are satisfied that our assumptions in Section 3.2.1 are reasonable. In Figure 4.4, we also present the quantile-quantile plots for each of these models for the bid side of the book. The analogous figures for the ask side of the book can be found in Appendix B.1. These compare each of these models' residuals with the theoretical quantiles given by Theorem 2.2.3. It is clear from Figure 4.4 that the conclusions we reached from comparing the AIC and BIC criteria for each of the models are supported by the residual processes of the models. The models do not, however, provide perfect fits to the data and it would be worth exploring the fit obtained when using more complicated kernel functions (see, for example, [6] for an example of non-parametric kernel estimation). We also note that the spectral radius for Model (iii) when fit to the first three queues on the bid side of the book is 0.83, and on the ask side of the book it is 0.67, so both of the models satisfy the stability condition we have imposed throughout this paper.

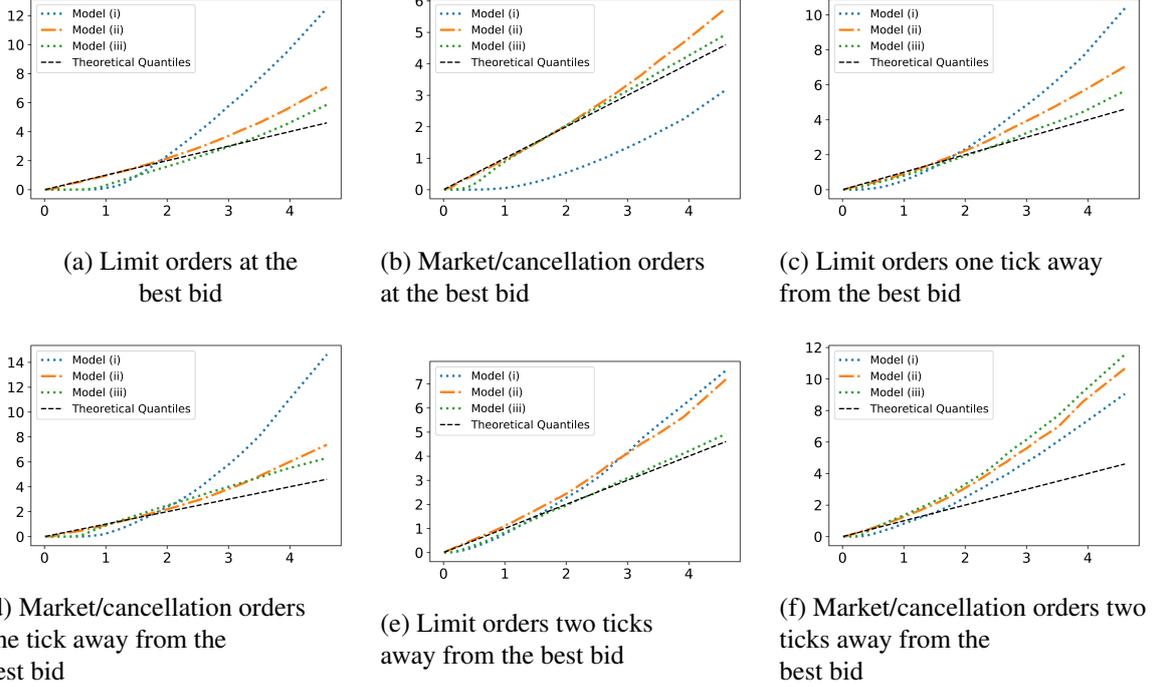


Figure 4.4: Quantile-Quantile Plots for various counting processes to model order arrivals at the first three queues on the bid side of the book.

4.4 Simulating the Mesoscopic Model

In this section, we demonstrate how a straightforward implementation of the model we gave in Section 3.2 can reproduce some of the characteristics of the limit order book profile. For comparison, we will also implement the model in [18] and remark on the differences between the two models.

4.4.1 The Model

We will implement the case where, between price movements, the bid and ask sides of the order book (X^b and X^a , respectively) evolve according to the system of SDEs

$$dX_t^b = Cdt + \Sigma dW_t^b + d\eta_t^b,$$

$$dX_t^a = Cdt + \Sigma dW_t^a + d\eta_t^a,$$

for $i \in \{1, \dots, N-1\}$, where the $N-1$ -dimensional Brownian motions $W^b = (W^{b,1}, \dots, W^{b,N-1})$ and $W^a = (W^{a,1}, \dots, W^{a,N-1})$ are independent of one another and have the common covariance matrix Σ . We assume (as in [18]) that the order arrival rates depend only on the distance of the event from the mid, and not on the state of the book at the time of the event. Note that in this case we are working with a system of SODE's, and our discretisation is only with respect to time as we have not assumed a continuous state space. In [18], the order book profile evolves according to an SPDE and so both the time and state spaces are continuous and must be discretised. While this has little impact on

the implementation, it is worth noting the distinct difference in the underlying theoretical models. We also assume that the drift and volatility parameters C_i and σ_i are the same for both sides. We make this assumption due to symmetry of order arrival rates on the bid and ask sides of the book, a phenomenon which is clear in this data set.

Our chosen mechanism for price changes also follows [18]: we assume that price changes are driven by a combination of the imbalance between the volumes at the best bid and best ask prices and some exogenous factors. To formalise this, we have that, at time t , the rate of an upward price jump will be given by

$$\theta_u(X^b(t, \cdot), X^a(t, \cdot)) = \gamma \max \left(\int_0^\epsilon (X^b(t, x) - X^a(t, x)) ds, 0 \right) + \delta, \quad (4.1)$$

and the rate of a downward price jump is given by

$$\theta_d(X^b(t, \cdot), X^a(t, \cdot)) = \gamma \max \left(\int_0^\epsilon (X^a(t, x) - X^b(t, x)) ds, 0 \right) + \delta. \quad (4.2)$$

We also assume that, following a price change, the profiles of the book simply shift in the direction of the price change by one tick. This idea is formalised by taking the functions $R(\cdot, \cdot, u)$ and $R(\cdot, \cdot, d)$ defined in Section 3.3.2 to be the deterministic functions:

$$R(X^b, X^a, u) = \begin{cases} (0, X^a(i+1)) & \text{for } i = 0, \\ (X^b(i-1), X^a(i+1)) & \text{for } i \in \{1, \dots, N-1\}, \\ (X^b(i-1), 0) & \text{for } i = N, \text{ and} \end{cases} \quad (4.3)$$

$$R(X^b, X^a, d) = \begin{cases} (X^b(i+1), 0) & \text{for } i = 0, \\ (X^b(i+1), X^a(i-1)) & \text{for } i \in \{1, \dots, N-1\}, \\ (0, X^a(i-1)) & \text{for } i = N. \end{cases} \quad (4.4)$$

We remark that, while these equations have the natural interpretation of price changes corresponding to the depletion of available liquidity at a given level, these profiles violate the boundary conditions of the system of SDE's defined in Theorem 3.2.1. These equations can, however, be shown to have solutions in $C_0((0, 1))^+$ when the initial value is positive.

4.4.2 Numerical Setup

We will use the Euler-Maruyama scheme to simulate this system of equations. We will work on a time interval $[0, T]$ and discretise this interval into M evenly spaced steps $0 = t_0 < t_1 < \dots < t_M = T$, where $t_j = jT/M$. We will work with a system N SDE's for each side of the book, and we will re-scale our discrete state space to fit into the interval $[0, 1]$. We will denote the simulated volume of the i^{th} queue on the bid (resp. ask) side of the book at the j^{th} time step by $X^b(t_j, i)$ (resp. $X^a(t_j, i)$) and we will denote the simulated price process at time t_j by $p(t_j)$. To simulate the pinning conditions of our mesoscopic model, we will set $X^b(t_j, 0) = X^b(t_j, N) = X^a(t_j, 0) = X^a(t_j, N)$.

At each time step, before updating the order book process, we determine if a price change should occur and what the direction of this price change should be. We assume that all price changes are of one

tick in size and that the bid-ask spread remains fixed at two ticks. In order to determine if a price change should occur, we approximate (4.1) and (4.2) by

$$\pi^+(t_j) = \left(\max \left(\frac{\gamma}{2N} (X^b(t_j, 1) - X^a(t_j, 1)), 0 \right) + \delta \right) \times \frac{T}{M},$$

and

$$\pi^-(t_j) = \left(\max \left(\frac{\gamma}{2N} (X^a(t_j, 1) - X^b(t_j, 1)), 0 \right) + \delta \right) \times \frac{T}{M},$$

respectively. We will then simulate a uniform random variable on $(0, 1)$ denoted by $U(t_j)$. The value of $U(t_j)$ determines whether there is a price change at each time step and the functions (4.3) and (4.4) determine how the simulated profile is updated with each price movement as follows:

- (i) If $U(t_j) < \pi^+(t_j)$, then the price moves up by one tick, so we set

$$p(t_{j+1}) = p(t_j) + \frac{1}{N},$$

$$X^b(t_j, 0) = X^a(t_j, N) = 0.$$

For $i \in \{1, \dots, N\}$, we set

$$X^b(t_j, i) = X^b(t_j, i - 1),$$

and for $i \in \{0, \dots, N - 1\}$, we set

$$X^a(t_j, i) = X^a(t_j, i + 1).$$

- (ii) If $\pi^+(t_j) \leq U(t_j) < \pi^+(t_j) + \pi^-(t_j)$, the price moves down by one tick, so we set

$$p(t_{j+1}) = p(t_j) - \frac{1}{N}.$$

For $i \in \{1, \dots, N\}$, we set

$$X^a(t_j, 0) = X^b(t_j, N) = 0,$$

$$X^a(t_j, i) = X^b(t_j, i - 1),$$

and for $i \in \{0, \dots, N - 1\}$, we set

$$X^b(t_j, i) = X^a(t_j, i + 1).$$

- (iii) If $\pi^+(t_j) + \pi^-(t_j) \leq U(t_j)$, then the price does not change, so we set

$$p(t_{j+1}) = p(t_j),$$

and we do not update the order book profile.

Once we have completed this process, we calculate the profiles on each side of the book using the Euler-Maruyama method as follows:

$$\mathbf{X}^b(t_{j+1}) = \max \left(\mathbf{X}^b(t_j) + C \frac{T}{M} + \sqrt{\frac{T}{M}} \boldsymbol{\Sigma} \mathbf{Z}_j^b, 0 \right),$$

and

$$\mathbf{X}^a(t_{j+1}) = \max \left(\mathbf{X}^a(t_j) + C \frac{T}{M} + \sqrt{\frac{T}{M}} \boldsymbol{\Sigma} \mathbf{Z}_j^a, 0 \right),$$

where \mathbf{Z}_j^b and \mathbf{Z}_j^a are $(N - 1)$ -dimensional normal random vectors. The maximum with zero replicates the effect of the reflection measure at each time step.

4.4.3 Parameter Estimation

In this section, we describe how we obtain estimates for the parameters of the model described above from our data set. Throughout this section, we will denote the order volume at the i^{th} price point below (resp. above) the best bid (resp. ask) price of the book at the j^{th} time-step of our observed data set by $x_{j,i}^b$ (resp. $x_{j,i}^a$). We will use the same scaling as in [18], i.e. we will measure order volumes in units of 10^4 and we will measure time in minutes. Moreover, we will consider 50 queues on each side of the book, which we will map to the interval $[0, 1]$ i.e. the i^{th} queue will be mapped to position $i/50$. Since the i^{th} queue of our data set corresponds to the queue i ticks away from the mid, and our tick size is 1 cent, a spatial increment of $1/50$ will correspond to a price change of 1 cent.

We begin by fitting the drift and covariance estimates for our static order book dynamics. Since we have assumed that the dynamics of the profile of the order book are independent of the price-changing mechanism, we fit these estimates on the static intervals in our data set. Let n_b (resp. n_a) be the number of time intervals during which the best bid (resp. ask) price remains constant, respectively. For each $k \in \{1, \dots, n_b\}$, let $(t_{1,i}^b, \dots, t_{l_k,i}^b)$ be the observed time steps in this static interval, where $t_{j,i}^b$ corresponds to the time stamp of the j^{th} event at the i^{th} queue on the bid side of the book, and let $(x_{1,i}^b, \dots, x_{l_k,i}^b)$ denote the corresponding volumes i ticks from the mid.

To estimate the drift, we calculate the least squares estimate for each queue on each side of the book (as described in Section 10.3.1 in [30]). The procedure is as follows: for the i^{th} queue on the bid side of the book, the least squares estimate of C_i is given by

$$\hat{C}_i^b = \frac{\sum_j (x_{j,i}^b - x_{j-1,i}^b)}{\sum_j (t_{j,i}^b - t_{j-1,i}^b)}.$$

Note that we only consider this sum over intervals where the best bid price does not change, i.e.

$$\sum_j (x_{j,i}^b - x_{j-1,i}^b) = \sum_{k=1}^{n_b} \sum_{j=1}^{l_k} (x_{l_k,i}^b - x_{l_k-1,i}^b),$$

and

$$\sum_j (t_{j,i}^b - t_{j-1,i}^b) = \sum_{k=1}^{n_b} \sum_{j=1}^{l_k} (t_{j,i}^b - t_{j-1,i}^b).$$

The drift on the ask side of the book is calculated analogously. For each queue, our drift estimate is then given by

$$\hat{C}_i = \frac{1}{2} \left(\hat{C}_i^b + \hat{C}_i^a \right) \times 60,$$

where the factor 60 appears because we have chosen minutes as our unit of time.

Now, let $\omega_{i,j}^b$ (resp. $\omega_{i,j}^a$) denote the size of the order placed in the i^{th} queue on the bid (resp. ask) side of the book at time t_j . To estimate the covariance matrix, Σ^2 , we equate, for $m, n \in \{1, \dots, 50\}$,

$$\hat{\sigma}_{mn}^2 = \frac{1}{60 \times 2} \sum_{j=1}^J \sum \left(\frac{\omega_{m,j}^{k_1}}{10^4} \times \frac{\omega_{n,j}^{k_2}}{10^4} \times \mathbb{1}_{\{k_1=k_2\}} \right), \quad (4.5)$$

k_1 and k_2 indicate which side of the book the order is placed on. The inner sum is taken over all orders in queues m and n that occur at time t_j . The indicator function ensures that we are only considering covariance between orders on the same side of the book. To ensure our estimated covariance has the correct sign, we also adopt the convention that the sizes of orders that increase liquidity at a given queue (limit orders) are positive, and the sizes of orders that decrease liquidity at a given queue (cancellations/market orders) are negative. This procedure is equivalent to calculating the cross variation between individual queues over the hour-long period (excluding the cross variation that results from price changes as we model this separately in this setup). The factor of 60 is because we are calculating the quadratic variation over the full hour and the factor of 2 is because we are considering the average over both sides of the book. Note that when $m = n$, $\sigma_{mn} = \sigma_m^2$, and (4.5) reduces to

$$\hat{\sigma}_m^2 = \frac{1}{60 \times 2} \sum_{j=1}^J \left(\frac{\omega_{m,j}^k}{10^4} \right)^2,$$

where this sum is calculated for all orders on both sides of the book. This estimation yields a negative covariance between any two adjacent queues, which is what we would expect given our hypothesis that cancellations/market orders in a given queue excite limit orders in the adjacent queue nearer the mid.

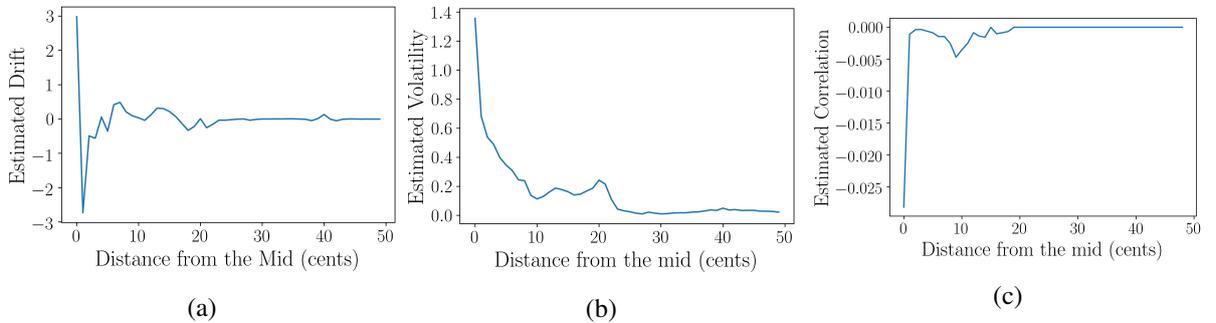


Figure 4.5: Estimated drift 4.5(a), volatility 4.5(b) and correlation 4.5(c) functions.

Figures 4.5(b) and 4.5(a) show the estimated drift and volatility functions for the data, and Figure 4.5(c) gives an illustration of the estimated correlation between adjacent queues. The fact that both the estimated volatility and correlation are much larger near the mid is to be expected as a large proportion of the total activity of the book (90.82% for this data set) happens in the first 10 queues.

To estimate the parameters γ and δ in equations (4.1) and (4.2), we will use the same method as [18]. Let $P(t)$ denote the observed mid at time t , measured in cents. Let I be the average local imbalance of the data over the hour long period under consideration, i.e.

$$I = \frac{1}{J \times 2 \times 50 \times 10^4} \sum_{j=1}^J (x_{j,1}^b - x_{j,i}^a),$$

where J is the total number of time steps in our data set and the factors 50 and 10^4 appear because of our chosen units for order sizes and tick sizes. We then calculate our estimate $\hat{\gamma}$ for γ by equating

$$P(1) - P(0) = 60 \times \hat{\gamma} \times I,$$

i.e. we choose $\hat{\gamma}$ so that the price change over the entire period is a function of the average local imbalance over the period. The factor of 60 appears again because our chosen time unit is minutes.

In order to find an estimate for δ , the rate of price movements due to exogenous factors, we equate (a scaled version of) our estimate $\hat{\delta}$ to the difference between the observed quadratic variation and the expected quadratic variation due to local imbalance. The expected quadratic variation due to the local imbalance is given by $60\hat{\gamma}\tilde{I}$, where

$$\tilde{I} = \frac{1}{J \times 2 \times 50 \times 10^4} \sum_{j=1}^J |x_{j,1}^b - x_{j,i}^a|.$$

For a given δ , we would expect $60 \times \delta$ price changes *in each direction* to occur because of exogenous factors on top of this, so our estimate $\hat{\delta}$ is given by

$$120\hat{\delta} = \left[\sum_{j=1}^J (P(j/J) - P((j-1)/J))^2 \right] - 60\hat{\gamma}\tilde{I}.$$

Calculating these values for our data set yields

$$\hat{\gamma} = 641.25, \tag{4.6}$$

and

$$\hat{\delta} = 48.38. \tag{4.7}$$

4.4.4 Results

The aim of this section is to demonstrate the results of the simulation with the relevant parameters as above. In order to demonstrate the goodness of fit of each of these models, we will give several graphs of the simulations of our SDE model from Section 3.3.2, the SPDE model given in [18], and the observed values from the data set. For both the SDE and SPDE models, we will simulate the limit order book profile for one hour using 1500000 time steps, and we simulate the data over an hour, so we take $T = 60$.

We begin by comparing the price processes for each of these models.

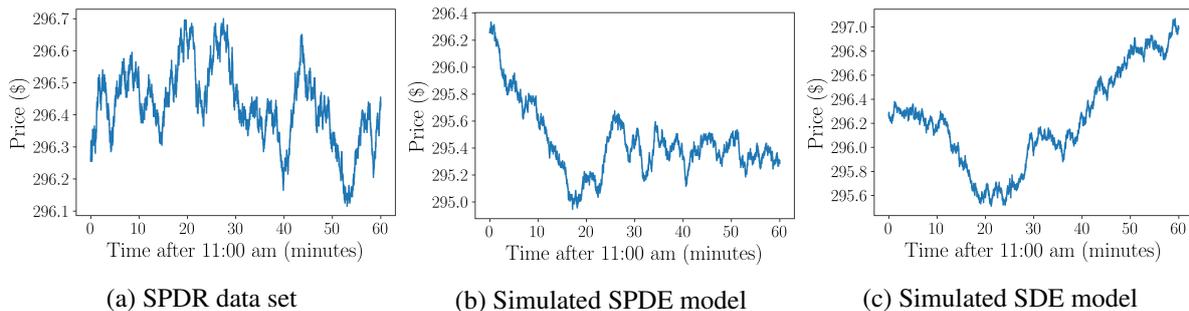


Figure 4.6: The price process over the hour from 11:00am to 12:00pm.

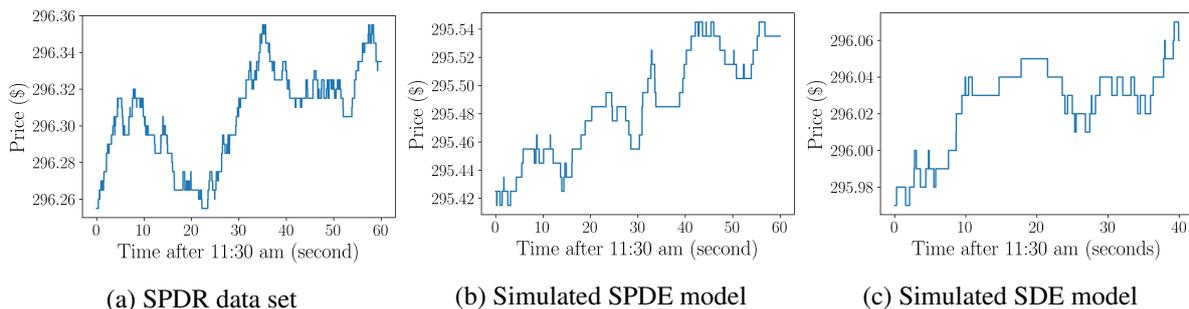


Figure 4.7: The price process over the first forty seconds after 11:30am.

In Figure 4.6, the price series over the hour for each of the simulated models and the actual data is given, while Figure 4.7 displays snapshots of the prices over the first forty seconds after 11:30am. When measured in dollars, the quadratic variations of the observed price process, the simulated SPDE model and the simulated SDE model are 0.5844, 0.5923, and 0.5771, respectively. From these observations, it is clear that we get somewhat reasonable simulations of the price process from both models. Since the price changing mechanism and method of parameter estimation here follows the work done in [18], it is not surprising that, in terms of the price processes, similar results are produced by both sets of simulations.

The more interesting case is the simulation of the order book profiles over the hour. Figure 4.8 displays the average profiles of the ask side of the order book over the hour under consideration, where Figure 4.9 gives snapshots of the order book profiles on the bid side of the book at 4 evenly spaced

intervals over the hour. The corresponding figures for the ask and bid sides of the book respectively can be found in Appendix B.2.

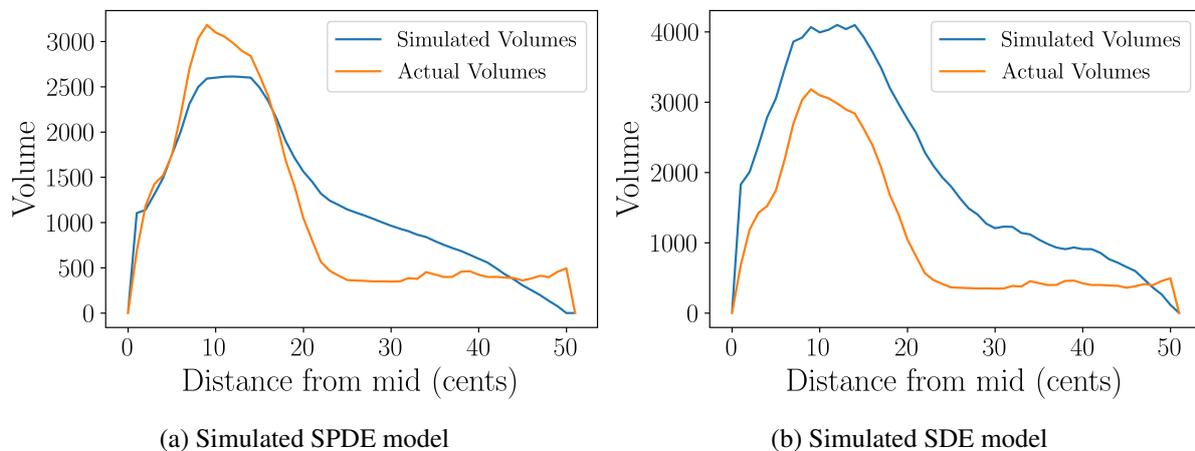


Figure 4.8: Average profile on the ask side of the order book over the hour from 11:00am to 12:00pm.

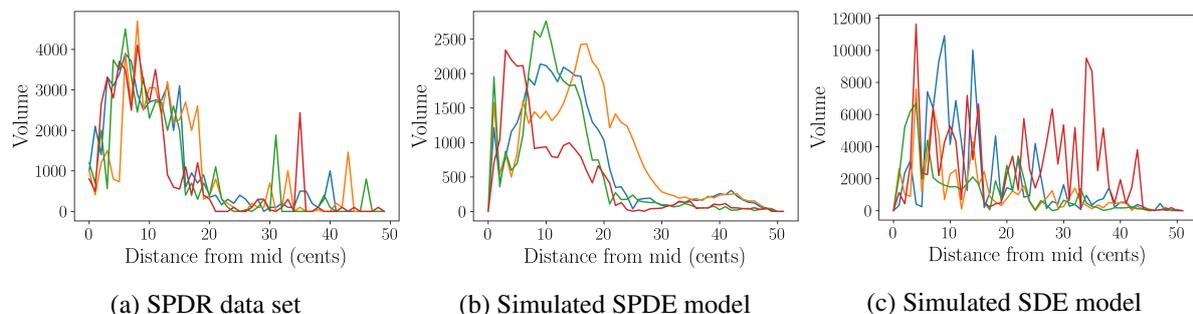


Figure 4.9: Snapshots of the profile on the bid side of the order book at 12 minute intervals in the hour from 11:00am to 12:00pm.

It is clear that, while the SPDE model slightly underestimates the average profile of the book, the SDE model is more volatile than both the observed data and the SPDE model and over estimates the average profile. This is not surprising as the addition of the laplacian in the SPDE model has a smoothing effect. It does, however, mean that in this straightforward implementation, the SDE model is not an improvement on the SPDE model. On the other hand, it is worth noting that because the SDE model is far more simple, the 1500000 simulations can be completed in less than 10% of the time required for the SPDE model. Artificially smoothing the SDE simulations (by multiplying the random term by 0.7 in each time step) does significantly improve the fit of the model in terms of the order book profile with little impact on its ability to model the price process. This suggests that exploring more robust calibration techniques might improve the performance the simulation. Indeed, the quadratic variation of the price process generated by the artificially smoothed SDE model is 0.5823 and the illustrations of the simulated profile on the bid side of the book are given in Figure 4.10. It is clear from these simulations that the smoothing achieved by the inclusion of the laplacian is important in simulating the order book

model. This suggests investigating a model that includes a laplacian term as well as the correlation we have included might yield better results.

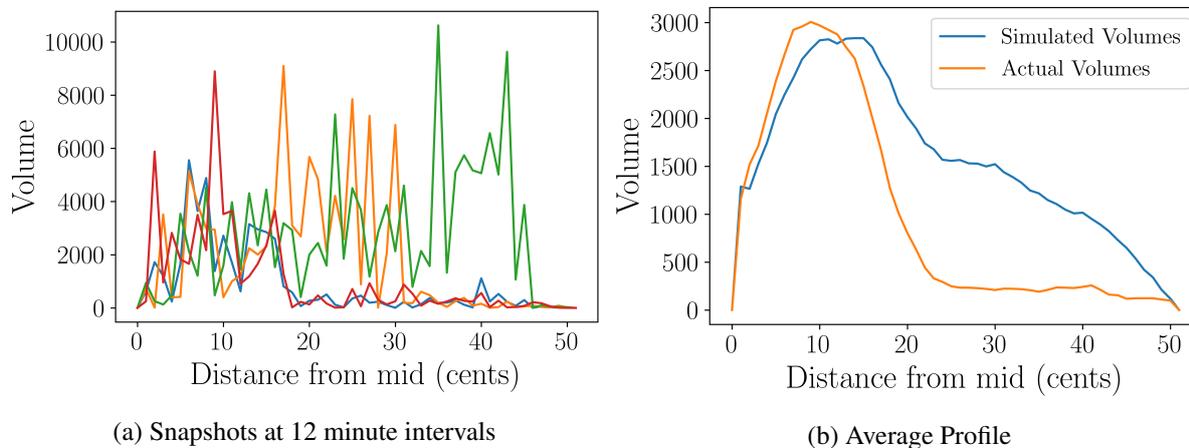


Figure 4.10: Simulated profiles for the bid side of the book (with smoothing).

4.4.5 A More Volatile Data Set

In this section, we perform a similar analysis on the SPDR Trust Series I for the same hour on 16 March 2020. 16 March 2020 represents a day of unusually high volatility, and performing our numerical analysis on this data set will give some indication of the robustness of our assumptions and methodologies.

This data set has different characteristics to the data set we have been considering until now. Indeed, the total number of limit orders, market orders and cancellation orders of the hour on 16 March 2020 are 372317, 40288, and 348731, respectively, where in the same hour on 25 September 2019, these values are 251383, 10833, 243697, respectively.

In this data set, there are nearly 7 times as many price changes than there are in the same hour on 25 September 2019 (152042 against 23338). This suggests that our assumption that the dynamics of the order book between price changes can be modelled independently of the price-changing mechanism may not be applicable in cases of unusually high volatility. The average spread is also much higher in the March data set (522.90 against 120.07), and Figure 4.11 illustrates the significant difference in the empirical distributions of the size of the bid/ask spread for the two data sets. Indeed, this distribution is concentrated around 200 ticks on 25 September 2019, but it is much more spread out on 16 March 2020. These observations indicate that our assumption that the bid/ask spread is a constant two ticks is also not valid in this case.

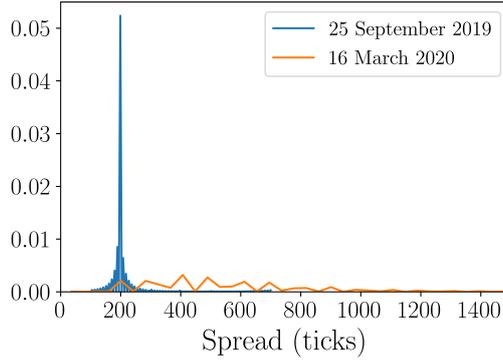


Figure 4.11: Empirical distributions of the bid/ask spread.

To assess the performance of our model on this data set, we fit the SDE model from Section 3.3.2 to the data using the procedure described above. We find that, in this case, the net order flow over the period provides a better estimate for the drift than the least squares estimate used in the previous section. Indeed, we fit the drift by equating

$$\hat{C}_i = \frac{1}{2 \times 60} (D_i^b + D_i^a - C_i^b - C_i^a),$$

where D_i^b (resp. C_i^b) represents the total number of limit (resp. market/cancellation) orders placed at the i^{th} queue on the bid side of the book over the hour long period. D_i^a and C_i^a are defined analogously for the ask side of the book. Once we have fit the drift, all other parameters are fit using the same methodology as the previous section. Graphs for the estimated drift and volatility functions are given in Appendix B.3. We remark here that the estimates for the parameters that determine the rates of price changes are much larger in magnitude in this case. Indeed, the estimates $\hat{\gamma}$ and $\hat{\delta}$ are -62761.59 and 1561.23 , respectively, for this data set, as opposed to 641.25 and 48.38 , for the September data set. This is an artefact of the fact that the quadratic variation (in dollar terms) is 18.62 for this data set and only 0.5844 for the less volatile data set. We then simulate the fitted model and compare our simulated price process and order book profiles with the observed data and with the simulations obtained from the SPDE model in [18]. Note that, again, we artificially smooth the SDE model by multiplying the random term by 0.6 . The quadratic variations of the simulated price processes of the SDE and SPDE models are 18.7447 and 18.7800 , respectively, demonstrating that this model still does a reasonable job of estimating the amount of variation in the price process. Plots of the simulated price processes are also given in Appendix B.3.

These models do not, however, reproduce the characteristics of the order book profiles particularly well, as can be seen in Figures 4.12 and 4.13. In each case, the simulated average order book profiles are much larger than their observed counterparts. It is clear from this analysis that it would be meaningful to investigate different techniques for modelling cases of extremely high volatility. The analogous plots for the bid side of the book are also given in Appendix B.3.

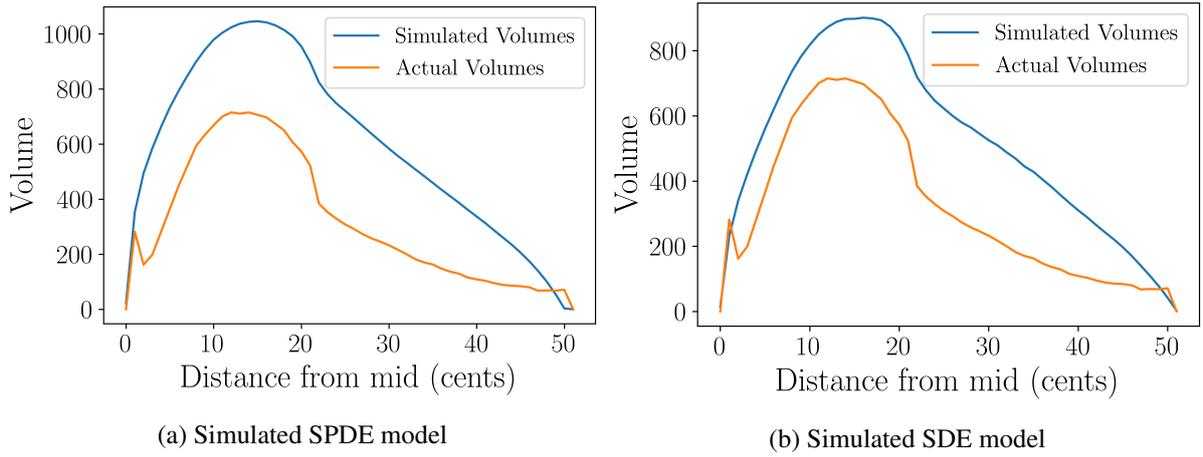


Figure 4.12: Average profile of the ask side of the order book over the hour from 11:00am to 12:00pm.

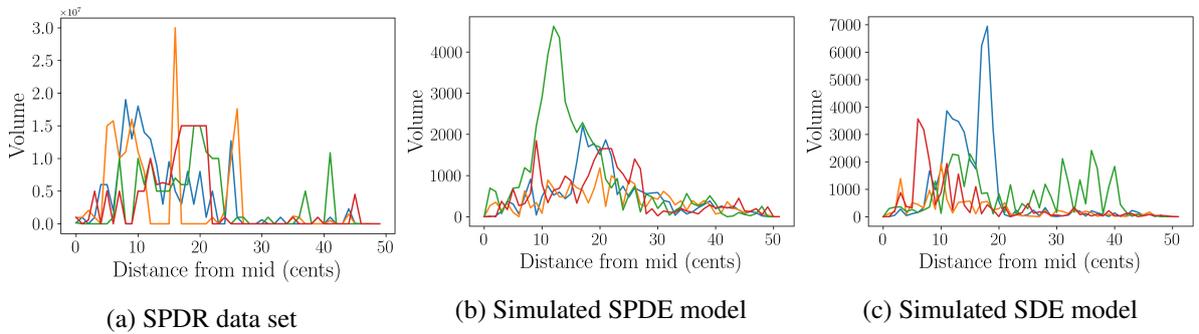


Figure 4.13: Snapshots of the profile of the ask side of the order book at 12 minute intervals in the hour from 11:00am to 12:00pm.

Section 5

Conclusion

5.1 Conclusion

In this dissertation, we demonstrated the applicability of Hawkes processes to model order arrival times in a limit order book. We presented a microscopic order book model driven by Hawkes processes and, by allowing time to tend to infinity and order sizes to tend to 0, we derived an SDE model for the limit order book in continuous time. We then implemented our model by calibrating it to an observed data set and demonstrated its ability to reproduce realistic order book profiles and price processes. We compared these results with those obtained by using the SPDE model given in [18] and found the SDE model does not perform better than its SPDE counterpart, but it is much more computationally efficient to simulate. We also perform this analysis on a more volatile data set and find that, in this case, a different modelling approach might be necessary.

5.2 Further Research

It is clear that incorporating the correlation between queues in this model does not do enough to smooth the process. A natural extension of this model would be to include a laplacian term (as was done in [18]) and to determine how the inclusion of both the correlation between queues and the laplacian compares to the SPDE model. One could then take the limit of this model as tick size tends to zero to obtain an SPDE model for the limit order book dynamics.

Another interesting extension of this model would be to consider the scaling limit of the microscopic model when the underlying Hawkes processes are *nearly* unstable. Limits of this type have been investigated in [27]. In this paper, Jaisson and Rosenbaum found that when the underlying Hawkes process is nearly unstable, their asymptotic behaviour is the same as integrated Cox-Ingersoll-Ross ([14]) models. They also found that, when considering the difference of two counting processes driven by such Hawkes processes (as we have done in our mesoscopic limit), the scaling limit converges to a Heston model ([22]) (i.e. a model with stochastic volatility). The Hawkes processes we fit to the data in Section 4 have rather large spectral radii, and this is consistent with the nearly unstable Hawkes processes fit to limit

order book data [6] and [16]. Given this and the fact that our proposed model falls short when estimating the volatility of the order book process, an investigation in this direction is also worth considering.

Appendix A

A.1 Proofs of Theorems in Section 2.3

In this appendix we give the outline of the proofs of the Functional Central Limit Theorem and Law of Large Numbers in Section 2.3. These are included for completeness and are not very different from the proofs in [4]. We refer the reader to [4] for details.

Let $N = (N_t^1, \dots, N_t^d)_{t \geq 0}$ be a d -dimensional process with intensity

$$\lambda_i(t) = \mu_i(q(t^-)) + \sum_{j=1}^d \int_0^t \alpha_{ij} e^{-\beta_{ij}(t-s)} dN_{j,s}.$$

Let the matrices $\boldsymbol{\mu}(q)$, $\boldsymbol{\Phi}$ and \boldsymbol{K} be as defined in 2.5 and 2.6, respectively, i.e. $\boldsymbol{\mu}(q) = (\mu_i(q))_{i \in \{1, \dots, d\}}$,

$$\boldsymbol{\Phi} = (\phi_{ij})_{i,j \in \{1, \dots, d\}}, \text{ and}$$

$$\boldsymbol{K} := \int_0^\infty \boldsymbol{\Phi}(t) dt = \left(\int_0^\infty \phi_{ij}(t) dt \right)_{1 \leq i, j \leq d}.$$

Assume that the stability condition in Proposition (2.1.3) holds, i.e. the spectral radius of \boldsymbol{K} is less than one, and assume that $\boldsymbol{\mu}(q)$ is finite and square integrable. In this case, we can give the versions of the proofs of Theorems (2.3.1) and (2.3.2) that have been adapted to include the state-dependent exogenous intensity.

We begin with the following lemma (Lemma 2 in [4]):

Lemma A.1.1. For all finite stopping times S , one has:

$$\mathbb{E}[N_S] = \mathbb{E} \left[\int_0^S \boldsymbol{\mu}(q(t^-)) dt \right] + \mathbb{E} \left[\int_0^S \boldsymbol{\Phi}(S-t) N_t dt \right], \text{ and} \quad (\text{A.1})$$

$$\mathbb{E}[N_S] \leq (\mathbf{Id} - \boldsymbol{K})^{-1} \mathbb{E} \left[\int_0^S \boldsymbol{\mu}(q(t^-)) dt \right] \text{ component-wise.} \quad (\text{A.2})$$

Proof. Let $(T_p)_{p \in \mathbb{N}}$ denote the successive jump times of N and set $S_p = S \wedge T_p$. Since the stochastic intensities λ_i are given by (2.7), we have that

$$\mathbb{E}[N_{S_p}] = \mathbb{E} \left[\int_0^{S_p} \boldsymbol{\mu}(q(t^-)) dt \right] + \mathbb{E} \left[\int_0^{S_p} dt \int_0^t \boldsymbol{\Phi}(t-s) dN_s \right].$$

By Fubini's Theorem, we have that

$$\begin{aligned}\int_0^{S_p} dt \int_0^t \Phi(t-s) dN_s &= \int_0^{S_p} \left(\int_s^{S_p} \Phi(t-s) dt \right) dN_s \\ &= \int_0^{S_p} \left(\int_0^{S_p-s} \Phi(t) dt \right) dN_s.\end{aligned}$$

Integrating by parts, using $\psi(t) := \int_0^t \Phi(s) ds$, we get

$$0 = \psi(0)N_{S_p} - \psi(S_p)N_0 = \int_0^{S_p} \psi(S_p-t) dN_t - \int_0^{S_p} \Phi(S_p-t) N_t dt.$$

Since $\sum_{i=1}^d N_{i,S_p} \leq p$, the right hand side of the equation above is finite, so

$$\mathbb{E}[N_{S_p}] = \mathbb{E} \left[\int_0^{S_p} \mu(q(t^-)) dt \right] + \mathbb{E} \left[\int_0^{S_p} \Phi(S_p-t) N_t dt \right].$$

Since $N_{S_p} \rightarrow N_S$ as $p \rightarrow \infty$, we have that

$$\int_0^{S_p} \Phi(S_p-t) dN_t = \int_0^{S_p} \Phi(t) N_{S_p-t} dt \uparrow \int_0^S \Phi(t) N_{S-t} dt = \int_0^S \Phi(S-t) N_t dt.$$

To see that (A.2) holds, note that

$$\begin{aligned}\mathbb{E}[N_{S_p}] &= \mathbb{E} \left[\int_0^{S_p} \mu(q(t^-)) dt \right] + \mathbb{E} \left[\int_0^{S_p} dt \int_0^t \Phi(t-s) dN_s \right] \\ &\leq \mathbb{E} \left[\int_0^{S_p} \mu(q(t^-)) dt \right] + \mathbb{E} \left[\int_0^\infty dt \int_0^t \Phi(t-s) dN_s \right], \text{ component-wise} \\ &= \mathbb{E} \left[\int_0^{S_p} \mu(q(t^-)) dt \right] + \mathbf{K} \mathbb{E}[N_{S_p}].\end{aligned}$$

By induction,

$$\mathbb{E}[N_{S_p}] \leq (\mathbf{Id} + \mathbf{K} + \dots + \mathbf{K}^{n-1}) \mathbb{E} \left[\int_0^{S_p} \mu(q(t^-)) dt \right] + \mathbf{K}^n \mathbb{E}[N_{S_p}],$$

component-wise, for all $n \in \mathbb{N}$. Since the spectral radius of \mathbf{K} is less than one, we have that $\mathbf{K}^n \rightarrow 0$ as $n \rightarrow \infty$, and

$$\sum_{n=0}^{\infty} \mathbf{K}^n = (\mathbf{Id} - \mathbf{K})^{-1}.$$

We also have that $\sum_{i=1}^d N_{i,S_p} \leq p$, so

$$\mathbb{E}[N_{S_p}] \leq (\mathbf{Id} - \mathbf{K})^{-1} \mathbb{E} \left[\int_0^{S_p} \mu(q(t^-)) dt \right].$$

Since $\mathbb{E}[N_S] = \lim_p \mathbb{E}[N_{S_p}]$, we have

$$\mathbb{E}[N_S] \leq (\mathbf{Id} - \mathbf{K})^{-1} \mathbb{E} \left[\int_0^{S_p} \mu(q(t^-)) dt \right] \text{ component-wise,}$$

as required. □

From now on, unless stated otherwise, the proofs are the same as in [4], except for the difference in the expression of the expectation of N_s due to Lemma A.1.1.

Note that when t is deterministic, it follows from Fubini's Theorem that equations (A.1) and (A.2) become

$$\begin{aligned}\mathbb{E}[N_t] &= t\mathbb{E}[\boldsymbol{\mu}(q)] + \mathbb{E}\left[\int_0^t \boldsymbol{\Phi}(t-s)N_s ds\right], \text{ and} \\ \mathbb{E}[N_t] &\leq (\mathbf{Id} - \mathbf{K})^{-1}\mathbb{E}[\boldsymbol{\mu}(q)]t \text{ component-wise.}\end{aligned}$$

Now, for $n \geq 1$, let $\boldsymbol{\Phi}_n$ be the $d \times d$ -matrix valued functions on \mathbb{R}_+ defined recursively by

$$\boldsymbol{\Phi}_1 = \boldsymbol{\Phi}, \text{ and } \boldsymbol{\Phi}_{n+1}(t) = \int_0^t \boldsymbol{\Phi}(t-s)\boldsymbol{\Phi}_n(s)ds.$$

By induction, we have that $\int_0^\infty \boldsymbol{\Phi}_n(t)dt = \mathbf{K}^n$, for all n . Indeed, $\int_0^\infty \boldsymbol{\Phi}_1(t)dt = \mathbf{K}$, by definition and if $\int_0^\infty \boldsymbol{\Phi}_n(t)dt = \mathbf{K}^n$, then

$$\begin{aligned}\int_0^\infty \boldsymbol{\Phi}_{n+1}(t)dt &= \int_0^\infty \left(\int_0^t \boldsymbol{\Phi}(t-s)\boldsymbol{\Phi}_n(s)ds\right) dt \\ &= \int_0^\infty \left(\int_0^\infty \mathbb{1}_{\{s \leq t\}} \boldsymbol{\Phi}(t-s)\boldsymbol{\Phi}_n(s)ds\right) dt \\ &= \int_0^\infty \boldsymbol{\Phi}_n(s) \left(\int_0^\infty \boldsymbol{\Phi}(t-s)dt\right) ds \\ &= \mathbf{K}^{n+1}.\end{aligned}$$

Let

$$\boldsymbol{\Psi} := \sum_{n \geq 1} \boldsymbol{\Phi}_n.$$

Then our stability assumption ensures that $\int_0^\infty \boldsymbol{\Psi}(t)dt = \sum_{n \geq 1} \mathbf{K}^n$ is finite component-wise.

The next result we need is a multivariate version of the renewal equation. It is Lemma 3 in [4].

Lemma A.1.2. Let h be a Borel and locally bounded function from $\mathbb{R}_+ \rightarrow \mathbb{R}^d$. Then there exists a unique, locally bounded function $f : \mathbb{R}_+ \rightarrow \mathbb{R}^d$ such that f is a solution to

$$f(t) = h(t) + \int_0^t \boldsymbol{\Phi}(t-s)f(s)ds, \text{ for all } t > 0, \tag{A.3}$$

and it is given by

$$f_h(t) = h(t) + \int_0^t \boldsymbol{\Psi}(t-s)h(s)ds.$$

Now, let $(M_t)_{t \in \mathbb{R}_+}$ be the d -dimensional martingale defined by

$$M_t = N_t - \Lambda(t),$$

where $\Lambda(t) = \int_0^t \lambda(s)ds$ is the compensator defined in (2.1) and $\lambda = (\lambda_i)_{i \in \{1, \dots, d\}}$. Then we have

Lemma A.1.3. For all $t \geq 0$,

$$\mathbb{E}[N_t] = t\mathbb{E}[\boldsymbol{\mu}(q)] + \left(\int_0^t \boldsymbol{\Psi}(t-s) ds \right) \mathbb{E}[\boldsymbol{\mu}(q)], \text{ and} \quad (\text{A.4})$$

$$N_t - \mathbb{E}[N_t] = M_t + \int_0^t \boldsymbol{\Psi}(t-s) dM_s ds \quad (\text{A.5})$$

Lemma A.1.4. Let $p \in [0, 1]$ and assume that $\int_0^\infty t^p \boldsymbol{\Phi}(t) dt < \infty$ component-wise. Then

(i) If $p < 1$, we have

$$T^p (T^{-1}\mathbb{E}[N_{Tv}] - v(\mathbf{Id} - \mathbf{K})^{-1}\mathbb{E}[\boldsymbol{\mu}(q)]) \rightarrow 0 \text{ as } T \rightarrow \infty$$

uniformly in $v \in [0, 1]$.

(ii) If $p = 1$, we have

$$\begin{aligned} & T (T^{-1}\mathbb{E}[N_T] - (\mathbf{Id} - \mathbf{K})^{-1}\mathbb{E}[\boldsymbol{\mu}(q)]) \\ & \rightarrow -(\mathbf{Id} - \mathbf{K})^{-1} \left(\int_0^\infty t \boldsymbol{\Phi}(t) dt \right) (\mathbf{Id} - \mathbf{K})^{-1}\mathbb{E}[\boldsymbol{\mu}(q)] \text{ as } T \rightarrow \infty. \end{aligned}$$

In what follows, $\|\cdot\|$ denotes the Euclidean norm in the relevant space (i.e. in \mathbb{R}^d or the set of $d \times d$ matrices).

Lemma A.1.5. There exists a constant $C_{\mu, \boldsymbol{\Phi}}$ such that for all $t, \Delta > 0$,

$$\mathbb{E} \left[\sup_{t \leq s \leq t+\Delta} \|M_s - M_t\|^2 \right] \leq C_{\mu, \boldsymbol{\Phi}} \Delta.$$

Now, let $W = (W^1, \dots, W^d)$ be a standard d -dimensional Brownian motion and for $i \in \{1, \dots, d\}$, let $\sigma_i = (\Sigma_{ii})^{1/2}$. We then have the following lemma:

Lemma A.1.6. The martingales $M^{(T)} := (T^{-1/2} M_{Tv})_{v \in [0,1]}$ converge in law for the Skorokhod topology to $(\sigma_1 W^1, \dots, \sigma_d W^d)$.

To prove Theorem 2.3.2, we note that it is a special case of the following more general Theorem (adapted from Theorem 2 in [4]).

Theorem A.1.7. *The processes*

$$\frac{1}{\sqrt{T}} (N_{Tv} - \mathbb{E}[N_{Tv}]), v \in [0, 1]$$

converges in law for the Skorokhod topology to

$$(\mathbf{Id} - \mathbf{K})^{-1} \boldsymbol{\Sigma}^{1/2} W_v, v \in [0, 1]$$

as $T \rightarrow \infty$, where $(W_v)_{v \in [0,1]}$ is a standard d -dimensional Brownian motion and $\boldsymbol{\Sigma}$ is the diagonal matrix such that $\Sigma_{ii} = ((\mathbf{Id} - \mathbf{K})^{-1}\mathbb{E}[\boldsymbol{\mu}(q)])_i$.

We are now in a position to prove Theorem 2.3.2:

Proof of Theorem 2.3.2. Recall that in this setting the kernel matrix of the Hawkes process takes the form

$$\Phi(t) = (\alpha_{ij}e^{-\beta_{ij}t})_{i,j \in \{1, \dots, d\}}.$$

It is easy to see that for every $i, j \in \{1, \dots, d\}$, $\int_0^\infty \alpha_{ij}e^{-\beta_{ij}t}t^{1/2}dt < \infty$. Using this fact, Lemma A.1.4 with $p = \frac{1}{2}$ gives

$$T^{1/2} (T^{-1}\mathbb{E}[N_{Tv}] - v(\mathbf{Id} - \mathbf{K})^{-1}\mathbb{E}[\boldsymbol{\mu}(q)]) \rightarrow 0 \text{ as } T \rightarrow \infty$$

uniformly in $v \in [0, 1]$. From Theorem A.1.7 we have that for $v \in [0, 1]$

$$\frac{1}{\sqrt{T}} (N_{Tv} - \mathbb{E}[N_{Tv}]) \rightarrow (\mathbf{Id} - \mathbf{K})^{-1}\boldsymbol{\Sigma}^{1/2}W_v$$

in distribution as $T \rightarrow \infty$. Putting these together gives the result. \square

A.2 Mesoscopic Model in Four Dimensions

In this appendix, we give explicit expressions for the drift and covariance parameters in the scaling limit for the Hawkes process in section 3.2.2. For ease of notation, we consider the four dimensional case of Theorem 3.2.1. The matrix of kernel norms is then equal to

$$\mathbf{K} = \begin{bmatrix} \frac{\alpha_{11}}{\beta_{11}} & \frac{\alpha_{12}}{\beta_{12}} & 0 & \frac{\alpha_{14}}{\beta_{14}} \\ \frac{\alpha_{21}}{\beta_{21}} & \frac{\alpha_{22}}{\beta_{22}} & 0 & 0 \\ 0 & 0 & \frac{\alpha_{33}}{\beta_{33}} & \frac{\alpha_{34}}{\beta_{34}} \\ 0 & 0 & \frac{\alpha_{43}}{\beta_{43}} & \frac{\alpha_{44}}{\beta_{44}} \end{bmatrix}.$$

Denote $\frac{\alpha_{ij}}{\beta_{ij}}$ as k_{ij} , then

$$(\mathbf{Id} - \mathbf{K})^{-1} = \frac{1}{K} (c_{ij})_{1 \leq i, j \leq 4},$$

where

$$K = (1 - k_{11})[(1 - k_{11})(1 - k_{22}) - k_{12}k_{21}] [(1 - k_{33})(1 - k_{44}) - k_{34}k_{43}]$$

and

$$\begin{aligned}
c_{11} &= (1 - k_{11})(1 - k_{22})[(1 - k_{34})(1 - k_{44}) - k_{43}k_{44}], \\
c_{12} &= k_{12}(1 - k_{11})[(1 - k_{34})(1 - k_{44}) - k_{43}k_{44}], \\
c_{13} &= k_{14}k_{43}(1 - k_{11})(1 - k_{22}), \\
c_{14} &= -k_{14}(k_{11} - 1)(k_{22} - 1)(k_{34} - 1), \\
\\
c_{21} &= k_{21}(1 - k_{11})[(k_{34} - 1)(k_{44} - 1) - k_{43}k_{44}], \\
c_{22} &= (k_{11} - 1)^2[(k_{34} - 1)(k_{44} - 1) - k_{43}k_{44}], \\
c_{23} &= k_{14}k_{21}k_{43}(1 - k_{11}), \\
c_{24} &= k_{14}k_{21}(k_{11} - 1)(k_{34} - 1), \\
\\
c_{31} &= 0, \\
c_{32} &= 0, \\
c_{33} &= (1 - k_{11})(1 - k_{44})[(1 - k_{11})(1 - k_{22}) - k_{12}k_{21}], \\
c_{34} &= k_{44}(1 - k_{11})[(1 - k_{11})(1 - k_{22}) - k_{12}k_{21}], \\
\\
c_{41} &= 0, \\
c_{42} &= 0, \\
c_{43} &= k_{43}(1 - k_{11})[(1 - k_{11})(1 - k_{22}) - k_{12}k_{21}], \text{ and} \\
c_{44} &= (1 - k_{11})(1 - k_{34})[(1 - k_{11})(1 - k_{22}) - k_{12}k_{21}].
\end{aligned}$$

From this, it follows that

$$(\mathbf{Id} - \mathbf{K})^{-1} \Sigma^{\frac{1}{2}} W_t = \begin{bmatrix} \sum_{j=1}^4 \frac{c_{1j}\sigma_j}{K} W_t^j \\ \sum_{j=1}^4 \frac{c_{2j}\sigma_j}{K} W_t^j \\ \sum_{j=1}^4 \frac{c_{3j}\sigma_j}{K} W_t^j \\ \sum_{j=1}^4 \frac{c_{4j}\sigma_j}{K} W_t^j \end{bmatrix}$$

where

$$\sigma_i^2 = \frac{\sum_{j=1}^4 c_{ij} \hat{\mu}_j}{K}.$$

Finally, we have that $(\frac{1}{\sqrt{n}} Z_{tn}^1 - \sqrt{nt}(\sigma_1^2 - \sigma_2^2), \frac{1}{\sqrt{n}} Z_{tn}^2 - \sqrt{nt}(\sigma_3^2 - \sigma_4^2))$ converges in distribution to $(\zeta_1 B_t^1, \zeta_2 B_t^2)$, where

$$\zeta_1 B_t^1 = \sum_{j=1}^4 \frac{(c_{1j} - c_{2j})\sigma_j}{K} W_t^j,$$

$$\zeta_2 B_t^2 = \sum_{j=1}^4 \frac{(c_{3j} - c_{4j})\sigma_j}{K} W_t^j,$$

so

$$\zeta_1 = \sqrt{\sum_{j=1}^4 \frac{(c_{1j} - c_{2j})^2 \sigma_j^2}{K^2}},$$

$$\zeta_2 = \sqrt{\sum_{j=1}^4 \frac{(c_{3j} - c_{4j})^2 \sigma_j^2}{K^2}},$$

and

$$\langle B^1, B^2 \rangle_t = \sum_{j=1}^4 \frac{(c_{1j} - c_{2j})(c_{3j} - c_{4j}) \sigma_j^2}{\zeta_1 \zeta_2 K^2} t.$$

Appendix B

In this section we include plots of our numerical results that were not included in Section 4.

B.1 Goodness of Fit of Estimated Hawkes Processes

We begin by giving the quantile-quantile plots for the counting processes we fit to the first three queues on the ask side of the limit order book.

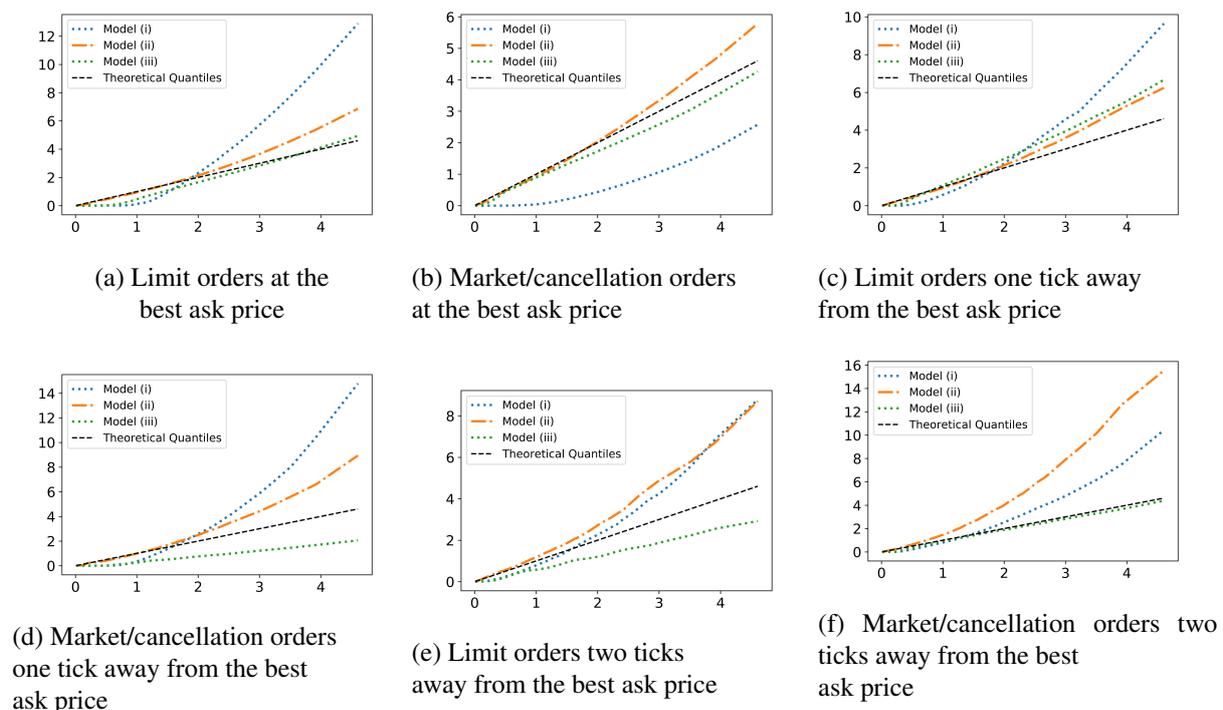


Figure B.1: Quantile-Quantile plots for various counting processes to model event arrivals at the first three queues on the ask side of the book.

B.2 Simulation Results

In this section we give the results of the simulations we performed in Section 4.4. We first give the comparison of the average order book profiles on the bid side of the book in Figure B.2.

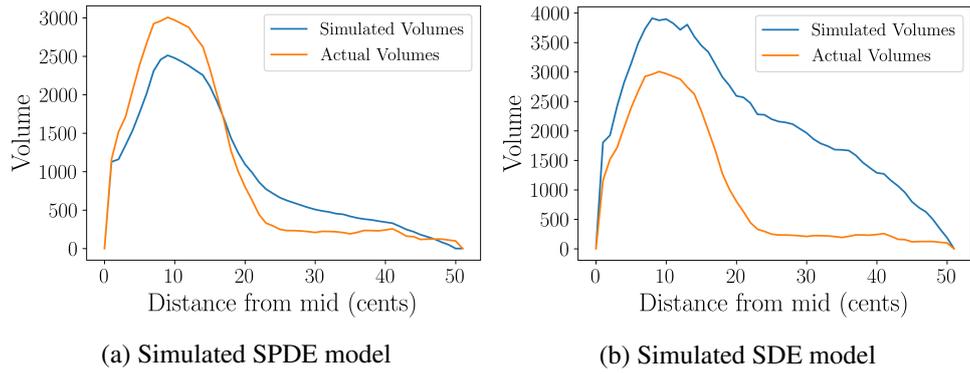


Figure B.2: Average profile of the bid side of the order book over the hour from 11:00am to 12:00pm.

Figure B.3 gives a comparison of the 12-minute snapshots of the simulated profiles of the ask side of the book against the observed data set. Figure B.4 gives these illustrations for the SDE simulations of the ask side of the book when we artificially smooth the volatility.

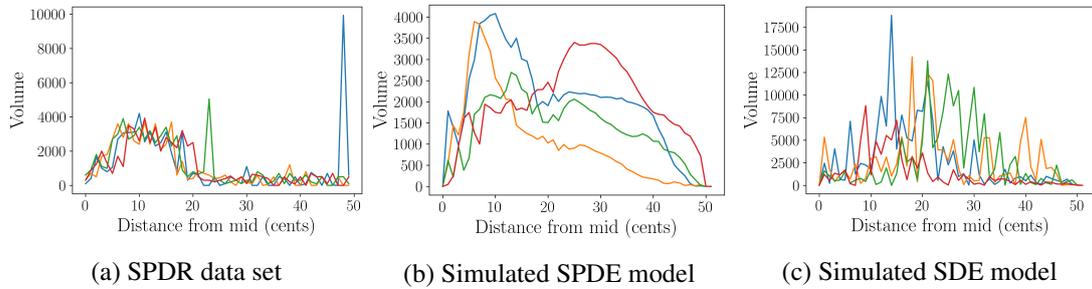


Figure B.3: Snapshots of the profile of the ask side of the order book at 12 minute intervals in the hour from 11:00am to 12:00pm.

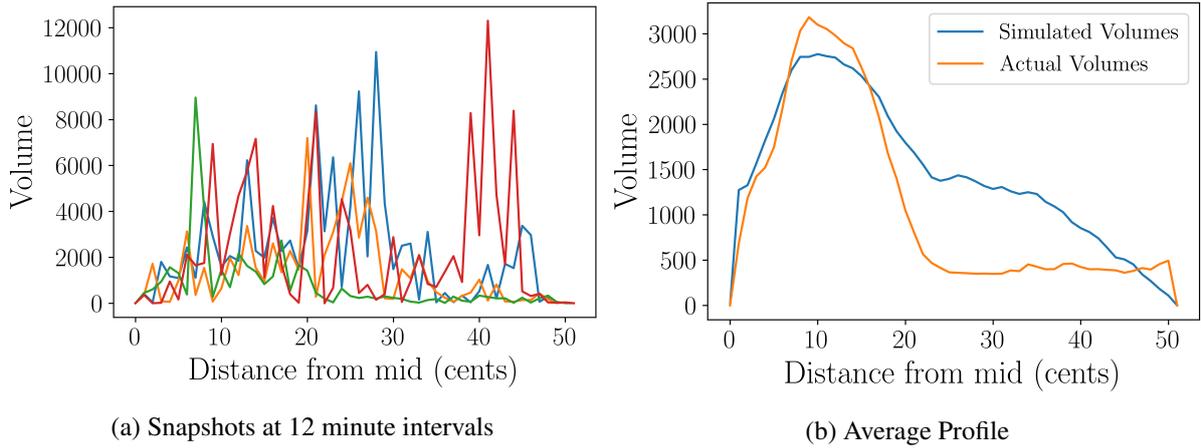


Figure B.4: Simulated profiles for the ask side of the book (with smoothing).

B.3 Estimated Parameters and Simulation Results for March Data Set

In this section we give some of the illustrations of the results of our analysis when performed on a more volatile data set.

B.3.1 Estimated Drift and Covariance Parameters

We begin by plotting the calibrated parameters for each queue in Figure B.5. It is clear from Figure B.5(c) that the correlation is significantly higher in the March data set when compared to the September dataset (Figure 4.5(c)). This is another distinct difference between the two data sets. The estimated drift and volatility are also both far less smooth than their September counterparts (Figures B.5(a) and B.5(b) against 4.5(a) and 4.5(b)).

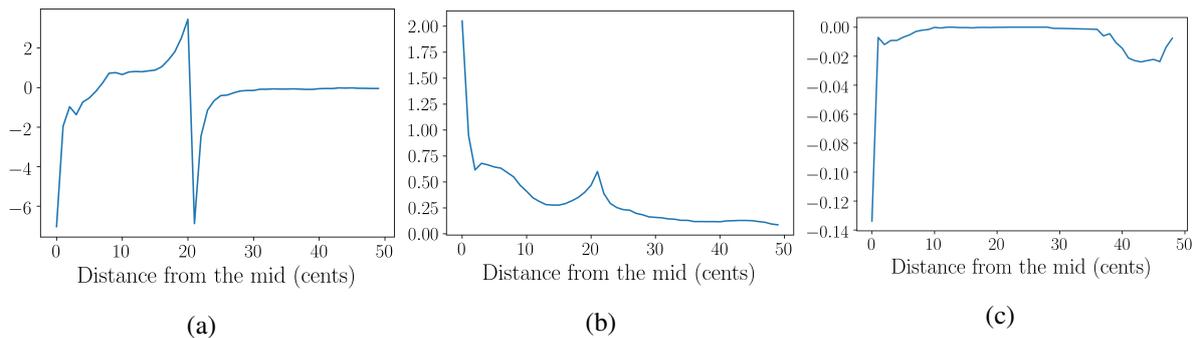


Figure B.5: Estimated drift B.5(a), volatility B.5(b), and correlation B.5(c) functions.

B.3.2 Simulation Results

Finally, we give some of the illustrations of the simulation results of each of the models when applied to the March data set. Figures B.6 and B.7 give the simulated price processes of each model over the hour

and over forty seconds after 11:30am, respectively. We also give the actual price processes over these intervals for reference.

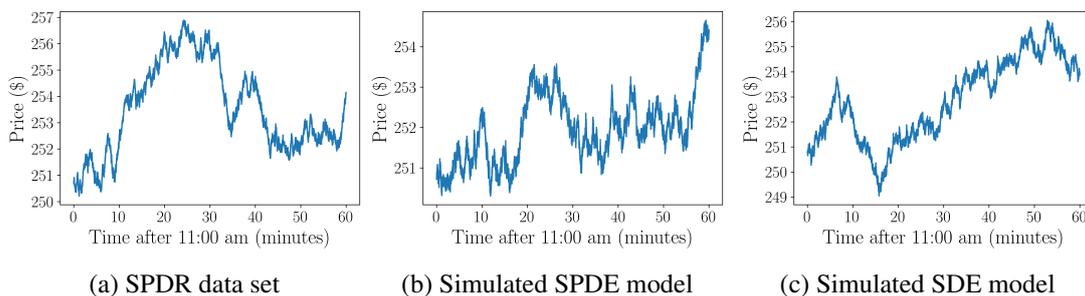


Figure B.6: The price process over the hour from 11:00am to 12:00pm

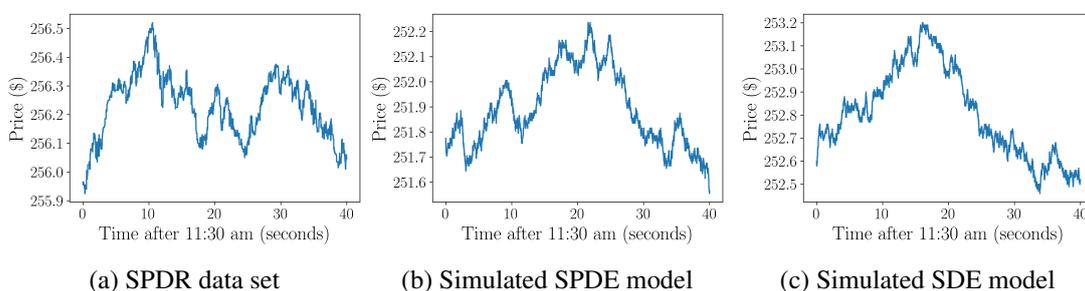


Figure B.7: The price process over the first forty seconds after 11 : 30am

Figures B.8 and B.9 give the simulation results for both models for the bid side of the book.

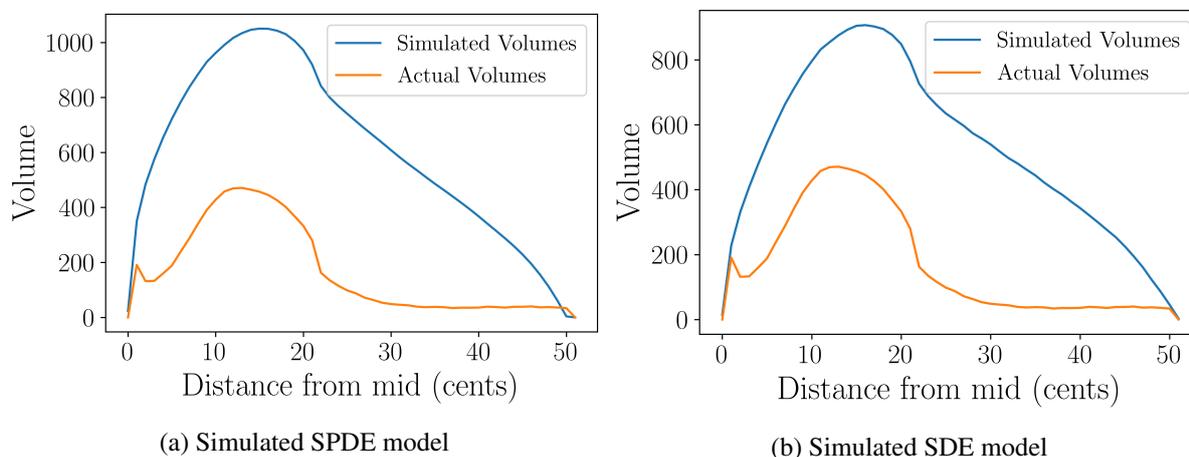


Figure B.8: Average profile of the bid side of the order book over the hour from 11:00am to 12:00pm.

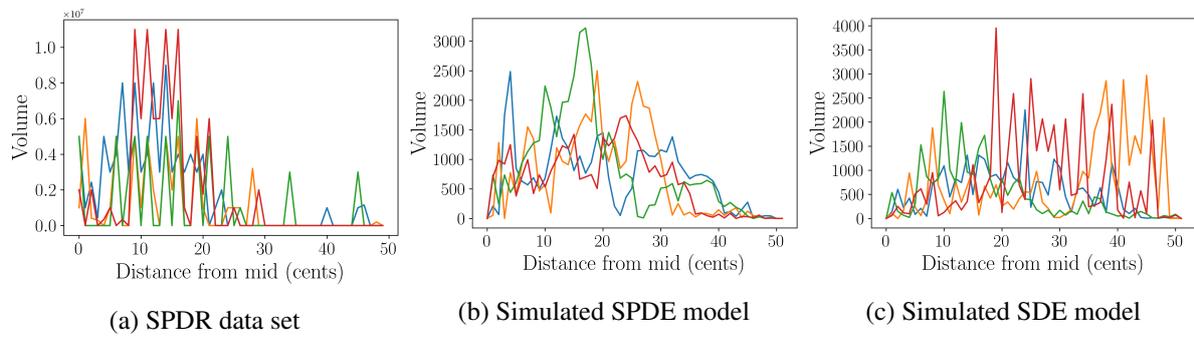


Figure B.9: Snapshots of the profile of the bid side of the order book at 12 minute intervals in the hour from 11:00am to 12:00pm.

Appendix C

Code

C.1 Maximum Likelihood Estimation

In this section we give the code for the maximum likelihood estimation of the state dependent Hawkes process. We use the `mpoints` python package developed in [31] as a framework for this implementation. The implementation is, however, quite different from the `mpoints` because of the distinct differences in the way state-dependence has been included in the exponential Hawkes processes. We begin by giving the functions we used to compute the log-likelihood function. We have used `cython` to improve the speed of these functions.

```
%load_ext cython
import numpy as np
import numpy as np
from libc.math import exp
from libc.math import log

DTYPEf = np.float
DTYPEi = np.int
ctypedef np.float_t DTYPEf_t
ctypedef np.int_t DTYPEi_t

def partial_log_likelihood(int event_type ,
    np.ndarray[DTYPEf_t, ndim=1] mu,
    np.ndarray[DTYPEf_t, ndim=1] alphas ,
    np.ndarray[DTYPEf_t, ndim=1] betas ,
    int n_events ,
    int n_states ,
    np.ndarray[DTYPEf_t, ndim=1] times ,
    np.ndarray[DTYPEi_t, ndim=1] events ,
    np.ndarray[DTYPEi_t, ndim=1] states ,
    np.float start_time ,
    np.float end_time):
    ,,,
    This function calculates the log-likelihood
```

function for a given event.

event_type: The event the log-likelihood function must be calculated for.

mu: vector of base rates (size: n_states x 1)

alphas: vector of alphas (size: n_events x 1)

betas: vector of betas (size: n_events x 1)

n_events: number of event types

n_states: number of states

times: vector of times of event arrivals

events: vector indicating type of event at each time

states: vector indicating state at each time

'''

'''We will use partial sums to store values that we'll use to calculate the log likelihood recursively.'''

```
cdef np.ndarray[DTYPEf_t, ndim=1] partial_sums
```

```
= np.zeros((n_events), dtype=DTYPEf)
```

```
cdef int n, event, state, e, e1, x, index_end
```

```
cdef double time, previous_time, intensity_of_the_event
```

```
cdef DTYPef_t alpha, beta, ratio,
```

```
time_increment, time_increment_2
```

```
# calculating the ratio alpha/beta
```

```
cdef np.ndarray[DTYPef_t, ndim=1] ratios = \
```

```
np.zeros((n_events), dtype=DTYPef)
```

```
for e1 in range(n_events):
```

```
    alpha = alphas[e1]
```

```
    beta = betas[e1]
```

```
    ratios[e1] = alpha / beta
```

```
cdef double log_likelihood = 0
```

```
previous_time = start_time
```

```
index_end = times.shape[0]
```

```
for n in range(0, index_end):
```

```
    time = times[n]
```

```
    event = events[n]
```

```
    state = states[n]
```

```
#Update the partial sums
```

```
time_increment = time - previous_time
```

```
for e1 in range(n_events):
```

```

        beta = betas[e1]
        partial_sums[e1] *= exp(-beta * time_increment)

#Compute the first term of the log-likelihood

    if event == event_type:
        intensity_of_the_event = mu[state]
        for e in range(n_events):
            intensity_of_the_event += partial_sums[e]
            log_likelihood += log(intensity_of_the_event)

#Update the partial sums

    alpha = alphas[event]
    partial_sums[event] += alpha
    previous_time = time

# Compute the second term of the likelihood

    log_likelihood -= mu[state]*time_increment

#Compute the third term of the likelihood

    time_increment = end_time - time
    beta = betas[event]
    ratio = ratios[event]
    log_likelihood -= ratio *
    (1 - exp(-beta * time_increment))

    log_likelihood -= mu[state]*time_increment
    return log_likelihood

def partial_gradient(int event_type ,
                    np.ndarray[DTYPEf_t, ndim=1] mus ,
                    np.ndarray[DTYPEf_t, ndim=1] alphas ,
                    np.ndarray[DTYPEf_t, ndim=1] betas ,
                    int n_events ,
                    int n_states ,
                    np.ndarray[DTYPEf_t, ndim=1] times ,
                    np.ndarray[DTYPEi_t, ndim=1] events ,
                    np.ndarray[DTYPEi_t, ndim=1] states ,
                    np.float start_time ,
                    np.float end_time):
    ...

This function calculates the gradient
function for a given event.

event_type: The event the log-likelihood function

```

```

must be calculated for.
mu: vector of base rates (size: n_states x 1)
alphas: vector of alphas (size: n_events x 1)
betas: vector of betas (size: n_events x 1)
n_events: number of event types
n_states: number of states
times: vector of times of event arrivals
events: vector indicating type of event at
each time
states: vector indicating state at each time

'''
cdef int n, event, state, e, e1, x, e2, index_end
cdef double time, previous_time
cdef DTYPEf_t alpha, beta, ratio, time_increment, \
time_increment_2, sample_duration, a, b, c, \
decay, intensity_of_the_event
sample_duration = end_time - start_time

cdef np.ndarray[DTYPEf_t, ndim=1] gradient_mus = \
    np.zeros((n_states))
cdef np.ndarray[DTYPEf_t, ndim=1] gradient_alphas = \
    np.zeros((n_events))
cdef np.ndarray[DTYPEf_t, ndim=1] gradient_betas = \
    np.zeros((n_events))

# calculating the ratio alpha/beta
cdef np.ndarray[DTYPEf_t, ndim=1] ratios = \
    np.zeros((n_events), dtype=DTYPEf)
for e1 in range(n_events):
    alpha = alphas[e1]
    beta = betas[e1]
    ratios[e1] = alpha / beta

#We'll use the partial sums in the recursion
cdef np.ndarray[DTYPEf_t, ndim=1] partial_sums = \
    np.zeros((n_events), dtype=DTYPEf)
cdef np.ndarray[DTYPEf_t, ndim=1] partial_sums_1 = \
    np.zeros((n_events), dtype=DTYPEf)

previous_time = start_time
index_end = times.shape[0]
for n in range(0, index_end):
    time = times[n]
    event = events[n]
    state = states[n]

    #Update the partial sums

```

```

time_increment = time - previous_time
for e1 in range(n_events):
    beta = betas[e1]
    partial_sums_1[e1] += time_increment * \
    partial_sums[e1]
    decay = exp(-beta * time_increment)
    partial_sums_1[e1] *= decay
    partial_sums[e1] *= decay

#Update the gradients
if event == event_type:
    intensity_of_the_event = mus[state]
    for e in range(n_events):
        intensity_of_the_event += partial_sums[e]
        gradient_mus[state] += 1 / intensity_of_the_event

    for e in range(n_events):
        alpha = alphas[e]
        if alpha != 0:
            gradient_alphas[e] += \
            (partial_sums[e] / alpha) / intensity_of_the_event
            gradient_betas[e] -= \
            partial_sums_1[e] / intensity_of_the_event
#Update the partial sums
    gradient_mus[state] -= time_increment
    alpha = alphas[event]
    partial_sums[event] += alpha
    previous_time = time

time_increment = end_time - time
beta = betas[event]
ratio = ratios[event]
c = 1 - exp(-beta * time_increment)
gradient_alphas[event] -= c / beta
gradient_betas[event] -= \
ratio * time_increment * (1 - c)
gradient_betas[event] -= - ratio * c / beta
gradient_mus[state] -= time_increment

return gradient_mus, gradient_alphas, gradient_betas

```

```

def residuals(np.ndarray[DTYPERf_t, ndim=2] mus,
              np.ndarray[DTYPERf_t, ndim=2] alphas,
              np.ndarray[DTYPERf_t, ndim=2] betas,
              int n_events,
              int n_states,
              np.ndarray[DTYPERf_t, ndim=1] times,
              np.ndarray[DTYPERi_t, ndim=1] events,

```

```

np.ndarray[DTYPEi_t, ndim=1] states ,
DTYPEf_t start_time ,
np.ndarray[DTYPEf_t, ndim=2] initial_partial_sums ):
    , , ,

```

This function calculates the residuals of a given model.

event_type: The event the log-likelihood function must be calculated for.

mu: vector of base rates (size: n_states x 1)

alphas: vector of alphas (size: n_events x 1)

betas: vector of betas (size: n_events x 1)

n_events: number of event types

n_states: number of states

times: vector of times of event arrivals

events: vector indicating type of event at each time

states: vector indicating state at each time

```

    , , ,

cdef int length = times.shape[0]
cdef np.ndarray[DTYPEf_t, ndim=2] residuals = \
np.zeros((n_events, length-index_start + 1), dtype=DTYPEf)

cdef np.ndarray[DTYPEi_t, ndim=1] residuals_lengths \
= np.zeros(n_events, dtype=DTYPEi)
cdef np.ndarray[DTYPEf_t, ndim=1] previous_times = \
start_time*np.ones(n_events, dtype=DTYPEf)
cdef np.ndarray[DTYPEf_t, ndim=2] partial_sums = \
np.zeros((n_events, n_events), dtype=DTYPEf)
cdef np.ndarray[DTYPEf_t, ndim=2] partial_sums_old = \
np.zeros((n_events, n_events), dtype=DTYPEf)
cdef np.ndarray[DTYPEf_t, ndim=2] ratios = \
np.zeros((n_events, n_events), dtype=DTYPEf)
cdef int e1, x, e2, n, e, event, state, i, pos
cdef DTYPEf_t alpha, beta, time, time_last

#Compute ratios alpha/beta
for e1 in range(n_events):
    for e2 in range(n_events):
        alpha = alphas[e1, e2]
        beta = betas[e1, e2]
        if(beta!=0):
            ratios[e1, e2] = alpha / beta

for e1 in range(n_events):
    for e2 in range(n_events):
        partial_sums[e1, e2] += /

```

```

        initial_partial_sums[e1, e2]
        partial_sums_old[e1, e2] = partial_sums[e1, e2]

#Calculating the residuals
time_last = start_time

for n in range(index_start, length):
    time = times[n]
    event = events[n]
    state = states[n]
    pos = residuals_lengths[event]

    residuals[event, pos] += /
    (time - previous_times[event])*mus[event, state]

    for e in range(n_events):
        if e != event:
            i = residuals_lengths[e]
            residuals[e, i] += ratios[event, e]
        if e == event:
            # this event contributes to the next residual
            residuals[e, pos+1] += ratios[event, e]
    #Update partial sums up to current time (excluding):
    for e1 in range(n_events):
        for e2 in range(n_events):
            partial_sums[e1, e2] *= /
            exp(-betas[e1, e2] * (time - time_last))
    #Add the partial sums
    for e in range(n_events):
        residuals[event, pos] += \
        partial_sums_old[e, event] \
        - partial_sums[e, event]
        # save new partial sums
        partial_sums_old[e, event] =\
        partial_sums[e, event]

    for e in range(n_events):
        partial_sums[event, e] += ratios[event, e]

#Update position
time_last = time
previous_times[event] = time
residuals_lengths[event] += 1

result = []
for e in range(n_events):
    length = residuals_lengths[e]
    result.append(residuals[e, 0:length])
return result

```

We used the following class to implement the functions above. Note that this is the class used to fit the six-dimensional state-dependent Hawkes process.

```
import numpy as np
import math
import copy
import scipy.optimize as opt

class HawkesExp:

    def __init__(self, n_events, n_states, events_labels, \
                 states_labels, forced_zeros):

        self.n_events = n_events
        self.n_states = n_states
        self.events_labels = events_labels
        self.states_labels = states_labels
        self.forced_zeros = forced_zeros
        self.mus = np.zeros((n_events, n_states))
        self.alphas = np.zeros((n_events, n_events))
        self.betas = np.zeros((n_events, n_events))
        self.ratios = np.zeros((n_events, n_events))

    def set_hawkes_parameters(self, mus, alphas, betas):

        self.mus = copy.copy(mus)
        self.alphas = copy.copy(alphas)
        self.betas = copy.copy(betas)
        self.ratios = np.divide(alphas, betas)

    def estimate_hawkes_parameters(self, times, events, states,
                                   start_time, end_time, maximum_number_of_iterations=2000,
                                   method='TNC', parameters_lower_bound=10**(-6),
                                   parameters_upper_bound=None,
                                   min_decay_coefficient=0.5, max_decay_coefficient=100):

        #Generate initial guesses
        guesses = []
        if np.shape(min_decay_coefficient) == ():
            # if a scalar was given instead of a matrix
            min_betas = min_decay_coefficient * \
                np.ones((self.n_events, self.n_events))
        if np.shape(max_decay_coefficient) == ():
            # if a scalar was given instead of a matrix
            max_betas = max_decay_coefficient * \
                np.ones((self.n_events, self.n_events))
```

```

#Generate some random initial values
#average intensities
average_intensities = \
np.zeros((self.n_events, self.n_states))
for n in range(len(times)):
    e = events[n]
    s = states[n]
    average_intensities[e,s] += 1
average_intensities = \
np.divide(average_intensities, end_time - start_time)

#mus
guess_mus = \
np.zeros((self.n_events, self.n_states))

for e in range(self.n_events):
    for x in range(self.n_states):
        guess_mus[e,x] = average_intensities[e,x]/2

#betas
guess_betas = \
np.zeros((self.n_events, self.n_events))
for e1 in range(self.n_events):
    for e2 in range(self.n_events):
        u_min = math.log10(min_betas[e1, e2])
        u_max = math.log10(max_betas[e1, e2])
        u = np.random.uniform(u_min, u_max)
        beta = 10 ** u
        guess_betas[e1, e2] = beta

#alphas
guess_alphas = \
np.zeros((self.n_events, self.n_events))
for e1 in range(self.n_events):
    for e2 in range(self.n_events):
        u = np.random.uniform(0, 1)
        alpha = u * guess_betas[e1, e2]
        guess_alphas[e1, e2] = alpha

#list of random guess
g = self.parameters_to_array(guess_mus,
guess_alphas, guess_betas)

dimension = self.n_states + 2 * self.n_events
bounds = \
[(parameters_lower_bound, parameters_upper_bound)]\
* dimension

```

```

opt_mus = np.zeros((self.n_events, self.n_states))
opt_alphas = \
np.zeros((self.n_events, self.n_events))
opt_betas = \
np.zeros((self.n_events, self.n_events))
success = True
successes = []
status = -999
statuses = []
message = \
'Multiple messages because parallel estimation'
messages = []
fun = 0
jacs = []
hesss = []
nfev = 0
nit = 0
kinds_of_best_initial_guesses = ''
for e in range(self.n_events):
    #define negative likelihood
    def likelihood_minus(parameters):
        result = - self.log_likelihood_partial(e,
        parameters, times, events, states,
        start_time, end_time)
        return result
    def gradient_of_likelihood_minus(parameters):
        result = - self.gradient_partial(e,
        parameters, times, events, states,
        start_time, end_time)
        return result
    # optimise likelihood
    optimal_results = []

    guess_mus, guess_alphas, guess_betas = \
self.array_to_parameters(g, self.n_events,
self.n_states, self.n_events)
    g_partial = \
self.parameters_to_array(guess_mus[e:e+1,:],
guess_alphas[:,e:e+1], guess_betas[:,e:e+1])

    o = opt.minimize(likelihood_minus, g_partial,
method=method, bounds=bounds,
jac=gradient_of_likelihood_minus,
options={'maxiter':
maximum_number_of_iterations})
    optimal_results.append(o)

# Save optimal parameters
v, a, b = self.array_to_parameters(o.x,

```

```

self.n_events, self.n_states, 1, e)
opt_mus[e:e+1,:] = v
opt_alphas[:,e:e+1] = a
opt_betas[:,e:e+1] = b

# Save optimiser information
successes.append(o.success)
statuses.append(o.status)
messages.append(o.message)
if success and not o.success:
    success = False
    status = o.status
fun += o.fun
jacs.append(o.jac)
nfev += o.nfev
nit += o.nit

o = opt.OptimizeResult()
x = self.parameters_to_array(opt_mus, opt_alphas,
opt_betas)
o['x'] = x
o['success'] = success
o['successes'] = successes
o['status'] = status
o['statuses'] = statuses
o['message'] = message
o['messages'] = messages
o['fun'] = fun
o['jacs'] = jacs
o['hesss'] = hesss
o['nfev'] = nfev
o['nit'] = nit
return o, g

```

```

def gradient_partial(self, event_type, parameters, times,
events, states, start_time, end_time):
    n_events = self.n_events
    n_states = self.n_states
    mu, alphas, betas = \
        self.array_to_parameters(parameters, n_events,
n_states, 1, event_type)
    g_mu, g_alphas, g_betas = \
        partial_gradient(event_type, mu[0,:], alphas[:,0],
betas[:,0], n_events, n_states, times, events,
states, np.float(start_time), np.float(end_time))

a = np.zeros((n_events, 1))
b = np.zeros((n_events, 1))

```

```

c = np.zeros((1, n_states))
a[:, 0] = g_alphas
b[:, 0] = g_betas
c[0, :] = g_mu

return self.parameters_to_array(c, a, b)

def log_likelihood_partial(self, event_type, parameters,
times, events, states, start_time, end_time):

n_events = self.n_events
n_states = self.n_states
mu, alphas, betas = \
    self.array_to_parameters(parameters, n_events,
n_states, 1, event_type)
return partial_log_likelihood(event_type, mu[0, :],
alphas[:,0], betas[:,0], n_events, n_states,
times, events, states, np.float(start_time),
np.float(end_time))

def array_to_parameters(self, array, n_events_1, n_states,
n_events_2=0, event_type=5):

if n_events_2 == 0:
n_events_2 = n_events_1
mus = np.zeros((n_events_2, n_states))
for n in range(n_events_2):
for s in range(n_states):
index = n*n_states + s
mus[n,s] = array[index]

alphas = np.zeros((n_events_1, n_events_2))
for i in range(n_events_1):
for k in range(n_events_2):
index = n_events_2*n_states+i * n_events_2 + k
alphas[i, k] = array[index]

betas = np.zeros((n_events_1, n_events_2))
for i in range(n_events_1):
for k in range(n_events_2):
index = n_events_2 * n_states
index += n_events_1 * n_events_2
index += i * n_events_2 + k
betas[i, k] = array[index]

if(self.forced_zeros):
#Setting these values to (very close to) zero
#doesn't change the value of log-likelihood
#if these values are ignored.

```

```

if (n_events_2==1):
    if(event_type == 0):

        alphas [2] = 10**-12
        betas [2] = 1
        alphas [4:] = 10**-12
        betas [4:] = 1

    elif(event_type == 1):
        alphas [2:] = 10**-12
        betas [2:] = 1

    elif(event_type == 2):
        alphas [0:2] = 10**-12
        alphas [4] = 10**-12
        betas [0:2] = 1
        betas [4] = 1

    elif(event_type == 3):
        alphas [0:2] = 10**-12
        alphas [4:] = 10**-12
        betas [0:2] = 1
        betas [4:] = 1

    elif(event_type == 4):
        alphas [0:4] = 10**-12
        betas [0:4] = 1

    elif(event_type == 5):
        alphas [0:4] = 10**-12
        betas [0:4] = 1

elif (n_events_2==6):

    alphas [0 ,2:] = 10**-12
    alphas [1 ,2:] = 10**-12
    alphas [2 ,0:2] = 10**-12
    alphas [2 ,4:] = 10**-12
    alphas [3 ,1] = 10**-12
    alphas [3 ,4:] = 10**-12
    alphas [4 ,0:4] = 10**-12
    alphas [5 ,0:2] = 10**-12
    alphas [5 ,3] = 10**-12

    temp = np.where( alphas == 10**-12)
    betas [temp] = 1

return mus, alphas , betas

```

```

def parameters_to_array(self, mus, alphas, betas):

    s = np.shape(alphas)
    t = np.shape(mus)
    n_events_1 = s[0]
    n_events_2 = s[1]
    n_states = t[1]

    result =\
        np.zeros(n_events_2 * n_states + \
            2 * n_events_1 * n_events_2)
    for n in range(n_events_2):
        for j in range(n_states):
            index = n*n_states + j
            result[index] = mus[n,j]
    for i in range(n_events_1):
        for k in range(n_events_2):
            index = n_events_2*n_states
            index += i * n_events_2 + k
            result[index] = alphas[i, k]
    for i in range(n_events_1):
        for k in range(n_events_2):
            index = n_events_2 * n_states
            index += n_events_1 * n_events_2
            index += i * n_events_2 + k
            result[index] = betas[i, k]

    return result
def compute_events_residuals(self, times, events, states,
start_time, initial_partial_sums=0):

    # Check if no initial partial sums if given
    s = np.zeros((self.n_events, self.n_events))
    if len(np.shape(initial_partial_sums)) != 0:
        s = initial_partial_sums
        s = np.divide(s, self.betas)
    return residuals(self.mus, self.alphas, self.betas,
self.n_events, self.n_states,
times, events, states, start_time, s)

```

Bibliography

- [1] Abergel, F. and Jedidi, A. (2013). "A Mathematical Approach to Order Book Modeling." *Economics of Order Driven Markets*, Springer,93-107. 2013.
- [2] Abergel, F. and Jedidi, A. (2015). "Long Time Behaviour of a Hawkes Process-Based Limit Order Book." *SIAM Journal on Financial Mathematics*, 6(1):1026-1043.
- [3] Abergel, F. et al. (2016). *Limit Order Books*. Cambridge University Press.
- [4] Bacry, E., Delattre, S., Hoffmann, M., and Muzy, J-F. (2012). "Scaling Limits for Hawkes Processes and Application to Financial Statistics." Available at: arXiv:1202.0842.
- [5] Bacry, E., and Mastromatteo, I. (2012). "Hawkes Processes in Finance." *Market Microstructure and Liquidity*, 1(1):1550005, 59 pages.
- [6] Bacry, E., Jaisson, T., and Muzy, J-F. (2016). "Estimation of slowly decreasing Hawkes kernels: application to high-frequency order book dynamics." *Quantitative Finance*, 16(8): 1179-1201. DOI: 10.1080/14697688.2015.1123287.
- [7] Bacry, E., Bompairé, M., Gaïffas, S., and Poulsen, S. (2018). "tick: a Python library for statistical learning, with a particular emphasis on time-dependent modeling." *The Journal of Machine Learning Research* 13(1):1-5.
- [8] Bayer, C., Horst, U. and Qiu, J. (2017). "A Functional Limit Theorem for Limit Order Books with State Dependent Price Dynamics." *Annals of Applied Probability* 27(5): 2753-806. DOI:10.1214/16-AAP1265
- [9] Chen, Y. (2017). "Modelling Limit Order Book Dynamics Using Hawkes Processes." PhD thesis, Florida State University College of Arts and Sciences.
- [10] Cont, R., Stoikov, S. and Talreja, R. (2010). "A Stochastic Model for Order Book Dynamics." *Operations Research* 58(3):549-63. Available at JSTOR, www.jstor.org/stable/40792679.
- [11] Cont, R. and De Larrard, A. (2012). "Order book dynamics in liquid markets: limit theorems and diffusion approximations." arXiv preprint, arXiv:1202.6412.

- [12] Cont, R., and De Larrard, A. (2013). "Price Dynamics in a Markovian Limit Order Market." *SIAM Journal on Financial Mathematics* 4(1): 1-25. Available at SSRN: <https://ssrn.com/abstract=1735338> or <http://dx.doi.org/10.2139/ssrn.1735338>
- [13] Cont, R., and Marvin M. (2019). "A Stochastic Pde Model For Limit Order Book Dynamics." IDEAS Working Paper Series from RePEc (2019): IDEAS Working Paper Series from RePEc.
- [14] Cox, J. C., Ingersoll Jr, J. E. and Ross, S. A. (1985). "A theory of the term structure of interest rates." *Econometrica* 53:385–407.
- [15] Daley, D.J., and Vere-Jones, D. (2003). *An Introduction to the Theory of Point Processes Volume 1*. Springer.
- [16] Filimonov, V. and Sornette, D. (2012). "Quantifying reflexivity in financial markets: Toward a prediction of flash crashes." *Quantitative Finance*, 15(8):1293-1314, DOI: 10.1080/14697688.2015.1032544
- [17] Gould, M. D., M. A. Porter, S. Williams, M. McDonald, D. J. Fenn, and S. D. Howison. (2013). "Limit Order Books." *Quantitative Finance* 13(11): 1709-742. DOI: 10.1080/14697688.2013.803148.
- [18] Hambly, B., Kalsi, J., and Newbury, J. (2020). "Limit order books, diffusion approximations and reflected SPDEs: from microscopic to macroscopic models." *Applied Mathematical Finance* 27(1-2):132-170. DOI: 10.1080/1350486X.2020.1758176.
- [19] Hawkes, A. G.(1971). "Point spectra of some mutually exciting point processes." *Biometrika*, 58(1):83–90.
- [20] Hawkes, A. G. and Oakes, D. (1974). "A cluster process representation of a self- exciting process." *Journal of Applied Probability*, 11(3):493–503.
- [21] Hawkes, A. G. (2018). "Hawkes Processes and Their Applications to Finance: A Review." *Quantitative Finance* 18(2):193-198. DOI: 10.1080/14697688.2017.1403131.
- [22] Heston, S. L. (1993). "A closed-form solution for options with stochastic volatility with applications to bond and currency options." *Review of Financial Studies* 6:327–343.
- [23] Hewlett, P. (2006). "Clustering of order arrivals, price impact and trade path optimisation." *Workshop on Financial Modeling with Jump Processes, Ecole Polytechnique*, pages 6–8.
- [24] Huang, W., Lehalle, C-A. and Rosenbaum, M. (2015). "Simulating and analyzing order book data: the queue-reactive model". *Journal of the American Statistical Association* 110(509):107-122. DOI: 10.1080/01621459.2014.982278.

- [25] Horst, U, and Paulsen, M. (2015). "A Law of Large Numbers for Limit Order Books." IDEAS Working Paper Series from RePEc (2015): IDEAS Working Paper Series from RePEc. Available at <https://doi.org/10.1287/moor.2017.0848>.
- [26] Horst, U, and Xu, W. (2017). "A Scaling Limit for Limit Order Books Driven by Hawkes Processes." *SIAM Journal on Financial Mathematics*, 10(2), 350–393. Available at <https://doi.org/10.1137/17M1148682>.
- [27] Jaisson, T., and Rosenbaum, M. (2015). "Limit Theorems for Nearly Unstable Hawkes Processes." *Annals of Applied Probability* 25(2): 600-31. Available at www.jstor.org/stable/24519929.
- [28] Keller-Ressel, M. and Muller, M. (2016). "A Stefan-type stochastic moving boundary problem." *Stochastics and Partial Differential Equations: Analysis and Computations* 4: 746–790. Available at <https://doi.org/10.1007/s40072-016-0076-z>.
- [29] Huang, R., and Polak., T. (2011). "LOBSTER: Limit Order Book Reconstruction System." *SSRN Electronic Journal* (2011). Available at SSRN: <https://ssrn.com/abstract=1977207> or <http://dx.doi.org/10.2139/ssrn.1977207>
- [30] Mishura, Y., and Shevchenko, G. (2017). *Theory and Statistical Applications of Stochastic Processes*, Wiley.
- [31] Morariu-Patrichi, M., and Pakkanen, M.S. (2018). "State-dependent Hawkes Processes and Their Application to Limit Order Book Modelling." Available at arXiv:1809.08060.
- [32] Parlour, C. and Seppi, D.J.. (2008). "Limit order markets: A survey." *Handbook of financial intermediation and banking* 5: 63-95.
- [33] P.Wu, M. Rambaldi, J.Muzy, E.Bacry. (2019). "Queue-reactive Hawkes models for the order flow". Preprint. Available at arXiv:1901.08938v1.