Master of Science in Advanced Analytics and Decision Sciences

Calibrating a Latent Order Book Model to Market Data

Michael Gant

Supervisor: A/Prof. T. Gebbie

Department of Statistical Sciences,

University of Cape Town, Rondebosch

May 18, 2022

Abstract

We investigate the formulation of the Latent Order Book (LOB) as a reaction-diffusion Partial Differential Equation (PDE) and its subsequent numerical solution through an explicit method based on discrete stochastic processes. The numerical solution is calibrated using likelihood-free methods, Approximate Bayesian Computation (ABC) and an iterative extension, Population Monte-Carlo ABC (PMC-ABC) as well as a Black-box approach using the Nelder-Mead algorithm. We show that in the diffusion limit, the master equation becomes the LOB reaction-diffusion PDE and certain free-parameters are recoverable with the iterative calibration techniques.

Acknowledgements

I would like to thank my supervisor, Tim Gebbie, for his understanding and support over a tumultuous period. His discussions, knowledge and the people he introduced me to were incredibly valuable in my understanding of the topic. His patience is also something that I hugely appreciate and can only hope to emulate with others.

Additional thank you's are extended to my peers who helped me along this journey. Specifically, Donovan Platt for his interest, advice and support early on. A massive thank you to Kelly Goosen for the proofreading, various technical discussions and most importantly, the motivational advice to finish this dissertation.

One last thank you to all my colleagues, friends and family who were still there for me over this long and bumpy journey.

Computations were performed using facilities provided by the University of Cape Town's ICTS High Performance Computing team: hpc.uct.ac.za.

Contents

List of Figures

List of Tables

1 Introduction

The modelling of financial time series is a significant area of research in both academia and industry as it is the dominant approach towards finding explanations of past market events, and to help predict and plan for future events and outcomes (Carmona, 2014; Sewell, 2011). This is a vast body of knowledge that spans Econometrics, much of Statistics and Applied Mathematics, but it is generally grounded in the estimation of time series models that assume shared statistical properties relating to residuals in specified time series models, rather than approaches encoding shared or universal properties of the system itself that can be represented as partial differential equations. Studying potential hidden mechanisms, universal features, and collective structures of the market has been seen to not only led to profitable insights and the satisfaction of overcoming uncertainty, but also as a means of beating competitors and managing risk. Ever since the derivation of the Black-Scholes-Merton equation and its calibration, there has been a quest for similar methods to capture market dynamics and to find appropriate representations of financial market phenomenology that can be recovered from or reduced to stochastic continuous-time representations (Angstmann et al., 2016; Tóth et al., 2011; Cartea et al., 2015).

However, a key feature of financial markets is that they adaptively emerge through time as the result of vast numbers of interacting strategic agents. This makes experimentation in financial markets difficult. The problem is the modelling and understanding of financial markets is an observational science, and there is only a single series of observations over time. There is no opportunity to restart the systems to explore parameters, or to perfectly replicate environments for double-blind studies. This makes financial time series modelling one of the most perplexing areas for statistical modelling. The standard statistical assumptions of independent and identically distributed observations do not hold and one has to be cautious of over-fitting as well data contamination from model inferences. On long horizons, financial market time series are represented by sequences of daily sampled market closing prices. These prices are the result of the end of the day closing auction where participants submit buy and sell orders which are then matched in a Walrasian auction that results in an inadequate equilibrium price where supply and demand are reasonably matched. This does not conform to how prices are discovered and realised in intraday trading.

This makes the modelling of intraday trading, continuous time trading between the market opening and the market close each day, a particularly troubling problem. An interesting and fundamental phenomenon in the trading domain is the effect of an executed trade on the price of a stock. This phenomenon is called price impact (Bouchaud, 2009). Although the phenomenon of price impact is empirically well understood, there is not an explicitly defined theoretical framework that allows one to derive the measured mechanism in the stock market from some underlying simplifying principles. Prices for stocks are not fixed by the seller or buyer, rather, the price fluctuates based on information arrival, supply and demand, and the vagaries of agent-feedbacks and strategic decision making. These fluctuations are facilitated by the environment in which stocks are exchanged, with orders to buy and sell being stored and matched on a limit order book.

The limit order book (Roşu, 2009) is a centralised store of traders' intentions to buy and sell stocks. The limit order book is made up of Limit Orders (LO) and Market Orders (MO), each with a specified intention to buy or sell an amount of a certain

stock. A limit order is an offer to buy (bid) or sell (ask) a specified amount of the stock for a specified price. A market order is the other half to a limit order, it is a request to buy or sell a specified amount of stock at the current best price. A market order will match with the relevant limit order or limit orders immediately after being made, concluding the trade event. The limit-order book we can observe in the market is the lit order book. These are the actual orders placed by traders in the market and can be seen by other traders. However, there exists the latent demand of the market more generally, the latent intentions of mutual funds, traders and other market participants. This is represented by the latent order book.

This dissertation aims to formulate a model which can be used to study the price impact of trades in a limit order book environment. We formulate a simple model of the latent order book as a reaction-diffusion partial differential equation, which we then numerically solve using an explicit method based on discrete stochastic processes. This numerical solution is calibrated to synthetic and market data using likelihood-free methods: Approximate Bayesian Computation (ABC) and an iterative extension, Population Monte-Carlo ABC (PMC-ABC), as well as a Black-box approach using the Nelder-Mead algorithm. With a model calibrated to market data, we then propose the addition of scheduled market orders of varying sizes into the latent order book model to calculate price impact across trade volume.

The dissertation is presented as follows. In Section 2, we introduce the latent order book and formulate its mathematical representation by deriving the reaction-diffusion partial differential equation. We then explore and extend the numerical solution proposed in Angstmann et al. (2016) to solve the latent order book PDE in Section 3. We show that in the diffusion limit, the master equation of the numerical scheme becomes the latent order book PDE. Section 4, formulates the Latent Order Book (LOB) model and its extension to facilitate shocks, the Sequential Latent Order Book (SLOB) model. We describe the input configuration parameters, the free-parameters and the outputs of each model.

With the model formulation, we can then proceed to model calibration. In Section 5, we define the calibration framework and its four components. We detail the likelihood-free calibration techniques, the summary statistics, the distance functions and the simulator function. In Section 6, we evaluate the calibration framework with synthetic data generated by the underlying model and decide on which likelihood-free calibration techniques will be used when calibrating the LOB and SLOB models to market data. In Section 7, we calibrate the LOB and SLOB models to market data from the Johannesburg Stock Exchange (JSE). We plot stylised facts for the market data as well as for synthetic data from the calibrated models which we use to evaluate how well the models simulate a real mid-price path. Finally, in Section 8, we summarise the results of the LOB and SLOB calibrations to market data and conclude on modelling the latent order book, likelihood-free calibration techniques and the use of the LOB and SLOB models when investigating price impact.

2 The Latent Order Book

2.1 Latent Order Book Definition

The latent order book is a representation of the latent liquidity in an order book (Tóth et al., 2011) which differs from the standard limit order book. In the standard limit order book, orders to buy and sell an asset are registered as bids (buys) and asks (sells) respectively. However, these observed orders are not the only intentions that exist in the market at any time. Remember that as a trader, you are not incentivised to reveal your intentions as this information could allow a competing trader to take advantage of your position and future trades. Thus, a trader will only reveal their intentions when a set of conditions is satisfied, such as the price moving a sufficient amount. These revealed and hidden intentions are the latent liquidity in the latent order book.

Having established the reason for the existence of latent liquidity, we further need to justify its significance when compared to the actual liquidity in a stock market. If one considers the volume of orders present at any one time in a limit order book, and then compares that volume to the total transaction volume over one day, the total transaction volume over one day is 10 to 100 times more than the instantaneously available volume (Tóth et al., 2011; Donier et al., 2015). This implies that the liquidity is revealed dynamically over time and the magnitude is clearly significant enough to justify the study of latent liquidity and the latent order book.

To represent the latent liquidity in the latent order book, we define $\varphi(x,t)$ as the latent liquidity or latent density at price x and time t. The latent liquidity can be further decomposed as the difference between the latent ask liquidity, $\rho_A(x,t)$ and latent bid liquidity, $\rho_B(x,t)$:

$$\varphi(x,t) = \rho_B(x,t) - \rho_A(x,t) \tag{1}$$

Thus, positive values of $\varphi(x,t)$ represent latent bid order density and negative values of $\varphi(x,t)$ represent latent ask order density. The mid-price is defined as the value between the best ask and the best bid, which is the x value in the latent order book where $\varphi(x,t) = 0 = \rho_B(x,t) - \rho_A(x,t)$. Visually, the latent order book can be pictured in conjunction with a representation of the limit order book as shown in Figure 1 below:

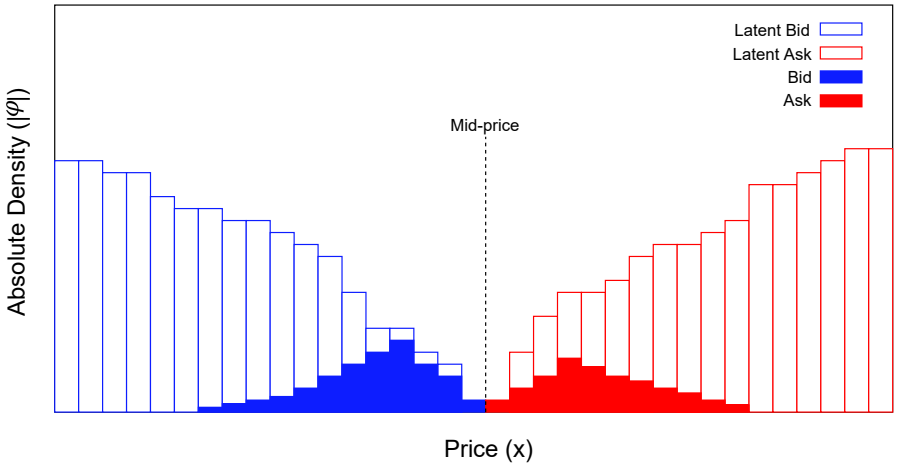


Figure 1: A simplified visual representation of the limit order book and the latent order book. The solid colour bars represent the observable bid and ask orders in the limit order book and the white-filled bars represent the latent bid and ask orders in the latent order book.

The intuition of the latent order book should become clear from this figure. One can see the magnitude of the latent liquidity and how it becomes revealed closer to the mid-price. However, this figure is not an accurate representation of the latent order book and should be viewed as a resource to assist with understanding the intuition of the latent and limit order books.

With a model for latent liquidity, we move on to describing the dynamics that move the latent liquidity within the price and time domain. We follow the derivation in Tóth et al. (2011) and Donier et al. (2015) and begin with the zero-intelligence model of Smith et al. (2003) within the context of the latent order book. Consider how the latent density changes over a small time period Δt (between time t and $t + \Delta t$). New bids and asks can arrive at various prices, existing orders can be re-evaluated and placed at new prices, existing orders can be cancelled and lastly, matches between bids and asks resulting in trades can occur. These four core components are enough to construct a reaction-diffusion model of the latent order book.

2.2 Deriving the Latent Order Book Reaction-Diffusion Equation

The latent order book reaction-diffusion model in two dimensions (price and time), consists of two types of particles, A and B particles, representing latent ask and latent bid orders respectively. The particles diffuse along the price domain and disappear when they meet a particle of the other type, $A + B = \varnothing$. Here the four components of the latent liquidity dynamics and the reaction-diffusion model give us two partial differential equations (Donier et al., 2015):

$$\frac{\partial \rho_B(x,t)}{\partial t} = D\frac{\partial^2 \rho_B(x,t)}{\partial x^2} - V_t\frac{\partial \rho_B(x,t)}{\partial x} - \nu\rho_B(x,t) + s_B(x,t) - \kappa R_{AB}(x,t) \qquad (2)$$

$$\frac{\partial \rho_A(x,t)}{\partial t} = \underbrace{D\frac{\partial^2 \rho_A(x,t)}{\partial x^2} - V_t \frac{\partial \rho_A(x,t)}{\partial x}}_{\text{1. Drift-Diffusion}} - \underbrace{\nu \rho_A(x,t)}_{\text{2. Cancellation}} + \underbrace{s_A(x,t)}_{\text{3. Deposition}} - \underbrace{\kappa R_{AB}(x,t)}_{\text{4. Reaction}} \tag{3}$$

The terms correspond to the previously discussed latent liquidity dynamics and are further elaborated on below:

1. *Drift-Diffusion*: The first two terms describe order re-evaluation which can be due to a variety of reasons such as the arrival of new information, changes to other assets causing portfolio adjustments or price movements themselves. The diffusion term with diffusion constant, D, controls the rate of diffusion and thus price volatility. The drift term with the stochastic variable V_t, for simplicity we assume to be a white noise process ($V_t \sim N(0, \sigma)$), ensures the mid-price path, p_t, is a diffusive random walk.
2. *Cancellation*: Latent liquidity is removed (partially or completely) proportionally to the cancellation rate, ν, and is assumed to be independent of the price level, x.
3. *Deposition*: The source terms, $s_B(x,t)$ and $s_A(x,t)$, correspond to the arrival of new latent liquidity in the latent order book. The source term in Donier et al. (2015) is an arbitrary increasing function ($\Theta(x-p_t)$ and $\Theta(p_t-x)$) modulated by free-parameter λ. Essentially, this source term should place buy order density below the best bid and place sell order density above the best ask with more density placed further away from p_t. Although the authors in Donier et al. (2015) decide on a simple step function due to the irrelevance of the source term in their paper, we consider a bid-ask agnostic two free-parameter (λ and μ) source term as:

$$s(x,t) = \lambda \tanh\left(\mu(p_t - x)\right) = s_B(x,t) - s_A(x,t) \tag{4}$$

   which allows us to control the magnitude of arriving density near the mid-price and towards the ends of the latent order book. Free-parameter μ controls the "steepness" of the arriving density around the mid-price and λ controls the total amount of liquidity arriving over all price levels.
4. *Reaction*: The reaction term corresponds to when two orders meet with reaction rate, κ, where $R_{AB}(x,t)$ is the average of the product of the density of A and B particles, $R_{AB}(x,t) \approx \rho_A(x,t)\rho_B(x,t)$ (Donier et al., 2015). However, this term disappears when we consider the latent order book density, $\varphi(x,t) = \rho_B(x,t) - \rho_A(x,t)$, with the derivation shown in Appendix A.

Using Equation 1 and the two partial differential Equations, 2 and 3, we can obtain the latent order book reaction-diffusion partial differential equation:

$$\frac{\partial \varphi(x,t)}{\partial t} = D\frac{\partial^2 \varphi(x,t)}{\partial x^2} - V_t \frac{\partial \varphi(x,t)}{\partial x} - \nu\varphi(x,t) + s(x,t) \tag{5}$$

where the mid-price, p_t, is the value of x which satisfies $\varphi(x,t) = 0$. Here we have a PDE which can be solved over time, giving us the latent order book density, $\varphi(x,t) \ \ \forall t \in [0,T]$ and the mid-price path, $p_t \ \ \forall t \in [0,T]$.

3 Numerical Solution for the Latent Order Book

We propose a numerical solution for the latent order book reaction-diffusion Equation 5 by following the Discrete Time Random Walk (DTRW) scheme in Angstmann et al. (2020) and additional methods in Angstmann et al. (2016). We derive the scheme from scratch to adapt the method to include additional terms in the latent order book reaction-diffusion PDE.

3.1 Master Equation

As in Angstmann et al. (2020), we consider a discrete time random walk on a one-dimensional lattice. On this lattice, particles exist and at each time step, they can jump left or right to a neighbouring lattice site. We extend the DTRW method by allowing particles to self jump in addition to the left and right jumps.

Following the notation in Angstmann et al. (2020), we define the probability of a particle being at lattice site i at time step n as $U(i,n)$. This probability mass, $U(i,n)$, is made up of three jump probabilities, a cancellation term and a source term. The probability that a particle on the i^{th} lattice site will jump to the left ($i^{th}+1$ lattice site) on the n^{th} time step is defined as $P_\ell(i,n)$. Similarly, the right jump probability on the i^{th} lattice site on the n^{th} time step is defined as $P_r(i,n)$ and the self jump probability on the i^{th} lattice site on the n^{th} time step is defined as $P(i,n)$.

The cancellation term is defined proportionally to the temporal grid spacing and current lattice probability as:

$$\nu\Delta t U(i,n)$$

The source term is defined as a general function of the lattice site i and time step n that is also proportional to the temporal grid spacing:

$$S(i,n)\Delta t$$

Thus, the master equation is:

$$\begin{aligned} U(i,n+1) &= P_r(i,n)U(i-1,n) + P_\ell(i,n)U(i+1,n) + P(i,n)U(i,n) \\ &\quad -\nu\Delta t U(i,n+1) + S(i,n)\Delta t \end{aligned} \tag{6}$$

In the diffusion limit, Angstmann et al. (2020) show that the master equation becomes an advection-diffusion PDE. Similarly, we will show that in the diffusion limit, as the spatial grid spacing, Δx, and the temporal grid spacing, Δt, approach 0, Equation 6 becomes the latent order book reaction-diffusion partial differential Equation 5. This requires that the following limit exists (Angstmann et al., 2016):

$$D = \lim_{\Delta x, \Delta t \to 0} \frac{\Delta x^2}{2\Delta t} \tag{7}$$

3.2 Taking the Master Equation to the Diffusion Limit

We begin by associating each discrete function from the preceding section with a corresponding continuous function, denoted by a Δ subscript (Angstmann et al., 2020).

These associations are listed below:

$$u_\Delta(i\Delta x, n\Delta t) = U(i,n)$$

$$p_{r,\Delta}(i\Delta x, n\Delta t) = P_r(i,n)$$

$$p_{\ell,\Delta}(i\Delta x, n\Delta t) = P_\ell(i,n)$$

$$p_\Delta(i\Delta x, n\Delta t) = P(i,n)$$

$$s_\Delta(i\Delta x, n\Delta t) = S(i,n)$$

At each point, the jump probabilities sum to one:

$$1 = p_{r,\Delta}(i\Delta x, n\Delta t) + p_\Delta(i\Delta x, n\Delta t) + p_{\ell,\Delta}(i\Delta x, n\Delta t) \tag{8}$$

We can then substitute into Master Equation 6 the continuous functions and $x = i\Delta x$ and $t = n\Delta t$ to obtain:

$$\begin{aligned} u_\Delta(x, t+\Delta t) = \; & p_{r,\Delta}(x-\Delta x, t)u_\Delta(x-\Delta x, t) \\ & + p_{\ell,\Delta}(x+\Delta x, t)u_\Delta(x+\Delta x, t) \\ & + p_\Delta(x,t)u_\Delta(x,t) \\ & - \nu\Delta t u_\Delta(x,t) + s_\Delta(x,t)\Delta t \end{aligned} \tag{9}$$

As is done in Angstmann et al. (2020), we define a force function, $f_\Delta(x,t)$ as:

$$f_\Delta(x,t) = p_{r,\Delta}(x,t) - p_{\ell,\Delta}(x,t) \tag{10}$$

By further manipulation of Equations 10 and 8, we obtain the following two equations:

$$p_{r,\Delta}(x,t) = \frac{1 + f_\Delta(x,t) - p_\Delta(x,t)}{2} \tag{11}$$

and

$$p_{\ell,\Delta}(x,t) = \frac{1 - f_\Delta(x,t) - p_\Delta(x,t)}{2} \tag{12}$$

Substituting Equations 11 and 12 into Equation 9 gives:

$$\begin{aligned} u_\Delta(x, t+\Delta t) = \; & \frac{1}{2}u_\Delta(x-\Delta x, t)\Big(1 + f_\Delta(x-\Delta x, t) - p_\Delta(x-\Delta x, t)\Big) \\ & + \frac{1}{2}u_\Delta(x+\Delta x, t)\Big(1 - f_\Delta(x+\Delta x, t) - p_\Delta(x+\Delta x, t)\Big) \\ & + p_\Delta(x,t)u_\Delta(x,t) - \nu\Delta t u_\Delta(x,t) + s_\Delta(x,t)\Delta t \end{aligned}$$

We then subtract $u_\Delta(x,t)$ from both sides:

$$\begin{aligned} u_\Delta(x, t+\Delta t) - u_\Delta(x,t) = \; & \frac{1}{2}\Big(u_\Delta(x+\Delta x, t) - 2u_\Delta(x,t) + u_\Delta(x-\Delta x, t)\Big) \\ & - \frac{1}{2}\Big(f_\Delta(x+\Delta x, t)u_\Delta(x+\Delta x, t) - f_\Delta(x-\Delta x, t)u_\Delta(x-\Delta x, t)\Big) \\ & - \frac{1}{2}\Big(p_\Delta(x+\Delta x, t)u_\Delta(x+\Delta x, t) + p_\Delta(x-\Delta x, t)u_\Delta(x-\Delta x, t)\Big) \\ & + p_\Delta(x,t)u_\Delta(x,t) - \nu\Delta t u_\Delta(x,t) + s_\Delta(x,t)\Delta t \end{aligned}$$

Furthermore, we divide both sides by Δt and multiply certain terms by $\frac{\Delta x^2}{\Delta x^2}$ to setup derivatives when the diffusion limit is applied:

$$\begin{aligned}\frac{u_\Delta(x,t+\Delta t)-u_\Delta(x,t)}{\Delta t} &= \frac{\Delta x^2}{2\Delta t}\frac{u_\Delta(x+\Delta x,t)-2u_\Delta(x,t)+u_\Delta(x-\Delta x,t)}{\Delta x^2} \\ &-\frac{\Delta x^2}{2\Delta t}\frac{f_\Delta(x+\Delta x,t)u_\Delta(x+\Delta x,t)-f_\Delta(x-\Delta x,t)u_\Delta(x-\Delta x,t)}{\Delta x^2} \\ &+\frac{p_\Delta(x,t)u_\Delta(x,t)-p_\Delta(x+\Delta x,t)u_\Delta(x+\Delta x,t)}{2\Delta t} \\ &+\frac{p_\Delta(x,t)u_\Delta(x,t)-p_\Delta(x-\Delta x,t)u_\Delta(x-\Delta x,t)}{2\Delta t} \\ &-\nu u_\Delta(x,t)+s_\Delta(x,t)\end{aligned} \tag{13}$$

We can now take the diffusion limit ($\Delta x \to 0$ and $\Delta t \to 0$) such that the limit in Equation 7 exists. The limit of the continuum representation of the discrete process is:

$$\lim_{\Delta x,\Delta t\to 0} u_\Delta(x,t) = u(x,t) \tag{14}$$

Thus the diffusion limit of Equation 13 is:

$$\frac{\partial u(x,t)}{\partial t} = D\frac{\partial^2 u(x,t)}{\partial x^2} - 2\beta D\frac{\partial}{\partial x}\Big(F(x,t)u(x,t)\Big) - \nu u(x,t) + s(x,t) \tag{15}$$

where β is related to the "temperature" of the system and,

$$\beta F(x,t) = \lim_{\Delta x\to 0}\frac{f_\Delta(x,t)}{\Delta x} \tag{16}$$

We choose the forms of the jump probabilities so that Equation 15 matches Equation 5. Our choice requires that in the diffusion limit,

$$F(x,t) = \lim_{\Delta x\to 0}\frac{f_\Delta(x,t)}{\beta\Delta x} = \lim_{\Delta x\to 0}\frac{p_{r,\Delta}(x,t)-p_{\ell,\Delta}(x,t)}{\beta\Delta x} \tag{17}$$

By constructing the jump probabilities from Boltzmann weights (Henry et al., 2010), we can satisfy the above equation and prevent an upper bound limitation on Δx. Following Angstmann et al. (2020), we consider a diffusing particle at equilibrium in potential $V(x,t)$, where,

$$F(x,t) = -\frac{\partial V(x,t)}{\partial x} \tag{18}$$

The position of such a particle will follow the Boltzmann distribution such that the probability of the particle being at position x at time t is proportional to $\exp(-\frac{3}{2}\beta V(x,t))$ where β is the temperature of the system. We restrict the system such that the particle can only be at $x+\Delta x$, $x-\Delta x$ or x. Thus, the normalization term is:

$$Z(x,t) = \exp(-\frac{3}{2}\beta V(x+\Delta x,t)) + \exp(-\frac{3}{2}\beta V(x-\Delta x,t)) + \exp(-\frac{3}{2}\beta V(x,t)) \tag{19}$$

The left, right and self jump probabilities, given that the particle started at position x, are the probabilities of the particles being at positions $x-\Delta x$, $x+\Delta x$ and x respectively.

$$p_{\ell,\Delta}(x,t) = \frac{\exp(-\frac{3}{2}\beta V(x-\Delta x,t))}{\exp(-\frac{3}{2}\beta V(x+\Delta x,t)) + \exp(-\frac{3}{2}\beta V(x-\Delta x,t)) + \exp(-\frac{3}{2}\beta V(x,t))} \tag{20}$$

$$p_{r,\Delta}(x,t) = \frac{\exp(-\frac{3}{2}\beta V(x+\Delta x,t))}{\exp(-\frac{3}{2}\beta V(x+\Delta x,t)) + \exp(-\frac{3}{2}\beta V(x-\Delta x,t)) + \exp(-\frac{3}{2}\beta V(x,t))} \tag{21}$$

$$p_{\Delta}(x,t) = \frac{\exp(-\frac{3}{2}\beta V(x,t))}{\exp(-\frac{3}{2}\beta V(x+\Delta x,t)) + \exp(-\frac{3}{2}\beta V(x-\Delta x,t)) + \exp(-\frac{3}{2}\beta V(x,t))} \tag{22}$$

The above functional forms will obey Equation 17 and guarantee that the jump probabilities are bounded by 0 and 1 for all Δx, β and $V(x,t)$. Now we choose $V(x,t)$ to match Equation 15 to Equation 5. Our choice is as follows:

$$V(x,t) = -\frac{V_t x}{2\beta D} \tag{23}$$

where V_t is stochastic variable indexed by t. Substituting Equation 23 into the jump probability equations and factoring out the common terms gives:

$$p_{r,\Delta}(x,t) = \frac{\exp(\frac{3V_t\Delta x}{4D})}{\exp(\frac{3V_t\Delta x}{4D}) + \exp(-\frac{3V_t\Delta x}{4D}) + 1} \tag{24}$$

$$p_{\ell,\Delta}(x,t) = \frac{\exp(-\frac{3V_t\Delta x}{4D})}{\exp(\frac{3V_t\Delta x}{4D}) + \exp(-\frac{3V_t\Delta x}{4D}) + 1} \tag{25}$$

$$p_{\Delta}(x,t) = \frac{1}{\exp(\frac{3V_t\Delta x}{4D}) + \exp(-\frac{3V_t\Delta x}{4D}) + 1} \tag{26}$$

In addition, the choice of $V(x,t)$ gives:

$$F(x,t) = \frac{V_t}{2\beta D} \tag{27}$$

We show in Appendix B that Equation 17 is satisfied with the above choice of $V(x,t)$. The final step is to substitute the value of $F(x,t)$ into Equation 15 to recover a reaction-diffusion partial differential equation of the form:

$$\frac{\partial u(x,t)}{\partial t} = D\frac{\partial^2 u(x,t)}{\partial x^2} - V_t\frac{\partial u(x,t)}{\partial x} - \nu u(x,t) + s(x,t) \tag{28}$$

which is the same as Equation 5 with a symbol change (u for φ). Thus we have formulated a numerical scheme that can iteratively solve the latent order book reaction-diffusion partial differential Equation 5.

3.3 Boundary Conditions

The Discrete Time Random Walk Numerical Scheme has been defined such that we can simulate the dynamics of density in the latent order book over two discrete dimensions, price (x) and time (t). However, the discrete lattice grid on which the particles exist will have boundary points and we must decide on suitable boundary conditions at the highest and lowest price points.

We make the choice of zero-flux Neumann boundary conditions (Mazumder, 2015) with "ghost points" defined outside of the domain. Let x_0 be the lowest price and x_M be the highest price on a lattice of $M+1$ price points. We can calculate the density value at discrete price point $x_i = x_0 + i\Delta x$ and time step $t_n = n\Delta t$ along the time domain as follows:

$$\varphi_{i,n} = \varphi(x_0 + i\Delta x, n\Delta t) \tag{29}$$

where Δx and Δt are the price and temporal grid space values respectively. We also define the two ghost points at x_{-1} and x_{M+1} with density values:

$$\varphi_{-1,n} = \varphi_{0,n} - \Delta x \varphi_x(x,t_n)|_{x=x_0} \tag{30}$$

$$\varphi_{M+1,n} = \varphi_{M,n} + \Delta x \varphi_x(x,t_n)|_{x=x_M} \tag{31}$$

and with the zero-flux Neumann boundary conditions where $\varphi_x(x,t_n)|_{x=x_0} = 0 = \varphi_x(x,t_n)|_{x=x_M}$, the ghost points are equal to their respective neighbouring lattice points.

3.4 Initial Conditions

After choosing suitable boundary conditions, we move on to solving the initial conditions. We define the initial conditions, $\varphi(x,0)$ by approximating the derivatives in Equation 5 with the finite difference approximations below:

$$\varphi_t(x_i,0) = 0 \tag{32}$$

$$\varphi_x(x_i,0) = \frac{\varphi(x_{i+1},0) - \varphi(x_{i-1},0)}{2\Delta x} \tag{33}$$

$$\varphi_{xx}(x_i,0) = \frac{\varphi(x_{i+1},0) - 2\varphi(x_i,0) + \varphi(x_{i-1},0)}{\Delta x^2} \tag{34}$$

Substituting the approximate derivatives into Equation 5 gives us:

$$\begin{aligned} -s(x_i,0) = & -\nu\varphi(x_i,0) + D\frac{\varphi(x_{i+1},0) - 2\varphi(x_i,0) + \varphi(x_{i-1},0)}{\Delta x^2} \\ & -V_0\frac{\varphi(x_{i+1},0) - \varphi(x_{i-1},0)}{2\Delta x} \end{aligned} \tag{35}$$

We can streamline the notation by using $\varphi_i = \varphi(x_i,0)$ and then we group terms:

$$\begin{aligned} -s(x_i,0) = & \left[\frac{D}{\Delta x^2} + \frac{V_0}{2\Delta x}\right]\varphi_{i-1} + \left[-\nu - \frac{2D}{\Delta x^2}\right]\varphi_i + \\ & \left[\frac{D}{\Delta x^2} - \frac{V_0}{2\Delta x}\right]\varphi_{i+1}. \end{aligned} \tag{36}$$

We can re-write this as a $1 \times (M+1)$ matrix equation:

$$\boldsymbol{s} = A\boldsymbol{\varphi} \tag{37}$$

Here the state vector, $\boldsymbol{\varphi}$, has i-th component φ_i and the source vector, $\mathbf{s}$, has i-th component, $-s(x_i,0)$ for $i = 0,1,2,\ldots,M$. The $(M+1)\times(M+1)$ matrix A is a tridiagonal matrix with terms corresponding to $\frac{D}{\Delta x^2} + \frac{V_0}{2\Delta x}$, $-\nu - \frac{2D}{\Delta x^2}$ or $\frac{D}{\Delta x^2} - \frac{V_0}{2\Delta x}$. We also include the ghost points which modifies the first and last rows of A and can be seen in Appendix C. Here a linear solver can then find the state vector $\boldsymbol{\varphi}$ of initial conditions with elements, $\varphi_{i,0}$ for $i = 0,1,2,\ldots,M$. A solution can be visualised as a plot against with the $\boldsymbol{x}$ vector along the x-axis and seen in Figure 2.

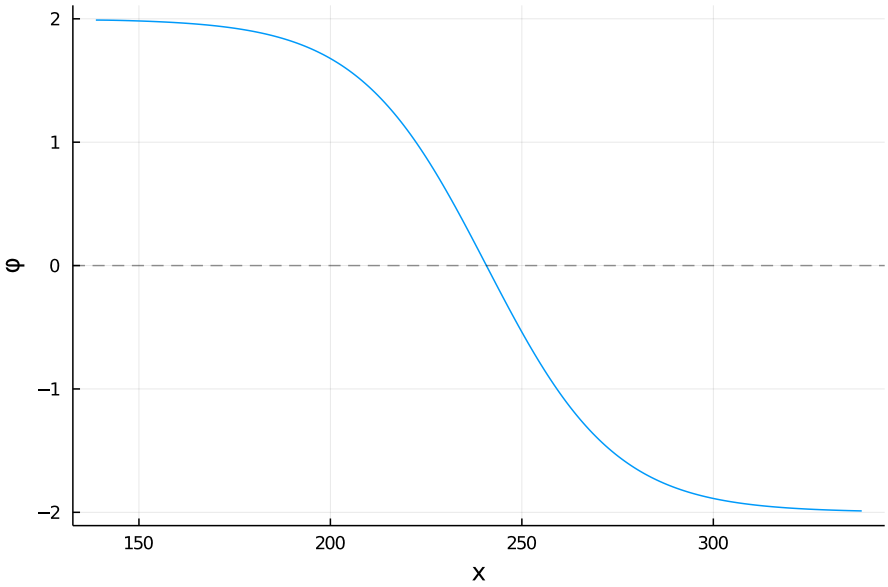


Figure 2: Plot of a solution for the initial conditions obtained by using a linear solver to find the state vector φ given a set of standard parameters. The parameters used were: $M = 800$, $P_0 = 238.75$, $L = 200$, $\lambda = 1.0$, $D = 0.5$, $\Delta x = 0.25$, $\sigma = 1.5$, $\mu = 0.03$, $\nu = 0.5$, $V_0 = 1.0$

3.5 Discrete Time Random Walk Numerical Scheme Summary

In Section 3 we construct a numerical solution for solving a form of reaction-diffusion partial differential equations based on the Discrete Time Random Walk method in Angstmann et al. (2016) and further work in Angstmann et al. (2020). Given a reaction-diffusion partial differential equation of the form:

$$\frac{\partial u(x,t)}{\partial t} = D\frac{\partial^2 u(x,t)}{\partial x^2} - V_t\frac{\partial u(x,t)}{\partial x} - \nu u(x,t) + s(x,t)$$

we can numerically solve for $u(x,t)$ on a two dimensional discrete lattice $(x,\, t)$ of shape $([0, M] \times [0, T])$, spaced according to Δx and Δt respectively. The numerical solution is initalised by solving the initial conditions set out in Equation 37.

The numerical scheme dictates how to iteratively solve for time step $t = t_{n+1}$ given the current time step is t_n until $t = T$. At each time step t_n, density at the subsequent time step t_{n+1} is calculated using the Master Equation 6, the jump probability Equations 24, 25 and 26 and the boundary conditions in Section 3.3.

We can visualise the density output in a surface plot on the price and time axes. Using default initial parameters, Figure 3 displays a sample of density values. As expected, the positive density values representing latent bid orders are towards the lower half of the price domain and negative density values representing latent ask orders are towards the upper half of the price domain. We can observe a smooth surface along the price domain whereas the time domain has more angular changes due to the sampling[1] performed to create this figure.

[1] points were sampled every 3000 time steps

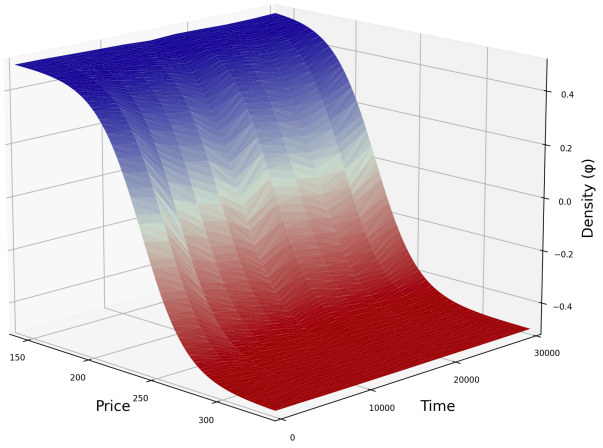


Figure 3: Latent order book density values, sampled every 3000 time steps, from the DTRW numerical scheme used to solve the latent order book reaction-diffusion partial differential equation. The DTRW scheme used the following initial parameters: $M = 800$, $T = 2299$, $P_0 = 238.75$, $L = 200$, $\lambda = 1.0$, $D = 0.5$, $\Delta x = 0.25$, $\Delta t = 0.0625$, $\sigma = 1.5$, $\mu = 0.03$, $\nu = 0.5$ and seed=5487.

4 The Latent Order Book Model

The Latent Order Book (LOB) model is the representation of the latent order book as reaction-diffusion partial differential Equation 5 and its numerical solution from Section 3. The LOB model aims to provide a realistic abstraction of a trading order book in the form of latent density of orders and an observed mid-price path, sampled at the end of each time period. Before we can begin model experimentation, there are a few areas that need to be explained.

4.1 Mid-Price Paths

The raw mid-prices, $\tilde{p}_t$, are calculated at each time step, t, by linear interpolation between the largest x_i value where $\varphi(x,t) > 0$ and the smallest x_i value where $\varphi(x,t) < 0$. This approximates the solution of x in $\varphi(x,t) = 0$ and corresponds to the mid-price which is the price halfway between the largest bid and smallest ask.

Each time step's corresponding mid-price $\tilde{p}_t$ is not necessarily the observed mid-price, p_t. Recall that mid-price path data is sampled periodically (every minute, every 5 minutes, every hour, etc) and thus the observed mid-price path should follow the same logic. Therefore, the observed mid-price, p_t, from the latent order book model, corresponds to the mid-price, $\tilde{p}_t$, at or before the closest integer value of $t \in [0, T]$. The observed mid-price path $\boldsymbol{p}$ from the LOB model is thus a $T + 1$ size vector of mid-prices, $p_t : \ t \in [0, T]$.

4.2 Stability

A key concern is numerical stability and the consistency required to ensure that the granularity of the lattice can reasonably capture the velocity of density propagation

so that the distance between the approximation and the numerical scheme is bounded for all n time steps subsequent to the application of initial and boundary conditions (Angstmann et al., 2020):

$$\sum_{i=1}^{M} |\varphi_{i,n} - \varphi(x_0 + i\Delta x, n\Delta t)| \leq C, \quad C \in \mathbb{R}^+ \tag{38}$$

This requires that the maximum probability of left and right jumps corresponds to the maximum velocity allowed by the lattice. The presence of viscosity ensures the maximum density necessarily occurs at $t = 0$ when initial conditions are computed (Angstmann et al., 2020). This can be used to ensure that over short-time periods, the approach remains stable. However, to ensure global consistency, universality and stability, we re-solve the initial conditions over a sequence of exponentially distributed waiting times $\tau \sim \exp(\frac{1}{\alpha})$ where α is the scale parameter.

With the introduction of re-solving the initial conditions, we create two types of latent order book models. The first model, which we name the latent order book model, solves the initial conditions once at the first time step. The second model, which we name the Sequential Latent Order Book (SLOB) model, solves the initial conditions multiple times according to a sequence of exponentially distributed waiting times and thus has an extra parameter, α, which controls the waiting time distribution.

4.3 Timescales for the Sequential LOB Implementation

When solving the initial conditions for the SLOB model, the waiting time is sampled from the Exponential distribution with scale parameter α. This waiting time represents the amount of time until the initial conditions are solved again. This splits the total time period into k sub-periods, each of time $\tau_k \sim \exp(\frac{1}{\alpha})$. The number of time steps in each sub-period, n_k, is calculated as:

$$n_k = \left\lfloor \frac{\tau_k}{\Delta t} \right\rfloor$$

Within this smaller sub-period, we essentially have a LOB model. Thus, the sequence of LOB models gives the SLOB model its name.

4.4 LOB and SLOB Model Overview

4.4.1 Inputs

The LOB and SLOB models take as input, configuration arguments and free-parameters. The configuration arguments for both models are detailed below:

1. T: An integer representing the total number of time periods in addition to the initial time period $t = 0$ or t_0.
2. M: An integer representing the number of discrete price points in addition to the initial mid-price point. The size of the price dimension in the discretised price lattice is $M + 1$.
3. p_0: The initial mid-price. This value is used as a starting point for the observed mid-price path and as a centre of the price lattice.
4. L: The length of the price dimension in the lattice.

5. Δx: The price grid spacing which is determined by:

$$\Delta x = \frac{L}{M}$$

6. Δt: The temporal grid spacing which is determined by configuration parameters and a free-parameter:

$$\Delta t = \frac{\Delta x^2}{2D}$$

As this is value is set indirectly, care should be taken when choosing M, L and D.

The free-parameters for both models are detailed below (α only applies to the SLOB model and if $\alpha = 0$, the SLOB model is a LOB model):

1. D: The diffusion constant which also determines the temporal grid spacing, Δt, due to Equation 7.
2. σ: The standard deviation parameter for the stochastic variable $V_t \sim N(0, \sigma)$.
3. ν: The cancellation rate which removes a proportion of density from the latent order book.
4. λ: The Source Term parameter which controls the total amount of density arriving. Larger values of λ result in larger amounts of latent density.
5. μ: The source term parameter which controls the slope of the density arrival around the mid-price. Larger values of μ result in steeper funnels around the mid-price, whereas smaller μ values create a more gradual decline in density magnitude around the mid-price.
6. α: The waiting time scale parameter. As the waiting time distribution is an Exponential distribution, the expectation is α, thus we can think of α as the average waiting time between latent order book initial condition solves.

4.4.2 Outputs

The LOB and SLOB models output:

1. An observed mid-price path, $\boldsymbol{p}$, of length $T + 1$
2. A raw mid-price path, $\tilde{\boldsymbol{p}}$, of length $\frac{T}{\Delta t} + 1$
3. The latent order book density matrix of shape $[M + 1, \frac{T}{\Delta t} + 1]$
4. Three vectors of length $\frac{T}{\Delta t}$ corresponding to the left, right and self jump probability values

The main focus in this dissertation will be on the observed mid-price path outputs from each model.

4.5 Initial Results From SLOB and LOB Model

The SLOB and LOB models are programmed in the language Julia (Bezanson et al., 2017) and the code to reproduce the results in this dissertation is available in the GitHub Repository (Gant and Gebbie, 2020b) and additional repositories are listed in Appendix G. To visualise a few initial mid-price paths that each model can generate, we intialise each model with a set of default configuration arguments and uncalibrated free-parameters which were found to produce suitable mid-price paths. The choices

are also informed by a real mid-price path dataset (Gebbie and Platt, 2019). The tables below display these configuration arguments and free-parameter values:

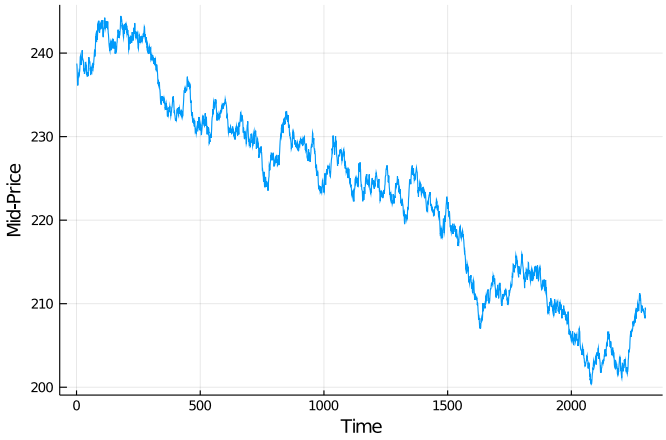
Table 1: The configuration arguments used in the LOB and SLOB models in this dissertation. These values are fixed and not a part of the calibration process, but were chosen based on initial model testing, computational time and observed real mid-price path data.

Configuration Argument	**Description**	**Value**
T	An integer representing the total number of time periods in addition to the initial time period $t = 0$ or t_0.	2229
M	An integer representing the number of discrete price points in addition to the initial mid-price point. The size of the price dimension in the discretised price lattice is $M + 1$.	400
p_0	The initial mid-price. This value is used as a starting point for the observed mid-price path and as a centre of the price lattice.	238.75
L	The length of the price dimension in the lattice.	200
Δx	The price grid spacing.	0.5
Δt	The temporal grid spacing which is determined by configuration parameters and a free-parameter.	0.125

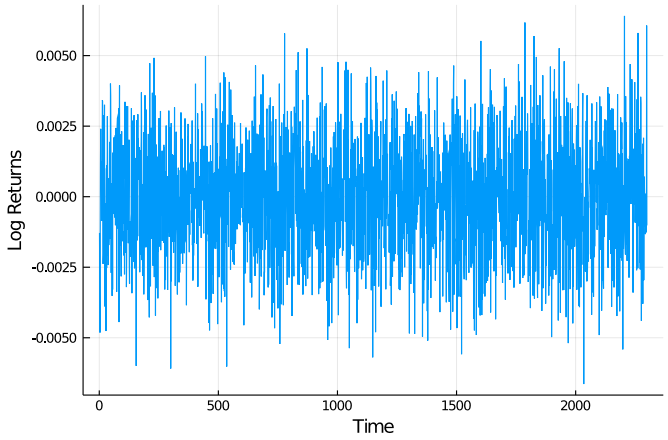
Table 2: A set of free-parameter values which are labelled as the default free-parameters to provide output from the LOB and SLOB models without needing any calibration. These values were chosen based on initial model testing and simplicity.

Free-Parameter	**Description**	**Value**
D	The diffusion constant which also determines the temporal grid spacing, Δt, due to Equation 7.	1.0
σ	The standard deviation parameter for the stochastic variable $V_t \sim N(0, \sigma)$.	1.5
ν	The cancellation rate which removes a proportion of density from the latent order book.	0.5
λ	The Source Term parameter which controls the total amount of density arriving. Larger values of λ result in larger amounts of latent density.	1.0
μ	The source term parameter which controls the slope of the density arrival around the mid-price. Larger values of μ result in steeper funnels around the mid-price, whereas smaller μ values create a more gradual decline in density magnitude around the mid-price.	1.0
α	The waiting time scale parameter. As the waiting time distribution is an Exponential distribution, the expectation is α, thus we can think of α as the average waiting time between latent order book initial condition solves.	100.0

With a set seed, we generate a mid-price path from each model and display the mid-price path and the respective log returns below:

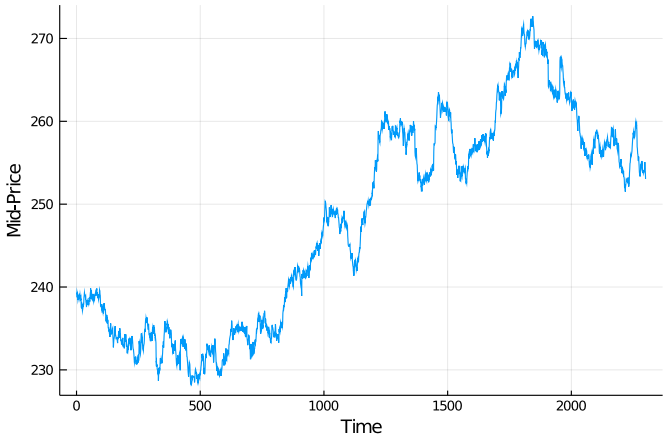


(a) LOB mid-price path generated with seed=7136 and configuration arguments and free-parameters in Table 1 and 2.

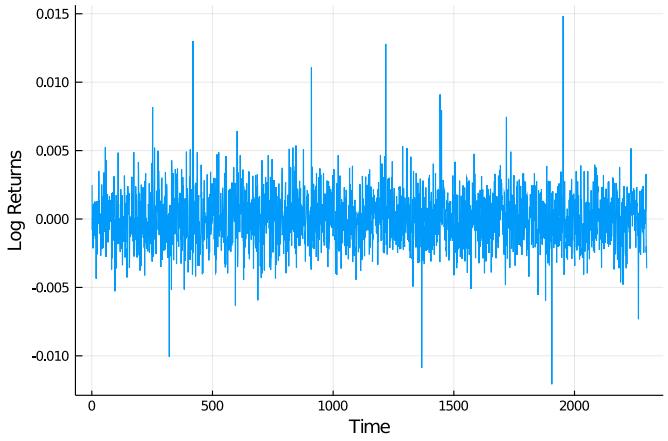


(b) LOB log returns calculated from a LOB mid-price path generated with seed=7136 and configuration arguments and free-parameters in Table 1 and 2.

Figure 4: A mid-price path and the corresponding log returns from one run of the LOB model. A GIF of this mid-price path and latent order book density evolution can be seen here (Gant and Gebbie, 2020b)



(a) SLOB mid-price path generated with seed=19822 and configuration arguments and free-parameters in Table 1 and 2.



(b) SLOB log returns calculated from a SLOB mid-price path generated with seed=19822 and configuration arguments and free-parameters in Table 1 and 2.

Figure 5: A mid-price path and the corresponding log returns from one run of the SLOB model. A GIF of this mid-price path and latent order book density evolution can be seen here (Gant and Gebbie, 2020b)

At a first glance, both models produce mid-price paths that seem realistic. The non-smooth, ragged time series move up and down in an irregular and unpredictable manner similar to observed mid-price paths from stock exchanges. However, the log return plot from the LOB model does not seem to exhibit the fat-tail distributions present in the real world, although the SLOB plot does seem to be more irregular with several extreme log returns observed. At this stage, we have two models which produce somewhat realistic mid-price paths, but we need to investigate the models and the effect of the free-parameters on the mid-price paths.

5 Calibration

5.1 Calibration Motivation

As a part of the investigation into the latent order book and its models, we are interested in whether we can fine-tune or calibrate the model by adjusting its free-parameters to obtain mid-price paths that have certain features. In doing so, a successful calibration process would confirm that the model's free-parameters are meaningful and provide us with a model which can do the following:

1. Generate desired features or stylised facts,
2. Provide a causal relationship between inputs and outputs,
3. Generate mid-price paths which are sufficiently similar to another mid-price path, thus recovering the source model that generated the other mid-price path. This also provides a large amount of new mid-price path data which is similar to real mid-price path data.

The first step in the calibration process is to test whether the LOB and SLOB model free-parameters affect emergent dynamics in a way that can be captured in the output. Due to the complexity, randomness and non-linearity present in the models, we cannot confirm analytically whether the free-parameters are meaningful. Therefore, we calibrate the model against synthetic data generated by the model using known free-parameters to confirm that the calibration technique is working and the free-parameters are recoverable. We follow the calibration assessment methodology introduced in Platt (2020) to understand and compare how well the proposed calibration techniques perform.

5.2 Likelihood-Free Calibration Techniques

The LOB and SLOB models do not have a trivial likelihood function and thus we consider two alternatives to maximum likelihood calibration, namely Approximate Bayesian Computation (ABC) (Marin et al., 2012; Forneron and Ng, 2018; Prangle et al., 2017) and black-box optimisation (Gao and Han, 2012) with summary statistics (Franke, 2009; Platt and Gebbie, 2018) and weighted distance functions (Prangle et al., 2017).

The approximate Bayesian computation and the black-box optimisation approaches that we consider both share the same summary statistics and distance function components, thus providing a like-for-like comparison between ABC sampling and black-box optimisation. We begin by first describing approximate Bayesian computation, then move on to the generic calibration framework and finally detail the specific components.

5.3 Approximate Bayesian Computation

Approximate Bayesian computation is a likelihood-free approach to model inference when the true likelihood function is impossible or infeasible to calculate (Prangle et al., 2017). This inability to know the likelihood function is common in complex models

(Marjoram et al., 2003) and thus many studies have been done to address this problem. Naturally, the thinking behind Bayes theorem applies well here, as we have some prior knowledge about the models' parameters and would like to know more about the posterior distribution of these parameters given some observed data. However, we do not know the likelihood function of the data given the parameters and thus cannot directly apply Bayes Theorem. This lack of likelihood function also means that standard Acceptance-Rejection sampling cannot be done.

To solve this likelihood issue, we consider the idea that random data generated by the same model with the same parameters will be similar to each other. Therefore, we can simulate new data, D', with parameters, θ', and compute the distance between D' and the observed data, D, as $\rho(D, D') \geq 0$. This distance function, $\rho(\cdot, \cdot)$, should quantify how similar two observations are, and ideally, if both D and D' were simulated using the same model and parameters θ, then $\rho(D, D') \approx 0$ (Marjoram et al., 2003).

When comparing two observations, the dimensionality of the observation can play a significant part in whether the distance computation is feasible or even accurate. Thus, we further consider comparing sets of sufficient summary statistics calculated from each observation (Marjoram et al., 2003). This compresses relevant information into a summarised and lower-dimensional version, making comparisons more general and computationally tractable (Marjoram et al., 2003).

5.3.1 ABC Rejection Sampling

ABC rejection sampling is the simplest ABC technique and is very similar to the classical Monte Carlo technique, rejection sampling. The difference with ABC rejection sampling is that the decision to accept or reject a sampled value is based on a distance function, $\rho(\cdot, \cdot)$, and a threshold, ϵ as opposed to a fraction of two probability density functions. This distance function is either the distance between actual observations, $\boldsymbol{D}$ vs. $\boldsymbol{D'}$, or the distance between sets of summary statistics, $\boldsymbol{S}$ vs. $\boldsymbol{S'}$. In our case, the observations are high dimensional stochastic variables and thus distance calculations between the raw observations are impractical (Marjoram et al., 2003).

The choice of summary statistics should ideally be sufficient for $\boldsymbol{\theta}$ (Marjoram et al., 2003) and can be informed by prior work in time series modelling and agent-based model calibration (Platt and Gebbie, 2018; Donier et al., 2015). Here we can describe the density function from which we are sampling as:

$$f(\boldsymbol{\theta} | \rho(\boldsymbol{S}, \boldsymbol{S'}) \leq \epsilon)$$

where $\boldsymbol{S}$ is the set of observed summary statistics calculated from the observed data $\boldsymbol{D}$ and $\boldsymbol{S'}$ is the set of simulated summary statistics calculated from the simulated data $\boldsymbol{D'}$ which was generated by model M with parameters $\boldsymbol{\theta'}$. Therefore, the ABC rejection sampling algorithm is:

Algorithm 1 Approximate Bayesian Computation Rejection Sampling.

Require:

(a) Simulator model, $M(\cdot)$, which returns data, $\boldsymbol{D}$, given input parameters, $\boldsymbol{\theta}$

(b) Prior probability distribution, $\pi(\boldsymbol{\theta})$

(c) Observed stochastic variable, $\boldsymbol{D}_{obs}$, of size T

(d) Summarisation function to map data to a set of summary statistics, $\boldsymbol{S}$

(e) Distance function that quantifies similarity/dissimilarity between two sets of summary statistics, $\rho(\boldsymbol{S}, \boldsymbol{S}')$

(f) The total number of simulations N

(g) Acceptance threshold, $\epsilon \geq 0$ used with the distance function to decide whether the sample is accepted or rejected.

for $i = 1$ to N **do**
  Sample $\boldsymbol{\theta}'$ from $\pi(\boldsymbol{\theta})$
  Simulate $\boldsymbol{D}'$ from $M(\boldsymbol{\theta}')$
  Calculate $\boldsymbol{S}'$ from $\boldsymbol{D}'$
  Calculate $\rho(\boldsymbol{S}_{obs}, \boldsymbol{S}')$
  Accept $\boldsymbol{\theta}'$ if $\rho(\boldsymbol{S}_{obs}, \boldsymbol{S}') \leq \epsilon$
end for i$^{\text{th}}$ sample
return $\{\boldsymbol{\theta}'_i | \rho(\boldsymbol{S}, \boldsymbol{S}'_i) \leq \epsilon\}$

If the distance function is suitable and the acceptance threshold is sufficiently close to 0, this straightforward approach allows us to sample from a close approximation to the target posterior distribution. We can tune ϵ to fit the desired accuracy and efficiency of the sampling, with higher ϵ values resulting in more acceptances but at the cost of an approximate posterior that is further away from the target posterior.

An important concern is the computational time taken to accept N samples. As each sample is independently and identically distributed, a low acceptance rate will mean many simulations are thrown away and future sampling efficiency does not improve as more acceptances are made. To address some of the shortcomings of this simple approach, we consider an iterative ABC method with an adaptive distance function proposed in Prangle et al. (2017) called ABC-Population Monte Carlo (ABC-PMC)

5.3.2 ABC-Population Monte Carlo

Iterative extensions to the simple ABC rejection sampling algorithm have been proposed by Sisson et al. (2007), Beaumont et al. (2009) and Toni et al. (2009) which are referenced in creation of a new iterative ABC method with an adaptive distance function. Specifically, we investigate and use Algorithm 4 ABC-PMC with adaptive h_t and $d_t(\cdot, \cdot)$ (ABC-PMC-4) from Prangle et al. (2017). The notation in Prangle et al. (2017) is slightly different to the ABC rejection sampling algorithm (Marjoram et al., 2003) and thus we present the ABC-PMC algorithm using the already defined symbols and nomenclature of this dissertation.

The iterative approach to ABC introduces a number of new concepts. Intuitively, the iterative ABC method is similar to a sequence of ABC rejection algorithms, but

with each iteration's prior density being dependent on the previous iteration's accepted samples and their respective weights. The weights allow the algorithm to place more importance on samples that are closer to the observed data and thus should provide better, more efficient sampling in future iterations. Combine this with a sequence of acceptance thresholds, $(\epsilon_1, \epsilon_2, ..., \epsilon_t)$, and the resulting samples from the final iteration should be better than the accepted samples from the ABC rejection method.

The ABC-PMC-4 algorithm makes use of an importance density function, $q_t(\boldsymbol{\theta})$, to more efficiently sample parameter vectors. This is achieved through the accepted parameter weights, w_i^t and a kernel, K_t, which perturbs the parameter vector. The importance density function is defined as:

$$q_t(\boldsymbol{\theta}) = \begin{cases} \pi(\boldsymbol{\theta}) & \text{if } t = 1, \text{ or } t = 2 \text{ and } \epsilon_1 = \inf \\ \sum_{i=1}^{N} w_i^{t-1} K_t(\boldsymbol{\theta} | \boldsymbol{\theta}_i^{t-1}) / \sum_{i=1}^{N} w_i^{t-1} & \text{otherwise} \end{cases} \tag{39}$$

In the first iteration, $t = 1$, parameters are sampled from the prior density. In the subsequent iterations, assuming that the acceptance threshold is not infinity, parameters are sampled from the previous iteration's weighted population of accepted parameters and then perturbed by the kernel, K_t. Following Beaumont et al. (2009) and Prangle et al. (2017), we choose a Normal density function, $\phi(\boldsymbol{\theta'}, 2\Sigma_{t-1})$, as the kernel, $K_t(\boldsymbol{\theta}|\boldsymbol{\theta'})$, where Σ_{t-1} is the empirical covariance matrix calculated from the previous iteration's accepted parameters, $\{\boldsymbol{\theta}_i^{t-1} : 1 \leq i \leq N\}$, and their respective sampling weights, $\{w_i^{t-1} : 1 \leq i \leq N\}$.

The adaptive distance function, $\rho_t(\cdot, \cdot)$, is a weighted Euclidean function whose weights are updated after each iteration. The weights can be chosen in a number of ways, such as the inverse of the empirical standard deviation or median absolute deviation of the summary statistics from both accepted and rejected simulated data in each iteration.

The acceptance threshold, ϵ_t, at each iteration is calculated as the α quantile of the distances from the previous iteration's accepted parameters. In the first iteration, ϵ_1 is initialised to infinity and all sampled parameters are accepted. From this initial population, we can calculate the distances and then take the α quantile as the next acceptance threshold. We formally describe the ABC-PMC-4 algorithm below:

Algorithm 2 Iterative ABC-PMC with adaptive distance function.

Require:

(a) Simulator model, $M(\cdot)$, which returns data, $\boldsymbol{D}$, given input parameters, $\boldsymbol{\theta}$.
(b) Prior probability distribution, $\pi(\boldsymbol{\theta})$.
(c) Observed stochastic variable, $\boldsymbol{D}_{obs}$, of size T.
(d) Summarisation function to map data to a set of summary statistics, $\boldsymbol{S}$.
(e) Adaptive distance function that quantifies similarity/dissimilarity between two sets of summary statistics, $\rho(\boldsymbol{S}, \boldsymbol{S'})$ and updates weights after each iteration.
(f) The total number of iterations, N_{iter}.
(g) The number of acceptances required per iteration, N_{accept}.
(h) The acceptance threshold quantile parameter, α.

for $t = 1$ to N_{iter} **do**
  for $i = 1$ to N_{accept} **do**
    repeat
      Sample $\boldsymbol{\theta'}$ from $q_t(\boldsymbol{\theta})$
      Simulate $\boldsymbol{D'}$ from $M(\boldsymbol{\theta'})$
      Calculate $\boldsymbol{S'}$ from $\boldsymbol{D'}$
      Calculate $\rho_t(\boldsymbol{S}, \boldsymbol{S'})$
      Accept $\boldsymbol{\theta'}$ if $\rho_t(\boldsymbol{S}_{obs}, \boldsymbol{S'}) \leq \epsilon_t$
    until acceptance
  end for i$^{\text{th}}$ acceptance
  Update distance function for next iteration ρ_{t+1}
  Calculate sample weights, $w_i^t = \pi(\boldsymbol{\theta}_i)/q_t(\boldsymbol{\theta}_i)$
  Set ϵ_{t+1} as the α quantile from distance values for all acceptances
end for t$^{\text{th}}$ iteration
return $\{\boldsymbol{\theta}'_i | \rho_t(\boldsymbol{S}, \boldsymbol{S}'_i) \leq \epsilon_t\}$

As each iteration has a different distance function and acceptance threshold, it is possible for a parameter to be accepted in iteration t but not in iteration $t-1$. To avoid this, we require that parameters only be accepted if their distance in the current iteration and every previous iteration is less than or equal to the acceptance threshold for that iteration.

5.4 Calibration Framework

The calibration framework for the LOB and SLOB models can be split into four sections:

1. **Simulation function** - The LOB or SLOB model which takes in a set of free-parameters and outputs a mid-price vector.
2. **Summarisation function** - A function that compresses a vector of values into a set of summary statistics to capture certain features in the underlying data.
3. **Distance function** - A function that quantifies the similarity or dissimilarity between the observed or known set of summary statistics and the set of simulated summary statistics.

4. **Sampling algorithm** - This algorithm determines what new free-parameters should be sampled or selected based on the value from the distance function and the related free-parameters.

Both ABC rejection sampling and ABC-Population Monte Carlo fit well into the above framework. Furthermore, the ABC posterior distributions of the free-parameters are far more informative and useful than point estimates from standard optimisation algorithms. To further justify the use of ABC, we compare its performance to a black-box optimisation approach in the form of the above framework.

5.4.1 Simulation Function

We abstract the LOB and SLOB models as generic simulation functions which take in a set of free-parameters and output a vector of values. This generic approach then allows us to verify the performance of the calibration approach on simpler toy models which are investigated in later sections. More formally, we define the simulator function as follows:

$$f(\boldsymbol{\theta}|\boldsymbol{\Pi}) = \boldsymbol{x} \tag{40}$$

where $f()$ is the simulation function, $\boldsymbol{\theta}$ is the set of free-parameters, $\boldsymbol{\Pi}$ is the set of fixed configuration parameters and $\boldsymbol{x}$ is the output vector of values. Practically, we would also input a seed into the simulation function to ensure that the randomness in the function is replicable.

5.4.2 Summarisation Function

Comparing a vector of observations can be difficult when similarity does not necessarily imply that values are element-wise equal. In the case of mid-price paths, two observed mid-price paths could be generated from the same source but due to randomness, not be equal at each time step. However, the observations would share similar features when looking at summary statistics like average log returns, log return volatility or log return auto-correlation.

Therefore, we require a generic approach that summarises the relevant information in the vector in a way that is conducive for like-for-like comparisons. The approach, although generic, should be informed by the input data, which are financial mid-price time series data. The approach is essentially a mapping of a vector of mid-prices into a set of summary statistics. Our choice of summary statistics should aim to be sufficient but this is rarely the case in complex cases (Marin et al., 2012). With this in mind, the choice of summary statistics is largely based on the use case and combine the summary statistics used in Franke (2009), Winker et al. (2007) and Platt (2020). Each summary statistic, s_i, is detailed below:

1. **Mean log returns** (s_1):
   The sample mean of the log returns ($\bar{r}$) measures the average magnitude of the Log Returns (r_t).
   $$\bar{r} = \frac{1}{T}\sum_{t=1}^{T} r_t, \qquad r_t = \log(\frac{p_t}{p_{t-1}})$$
2. **Log returns standard deviation** (s_2):
   The sample standard deviation of the log returns (s) measures the average mag-

nitude deviation of the log returns from the sample mean.

$$s = \sqrt{\frac{\sum_{t=1}^{T}(r_t - \bar{r})^2}{T-1}}$$

3. **Excess kurtosis of log returns** (s_3):
The excess kurtosis of log returns is a measure of the size of the tails of the log return empirical distribution in relation to the Normal Distribution.

$$\text{kurtosis} = \frac{\sum_{t=1}^{T}(r_t - \bar{r})^4}{T \cdot s^4} - 3$$

4. **Kolmogorov-Smirnov two sample test statistic** (s_4):
The Kolmogorov-Smirnov two sample test quantifies whether the underlying probability distributions of two samples differ.

$$\text{KS-Test} = \sup_x |F_A(x) - F_B(x)|$$

where $F_A(x)$ and $F_B(x)$ are the empirical cumulative distribution functions of the two log return samples, A and B.

5. **Generalised Hurst exponent** (s_5):
The generalised Hurst exponent, H_q, is a measure of the fractal nature of a time series (Hurst, 1951; Preis et al., 2009), in this case, the log returns of a mid-price path. When $H_q = 0.5$, the time series is a random walk. When $H_q < 0.5$, the time series is mean-reverting and when $H_q > 0.5$, the time series is trend following or super-diffusive.

6. **Autocorrelation function at lags $q = [1, 5]$ of log returns, squared log returns and absolute log returns** (s_6 **to** s_{11}):
The autocorrelation function measures the correlation of a time series with a delayed copy of itself.

$$\text{ACF}_q = \frac{1}{s}\sum_{t=1}^{T-q}(r_t - \bar{r})(r_{t+q} - \bar{r})$$

7. **Partial autocorrelation function at lags $q = [1, 5]$ of log returns, Squared log returns and absolute log returns** (s_{12} **to** s_{17}):
The Partial Autocorrelation Function (PACF) measures the correlation of a time series with a delayed copy of itself and with the relationships at prior time steps removed.

With the chosen summary statistics, we can formally define the summarisation function as a mapping of the mid-price vector, $\boldsymbol{p}$, to a seventeen element vector of summary statistics, $\boldsymbol{s} = (s_1, s_2, ..., s_{17})$, as defined in the above list.

5.4.3 Distance Function

Quantifying the similarity of two summary statistic vectors can be as simple as calculating the sum total difference between each pair of elements. However, this then assumes that the magnitude of differences in each of the summary statistics is equivalent. This is not correct as the individual summary statistics vary on differing scales. Thus, we consider the use of a weighted distance function, which should standardise the element-wise distances for each summary statistic such that the contributions are

equivalent.

We investigate two different weighted distance functions, namely the Block Bootstrap Weight Matrix (BBWM) distance function (Winker et al., 2007; Platt and Gebbie, 2018) and the Median Absolute Deviation Weighted Euclidean (MADWE) distance function (Csilléry et al., 2012; Prangle et al., 2017).

Formally, the generic weighted distance function, $\rho(\boldsymbol{x}, \boldsymbol{y})$ can be defined as:

$$\rho(\boldsymbol{x}, \boldsymbol{y}) = \left[(\boldsymbol{x} - \boldsymbol{y})^T \boldsymbol{W} (\boldsymbol{x} - \boldsymbol{y}) \right]^{1/L} \tag{41}$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are the vectors of interest, $\boldsymbol{W}$ is the weight matrix and $L = 1$ or $L = 2$ is a parameter that controls whether the square root is applied. The $\boldsymbol{x}$ vector will represent the observed summary statistics vector, $\boldsymbol{s}^{obs}$, and the $\boldsymbol{y}$ vector will represent each simulated summary statistics vector, $\boldsymbol{s}^{sim}$.

In the BBWM distance function, $L = 1$ and the weight matrix, $\boldsymbol{W}_{BB}$, is a 17×17 positive definite matrix calculated as the inverse estimate of the variance-covariance matrix from a bootstrapped sample of the observed summary statistics, $\boldsymbol{s}^{obs}$. We follow the methodology in Winker et al. (2007) and use 10 000 bootstrap samples but decrease the bootstrap window size to 100. The weight matrix estimator is then:

$$\boldsymbol{W}_{BB} = \hat{\boldsymbol{W}}(\boldsymbol{s}^{obs}) = \hat{\Sigma}_{BB}^{-1} \tag{42}$$

In the MADWE distance function, $L = 2$ and the weight matrix is a diagonal matrix with elements, $w_i = 1/\sigma_i$ where σ_i is the empirical standard deviation of the i^{th} summary statistic (Prangle et al., 2017). To calculate a robust estimate of the empirical standard deviation of a summary statistic, we follow Csilléry et al. (2012) and use the Median Absolute Deviation (MAD) due to its robustness to large outliers. Unlike the BBWM weight matrix, a large number of simulated summary statistics vectors are generated to calculate the MAD estimate of σ_i. The MAD estimator for summary statistic i given m simulations, $[s_1^{(i)}, s_2^{(i)}, ..., s_m^{(i)}]$, is:

$$\sigma_i^{MAD} = \text{Median}(\boldsymbol{s}_{\boldsymbol{\Delta}}^{(\boldsymbol{i})}) \tag{43}$$

where

$$\boldsymbol{s}_{\boldsymbol{\Delta}}^{(\boldsymbol{i})} = |s_j^{(i)} - \tilde{s}^{(i)}| \qquad \forall j \in [1, 2, 3, ..., m]$$

and $\tilde{s}^{(i)}$ is the median of the i^{th} summary statistic over m simulations. Thus the weight matrix estimator is:

$$\begin{aligned} \boldsymbol{W}_{MAD} &= \hat{\boldsymbol{W}}(\boldsymbol{\sigma}^{MAD}) \\ &= \begin{bmatrix} 1/\sigma_1^{MAD} & 0 & \dots & 0 \\ 0 & 1/\sigma_2^{MAD} & & \vdots \\ \vdots & & \ddots & \\ 0 & \dots & & 1/\sigma_{17}^{MAD} \end{bmatrix} \end{aligned}$$

5.4.4 Sampling Algorithm

The sampling step in approximate Bayesian computation can range from the simple rejection sampling (Prangle et al., 2017) to Markov Chain Monte Carlo (MCMC)

sampling (Marjoram et al., 2003) and then iterative or sequential methods such as Population Monte Carlo (PMC) (Sisson et al., 2007; Prangle et al., 2017). In this dissertation, we consider two ABC approaches, the simple rejection ABC sampling and the ABC-PMC Algorithm 4 (Prangle et al., 2017). The details of each algorithm are explained in Section 5.3.1 and 5.3.2.

For the Black-box approach, the sampling algorithm is essentially the optimisation algorithm. The algorithm takes in the set of free-parameters and the distance value and selects a new set of free-parameters that should return a lower distance value. We choose to make use of the Nelder-Mead algorithm for the Black-box approach (Gao and Han, 2012) due to its use in existing literature (Platt, 2020) and not requiring any derivatives of the objective function (distance function) with respect to the free-parameters.

5.5 Calibration Techniques Summary

The different combinations of the distance functions and sampling algorithms provide six different calibration techniques, which we summarise and label below:

1. **ABC rejection BBWM**: approximate Bayesian computation rejection sampling with the block bootstrap weight matrix distance function
2. **ABC rejection MADWE**: approximate Bayesian computation rejection sampling with the median absolute deviation weighted euclidean distance function
3. **ABC-PMC BBWM**: approximate Bayesian computation population Monte Carlo with the block bootstrap weight matrix distance function
4. **ABC-PMC MADWE**: approximate Bayesian computation population Monte Carlo with the median absolute deviation weighted euclidean distance function
5. **Nelder-Mead BBWM**: Nelder-Mead minimisation with the block bootstrap weight matrix distance function
6. **Nelder-Mead MADWE**: Nelder-Mead minimisation with the median absolute deviation weighted euclidean distance function

6 Synthetic Calibration Results

6.1 ARMA Calibration

To validate and benchmark the performance of each calibration technique, we set up the calibration of a simple synthetic log return time series generated by an Autoregressive–Moving-Average (ARMA) model (Whittle, 1951) with three known free-parameters ($\phi = 0.7$, $\theta = 0.3$, $\sigma = 0.01$). This exercise provides a performance baseline and validates the implementation of each calibration technique.

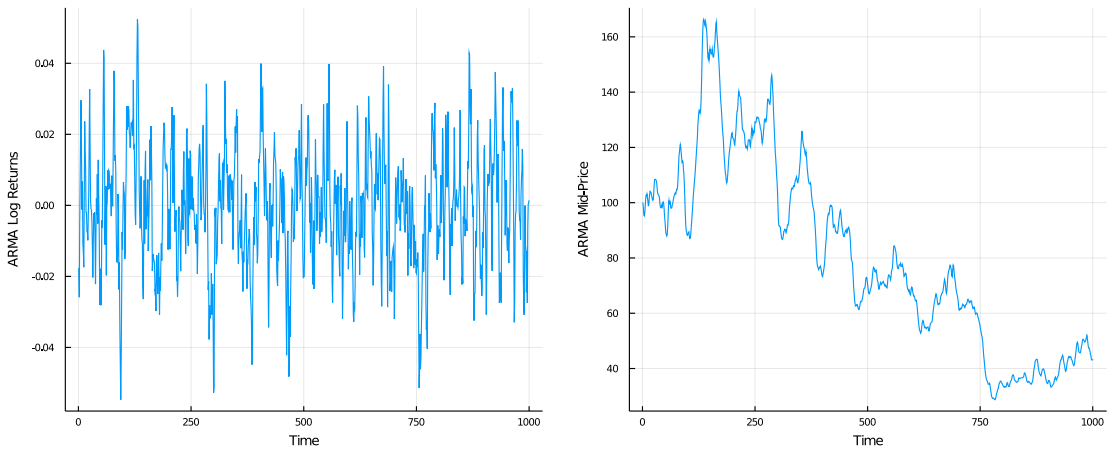


Figure 6: Observed log return time series from an Autoregressive–Moving-Average (ARMA) model comprised of an auto-regressive polynomial of order 1 and a moving average polynomial of order 1. The free-parameters of the ARMA model are $\phi = 0.7$, $\theta = 0.3$ and $\sigma = 0.01$ and the seed is 37162. The mid-price path is constructed as the cumulative sum of the log return time series.

The calibration techniques are compared based on the magnitude of the difference between the true free-parameters and the corresponding estimated free-parameters from the calibration. The table below summarises the calibrated free-parameter estimates with the absolute difference between the true free-parameter and the estimate shown in brackets:

Table 3: The calibration results of six techniques that attempted to recover the true free-parameters from a single time series generated from an ARMA model with $\phi = 0.7$, $\theta = 0.3$, $\sigma = 0.01$. The values in the table are the point estimate from each technique and their absolute difference from the relevant true free-parameter.

	Free-Parameter Estimates		
Calibration Technique	$\hat{\phi}$	$\hat{\theta}$	$\hat{\sigma}$
ABC rejection BBWM	0.61 (0.09)	0.52 (0.22)	0.009 (0.001)
ABC rejection MADWE	0.6 (0.1)	**0.4 (0.1)**	0.014 (0.004)
ABC-PMC BBWM	0.64 (0.06)	0.45 (0.15)	0.009 (0.001)
ABC-PMC MADWE	**0.67 (0.03)**	**0.4 (0.1)**	**0.01 (0)**
Nelder-Mead BBWM	0.8 (0.07)	0.65 (0.45)	0.006 (0.004)
Nelder-Mead MADWE	0.48 (0.22)	0.58 (0.28)	**0.01 (0)**

The ARMA synthetic data calibration results indicate that there is some level of calibration occurring, with the iterative ABC method generally performing better than the simple ABC rejection sampling technique. On the other hand, the Nelder-Mead method seems to struggle with the stochastic nature of the simulator model and the

approximate distance function. Additionally, the Nelder-Mead algorithm is very sensitive to the initial free-parameters it starts out with and becomes stuck in local minima quite frequently. Although there is room for further fine-tuning, the initial results validate the implementation of the calibration techniques and provide baseline expectations for the relative performances of each going forward.

We can visualise the distance surface with respect to two free-parameters by plotting the distance value for pairs of ϕ and θ free-parameters around the true free-parameter values. Figure 7 below displays the MADWE distance surface of ϕ vs. θ with $\sigma = 0.01$. The surface plot decreases around the true free-parameter ϕ and θ values, indicated by the red dot. This is further evidence that the ABC calibration techniques are better suited than black-box optimisation techniques for problems with stochastic components and approximate distance functions.

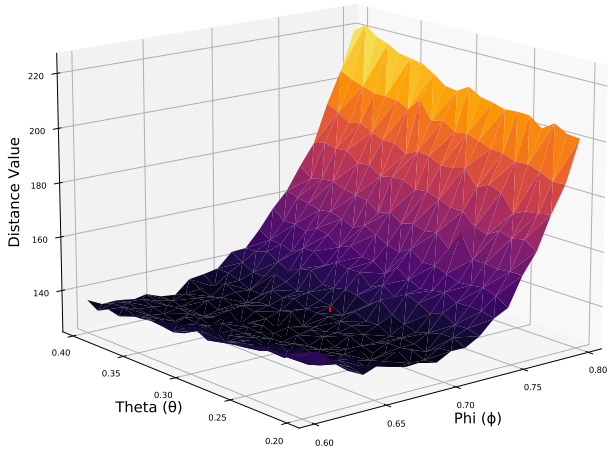


Figure 7: Surface plot of the MADWE distance function for θ and ϕ free-parameter values against a simulated log return time series from an ARMA(1,1) model with true free-parameters, ($\phi = 0.7$, $\theta = 0.3$ and $\sigma = 0.01$). The true free-parameter values for θ and ϕ are overlaid on the surface plot with a red dot.

6.2 Synthetic LOB Calibration

Having validated the implementation of the calibration techniques with a simple ARMA time series calibration exercise, we move on to a more complex synthetic time series, one generated by the LOB model. This synthetic calibration provides insight into whether the LOB model's free-parameters are degenerate or recoverable as well as further performance insight into the calibration techniques. We use the LOB mid-price path displayed in Figure 4 as the "observed" mid-price path which we calibrate against. The table below summarises the LOB synthetic calibration results for the six methods and four free-parameters.

Table 4: The calibration results of 6 techniques that attempted to recover the true free-parameters from a single time series generated from a LOB model with $D = 1.0$, $\sigma = 1.5$, $\nu = 0.5$ and $\mu = 1.0$. The values in the table are the point estimates from each technique and their absolute difference from the relevant true free-parameter in brackets. The best point-estimate values are highlighted in bold.

	Free-Parameter Estimates			
Calibration Technique	$\hat{D}$	$\hat{\sigma}$	$\hat{\nu}$	$\hat{\mu}$
ABC rejection BBWM	1.551 (0.551)	1.821 (0.321)	**0.503 (0.003)**	1.496 (0.496)
ABC rejection MADWE	1.407 (0.407)	1.945 (0.945)	0.546 (0.046)	1.838 (0.838)
ABC-PMC BBWM	**1.086 (0.086)**	**1.596 (0.096)**	0.457 (0.043)	**1.279 (0.279)**
ABC-PMC MADWE	1.182 (0.182)	1.767 (0.267)	0.566 (0.066)	1.75 (0.75)
Nelder-Mead BBWM	1.778 (0.778)	1.944 (0.444)	0.719 (0.219)	1.756 (0.756)
Nelder-Mead MADWE	2.373 (1.373)	1.924 (0.423)	0.52 (0.02)	1.295 (0.295)

With three of the most accurate point-wise estimates, and the fourth free-parameter close to the true value, the ABC-PMC BBWM calibration technique seems to have performed exceedingly well and recovered the true free-parameters of the LOB model. To further investigate the performance, we can plot the particles of certain iterations in the ABC-PMC method to both analyse the improvement over each iteration and the posterior distributions of the free-parameters.

In Figure 8, we observe the improvement in the calibration as the iterations progress. The darker blue particles are more closely clustered around the red particle, which represents the true free-parameters, compared to the spread of the lighter blue particles from earlier iterations. It is interesting to see the distinct shape of the particles in the σ scatter plots. It seems that this free-parameter has the most significant effect on the LOB model output and has been recovered the best.

From the histograms for the D, ν and μ free-parameters, although the mean (green line) and median (black line) in each plot are close to the true free-parameter (red line), we do not observe a peak in the density bars here. In the case of the D free-parameter, the density is clustered towards the left side of the uniform prior (blue line), possibly indicating that smaller values for D in the LOB model generate mid-price paths which are closer to the observed synthetic mid-price path.

For the ν and μ cases, the histograms are more evenly spread across the priors domain with some skewness present that could hint towards the possibility of free-parameter recovery. The three other ABC calibration matrix plots can be viewed in Appendix D.

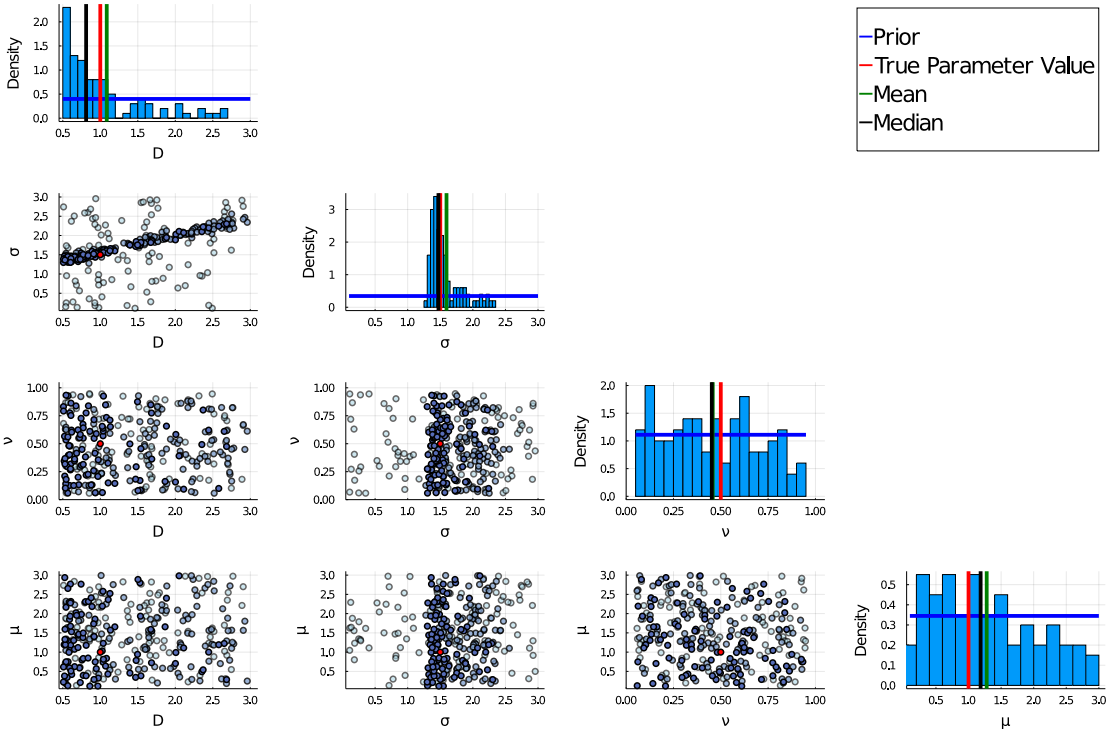


Figure 8: The free-parameter particle scatter plot and histogram matrix for the ABC-PMC calibration technique with the block bootstrap weight matrix (BBWM) distance function. A single observation from the LOB model with free-parameters ($D = 1.0$, $\sigma = 1.5$, $\nu = 0.5$, $\mu = 1.0$) plotted in red was the focus of the calibration. The iterations of the ABC-PMC method can be observed by the lighter blue dots, indicating early iterations, to darker blue dots, indicating later iterations. The histograms are created from the final iteration only.

6.3 Synthetic SLOB Calibration

Following on from the synthetic LOB model calibration, we perform the same process but with the more complex SLOB model, using the mid-price path displayed in Figure 5 as the "observed" mid-price path which we calibrate against. The true free-parameters which we attempt to recover are $D = 1.0$, $\sigma = 1.5$, $\nu = 0.5$, $\mu = 1.0$, $\alpha = 100.0$, similar to the LOB synthetic calibration but with a different α value. Table 5 displays the free-parameter point estimates and absolute differences from the true free-parameters for the 6 calibration techniques.

Table 5: The calibration results of 6 techniques that attempted to recover the true free-parameters from a single time series generated from a SLOB model with $D = 1.0$, $\sigma = 1.5$, $\nu = 0.5$, $\mu = 1.0$, $\alpha = 100.0$. The values in the table are the point estimates from each technique and their absolute difference from the relevant true free-parameter in brackets.

	Free-Parameter Estimates				
Calibration Technique	$\hat{D}$	$\hat{\sigma}$	$\hat{\nu}$	$\hat{\mu}$	$\hat{\alpha}$
ABC rejection BBWM	1.738 (0.738)	1.752 (0.252)	0.669 (0.169)	1.512 (0.512)	124.843 (24.843)
ABC rejection MADWE	1.806 (0.806)	1.749 (0.249)	0.525 (0.025)	1.466 (0.466)	120.572 (20.572)
ABC-PMC BBWM	1.873 (0.873)	1.788 (0.288)	0.625 (0.125)	1.477 (0.477)	138.597 (38.597)
ABC-PMC MADWE	1.819 (0.819)	1.748 (0.248)	**0.517 (0.017)**	**1.402 (0.402)**	**114.808 (14.808)**
Nelder-Mead BBWM	**1.735 (0.735)**	1.775 (0.275)	0.599 (0.099)	2.203 (1.203)	125.624 (25.624)
Nelder-Mead MADWE	1.819 (0.819)	**1.711 (0.211)**	0.608 (0.108)	1.786 (0.786)	143.291 (43.291)

The results of the SLOB synthetic calibration are quite different from the seemingly recoverable LOB synthetic calibration results. Firstly, the D free-parameter point estimates are all clustered between 1.7 and 1.9, indicating that the distance function could be incorrectly capturing information about this free-parameter from the summary statistics of the mid-price paths. This might be a flaw in the distance function or the chosen summary statistics. Looking at Figure 9, the free-parameter particle scatter plot and histogram matrix for the ABC-PMC MADWE technique, the spread of the particles in the D scatter plots and the somewhat even spread of density in the D histogram lead us to consider that the summary statistics are misguided. This could lead the distance functions for all techniques to the incorrect minimum, thus making the D free-parameter unrecoverable.

Secondly, the well recovered σ free-parameter in the LOB calibration has been estimated to be between 1.7 and 1.85 across all techniques in the SLOB calibration, indicating poor free-parameter recovery and a possible convergence around the 1.7 value. All six techniques performed similarly, with a slightly better point estimate coming from the Nelder-Mead MADWE technique.

Thirdly and surprisingly, the point estimates for the ν free-parameter are much closer to the true value when compared to the LOB synthetic calibration. Looking at the ν related plots in Figure 9 reveals clustering for the particles around the true value as well as a significant peak of the density bars around 0.5. This pattern, although not as apparent, can also be observed in the scatter plot and histogram matrix for the other techniques in Appendix E.

Overall, the synthetic calibration has revealed that a few of the free-parameters do seem recoverable, mainly when calibrating with the iterative ABC techniques. However, the complexity and non-linearity present in the LOB and SLOB models combined with the summary function, which can lose information, and approximate calibration means that calibrating to real data will be fraught with unknown pitfalls. Is the LOB or SLOB model unable to simulate a realistic mid-price path or is it the summary function that cannot capture the required information? These possibilities will be considered when evaluating the calibration results against market data.

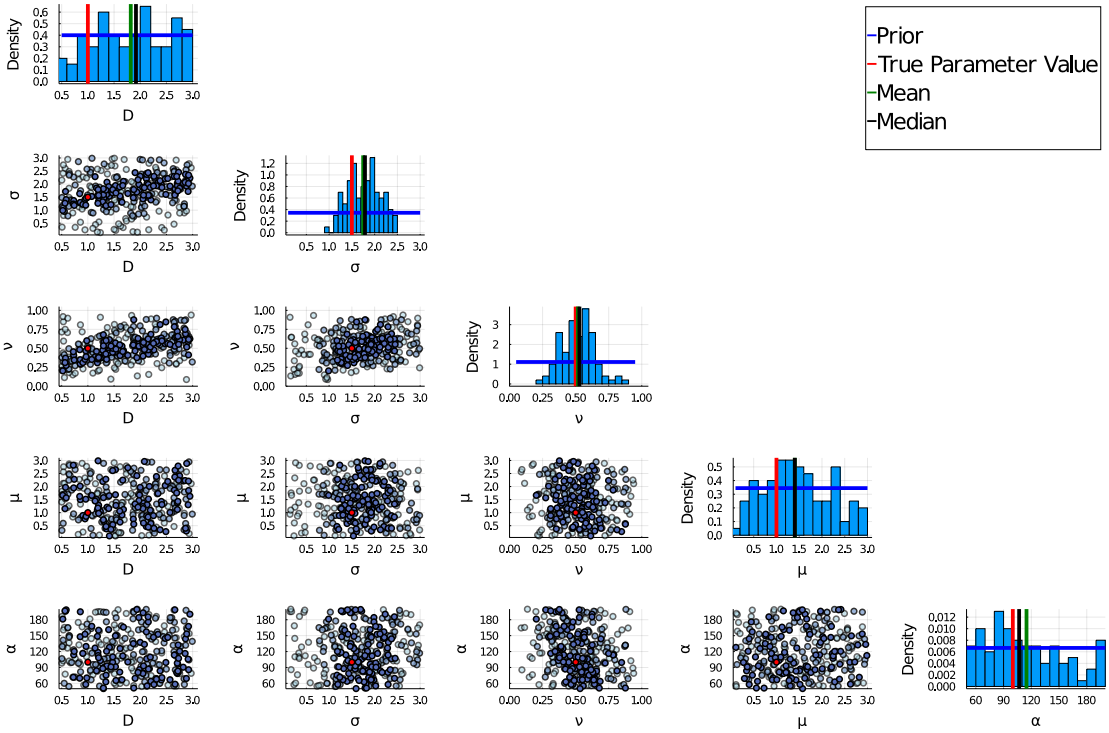


Figure 9: The free-parameter particle scatter plot and histogram matrix for the ABC-PMC calibration technique with the median absolute deviation weighted euclidean (MADWE) distance function. A single observation from the SLOB model with free-parameters ($D = 1.0$, $\sigma = 1.5$, $\nu = 0.5$, $\mu = 1.0$, $\alpha = 100.0$) plotted in red was the focus of the calibration. The iterations of the ABC-PMC method can be observed by the lighter blue dots, indicating early iterations, to darker blue dots, indicating later iterations. The histograms are created from the final iteration only.

7 Market Data Calibration

7.1 Market Data Overview

We consider the use of real world market data in the calibration of the LOB and SLOB models in order to recover a simulation model which can output realistic mid-price paths. This simulation model could then be used for many useful investigations, most notably, the study of price impact. The market data we use is five days of 1-minute mid-price data for Anglo American PLC (AGLJ.J) listed on the Johannesburg Stock Exchange and extracted from Thomson Reuters over the period 9:10 on 1 November 2013 and ending at 16:50 on 5 November 2013 (Gebbie and Platt, 2019). This dataset has been used in a number of papers and we consider it as a standard test set for comparisons to other papers. The data consists of 2700 mid-prices, thus the configuration parameters for the LOB and SLOB models do not need to change from their synthetic calibration runs.

7.2 Market Data LOB Calibration

We calibrate the LOB model to the AGLJ.J mid-price path using the two best performing techniques from section 6.2, the ABC-PMC BBWM and ABC-PMC MADWE techniques. The free-parameter estimates for each technique are summarised in Table 6 below. The ABC-PMC MADWE technique was unable to calibrate past two iterations due to a poorly constructed distance function. Unlike the BBMW distance function, the MADWE distance function's weights are calculated using the previous iterations particles and none of the potential particles generated in the third iteration were accepted. Thus the free-parameter estimates for the LOB ABC-MADWE run are drawn from the prior distribution and should not be considered much further.

In contrast, the calibration results from the BBWM method are very promising, with an extreme concentration for the σ free-parameter around 0.4. In Figure 10 this can be seen in the histogram, with the bars heavily peaked around the mean, and the particles in the scatter plots clustered together. Given the success of recovering the σ free-parameter from the synthetic calibrations, it seems that a good estimate for σ in the LOB model based on the AGLJ.J market dataset is 0.4. The other free-parameters do not share the same level of clustering, with a few peaks on the histograms being observed but not enough to consider a possible free-parameter recovery.

Table 6: Free-parameter estimates as posterior means for the LOB model from calibrations to the AGLJ.J market data using the ABC-PMC MADWE and ABC-PMC BBWM calibration techniques.

	Free-Parameter Estimates			
Calibration Technique	$\hat{D}$	$\hat{\sigma}$	$\hat{\nu}$	$\hat{\mu}$
ABC-PMC MADWE	1.784	1.455	0.496	1.547
ABC-PMC BBWM	1.79	0.401	0.533	1.518

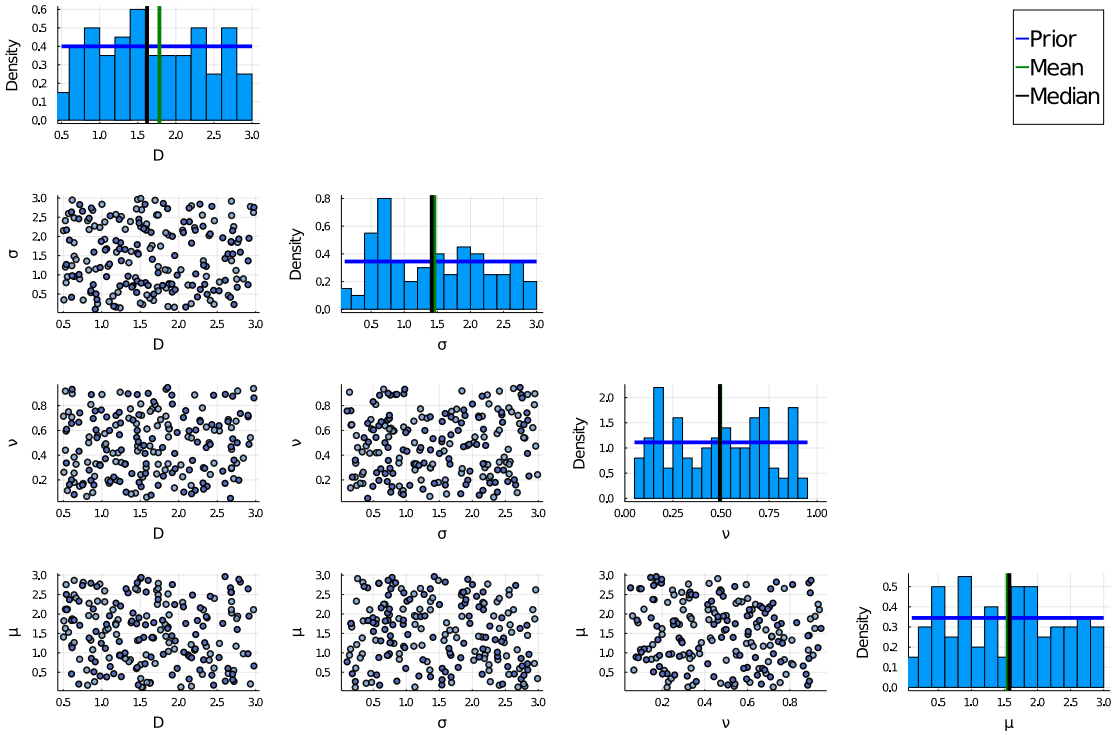


Figure 10: The free-parameter particle scatter plot and histogram matrix for the ABC-PMC calibration technique with the median absolute deviation weighted euclidean (MADWE) distance function. Five days of 1-minute mid-price data for Anglo American PLC (AGLJ.J) listed on the Johannesburg Stock Exchange was used as the observed data that the model was calibrated to. The iterations of the ABC-PMC method can be observed by the lighter blue dots, indicating early iterations, to darker blue dots, indicating later iterations. The histograms are created from the final iteration only.

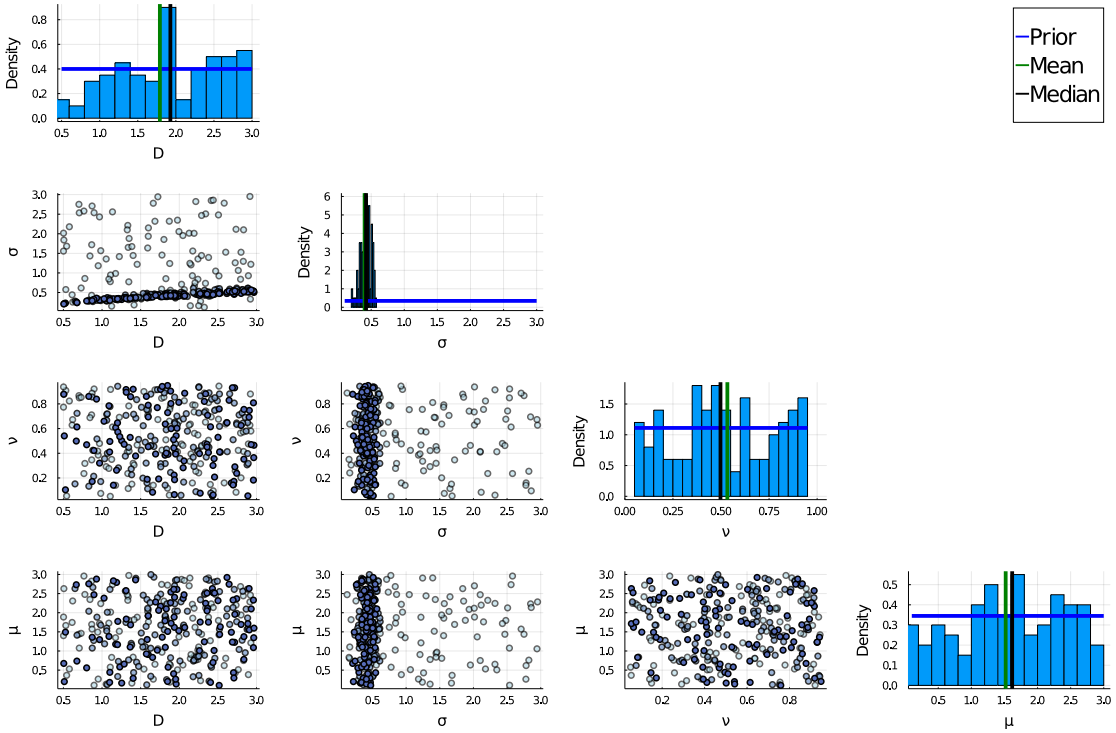


Figure 11: The free-parameter particle scatter plot and histogram matrix for the ABC-PMC calibration technique with the block bootstrap weight matrix (BBWM) distance function. Five days of 1-minute mid-price data for Anglo American PLC (AGLJ.J) listed on the Johannesburg Stock Exchange was used as the observed data that the model was calibrated to. The iterations of the ABC-PMC method can be observed by the lighter blue dots, indicating early iterations, to darker blue dots, indicating later iterations. The histograms are created from the final iteration only.

7.3 Market Data SLOB Calibration

Following the LOB model calibration to the AGLJ.J market dataset, we use the same two techniques to calibrate the SLOB model, with the resulting free-parameter estimates detailed in Table 7. Unlike the LOB market data calibration, the ABC-PMC MADWE technique did not get stuck during its third iteration and was able to generate free-parameter estimates from its tenth iteration.

The free-parameter estimates from the ABC-PMC MADWE calibration technique are mainly centred around the mean of the prior distributions, but the histograms in Figure 12, with spikes and skews, indicate that the posterior distributions deviate from the priors and there is possibly some level of free-parameter recovery. However, there seems to be only minimal clustering of the particles in the scatter plots, leading us to conclude that there is not enough evidence of a successful calibration to the data with this technique.

As with the LOB market data ABC-PMC BBWM calibration, the SLOB model equivalent also finds a very concentrated cluster of particles for the σ free-parameter which can be seen in Figure 13. Here we have a posterior mean value of 0.421, with the σ free-parameter in the SLOB model being very close to the estimate from the LOB

model, indicating a significant pattern and definite effect on the model outputs.

The ν free-parameter also exhibits a concentrated posterior distribution with a fairly narrow histogram around the mean value of 0.603 and scatter plots with noticeable clustering. Of note is the scatter plot between the ν and σ free-parameters where the contrast between the light blue early iteration particles and the dark blue final iteration particles can be seen. This convergence on a posterior distribution provides evidence of free-parameter recovery and impact on the model output from the free-parameters.

Table 7: Free-parameter estimates as posterior means for the SLOB model from calibrations to the AGLJ.J market data using the ABC-PMC MADWE and ABC-PMC BBWM calibration techniques.

	Free-Parameter Estimates				
Calibration Technique	$\hat{D}$	$\hat{\sigma}$	$\hat{\nu}$	$\hat{\mu}$	$\hat{\alpha}$
ABC-PMC MADWE	1.661	1.127	0.667	1.987	127.368
ABC-PMC BBWM	2.087	0.421	0.603	1.202	106.504

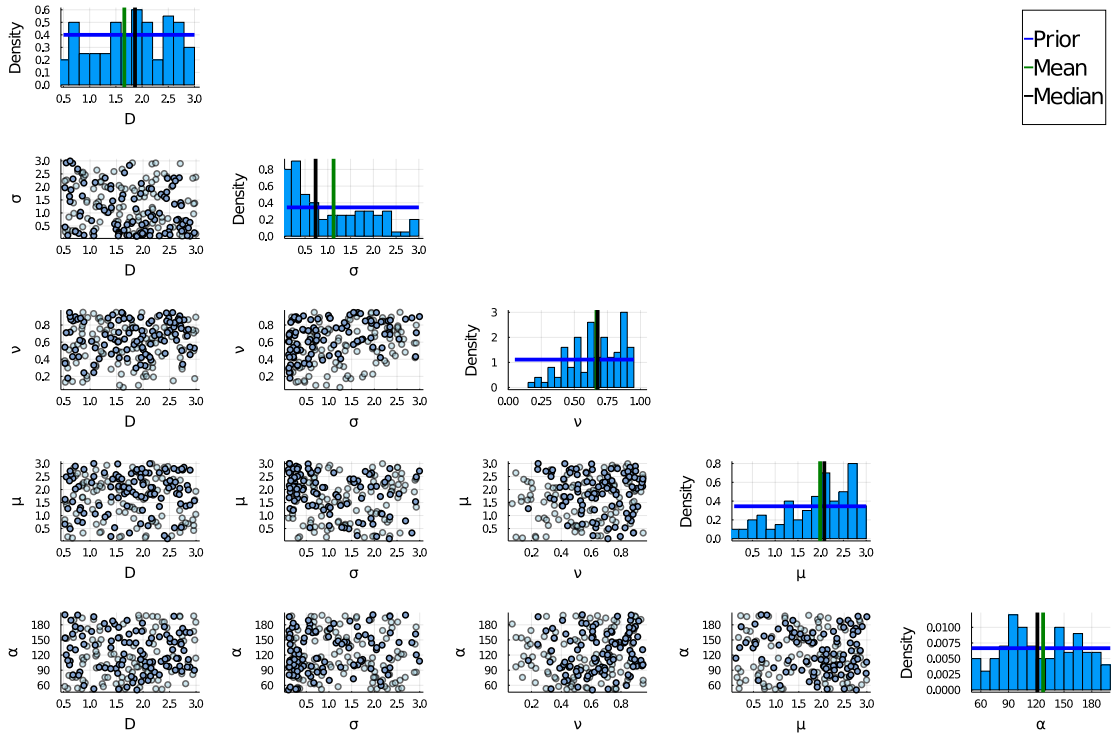


Figure 12: The free-parameter particle scatter plot and histogram matrix for the ABC-PMC calibration technique with the median absolute deviation weighted euclidean (MADWE) distance function. Five days of 1-minute mid-price data for Anglo American PLC (AGLJ.J) listed on the Johannesburg Stock Exchange was used as the observed data that the model was calibrated to. The iterations of the ABC-PMC method can be observed by the lighter blue dots, indicating early iterations, to darker blue dots, indicating later iterations. The histograms are created from the final iteration only.

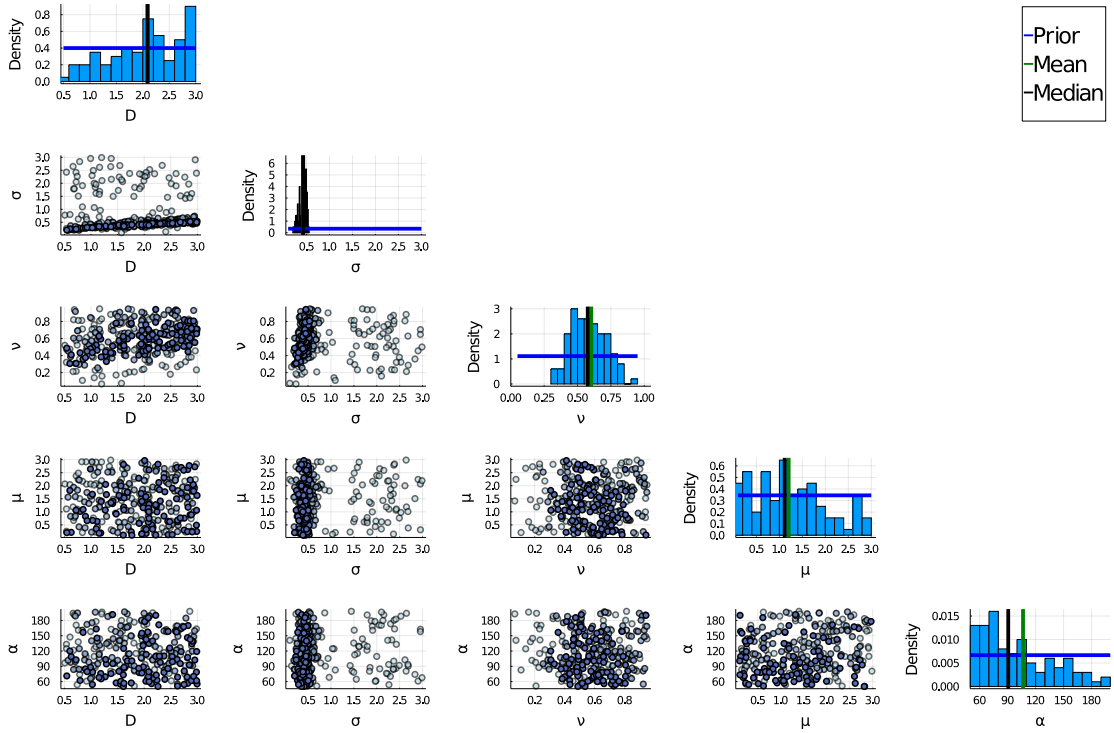


Figure 13: The free-parameter particle scatter plot and histogram matrix for the ABC-PMC calibration technique with the block bootstrap weight matrix (BBWM) distance function. Five days of 1-minute mid-price data for Anglo American PLC (AGLJ.J) listed on the Johannesburg Stock Exchange was used as the observed data that the model was calibrated to. The iterations of the ABC-PMC method can be observed by the lighter blue dots, indicating early iterations, to darker blue dots, indicating later iterations. The histograms are created from the final iteration only.

7.4 Stylised Facts Comparison

Stylised facts are empirically based statements that represent phenomenology that is consistently found in a given system and hence any reasonable model should recover these empirical features (Sewell, 2011). When analysing financial time series, stylised facts can be used to understand the characteristics of the time series and furthermore, compare multiple time series as to whether they share similar stylised facts.

Having completed the LOB and SLOB calibration to real market data results, we move on to comparing the observed AGKJ.J mid-price path time series to a simulated mid-price path time series from the ABC-PMC BBWM calibrated LOB and SLOB models.

We generate six plots for each mid-price path to compare and analyse three stylised facts, beginning with the Leptokurtic distribution of log returns.

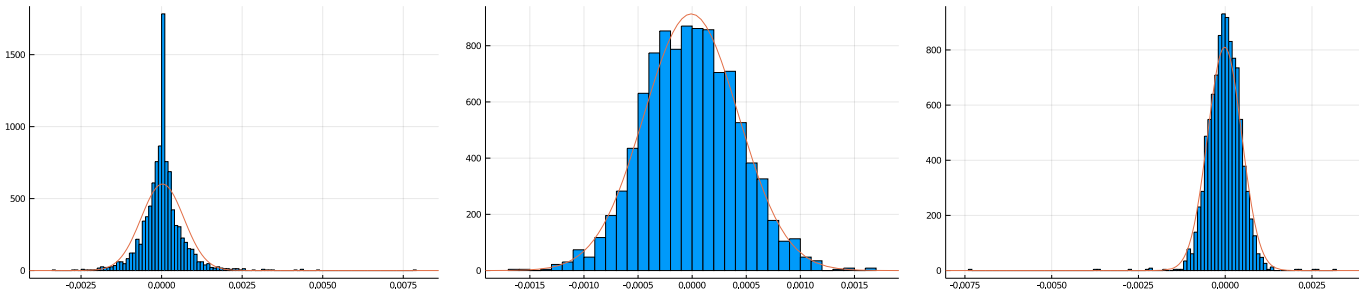
7.4.1 Leptokurtic Distribution of Log Returns

The distribution of returns and log returns is known to be approximately symmetric with fat tails and a higher peak around the mode (Sewell, 2011). In Figure 14, the

log returns of a mid-price path from each source are displayed as histograms with a normal distribution overlaid. The expected high peak is clearly observed with the AGKJ.J log returns as the bar in the middle of the histogram plot is far higher than the normal distribution curve.

In contrast, the LOB log returns seem to fit very well with the normal distribution curve, even at the tails. Further proof of this can be seen in the QQ-plot of the log returns in Figure 22c. This leads us to conclude that the calibrated LOB model is not able to simulate realistic log returns.

The SLOB model log returns are marginally better, with a slightly higher peak around the mode but with pronounced fat tails. The QQ plot in Figure 23c displays the fat tails but there is minimal curvature around the centre as can be observed with the AGKJ.J log returns in Figure 21c. Thus, the sequential nature of the SLOB model with the reset of the latent order book seems to have recovered part of the leptokurtic distribution of log returns.



(a) AGKJ.J log returns **(b)** LOB log returns **(c)** SLOB log returns

Figure 14: Histogram plots of the log returns from three mid-price paths, the observed AGKJ.J market data, the ABC-PMC BBWM LOB model and the ABC-PMC BBWM SLOB model. A normal distribution is overlaid on the histogram bars, with each distribution being fit based on the respective data.

7.4.2 Volatility Clustering

Volatility clustering is the observation of large absolute returns occurring in the same time window, seemingly following one another (Lux and Marchesi, 2000). This can be quantified by calculating the auto-correlation function of the absolute or squared returns, where we should observe significant values at lower lags and a slow decay towards zero as the lag increases.

In Figure 21e, the log return auto-correlation function is not significant at any lags but the absolute log return auto-correlation function displayed in Figure 21f is significant and slowly decreasing for the first ten lags. Further confirmation of the clustering can be seen in Figure 21b, where periods of large returns occur close together.

Although the LOB model simulates a mid-price path whose log returns are not auto-correlated, as can be seen in Figure 22e, it has failed to simulate log returns that display the volatility clustering. In Figure 22e, there are no significant lags nor is there any sequential decrease.

The SLOB model shares a similar lack of significant auto-correlation coefficients with

the LOB model. In Figure 23f, there doesn't appear to be any significant lags and this can be further ratified by considering Figure 23b, where large log returns are isolated and not clustered near one another. Thus we can conclude that the LOB and SLOB models were not able to recover the volatility clustering present in the real market data.

7.4.3 Order Flow Autocorrelation

The final stylised fact that we consider is the phenomenon where the time series of the trade order signs have a long memory property (Lillo and Farmer, 2004). Sell orders are more likely to be followed by sell orders and buy orders are more likely to follow buy orders. This positive correlation can be captured by an auto-correlation function calculated on the order sign time series. However, as we do not have the quote data for the AGKJ.J, LOB and SLOB mid-price paths we will need to infer the order signs for all three mid-price paths. We make use of a simple technique, the Tick Rule or Tick Test (Lee and Ready, 1991), where a mid-price path time series is classified based on the prior mid-price.

In all three time series, there was no significant auto-correlation of the order signs. Figures 21d, 22d, 23d show the coefficients within the confidence intervals. Due to the use of an order sign inference technique and the granularity of the time series, we cannot fully conclude that there is no order flow correlation. It is worth noting that there is a similarity between the observed order flow and the simulated order flow, and thus a possible success for the LOB and SLOB models.

8 Conclusion

Many approaches to simulating financial time series struggle to recover some or all of the stylised facts and thus cannot be used to investigate the underlying structure of the market (Platt and Gebbie, 2018). Our novel approach to this problem, through top-down modelling of the limit order book as a latent order book, has also met similar hurdles but still found a few successes.

We began with the theoretical concept of a latent order book (Tóth et al., 2011) and derived a reaction-diffusion partial differential equation in Section 2.2 following the work of Donier et al. (2015). In Section 3, we proposed an adaption of a numerical solution for reaction-diffusion equations (Angstmann et al., 2020) and proved that in the diffusion limit, the master equation from a discrete time random walk numerical scheme becomes the latent order book reaction-diffusion PDE. The defined initial and boundary conditions allowed us to iteratively solve the equation and obtain an evolution of solutions over time from which we could extract a mid-price path.

Further experimentation of the LOB model led to the resolving of the initial conditions after an exponentially distributed waiting time, giving us the sequential latent order book model. The hope of the introduction of this mechanism was to introduce shocks and information dislocations, similar to trading halts that occur in real limit order book markets.

With two models that could generate random walk seeming mid-price paths, we attempted to calibrate the models with the likelihood-free approach, approximate

Bayesian computation. Synthetic calibration of each model to a self-generated mid-price path revealed the strengths and flaws of both the models and the calibration techniques. Certain free-parameters were found to not affect the model output significantly whereas others were easily recoverable. Additionally, the posterior distributions for the free-parameters provided further insight into model dynamics, something that the black-box optimisation approach could not provide.

After confirming the potential of the ABC calibration approach in Section 6, we attempted to calibrate the LOB and SLOB models to market data, where the posterior distributions of the free-parameters served as a form of validation. We found success with a few free-parameters, the most significant being the σ free-parameter which controlled the variance of the normal distribution of the V_t term in the model. The varying posterior distributions were interesting to note and provided further justification for the SLOB model over the LOB model.

Our final set of analyses compared the stylised facts of the observed AGKJ.J mid-price path to simulated mid-price paths from the best calibrated LOB and SLOB models. Again, there was some success with the SLOB mid-price path displaying a leptokurtic distribution of log returns. However, the LOB and SLOB models failed to simulate the volatility clustering that is so commonly observed in stock markets. Improvements to the LOB and SLOB models could come from introducing terms that induce the missing stylised facts.

References

Angstmann, C. N., Donnelly, I. C., Henry, B. I., Jacobs, B., Langlands, T. A., and Nichols, J. A. (2016). From stochastic processes to numerical methods: A new scheme for solving reaction subdiffusion fractional partial differential equations. *Journal of Computational Physics*, 307:508–534.

Angstmann, C. N., Henry, B. I., Jacobs, B. A., and McGann, A. V. (2020). Numeric solution of advection–diffusion equations by a discrete time random walk scheme. *Numerical Methods for Partial Differential Equations*, 36(3):680–704.

Beaumont, M. A., Cornuet, J.-M., Marin, J.-M., and Robert, C. P. (2009). Adaptive approximate bayesian computation. *Biometrika*, 96(4):983–990.

Bezanson, J., Edelman, A., Karpinski, S., and Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM review*, 59(1):65–98.

Bouchaud, J.-P. (2009). Price impact. *arXiv preprint arXiv:0903.2428*.

Carmona, R. (2014). *Statistical analysis of financial data in R*, volume 2. Springer.

Cartea, Á., Jaimungal, S., and Penalva, J. (2015). *Algorithmic and high-frequency trading*. Cambridge University Press.

Csilléry, K., François, O., and Blum, M. G. (2012). abc: an r package for approximate bayesian computation (abc). *Methods in ecology and evolution*, 3(3):475–479.

Donier, J., Bonart, J., Mastromatteo, I., and Bouchaud, J.-P. (2015). A fully consistent, minimal model for non-linear market impact. *Quantitative finance*, 15(7):1109–1121.

Forneron, J.-J. and Ng, S. (2018). The abc of simulation estimation with auxiliary statistics. *Journal of Econometrics*, 205(1):112–139.

Franke, R. (2009). Applying the method of simulated moments to estimate a small agent-based asset pricing model. *Journal of Empirical Finance*, 16(5):804–815.

Gant, M. and Gebbie, T. (2020a). AdaptiveABC.jl. URL: https://github.com/GantZA/AdaptiveABC.jl. doi: https://doi.org/10.25375/uct.19127930.

Gant, M. and Gebbie, T. (2020b). Julia code: Dissertation. URL: https://github.com/GantZA/CALOBMTMD. doi: https://doi.org/10.25375/uct.19127855.

Gant, M. and Gebbie, T. (2020c). SequentialLOB.jl. URL: https://github.com/GantZA/SequentialLOB.jl. doi: https://doi.org/10.25375/uct.19127927.

Gant, M. and Gebbie, T. (2020d). StylizedFacts.jl. URL: https://github.com/GantZA/StylizedFacts.jl. doi: https://doi.org/10.25375/uct.19127933.

Gao, F. and Han, L. (2012). Implementing the nelder-mead simplex algorithm with adaptive parameters. *Computational Optimization and Applications*, 51(1):259–277.

Gebbie, T. and Platt, D. (2019). AGLJ.J 5 days of 1-minute mid-price bar test data. https://data.mendeley.com/datasets/nt8nw28h7c/1.

Henry, B. I., Langlands, T., and Straka, P. (2010). Fractional fokker-planck equations for subdiffusion with space-and time-dependent forces. *Physical review letters*, 105(17):170602.

Hurst, H. E. (1951). Long-term storage capacity of reservoirs. *Transactions of the American society of civil engineers*, 116(1):770–799.

Lee, C. M. and Ready, M. J. (1991). Inferring trade direction from intraday data. *The Journal of Finance*, 46(2):733–746.

Lillo, F. and Farmer, J. D. (2004). The long memory of the efficient market. *Studies in nonlinear dynamics & econometrics*, 8(3).

Lux, T. and Marchesi, M. (2000). Volatility clustering in financial markets: a microsimulation of interacting agents. *International journal of theoretical and applied finance*, 3(04):675–702.

Marin, J.-M., Pudlo, P., Robert, C. P., and Ryder, R. J. (2012). Approximate bayesian computational methods. *Statistics and Computing*, 22(6):1167–1180.

Marjoram, P., Molitor, J., Plagnol, V., and Tavaré, S. (2003). Markov chain monte carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 100(26):15324–15328.

Mazumder, S. (2015). *Numerical methods for partial differential equations: finite difference and finite volume methods*. Academic Press.

Platt, D. (2020). A comparison of economic agent-based model calibration methods. *Journal of Economic Dynamics and Control*, 113:103859.

Platt, D. and Gebbie, T. (2018). Can agent-based models probe market microstructure? *Physica A: Statistical Mechanics and its Applications*, 503:1092–1106.

Prangle, D. et al. (2017). Adapting the abc distance function. *Bayesian Analysis*, 12(1):289–309.

Preis, T., Virnau, P., Paul, W., and Schneider, J. J. (2009). Accelerated fluctuation analysis by graphic cards and complex pattern formation in financial markets. *New Journal of Physics*, 11(9):093024.

Roşu, I. (2009). A dynamic model of the limit order book. *The Review of Financial Studies*, 22(11):4601–4641.

Sewell, M. (2011). Characterization of financial time series. *Rn*, 11(01):01.

Sisson, S. A., Fan, Y., and Tanaka, M. M. (2007). Sequential monte carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 104(6):1760–1765.

Smith, E., Farmer, J. D., Gillemot, L. s., Krishnamurthy, S., et al. (2003). Statistical theory of the continuous double auction. *Quantitative finance*, 3(6):481–514.

Toni, T., Welch, D., Strelkowa, N., Ipsen, A., and Stumpf, M. P. (2009). Approximate bayesian computation scheme for parameter inference and model selection in dynamical systems. *Journal of the Royal Society Interface*, 6(31):187–202.

Tóth, B., Lemperiere, Y., Deremble, C., De Lataillade, J., Kockelkoren, J., and Bouchaud, J.-P. (2011). Anomalous price impact and the critical nature of liquidity in financial markets. *Physical Review X*, 1(2):021006.

Whittle, P. (1951). *Hypothesis testing in time series analysis*, volume 4. Almqvist & Wiksells boktr.

Winker, P., Gilli, M., and Jeleskovic, V. (2007). An objective function for simulation based inference on exchange rate data. *Journal of Economic Interaction and Coordination*, 2(2):125–145.

9 Appendices

A Simplification of Bid and Ask Partial Differential Equations

Given two partial differential equations:

$$\frac{\partial \rho_B(x,t)}{\partial t} = D\frac{\partial^2 \rho_B(x,t)}{\partial x^2} - V_t\frac{\partial \rho_B(x,t)}{\partial x} - \nu\rho_B(x,t) + s_B(x,t) - \kappa R_{AB}(x,t)$$

$$\frac{\partial \rho_A(x,t)}{\partial t} = D\frac{\partial^2 \rho_A(x,t)}{\partial x^2} - V_t\frac{\partial \rho_A(x,t)}{\partial x} - \nu\rho_A(x,t) + s_A(x,t) - \kappa R_{AB}(x,t)$$

with common term, $-\kappa R_{AB}(x,t)$, we can move terms and simplify to one equation as follows:

$$\begin{aligned}\frac{\partial \rho_B(x,t)}{\partial t} - D\frac{\partial^2 \rho_B(x,t)}{\partial x^2} + V_t\frac{\partial \rho_B(x,t)}{\partial x} + \nu\rho_B(x,t) - s_B(x,t) = \\ \frac{\partial \rho_A(x,t)}{\partial t} - D\frac{\partial^2 \rho_A(x,t)}{\partial x^2} + V_t\frac{\partial \rho_A(x,t)}{\partial x} + \nu\rho_A(x,t) - s_A(x,t)\end{aligned}$$

We then group terms with common factors

$$\begin{aligned}\frac{\partial \rho_B(x,t)}{\partial t} - \frac{\partial \rho_A(x,t)}{\partial t} &= D\big[\frac{\partial^2 \rho_B(x,t)}{\partial x^2} - \frac{\partial^2 \rho_A(x,t)}{\partial x^2}\big] - V_t\big[\frac{\partial \rho_B(x,t)}{\partial x} - \frac{\partial \rho_A(x,t)}{\partial x}\big] \\ &\quad -\nu\big(\rho_B(x,t) - \rho_A(x,t)\big) + s_B(x,t) - s_A(x,t) \\ \frac{\partial\big(\rho_B(x,t) - \rho_A(x,t)\big)}{\partial t} &= D\frac{\partial^2\big(\rho_B(x,t) - \rho_A(x,t)\big)}{\partial x^2} - V_t\frac{\partial\big(\rho_B(x,t) - \rho_A(x,t)\big)}{\partial x} \\ &\quad -\nu\big(\rho_B(x,t) - \rho_A(x,t)\big) + s_B(x,t) - s_A(x,t)\end{aligned}$$

and finally make substitutions $\varphi(x,t) = \rho_B(x,t) - \rho_A(x,t)$ and $s(x,t) = s_B(x,t) - s_A(x,t)$ to get Equation 5:

$$\frac{\partial \varphi(x,t)}{\partial t} = D\frac{\partial^2 \varphi(x,t)}{\partial x^2} - V_t\frac{\partial \varphi(x,t)}{\partial x} - \nu\varphi(x,t) + s(x,t)$$

B Numerical Scheme Diffusion Limit Requirement

During the formulation of the numerical scheme to solve the latent order book reaction-diffusion partial differential Equation 28, a condition was imposed in the diffusion limit, specifically,

$$F(x,t) = \lim_{\Delta x\to 0}\frac{f_\Delta(x,t)}{\beta\Delta x} = \lim_{\Delta x\to 0}\frac{p_{r,\Delta}(x,t) - p_{\ell,\Delta}(x,t)}{\beta\Delta x}$$

which we show to be true below, given the choices for the jump probabilities in Equations 20, 21 and 22 and the choice of $V(x,t)$ in Equation 23.

By substituting the above choices into the equation, we obtain:

$$\begin{aligned}
\frac{V_t}{2\beta D} &= \lim_{\Delta x\to 0}\frac{\exp(\frac{3V_t\Delta x}{4D})-\exp(-\frac{3V_t\Delta x}{4D})}{\exp(\frac{3V_t\Delta x}{4D})+\exp(-\frac{3V_t\Delta x}{4D})+1}\cdot\frac{1}{\beta\Delta x}\\
&= \lim_{\Delta x\to 0}\frac{\exp(\frac{3V_t\Delta x}{4D})}{\exp(\frac{3V_t\Delta x}{4D})}\cdot\frac{1-\exp(-\frac{3V_t\Delta x}{2D})}{1+\exp(-\frac{3V_t\Delta x}{2D})+\exp(-\frac{3V_t\Delta x}{4D})}\cdot\frac{1}{\beta\Delta x}\\
&= \lim_{\Delta x\to 0}\frac{1}{1+\exp(-\frac{3V_t\Delta x}{2D})+\exp(-\frac{3V_t\Delta x}{4D})}\cdot\lim_{\Delta x\to 0}\frac{1-\exp(-\frac{3V_t\Delta x}{2D})}{\beta\Delta x}\\
&= \frac{1}{3}\cdot\lim_{\Delta x\to 0}\frac{1-\exp(-\frac{3V_t\Delta x}{2D})}{\beta\Delta x}
\end{aligned}$$

We then apply L'Hôpital's Rule and take the derivative of the numerator and denominator with respect to Δx:

$$\begin{aligned}
\lim_{\Delta x\to 0}\frac{1-\exp(-\frac{3V_t\Delta x}{2D})}{\beta\Delta x} &\overset{\text{H}}{=} \lim_{\Delta x\to 0}\frac{-\exp(-\frac{3V_t\Delta x}{2D})\cdot-\frac{3V_t}{2D}}{\beta}\\
&= \frac{3V_t}{2D\beta}
\end{aligned}$$

Thus, we recover the LHS:

$$\begin{aligned}
&= \frac{1}{3}\cdot\frac{3V_t}{2D\beta}\\
&= \frac{V_t}{2D\beta} = \text{LHS}
\end{aligned}\tag{44}$$

C Matrix Formulation of Initial Conditions

To find the initial conditions given a state vector $\boldsymbol{\varphi}$ with i^{th} component φ_i, the source vector $\mathbf{s}$ with i^{th} component $-s(x_i,0)$ for $i=0,1,2,\ldots,M$ and matrix A we need to solve the linear equations, $\mathbf{s}=A\boldsymbol{\varphi}$. From Equation 36 the A tridiagonal matrix is:

$$A=\begin{bmatrix}
-\frac{2D}{\Delta x^2}-\nu & \frac{2D}{\Delta x^2} & & & & & \\
\frac{D}{\Delta x^2}+\frac{V_0}{2\Delta x} & -\frac{2D}{\Delta x^2}-\nu & \frac{D}{\Delta x^2}-\frac{V_0}{2\Delta x} & & & & \\
 & \frac{D}{\Delta x^2}+\frac{V_0}{2\Delta x} & -\frac{2D}{\Delta x^2}-\nu & \frac{D}{\Delta x^2}-\frac{V_0}{2\Delta x} & & & \\
 & & \ddots & \ddots & \ddots & & \\
 & & & \frac{D}{\Delta x^2}+\frac{V_0}{2\Delta x} & -\frac{2D}{\Delta x^2}-\nu & \frac{D}{\Delta x^2}-\frac{V_0}{2\Delta x} & \\
 & & & & \frac{2D}{\Delta x^2} & -\frac{2D}{\Delta x^2}-\nu &
\end{bmatrix}$$

D LOB Calibration Figures

D.1 LOB Calibration: ABC rejection BBWM

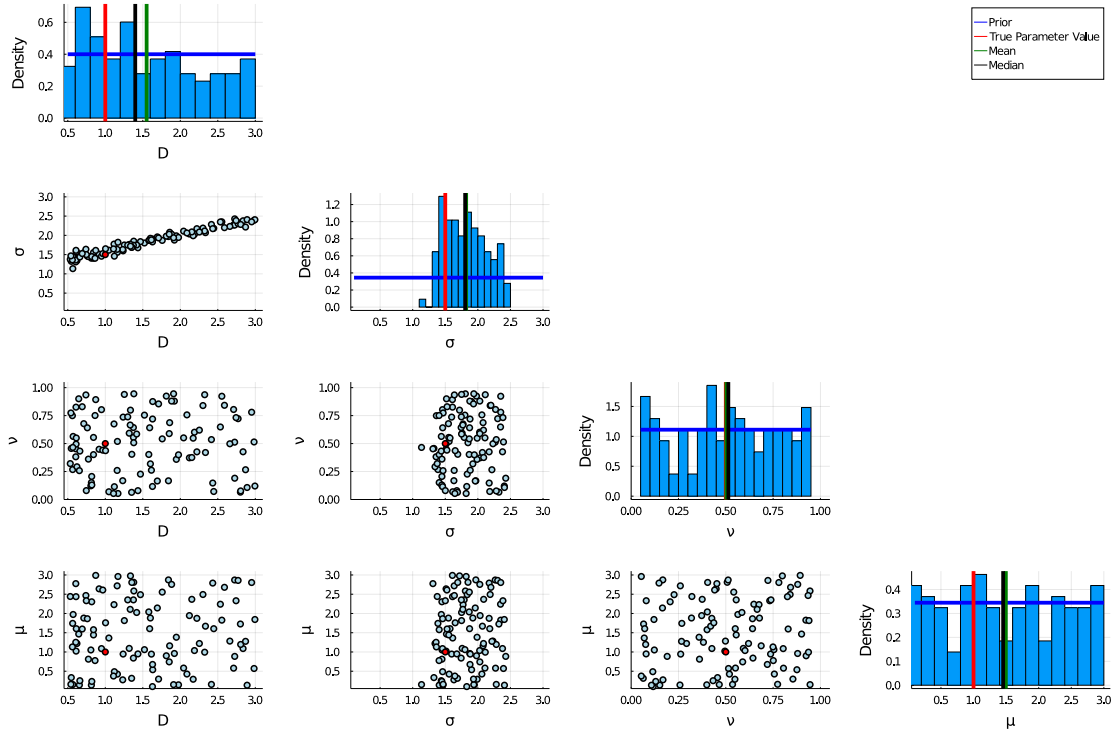


Figure 15: The free-parameter particle scatter plot and histogram matrix for the ABC rejection calibration technique with the block bootstrap weight matrix (BBWM) distance function. The LOB model with free-parameters ($D = 1.0$, $\sigma = 1.5$, $\nu = 0.5$, $\mu = 1.0$) plotted in red was the focus of the calibration.

D.2 LOB Calibration: ABC rejection MADWE

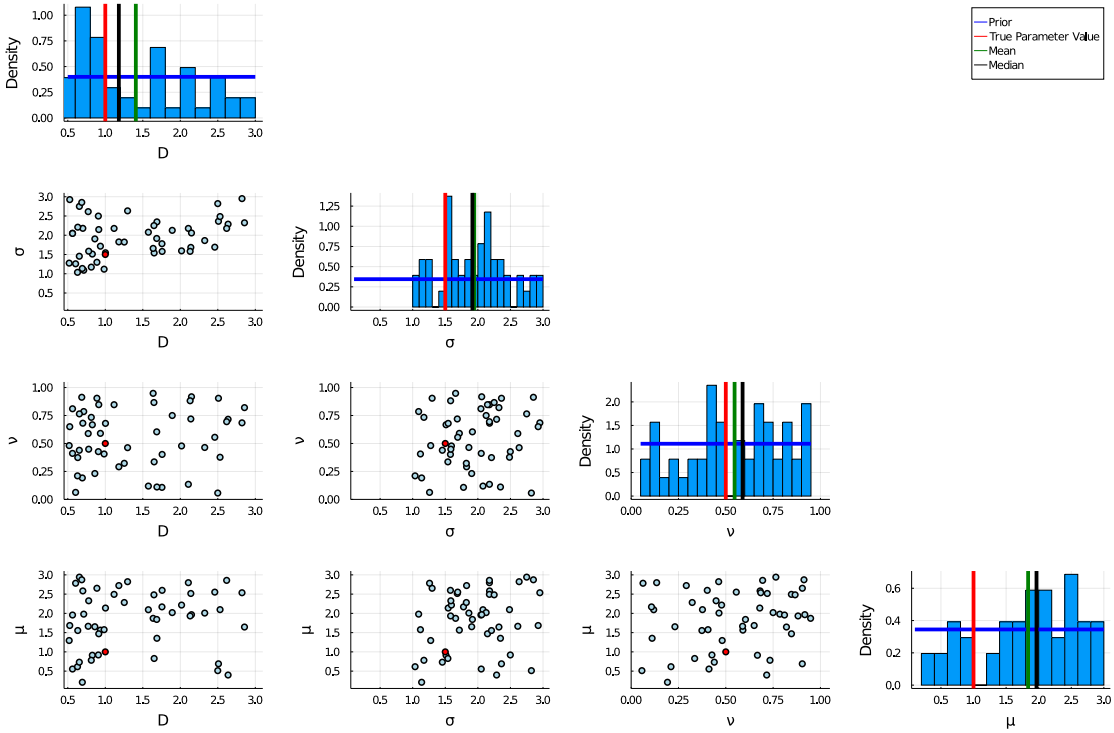


Figure 16: The free-parameter particle scatter plot and histogram matrix for the ABC-PMC calibration technique with the median absolute deviation weighted euclidean (MADWE) distance function. The LOB model with free-parameters ($D = 1.0$, $\sigma = 1.5$, $\nu = 0.5$, $\mu = 1.0$) plotted in red was the focus of the calibration.

D.3 LOB Calibration: ABC-PMC MADWE

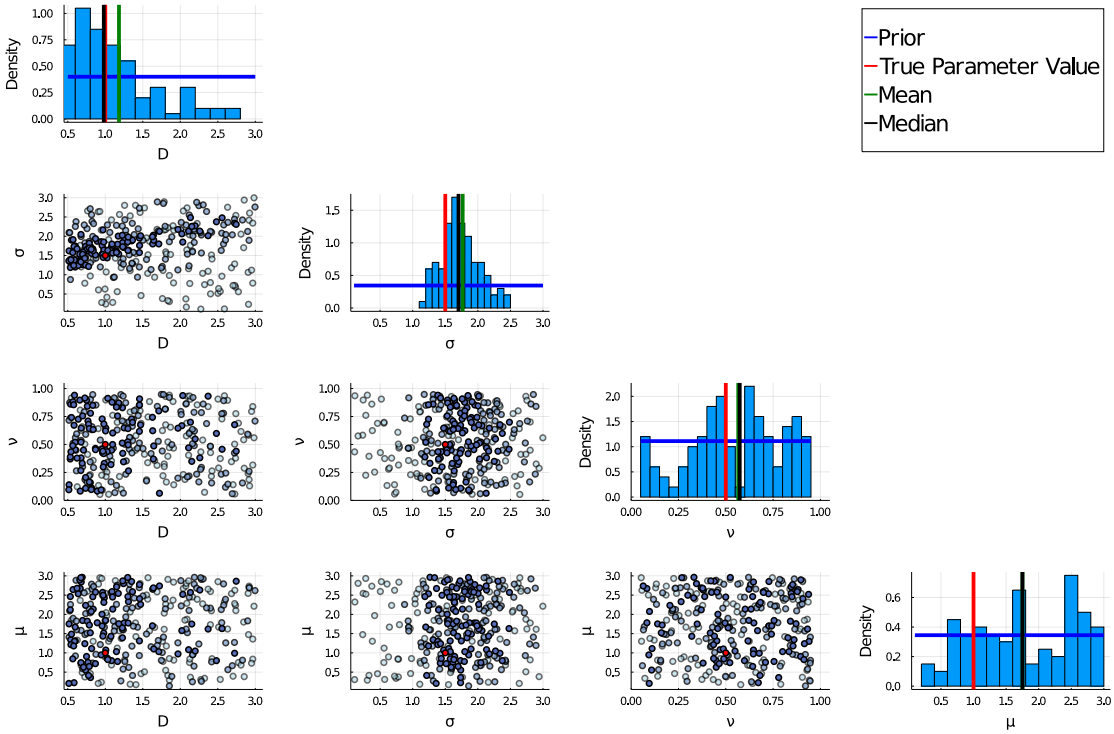


Figure 17: The free-parameter particle scatter plot and histogram matrix for the ABC-PMC calibration technique with the median absolute deviation weighted euclidean (MADWE) distance function. The LOB model with free-parameters ($D = 1.0$, $\sigma = 1.5$, $\nu = 0.5$, $\mu = 1.0$) plotted in red was the focus of the calibration. The iterations of the ABC-PMC method can be observed by the lighter blue dots, indicating early iterations, to darker blue dots, indicating later iterations. The histograms are created from the final iteration only.

E SLOB Calibration Figures

E.1 SLOB Calibration: ABC rejection BBWM

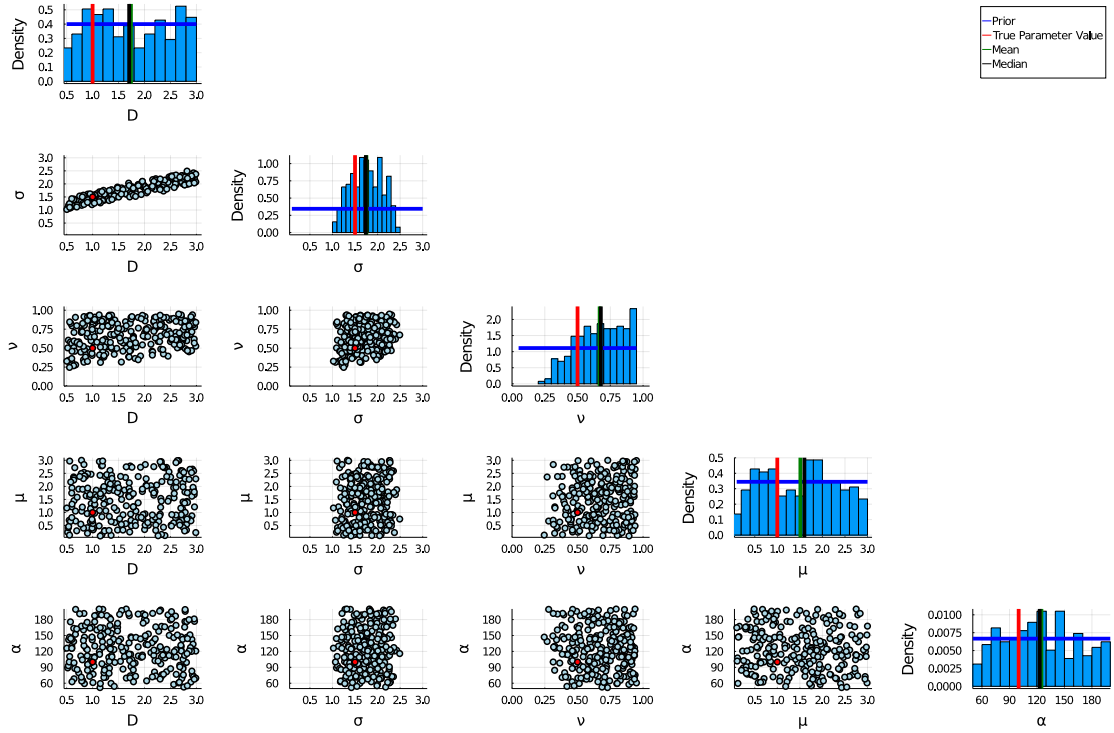


Figure 18: The free-parameter particle scatter plot and histogram matrix for the ABC rejection calibration technique with the block bootstrap weight matrix (BBWM) distance function. The SLOB model with free-parameters ($D = 1.0$, $\sigma = 1.5$, $\nu = 0.5$, $\mu = 1.0$, $\alpha = 100.0$) plotted in red was the focus of the calibration.

E.2 SLOB Calibration: ABC rejection MADWE

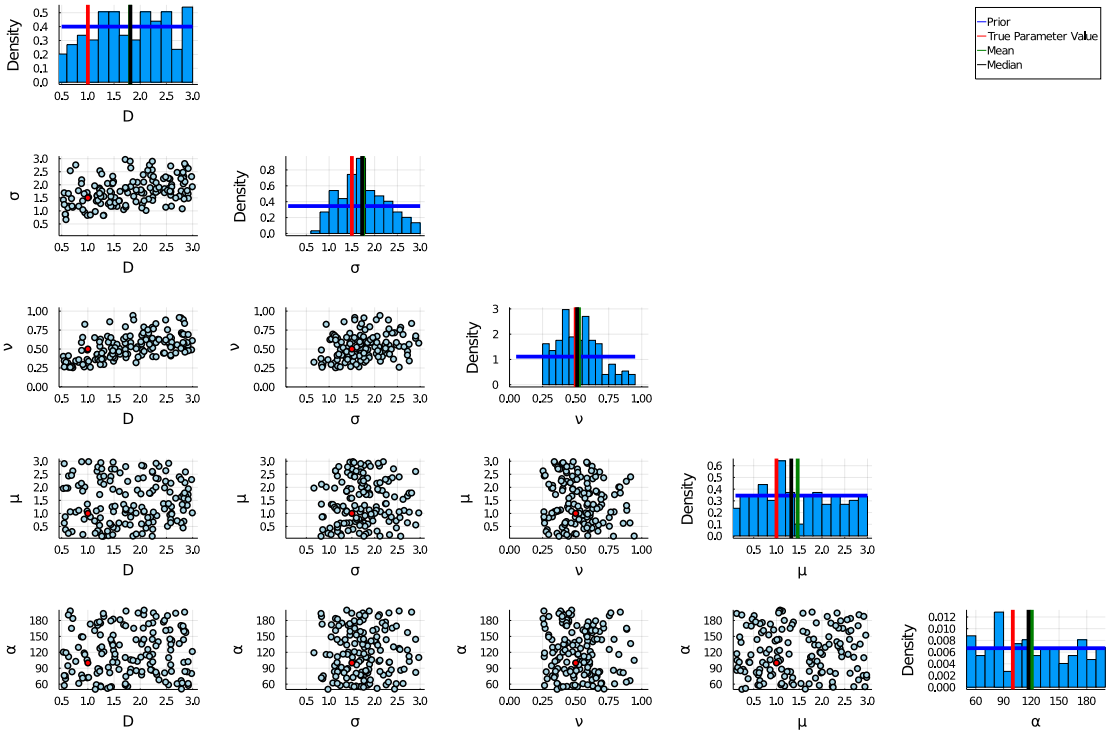


Figure 19: The free-parameter particle scatter plot and histogram matrix for the ABC-PMC calibration technique with the median absolute deviation weighted euclidean (MADWE) distance function. The SLOB model with free-parameters ($D = 1.0$, $\sigma = 1.5$, $\nu = 0.5$, $\mu = 1.0$, $\alpha = 100.0$) plotted in red was the focus of the calibration.

E.3 SLOB Calibration: ABC-PMC BBWM

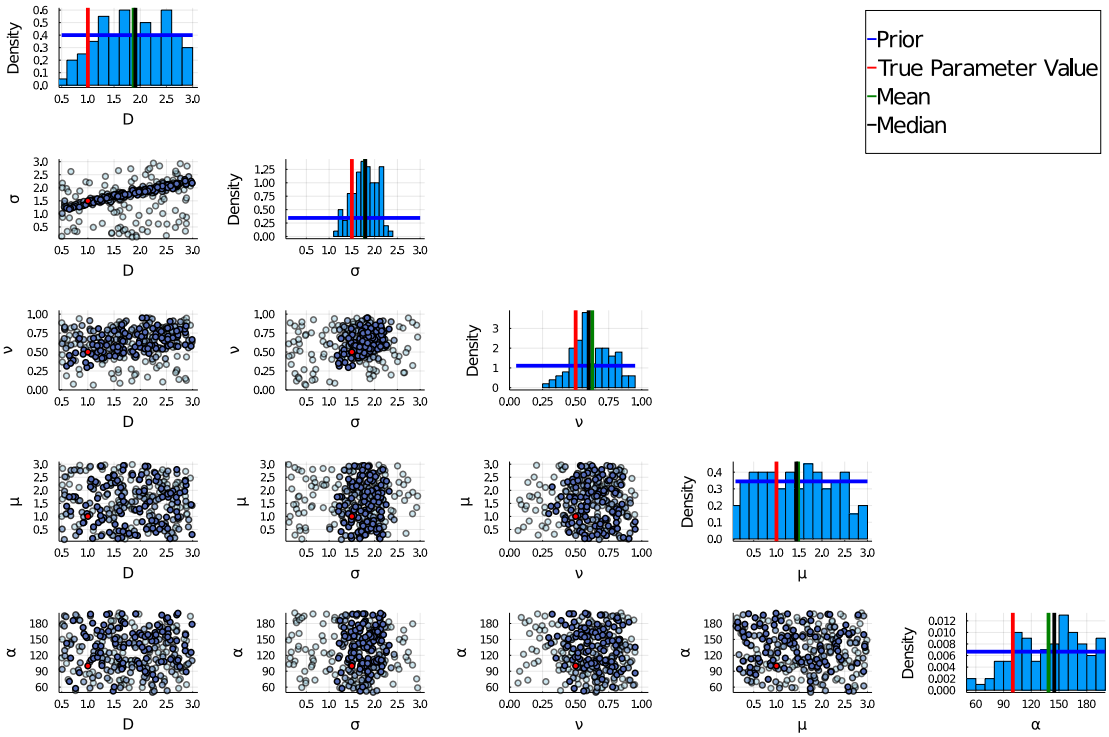
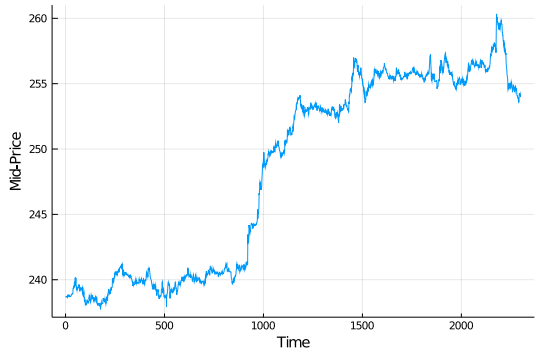


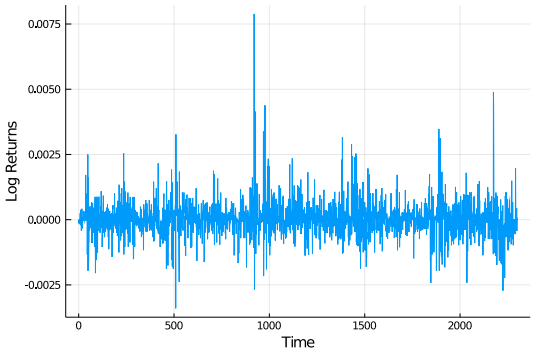
Figure 20: The free-parameter particle scatter plot and histogram matrix for the ABC-PMC calibration technique with the block bootstrap weight matrix (BBWM) distance function. The SLOB model with free-parameters ($D = 1.0$, $\sigma = 1.5$, $\nu = 0.5$, $\mu = 1.0$, $\alpha = 100.0$) plotted in red was the focus of the calibration. The iterations of the ABC-PMC method can be observed by the lighter blue dots, indicating early iterations, to darker blue dots, indicating later iterations. The histograms are created from the final iteration only.

F Stylised Facts

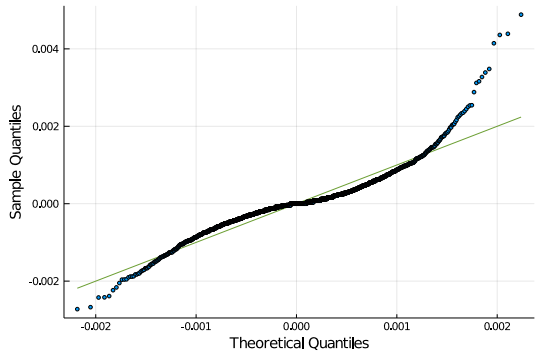
F.1 Market Data Stylised Fact Figures



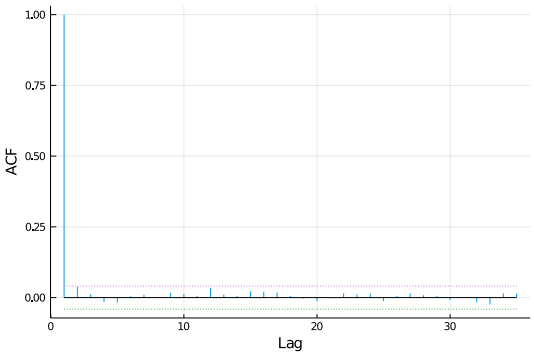
(a) Mid-price path



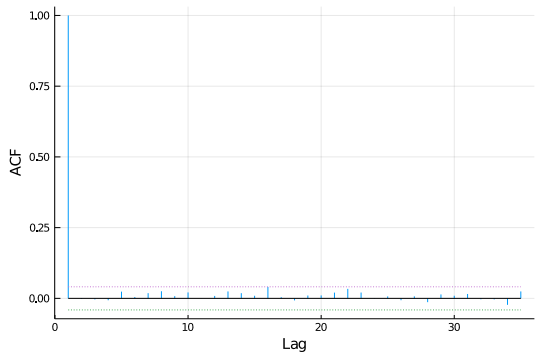
(b) Log returns



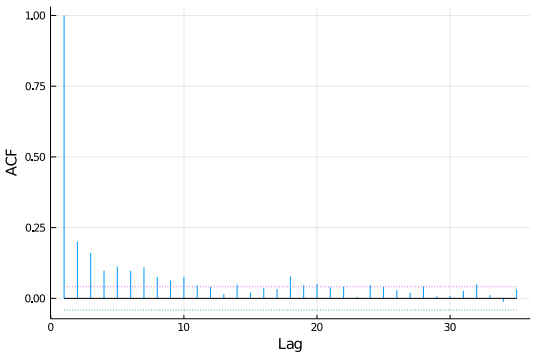
(c) QQ plot of log returns against Normal distribution



(d) Order flow (Tick Rule) auto-correlation function



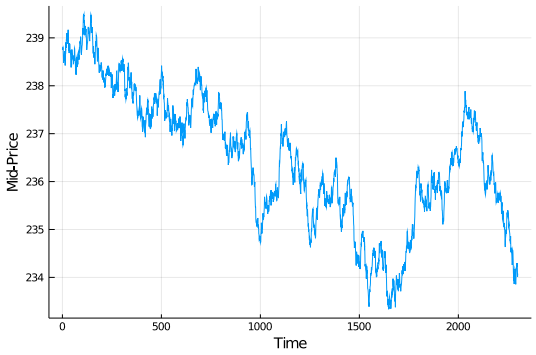
(e) Log returns auto-correlation function



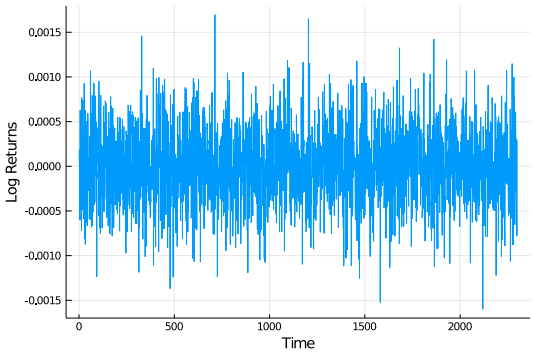
(f) Absolute log returns auto-correlation function

Figure 21: Stylised fact plots based on the AGLJ.J market data mid-price path.

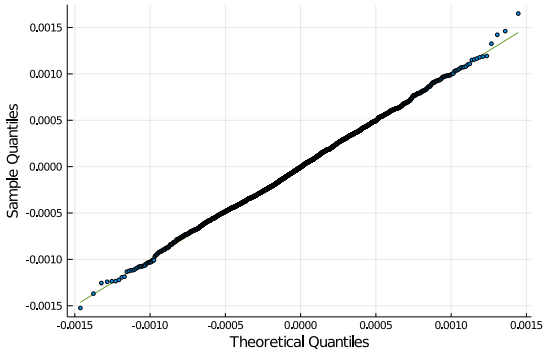
F.2 LOB Calibrated Model Stylised Fact Figures



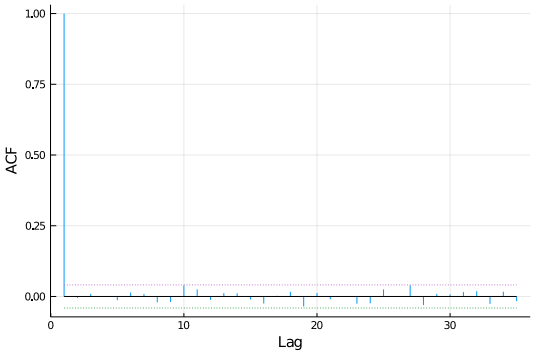
(a) Mid-price path



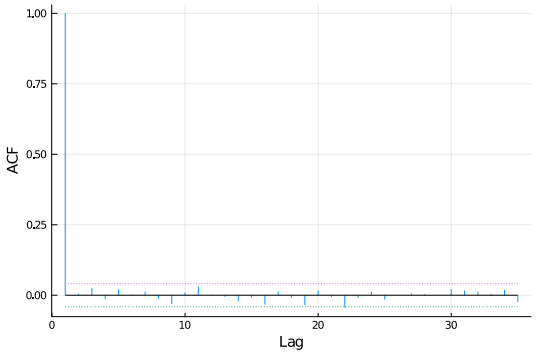
(b) Log returns



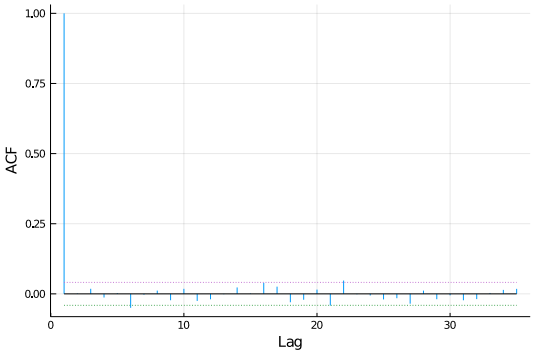
(c) QQ plot of log returns against Normal distribution



(d) Order flow (Tick Rule) auto-correlation function



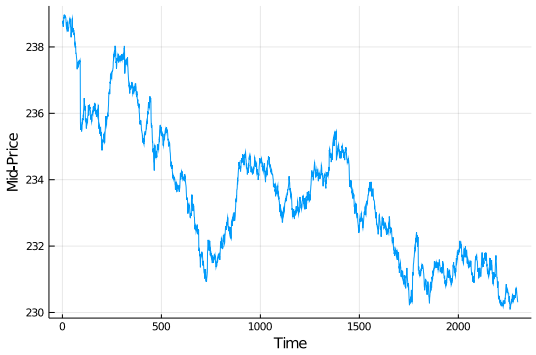
(e) Log returns auto-correlation function



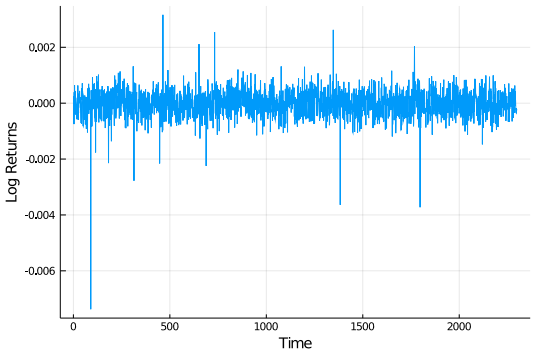
(f) Absolute log returns auto-correlation function

Figure 22: Stylised fact plots based on a single generated mid-price path from the ABC-PMC BBWM calibrated LOB model.

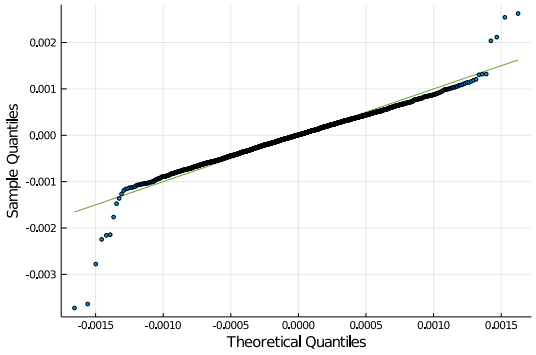
F.3 SLOB Calibrated Model Stylised Fact Figures



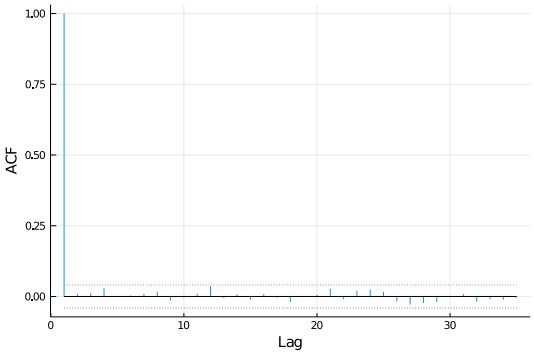
(a) Mid-price path



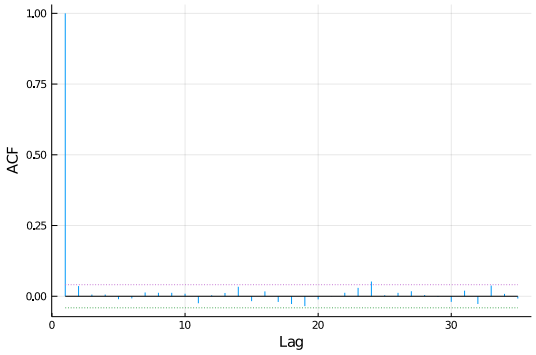
(b) Log returns



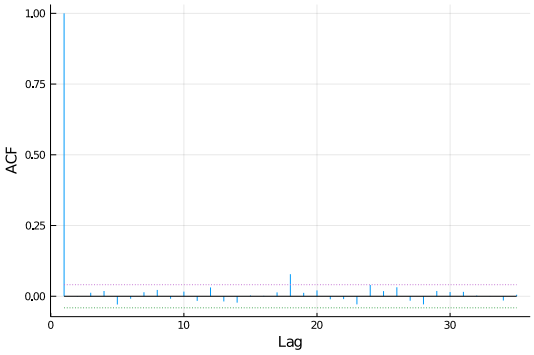
(c) QQ plot of log returns against Normal distribution



(d) Order flow (Tick Rule) auto-correlation function



(e) Log returns auto-correlation function



(f) Absolute log returns auto-correlation function

Figure 23: Stylised fact plots based on a single generated mid-price path from the ABC-PMC BBWM calibrated SLOB model.

G GitHub Repository

The Julia programming code for the models, calibration techniques and figures are stored in Github and cited in Zivahub with generated DOIs:

- Dissertation Scripts (Gant and Gebbie, 2020b)

- SequentialLOB.jl (Gant and Gebbie, 2020c)
- AdaptiveABC.jl (Gant and Gebbie, 2020a)
- StylizedFacts.jl (Gant and Gebbie, 2020d)