Msc Thesis in Applied Mathematics

A comparison of the limit order book and automated market maker

J.C. de Vries

September 2022

TU Delft

MSc thesis in Mathematics

A comparison of the Limit Order Book and Automated Marker Maker

J.C. de Vries

September 2022

to obtain the degree of Master of Science in Applied Mathematics at Delft University of Technology, to be defended publicly on September 14th, 2022.

The work in this thesis was carried out in the:

Financial Mathematics
Delft University of Technology

Student number	4594223	
Thesis committee:	Prof. Dr. A. Papapantoleon	TU Delft, Chair
	Dr. ir. J. H .M. Anderluh	TU Delft, Daily supervisor
	Dr. N. Parolya	TU Delft

Cover image: Mail & Guardian. (2021, July 14). [Stock exchange]. Where to Look for Forex Trading in South Africa. https://mg.co.za/special-reports/2021-07-14-where-to-look-for-forex-trading-in-south-africa/

Abstract

All investment firms in Europa must track the market quality of different exchanges and choose the best for their clients. Previous research has shown that a limit order book with additional market makers performs best in terms of market quality, followed by a pure limit order book and market maker. With the rise of cryptocurrencies, a new exchange called the automated market maker has appeared in the crypto markets. It is unclear how it compares to the limit order book regarding market quality. This thesis compares the automated market maker with the limit order book. We have built a simulation and tested seven different scenarios, varying in liquidity, information, price variability and fee. Six out of seven scenarios indicated that the automated market maker outperforms the limit order book.

Contents

Glossary

$\mathbb{N}$ The natural numbers, namely 1,2,3,... 11

best ask The lowest price for which any positive amount of assets can be bought directly. viii, 9, 10

best bid The highest price for which any positive amount of assets can be sold directly. viii, 9, 10

bid-ask spread The difference between the best ask and the best bid. This is often seen as the cost of a market trade. 10

daily dollar volume Total amount (represented in US dollars) of traders that occured in one day. 1

execution risk Risk than your order is never executed due factors like no available counterparty. 10

liquidity provider Any entity that provides liquidity in a market by either stating a price (range) at which he would like to buy or sell a predetermined amount of assets. When there is a contractual obligation, the therm market maker is usted instead. viii, 1, 9, 19

market capitalization Total value (in US dollars) of all outstanding shares. 1

market maker Any entity that makes a market by providing liquidity at a certain price (range). The term is often reserved for large entities that are regulated and contractually obligated to provide liquidity in the market. When there is no obligation, the term liquidity provider is used instead. viii, 1

market taker Any entity that takes liquidity by accepting offers from a liquidity provider or market maker. 9

midprice The average of the best ask and best bid. viii

spot price The price of an asset when an infinitely small portion is bought or sold directly. This often equals the midprice, but can also be computed differently. 21, 23–25

Acronyms

ADF Augemented Dickey-Fuller test 8

AMM Automated market maker 1–3, 7, 19, 20, 23, 24, 37, 40, 44, 47, 48, 51, 54, 58–61, 63, 64, 66, 69, 74, 76, 77

EMH Efficient Market Hypothesis 5, 6, 63

KPSS Kwiatkowski–Phillips–Schmidt–Shin test 8

LOB Limit order book 1–3, 8, 9, 13, 40, 44–46, 48, 51, 52, 58, 61, 63, 64, 66, 69, 74, 76, 77

LP liquidity pool 19, 20, 22–25, 47, 53

MiFID Markets in Financial Instruments Directive 1

1. Introduction

Any investment firm in Europe is required to 'take all reasonable steps to obtain, when executing orders, the best possible result for their clients taking into account price, costs, speed, likelihood of execution and settlement, size nature or any other consideration relevant to the execution of the order' by the Markets in Financial Instruments Directive (MiFID) [1]. In other words, any investment firm must trade according to their client's best interest by reducing costs where possible, decreasing waiting time and increasing execution likelihood. These goals are aggregated in market quality, defined as 'the ability to get your orders done subject to your best-execution requirements' [2]. Thus, an investment firm must track the market quality of the exchange they use.

The traditional stock markets have two main strands of exchanges. The first is the quote-driven market, in which a specialist like a market maker or dealer quotes the price he is willing to sell or buy at. Any market participant has to trade with the specialist and cannot set the price on his own. This exchange type is also called the price-driven market since the quoted price drives the exchange. A well-known example of a quote-driven market is the dealer market.

In the order-driven market, any market participant can propose or accept a deal. A market participant can, for instance, propose to sell a chosen number of shares for a specific price. However, the same market participant can also choose to sell his shares for the best offer. A well-known example is an exchange that works with a Limit order book (LOB). The exchange displays all non-matched trades in the LOB. New traders can either fill existing open orders or add a new one to the book.

Previous research has already measured and compared (several aspects of) market quality of the LOB and the market maker, the two main exchange structures. It is generally accepted that a combination of the LOB and market maker gives the best market quality, followed by the pure LOB and the market maker in last place [3]–[6]. However, in recent years another trading mechanism called the Automated market maker (AMM) has entered the cryptocurrency markets.

The AMM is neither a quote nor an order-driven market. Even though everyone can supply liquidity as in the order-driven market, the quoted prices are a function of the available liquidity and thus only indirectly influenced by the liquidity providers. Furthermore, the function only quotes one bid and ask price, which is more similar to a quote-driven market. Even though it is unclear which category the AMM falls into, the numbers suggest its popularity is rising in the cryptocurrency markets. Uniswap, the leading AMM platform, has seen an increase of almost 200% in their daily dollar volume in a little more than 12 months [7], [8]. Furthermore, 15,35% of the total market capitalization of cryptocurrencies takes place in decentralized finance, of which the AMM is an example, with a 4.8% increase in the last two months [9]–[11].

The AMM was initially proposed in the early 1990s to solve several issues caused by the (human) market makers. After the introduction of the LOB, the AMM disappeared until its reintroduction in 2018 in the cryptomarkets, where it solved several technical issues with the blockchain that the LOB faced [12]–[14]. Though several advantages and drawbacks have been empirically observed and theoretically shown, it is still unclear how well the AMM performs in market quality. Furthermore, it is unclear whether the increasing popularity of the AMM stems from technical superiority on the blockchain or the market quality, as this was never measured nor compared.

In this paper, we will compare the performance of the AMM with the LOB in terms of market quality. First, we introduce the minimal market requirements, the definition of market quality and several metrics in chapter 2. Furthermore, in the same chapter, we introduce the LOB and AMM and check whether they satisfy the basic requirements. Next, the simulation is described in chapter 3. After this, the simulation results and the outcome of the metrics are given in chapter 4. Finally, we conclude in chapter 5.

2. Theory

This chapter discusses the theory of markets and the LOB and AMM in particular. For this, we assume that every exchange trades assets against money, denoted by A and M respectively. First, we discuss the minimal requirements of a market in section 2.1. These requirements should be implemented in the market. After this, we discuss several desired properties in section 2.2. These properties can be measured and used to compare the two markets in chapter 4. Finally, we discuss how the LOB and AMM work, and how they meet the requirements in sections 2.3 and 2.4 respectively.

2.1. Market Requirements

In this section, it is described what is required for marketplace to function properly.

Requirement 1. *The given price should be the worst case price, when no other trade occurs in the mean time.*

The first requirement is transparency of price, meaning that you know a worst price at which you are guaranteed to trade. When you sell, this is a lower price, while it is an upper price when you buy. If a market doesn't have this characteristic, it functions like a lottery. Even though there still will be some interested parties, it cannot really be thought of as a exchange anymore.

Requirement 2. *Trades are made according to the agreed terms.*

The second requirements is the guarantee that the trade is made according to the conditions as soon as two parties agreed upon. If this were not the case, no trades can occur since the risks are very high.

Both requirements 1 and 2 are forced upon an exchange via legalization. Therefore, we assume that these are satisfied in any exchange.

Before we introduce the third and fourth requirement, we first give the pricing function of assets expressed in money in equation 2.1. Here, we denote the amount of assets paid and money received by A and M respectively. Therefore, $A, M > 0$ stands for a sell order while $A, M < 0$ is a buy order.

$$M(A) : \mathbb{R} \to \mathbb{R} \tag{2.1}$$

Requirement 3. *All gains should have positive costs, that is $M(A)$ should be an increasing function through the origin.*

Requirement 3 ensures that there is no arbitrage opportunity in the exchange. Since the price is increasing and it goes through the origin, we have $M(A) > 0 \iff A > 0$. Therefore, positive costs ($A > 0$) have positive gains ($M(A) > 0$) and visa versa. Furthermore, since the price is increasing, additional costs are guaranteed to give additional assets and via versa. The reason this in a requirement is twofold: First, the available liquidity in the market would dry up very fast as everything is 'bought' for non-positive costs, thus halting further trading very fast. Secondly, it is very unlikely that a counterparty for such a trade can be found as he only gives but doesn't receive anything. Therefore, no new liquidity will be added to the market and trading would halt.

Requirement 4. *The price does not decreases when a buy order is placed while it does not increase when a sell order is placed, that is*

$$\forall A_1, A_2, A_3 \in \mathbb{R} : \Big(0 < A_1 \leq A_2 \leq A_3 \vee A_1 \leq A_2 \leq A_3 \leq 0\Big)$$
$$\implies \frac{M(A_3) - M(A_2)}{A_3 - A_2} \leq \frac{M(A_2) - M(A_1)}{A_2 - A_1}$$

Requirement 4 is to create incentive to trade now rather than later. When $0 < A_1 \leq A_2 \leq A_3$, there are three consecutive sell market orders of size A_1, $A_2 - A_1$ and $A_3 - A_2$ respectively. Requirement 4 states that the price decreases as more assets are sold. Therefore, trading now rather than later is better for the trader. When $A_1 \leq A_2 \leq A_3 \leq 0$, there are three consecutive buy market orders of size A_3, $A_3 - A_2$ and $A_1 - A_3$ respectively[1]. Therefore, requirement 4 states that the price increases as more assets are bought. Therefore, trading now rather than later is better for the trader.

Theorem 2.1.1. *Requirement 4 holds if and only if $M(A)$ is concave on domains $\mathbb{R}_+ \cup \{0\}$ and $\mathbb{R}_- \cup \{0\}$.*

Proof. A function f is concave on interval I whenever

$$\forall x, y \in I, t \in (0,1) : f(x + t(y - x)) \geq f(x) + t(f(y) - f(x)). \tag{2.2}$$

Without lose of generality, assume $A_1 < A_3$. Furthermore, denote $A_2 = A_1 + t(A_3 - A_1)$. Then we have to prove

$$\forall t \in (0,1) \wedge A_1 \times A_3 \geq 0 : M(A_2) \geq M(A_1) + t(M(A_3) - M(A_1)).$$

Here, we require $A_1 \times A_3 \geq 0$ such that they have the same sign and we are on either domain for which we have to prove concavity.

$$\frac{M(A_3) - M(A_2)}{A_3 - A_2} \leq \frac{M(A_2) - M(A_1)}{A_2 - A_1}$$
$$\equiv \frac{M(A_3) - M(A_2)}{(1-t)(A_3 - A_1)} \leq \frac{M(A_2) - M(A_1)}{t(A_3 - A_1)}$$

[1]The order is now reversed, since the smaller number are of larger magnitude.

$$\equiv t(A_3 - A_1)\big(M(A_3) - M(A_2)\big) \leq (1-t)(A_3 - A_1)\big(M(A_2) - M(A_1)\big)$$
$$\equiv t\big(M(A_3) - M(A_2)\big) \leq (1-t)\big(M(A_2) - M(A_1)\big)$$
$$\equiv tM(A_3) + (1-t)M(A_1) \leq M(A_2)$$

□

Since requirement 4 holds if and only if the pricing formula is concave, we can test for this instead.

2.2. Market Quality

Market quality is ”the ability to get your orders done subject to your best-execution requirements” [2]. Therefore, market quality considers traders’ preferences, like price, transaction speed and execution risk. The higher the quality of the market, the more the demands of the different traders are met. High market quality is thus not a requirement but a contraction of desired properties. The first is called price discovery, which is the need for information transparency or, said differently, the incorporation of all relevant information in the market price. When an exchange has efficient price discovery, meaning that prices are updated as fast as possible, there is little to no information gap between different traders. Since all traders have the same knowledge and are thus on equal footing, there is a sense of ’fairness’. More importantly, this property makes it possible for a trader to weigh his trading preferences accurately. Furthermore, it gives some insight into the feasibility of the price aspect of the execution requirements. The second property, market liquidity, is defined as ’the degree to which an investor can buy or sell an asset quickly without incurring large transaction costs or exerting a material effect on the asset’s price’ [15]. Market liquidity covers the feasibility of execution requirements in terms of risk, time and costs. Other more subjective best-execution requirements, like the ease of use, are difficult to measure objectively and thus will not be considered in this thesis. They are, however, part of market quality and desired properties. In the following sections, we will discuss information incorporation and liquidity, how they are related, and the metrics used to measure market quality.

2.2.1. Information incorporation

One of the key aspects of a high-quality market is the speed at which an exchange incorporates new information. The process is called price discovery and has been defined in many ways. Nie [16] summarizes the different definitions as ’a dynamic process to reach a state of equilibrium with the rapidly adjusting market prices to replace the old equilibrium with the new one through new information.’ A market (or exchange) that encapsulates all relevant information about the asset in the current price is called efficient, and price discovery is the process that takes the market to its efficient state. Fama [17] first introduced the Efficient Market Hypothesis (EMH), which states that all markets are efficient and rational. When a market is efficient, it has already incorporated all relevant information in the current asset price. Therefore, all price movements are random and independent of the past since there is no new information on which they can be based. The price follows a random walk (for short time horizons) when an exchange is in its efficient state [18].

Therefore, the time series of the price can be split into two parts. First, there are times when there is no new information, and the market should be in equilibrium. In those times, the prices should behave like a random walk, meaning increments should be uncorrelated. The second part contains the times when not yet all information is incorporated. A positive price correlation is expected since the price should move to the new price as fast as possible without backwards steps. Therefore, the price should continuously decrease or increase towards the new price. Furthermore, the time, the number of trades and the volume to reach the new price can be measured for the speed of information incorporation and adaptation of the new price in the market.

2.2.2. Market liquidity

Market liquidity is defined as 'the degree to which an investor can buy or sell an asset quickly without incurring large transaction costs or exerting a material effect on the asset's price' [15]. A market is thus liquid when demand and supply are high and the bid-ask spread is low, which results in low transaction costs. In general, there are three categories of metrics used to measure market liquidity [19].

The first category consists of volume-based liquidity measures. These measures were presented in the early stages and often easily determined. An example is the daily volume. Fleming [20] claims that it is not a good measure since it only weakly correlates with liquidity, as can be shown quickly for the daily volume. On one side, a larger volume seems to point towards more liquidity. However, Karpoff [21] has also shown that the volume positively correlates to price volatility, which raises the transaction costs and thus reduces liquidity. This example illustrates that a definitive conclusion is hard to draw using a volume-based liquidity measure. Other drawbacks of the volume-based liquidity measures are that they fail to distinguish between a state of temporary and persistent illiquidity, are only based on the past, overestimate (underestimate) big (small) transactions and are only an indirect measure of liquidity [15], [21].

The second category are the price-variability indices, which include measures that derive market liquidity directly from price behaviour, either by looking at the variance or by looking at statistics. The two main methods that look at the variance are the liquidity ratio of Marsh and Rock [22] and the variance ratio. It is questionable whether something new is measured since the assumptions made by these measures are similar to those of the EMH. The second category of statistical measures is primarily based on event studies, which do not uniquely define how to measure liquidity [19]. Furthermore, the measures overlap with section 2.2.1 since they do not solely measure liquidity but market quality. Therefore, price-variability indices cannot give a conclusive answer regarding market liquidity without additional support.

The last category are the transaction cost measures, like the bid-ask spread. Transaction costs can be seen as both the price for a quick transaction and the compensation for liquidity provision. Therefore, transaction costs are deeply connected with market liquidity. The bid-ask spread consists mainly of costs due order processing, adverse information, and inventory [19]. However, not all costs are relevant to measure. The exchange structure does not cause order processing costs since they stem from implementation or additional fees for using the exchange. In the simulation, only costs caused by the exchange structure are relevant to measure. Therefore, we will not consider costs like additional fees or waiting times due to implementation in the simulation. The second cost caused by adverse information is already measured by measuring market efficiency (see section 2.2.1). Therefore, this does not have to be measured anymore. Only transaction costs caused by inventory considerations are not yet measured and are still very informative for liquidity. The bid-ask spread is such a measure. Fleming [20] shows that the bid-ask spread, complemented by the quote size, is the best indicator for market liquidity. The quote size is the amount that can be bought (or sold) at the best price and indicates how variable the price is. According to Hu *et al.* [15], larger volatilities indicate lower liquidity because 'prices tend to move more when there is less depth to execute large orders'. A larger quote size thus indicates lower volatility and higher liquidity.

However, the quote size is only reasonable to use when the prices and quantities offered are discretized, which is not the case for the AMM. Even though the price and quantities are discretized in computer models, the theory behind the automated market maker implies a price that continuously changes. The best ask, best bid and quote size are thus not well defined since the price changes with each trade regardless of size. Therefore, we generalize the bid-ask spread and quote size to look at one unit. The unit bid-ask spread is the price difference when buying and selling one unit of the asset. The unit quote size is the number of assets that must be sold (bought) to decrease (increase) the price by one unit.

2.2.3. Interaction of Market Efficiency and Market liquidity

Bernstein [23] notes that liquidity and efficiency are not always compatible in every situation. We can easily see this by looking at two examples.

First, we look at a situation where the true price is constant. The market price should stay at this constant true price for perfect price efficiency. Even more so, the bid-ask spread should be (near) zero. However, this perfect efficiency prohibits liquidity. Noise traders are traders that push the price away from equilibrium, either because they are misinformed or because they want to obtain a position. By doing this, noise traders provide depth, breadth and resiliency [24], since informed traders rush to restore the price. Thus, noise traders provide liquidity in the market by decreasing the price efficiency. A perfectly efficient market will thus decrease trading since there is no place for noise traders.

The second situation that displays the tension between liquidity and price efficiency is a price change due to new information. A quick price change is desired when new information is introduced, but a lot of liquidity might prevent this. For instance, many traders are needed whenever there is a lot of liquidity around the current price. In that case, the metric of quote size would indicate a very liquid market, but efficiency would decrease. If the market were less liquid, equilibrium would need a smaller traded volume.

Bernstein [23] therefore concludes that liquidity measures hold more weight when there is no new information. This suggests a simulation in which more weight is given to the liquidity measure when the true price is kept static. In contrast, price efficiency is more important when new information becomes available.

2.2.4. Metrics

There are two things the metrics for market quality should measure: information incorporation and liquidity in the market. As explained in section 2.2.3, we value the metrics of information incorporation more when new information is introduced, while metrics for liquidity are valued more when the exchange is somewhat in equilibrium. To precisely separate these parts, we define the time when new information is introduced until the new price has been reached (and possibly surpassed) as the time for price discovery. The remaining parts are assumed to be in equilibrium.

For information incorporation, the measures already mentioned in section 2.2.1 will be used. The first metric is to test whether or not the price behaves like a random walk. The Augemented Dickey-Fuller test (ADF) [25] tries to reject the null hypothesis that the time series is non-stationary and thus a random walk. Next to other values, it returns a p-value at a 5% significance level. Therefore, the ADF concludes that the series behaves like a random walk whenever the p-value exceeds 0.05. The ADF is often complemented by a Kwiatkowski–Phillips–Schmidt–Shin test (KPSS)[26], which does the exact opposite: It tests the null hypothesis that the series is stationary and thus not a random walk. The tests should indicate a random walk when the exchange is in equilibrium and indicate otherwise when new information is integrated. However, price-increment type tests like these are often used when there is just one single market[27]. Metrics of relative speed are used whenever two or more markets are compared [18]. Therefore, the time, volume and number of trades during price discovery are used as a metric to compare two exchanges. On the other hand, the random walk tests are used as validity and price incorporation checks.

As suggested in section 2.2.2, the unit bid-ask spread and the unit quote will be used as metrics for market liquidity.

Another metric that does not fall under either market liquidity or information incorporation will be used. It does, however, fall under market quality. We will call this metric the completion percentage, representing trader satisfaction. Simulation data includes the original goal of the trader, unlike real-world data. The goal is to buy assets or money (which is the same as selling stocks). Let us assume that the trader wanted to receive X assets (or money) and has received X' within his time frame. The completion percentage of this trader is then

$$C = \frac{X'}{X}. \tag{2.3}$$

Therefore, the trader is completely satisfied whenever $C >= 1$, while his goal could not be reached when $C < 1$. The market is thus of higher quality when the completion percentages of the traders are higher. For this, we look at the mean and the variance in the completion percentages. To see why the variance matters, let us take the following example: Two markets both have an average completion percentage of one. The first has zero variance, while the second has variance greater than zero. It should be clear that the first market is of perfect quality since everybody is completely satisfied, while there are unsatisfied traders in the second exchange. Therefore, the second exchange is of lesser quality.

2.3. Limit order book

In this section, we will introduce the Limit order book (LOB). Currently, the LOB is seen as the best pure trading mechanism and is often employed in traditional finance. In section 2.3.1, the LOB is introduced, couples with some examples for better understanding. After this, the LOB is formally defined in section 2.3.2. Finally, in section 2.3.3, it is shown how the LOB satisfies the base requirements described in 2.1.

2.3.1. Overview

A Limit order book (LOB) is a (digital) record of outstanding buy and sell orders that have not yet been matched to a second party. Alternatively, the LOB is a collection of prices for which a determined quantity can be bought/sold. The outstanding orders in the book are called limit orders since the order is both limited in price and quantity.

Example - A basic limit order book

An example of an order book is given in figure 2.1. In this example, two assets can be sold for €7 each, while the third to the tenth asset only yield €6. At the same time, an asset can be bought for €10, while an additional asset will cost €11.

Figure 2.1.: An example of a limit order book

There are three actions a market participant can take. The first action is a market order, in which he buys or sells the asset directly at the best ask or best bid, respectively. A market participant takes this action if time is of the essence.

Whenever a market participant has time, he can place a limit order. In this case, he wants a better price than currently available via a market order and has some time to wait for a counterparty. He decides on a price and quantity he wants to buy (or sell) and submits a limit order, which is placed in the order book. By doing this, the market participant raises the liquidity because he makes it easier for new market participants to find a counterparty. Furthermore, by adding to the stock of the order book, the order book becomes more resilient to large orders meaning that the price shifts due to large orders are dampened.

The third, and last action, is a limit order cancellation. Only market participants with an outstanding limit order can take this action.

From these three actions, a natural division of market participants follows. The first group are the market takers, which place market orders and thus take liquidity. The second group are the liquidity providers, who provide liquidity by placing limit orders. It should be noted that liquidity providers can also reduce liquidity when they place a limit order which can be matched immediately[2] or cancel a limit order.

[2] In this case, the liquidity taker can be seen as a market taker and will be categorized as such.

Example continued - Placing orders

We take the previous setting with an order book in the state as shown in figure 2.1. Now assume that a market maker wants to buy three assets. He will have to pay €32, and the order book will be updated as shown in figure 2.2.

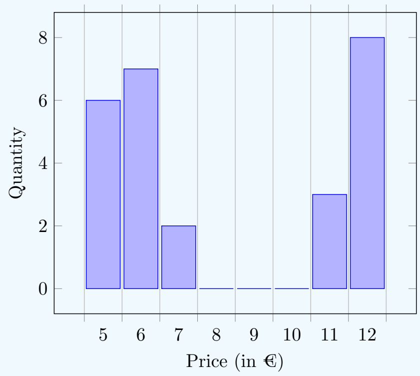


Figure 2.2.: The order book after a market participant bought 3 assets

The best ask has risen from €10 to €11. Suppose another market participant owns three assets and believes it is still profitable to sell them for €10, €11 and €11, respectively. In that case, he can place two limit orders (one of one asset for €10 and one of two assets for €11) to return to the original state as depicted in figure 2.1.

A liquidity provider gets a better price than he would have when he placed a market order at that moment. Sometimes, the bid-ask spread is seen as the costs (or merit in the case of the liquidity provider) of liquidity. However, there is a risk that his order is never executed, named the execution risk. The liquidity provider can influence his execution risk through the price he provides since this partly determines his place in the queue. When the liquidity provider provides a worse price than the market, he would be placed more in the back of the queue. A liquidity provider performs an undercut when he places an order with a better price than the market provides. Since his price is better than all other prices, he will be first in line to be matched with a market order, and his execution risk decreases. Another benefit is that the time to execution also decreases. An undercut is sometimes also called jumping the queue.

The LOB has several settings that can be altered per exchange. The first is the order in which limit orders are matched. It is clear from the example that a better price puts the order earlier in the queue, but it does not describe the order of limit orders with the same price. In practice, the first-in-first-out principle is used. Orders are thus ordered by price and then by time in the order book.

The second setting is the tick size which is the step size of the price. All wielded prices have to be a multiple of the tick size. For example, if the tick size were €0.01, a price of €0.91 would be possible, whereas a price of €1.231 is not. It should be noted that the asset sold can also have a tick size representing the step size of the quantity bought. In US markets, stocks are often bought in bundles of one hundred, but this is not strictly necessary. The tick size also limits the possibilities of an undercut since there are fewer steps between the best bid and best ask. Additionally, a larger tick size negatively influences the benefit of the undercut.

Example - Tick size and undercut

Assume a LOB with the order book in the state of figure 2.1 and tick size of €0.50. If a market participant wanted to sell an asset in little time, he could attempt an undercut to jump the queue. To do this, we would make a limit order for one asset at €9.50 to make him the first in the queue. However, if he wanted to be sure that nobody could perform an undercut on him, we would instead use the price of €8.50. Nobody could perform an undercut since this would require a price of €8.00, which can immediately be matched against a limit buy order. Since the market participant is ensured to be first in line, it would only take one market order that buys an asset to sell his asset. When there are many transactions on the exchange, he will sell for €8.50 instead of €8.00 in a relatively short time.

2.3.2. Formal definition

This section formally defines the LOB after several components are presented.

Definition 2.3.1. *A limit order is a tuple* (p, q, t) *such that* $p, q, t \in \mathbb{N}$*. Here,* p *is the price of the asset expressed in another asset,* q *is the number of assets, and* t *is the discretized time stamp*[3]*. Furthermore, we define* $(p, 0, t)$ *as the empty limit order with the property that* $O \cup (p, 0, t) = O$*, where* O *is a list of limit orders.*

In the definition of a limit order, it is unclear whether a sell or buy order is meant. As will be shown, it will be clear from the context whether it is a sell or buy order. Furthermore, even though the domain of natural numbers is taken for each variable, it can be generalized to take smaller steps in all variables.

Definition 2.3.2. *The sell side is a an ordered finite list* $(s_1, s_2, ..., s_n)$ *such that*

- $\forall i : s_i = (p_i, q_i, t_i)$ *is a limit order representing a buy order*[4]
- $\forall i, j : i < j \implies (p_i > p_j \vee (p_i = p_j \wedge t_i < t_j))$
- $\forall i, j : t_i = t_j \implies i = j$

Definition 2.3.3. *The buy side is a an ordered finite list* $(b_1, b_2, ..., b_n)$ *such that*

- $\forall i : b_i = (p_i, q_i, t_i)$ *is a limit order representing a sell order*
- $\forall i, j : i < j \implies (p_i < p_j \vee (p_i = p_j \wedge t_i < t_j))$
- $\forall i, j : t_i = t_j \implies i = j$

Definition 2.3.4. *A LOB is a tuple* (S, B, F) *such that*

- $S = (s_1, ..., s_n)$ *is a sell side, representing all limit orders that buy the asset.*

[3] t can also be seen as an increasing identification number

[4] This name sell side might be a bit confusing when there are only limit buy orders in it. However, seen from the market participants' perspective, one can directly sell his assets on this side of the book, hence the name.

- *$B = (b_1, ..., s_n)$ is a buy side, representing all limit orders that sell the asset.*
- *$p_{s1} > p_{b1}$, where $s_1 = (p_{s1}, q_{s1}, t_{s1})$ and $b_1 = (p_{b1}, q_{b1}, t_{b1})$*
- *$F = f_1, f_2, ...$ is the set of transformations such that $\forall f \in F : f(q, p, t, S, B) = (S', B', T')$, where (S', B', F) is a LOB again.*

Below, we give all the basic transformations possible for a LOB. All other possible transformations can be seen as a combination of the others. Here, we define t_n as the current time[5] and $Q_b = \sum_{(p_i,q_i,t_i)\in B} q_i$ and $Q_s = \sum_{(p_i,q_i,t_i)\in S} q_i$ as the total amount you can respectively buy and sell in the LOB. Finally, we denote $p_{bj}, q_{bj}, t_{bj}, p_{sj}, q_{sj}$ and t_{sj} as the price, quantity and time index of jth order on the buy and sell side of the LOB.

- $Q_b \geq q_b > 0, q_s = 0, p = -1$: this is a market order that wants to buy q_b assets. From this, we derive all components of the new LOB as follows:
  - First, we find index j such that $\sum_{i=1}^{j-1} q_{bi} \leq q_b < \sum_{i=1}^{j} q_{bi}$
  - $B' = B \setminus \bigcup_{i=1}^{j} b_i \cup (p_{bj}, \sum_{i=1}^{j} q_{bi} - q_b, t_{sj})$
  - $S' = S$
- $q_b = 0, Q_S \geq q_s > 0, p = -1$: This is a market order that wants to sell q_s assets. From this, we derive all components of the new LOB as follows:
  - First, we find index j such that $\sum_{i=1}^{j-1} q_{si} \leq q_s < \sum_{i=1}^{j} q_{si}$
  - $B' = B$
  - $S' = S \setminus \bigcup_{i=1}^{j} s_i \cup (p_{sj}, \sum_{i=1}^{j} q_{si} - q_s, t_{bj})$
- $q_b > 0, q_s = 0, p < p_{b1}$: This is a limit buy order that wants to buy q_b assets. From this we derive all components of the new LOB as follows:
  - $B' = B$
  - $S' = S \cup (p, q_b, t_n)$
- $q_b > 0, q_s = 0, p \geq p_{b1}$: This is a limit buy order that can directly be matched and thus converted to a market buy order.
- $q_b = 0, q_s > 0, p > p_{s1}$: This is a limit sell order that wants to sell q_s assets. From this, we derive all components of the new LOB as follows:
  - $B' = B \cup (p, q_s, t_n)$
  - $S' = S$
- $q_b = 0, q_s > 0, p \leq p_{s1}$: This is a limit sell order that can directly be matched and thus converted to a market sell order.
- $t < t_n, (p, q, t) \in B$: This represents the cancellation of a limit sell order, uniquely identified by t. From this, we derive all components of the new LOB as follows:
  - $B' = B \setminus (p, q, t)$
  - $S' = S$

[5] If t is seen as identifier, then t_n can be seen as the identifier the next order would get.

- $t < t_n, (p, q, t) \in S$: This represents the cancellation of a limit buy order, uniquely identified by t. From this, we derive all components of the new LOB as follows:
  - $B' = B$
  - $S' = S \setminus (p, q, t)$

2.3.3. Requirements

In this section, it will be shown that LOB satisfied the requirement posed on an exchange in section 2.1.

It is clear to see that the requirement 1 of transparency is satisfied. Although traders may hide their orders, this can only give a better price than the shown price.

Requirement 2 can only be enforced through legalization. However, a case can be made that the price and quantity remain the same, since neither party would otherwise agree to the exchange.

We know that requirements 3 and 4 are satisfied when several mathematical properties hold for the pricing function. We will first define the pricing function. For this, assume A assets are sold to receive M money. Furthermore, assume that limit orders with the same price have been taken together. Then we have

$$M(A) = \begin{cases} \sum_{i=1}^{j-1} p_{si} * q_{si} + p_{sj} * \left(A - \sum_{i=1}^{j-1} q_{si}\right) & \text{s.t. } \sum_{i=1}^{j-1} q_{si} \leq \quad A < \sum_{i=1}^{j} q_{si} \quad \wedge \quad A \geq 0 \\ -\sum_{i=1}^{j-1} p_{bi} * q_{bi} + p_{bj} * \left(A + \sum_{i=1}^{j-1} q_{bi}\right) & \text{s.t. } \sum_{i=1}^{j-1} q_{bi} \leq -A < \sum_{i=1}^{j} q_{bi} \quad \wedge \quad A < 0 \end{cases}. \tag{2.4}$$

Sometimes, the alternate form in equation 2.5 is used in the proofs.

$$M(A) = \begin{cases} \sum_{i=1}^{j} p_{si} * q_{si} - p_{sj} * \left(\sum_{i=1}^{j} q_{si} - A\right) & \text{s.t. } \sum_{i=1}^{j-1} q_{si} \leq \quad A < \sum_{i=1}^{j} q_{si} \quad \wedge \quad A \geq 0 \\ -\sum_{i=1}^{j} p_{bi} * q_{bi} + p_{bj} * \left(\sum_{i=1}^{j} q_{bi} + A\right) & \text{s.t. } \sum_{i=1}^{j-1} q_{bi} \leq -A < \sum_{i=1}^{j} q_{bi} \quad \wedge \quad A < 0 \end{cases}. \tag{2.5}$$

In the remaining of the section, we proof that the pricing formula is increasing and concave with theorems 2.3.9 and 2.3.11. In order to do this, we prove several lemmas first.

Lemma 2.3.5. *Whenever $0 \leq A_1 < A_2$, we have $j_1 \leq j_2$, where j_i is the index in equation 2.4.*

Proof. By definition of formula 2.4, we know that $\sum_{i=1}^{j_1-1} q_{si} \leq A_1 < A_2 < \sum_{i=1}^{j_2} q_{si}$. There are two missing terms, namely $\sum_{i=1}^{j_1} q_{si}$ and $\sum_{i=1}^{j_2-1} q_{si}$, which still need to be ordered. There are three possible orders:

- $\sum_{i=1}^{j_1-1} q_{si} = \sum_{i=1}^{j_2-1} q_{si} \leq A_1 < A_2 < \sum_{i=1}^{j_1} q_{si} = \sum_{i=1}^{j_2} q_{si}$,

- $\sum_{i=1}^{j_1-1} q_{si} \leq A_1 < \sum_{i=1}^{j_1} q_{si} \leq \sum_{i=1}^{j_2-1} q_{si} \leq A_2 < \sum_{i=1}^{j_2} q_{si}$,
- $\sum_{i=1}^{j_1-1} q_{si} \leq A_1 < \sum_{i=1}^{j_2-1} q_{si} < \sum_{i=1}^{j_1} q_{si} < A_2 < \sum_{i=1}^{j_2} q_{si}$.

The first case implies $j_1 = j_2$. The second case implies $j_1 \leq j_2 - 1$ and thus $j_1 < j_2$. The third case is a contradiction, since it implies $j_1 - 1 < j_2 - 1 < j_1$. Since the j_i's are indexes, they are integers and there is no integer number between two consecutive integers. Therefore $j_1 - 1 < j_2 - 1 < j_1$ is a contradiction and the third case cannot happen.

Thus we have proven $\forall 0 \leq A_1 < A_2 : j_1 \leq j_2$. □

Lemma 2.3.6. *Whenever $A_2 < A_1 \leq 0$, we have $j_1 \leq j_2$, where j_i is the index in equation 2.4.*

Proof. By definition of formula 2.4, we know that $\sum_{i=1}^{j_1-1} q_{bi} \leq -A_1 < -A_2 < \sum_{i=1}^{j_2} q_{bi}$. There are two missing terms, namely $\sum_{i=1}^{j_1} q_{bi}$ and $\sum_{i=1}^{j_2-1} q_{bi}$, which still need to be ordered. There are three possible orders:

- $\sum_{i=1}^{j_1-1} q_{bi} = \sum_{i=1}^{j_2-1} q_{bi} \leq A_1 < A_2 < \sum_{i=1}^{j_1} q_{bi} = \sum_{i=1}^{j_2} q_{bi}$,
- $\sum_{i=1}^{j_1-1} q_{bi} \leq A_1 < \sum_{i=1}^{j_1} q_{bi} \leq \sum_{i=1}^{j_2-1} q_{bi} \leq A_2 < \sum_{i=1}^{j_2} q_{bi}$,
- $\sum_{i=1}^{j_1-1} q_{bi} \leq A_1 < \sum_{i=1}^{j_2-1} q_{bi} < \sum_{i=1}^{j_1} q_{bi} < A_2 < \sum_{i=1}^{j_2} q_{bi}$.

The first case implies $j_1 = j_2$. The second case implies $j_1 \leq j_2 - 1$ and thus $j_1 < j_2$. The third case is a contradiction, since it implies $j_1 - 1 < j_2 - 1 < j_1$. Since the j_i's are indexes, they are integers and there is no integer number between two consecutive integers. Therefore $j_1 - 1 < j_2 - 1 < j_1$ is a contradiction and the third case cannot happen.

Thus we have proven $\forall A_2 < A_1 \leq 0 : j_1 \leq j_2$. □

Lemma 2.3.7.

$$M(A_2) - M(A_1) = \begin{cases} p_{sj_2}(A_2 - A_1) & 0 \leq A_1 < A_2 \wedge j_1 = j_2 \\ p_{bj_2}(A_2 - A_1) & A_1 < A_2 \leq 0 \wedge j_1 = j_2 \end{cases} \tag{2.6}$$

Proof. Assume $0 \leq A_1 < A_2$ and $j_1 = j_2$. Then we have

$$\begin{aligned} M(A_2) - M(A_1) &= \sum_{i=1}^{j_2} p_{si} * q_{si} - p_{sj_2} * (\sum_{i=1}^{j_2} q_{si} - A_2) - \sum_{i=1}^{j_1} p_{si} * q_{si} + p_{sj_1} * (\sum_{i=1}^{j_1} q_{si} - A_1) \\ &= \sum_{i=1}^{j_2} p_{si} * q_{si} - p_{sj_2} * (\sum_{i=1}^{j_2} q_{si} - A_2) - \sum_{i=1}^{j_2} p_{si} * q_{si} + p_{sj_2} * (\sum_{i=1}^{j_2} q_{si} - A_1) \\ &= -p_{sj_2} * (\sum_{i=1}^{j_2} q_{si} - A_2) + p_{sj_2} * (\sum_{i=1}^{j_2} q_{si} - A_1) \\ &= p_{sj_2} A_2 - p_{sj_2} A_1 \\ &= p_{sj_2}(A_2 - A_1) \end{aligned}$$

The proof for $A_1 < A_2 \leq 0$ and $j_1 = j_2$ follows along the same lines. □

Lemma 2.3.8.

$$M(A_2) - M(A_1) \geq \begin{cases} \sum_{i=j_1+1}^{j_2-1} p_{si} * q_{si} + p_{sj_1} * (\sum_{i=1}^{j_1} q_{si} - A_1) & 0 \leq A_1 < A_2 \wedge j_1 < j_2 \\ \sum_{i=j_2+1}^{j_1-1} p_{bi} * q_{bi} + p_{bj_2} * (\sum_{i=1}^{j_2} q_{bi} + A_2) & A_1 < A_2 \leq 0 \wedge j_1 < j_2 \end{cases} \tag{2.7}$$

Proof. Assume $0 \leq A_1 < A_2$ and $j_1 < j_2$. First note

$$\sum_{i=1}^{j_2-1} q_{si} \leq q_{A_2} < \sum_{i=1}^{j_2} q_{si} \implies \sum_{i=1}^{j_2} q_{si} - q_{A_2} \leq \sum_{i=1}^{j_2} q_{si} - \sum_{i=1}^{j_2-1} q_{si} = q_{sj_2}, \tag{2.8}$$

such that

$$\begin{aligned} M(A_2) - M(A_1) &= \sum_{i=1}^{j_2} p_{si} * q_{si} - p_{sj_2} * (\sum_{i=1}^{j_2} q_{si} - A_2) - \sum_{i=1}^{j_1} p_{si} * q_{si} + p_{sj_1} * (\sum_{i=1}^{j_1} q_{si} - A_1) \\ &= \sum_{i=j_1+1}^{j_2} p_{si} * q_{si} - p_{sj_2} * (\sum_{i=1}^{j_2} q_{si} - A_2) + p_{sj_1} * (\sum_{i=1}^{j_1} q_{si} - A_1) \\ &\geq \sum_{i=j_1+1}^{j_2} p_{si} * q_{si} - p_{sj_2} q_{sj_2} + p_{sj_1} * (\sum_{i=1}^{j_1} q_{si} - A_1) \\ &= \sum_{i=j_1+1}^{j_2-1} p_{si} * q_{si} + p_{sj_1} * (\sum_{i=1}^{j_1} q_{si} - A_1). \end{aligned}$$

Note that in the last line, the term $\sum_{i=j_1+1}^{j_2-1} p_{si} * q_{si}$ consists of zero terms whenever $j_2 = j_1 + 1$.

The proof for $A_1 < A_2 \leq 0$ and $j_1 < j_2$ follows along the same lines. □

Theorem 2.3.9. *Function 2.4 is increasing.*

Proof. To be increasing, we must prove $\forall A_1 < A_2 : M(A_2) - M(A_1) \geq 0$. Therefore, assume $A_1 < A_2$. We have three cases: $A_1 < A_2 \leq 0$, $0 \leq A_1 < A_2$, and $A_1 < 0 < A_2$. We will separately prove $M(A_2) - M(A_1) \geq 0$ for all these cases.

Assume $A_1 < A_2 \leq 0$. Lemma 2.3.6 tells us that $j_2 \leq j_1$. When $j_2 = j_1$, we have $M(A_2) - M(A_1) = p_{bj_2}(A_2 - A_1) > 0$ according to lemma 2.3.7, which is greater than zero since $A_1 < A_2 \iff A_2 - A_1 > 0$. When $j_2 < j_1$, we have $M(A_2) - M(A_1) \geq \sum_{i=j_2+1}^{j_1-1} p_{bi} * q_{bi} + p_{bj_2} * (\sum_{i=1}^{j_2} q_{bi} + A_2)$ according to lemma 2.3.8. Furthermore, we know that this value is greater than zero since the first sum consist of nonnegative elements while the second term is positive since $\sum_{i=1}^{j_2} p_{bi} * q_{bi} > -A_2$. Therefore $A_1 < A_2 \leq 0 \implies M(A_2) - M(A_1) \geq 0$.

Assume $0 \leq A_1 < A_2$. Lemma 2.3.5 tells us that $j_2 \geq j_1$. When $j_2 = j_1$, we have $M(A_2) - M(A_1) = p_{sj_2}(A_2 - A_1) > 0$ according to lemma 2.3.7, which is greater than zero since $A_1 < A_2 \iff A_2 - A_1 > 0$. When $j_1 < j_2$, we have $M(A_2) - M(A_1) \geq \sum_{i=j_1+1}^{j_2-1} p_{si} * q_{si} + p_{sj_1} * (\sum_{i=1}^{j_1} q_{si} - A_1)$ according to lemma 2.3.8. Furthermore, we know that this value is greater than zero since the first sum consist of nonnegative elements while the second term is positive since $\sum_{i=1}^{j_1} p_{si} * q_{si} > A_1$. Therefore $0 \leq A_1 < A_2 \implies M(A_2) - M(A_1) \geq 0$.

Finally, assume $A_1 < 0 \leq A_2$. Then $M(A_1) < 0$ since all terms in equation 2.4 are negative. Furthermore, since all terms are nonnegative, $M(A_2) \geq 0$. Therefore $M(A_2) - M(A_1) > 0$.

We have proven for arbitrary $A_1 > A_2$ that $M(A_2) - M(A_2) > 0$. Therefore, $M(A)$ is an increasing function for all $A \in \mathbb{R}$.

$\square$

Lemma 2.3.10.

$$\forall A_1 < A_2 : (A_1 * A_2 \geq 0) \implies p_2(A_2 - A_1) \leq M(A_2) - M(A_1) \leq p_1(A_2 - A_1) \tag{2.9}$$

Here, p_1 and p_2 are the prices at the border for A_1 and A_2 respectively. From the context, it can be derived whether these are p_{sj_1} or p_{bj_1} and p_{sj_2} or p_{bj_2}.

Proof. Since $A_1 * A_2 \geq 0$, we know that they both have the same sign. Therefore, there are four possible cases: Either $A_1 = 0$, $A_2 = 0$, $A_1 < A_2 < 0$ or $0 > A_1 < A_2$. Note that it is trivially true when $A_1 = 0$ or $A_2 = 0$. We will proof the formula to be true for the remaining cases.

Assume $A_1 < A_2 < 0$ such that $j_1 \geq j_2$ according to lemma 2.3.6. Then we compute

$$\begin{aligned}
&M(A_2) - M(A_1) \\
&= \left(-\sum_{i=1}^{j_2-1} p_{bi} * q_{bi} + p_{bj_2} * \big(A_2 + \sum_{i=1}^{j_2-1} q_{bi}\big) \right) - \left(-\sum_{i=1}^{j_1-1} p_{bi} * q_{bi} + p_{bj_1} * \big(A_1 + \sum_{i=1}^{j_1-1} q_{bi}\big) \right) \\
&= \sum_{i=j_2}^{j_1-1} p_{bi} * q_{bi} + p_{bj_2} * \left(A_2 + \sum_{i=1}^{j_2-1} q_{bi} \right) - p_{bj_1} * \left(A_1 + \sum_{i=1}^{j_1-1} q_{bi} \right) \\
&= \sum_{i=j_2+1}^{j_1} p_{bi} * q_{bi} + p_{bj_2} * \left(A_2 + \sum_{i=1}^{j_2} q_{bi} \right) - p_{bj_1} * \left(A_1 + \sum_{i=1}^{j_1} q_{bi} \right)
\end{aligned}$$

When $j_1 = j_2$, this simplifies to

$$\begin{aligned}
&M(A_2) - M(A_1) \\
&= \sum_{i=j_1+1}^{j_1} p_{bi} * q_{bi} + p_{bj_1} * \left(A_2 + \sum_{i=1}^{j_1} q_{bi} \right) - p_{bj_1} * \left(A_1 + \sum_{i=1}^{j_1} q_{bi} \right) \\
&= p_{bj_1}(A_2 - A_1) \\
&= p_{bj_2}(A_2 - A_1).
\end{aligned}$$

When $j_1 > j_2$, we can compute

$$\begin{aligned}
&M(A_2) - M(A_1) \\
&\leq \sum_{i=j_2+1}^{j_1} p_{bi} * q_{bi} + p_{bj_1} * \left(A_2 + \sum_{i=1}^{j_2} q_{bi} \right) - p_{bj_1} * \left(A_1 + \sum_{i=1}^{j_1} q_{bi} \right) && (2.10) \\
&= p_{bj_1}(A_2 - A_1) + \sum_{i=j_2+1}^{j_1} p_{bi} * q_{bi} - p_{bj_1} \sum_{i=j_2+1}^{j_1} q_{bi}
\end{aligned}$$

$$
\begin{aligned}
&= p_{bj_1}(A_2 - A_1) + \sum_{i=j_2+1}^{j_1} (p_{bi} - p_{bj_1}) * q_{bi} \\
&\leq p_{bj_1}(A_2 - A_1). \qquad (2.11)
\end{aligned}
$$

In equation 2.10, we used $p_{bj_1} > p_{bj_2}$ and $A_2 + \sum_{i=1}^{j_2} q_{bi} \geq 0$. Equation 2.11 follows from from the fact that $\forall i < j : p_{bi} < p_{bj}$.

Furthermore, using the same arguments as above, we can compute

$$
\begin{aligned}
&M(A_2) - M(A_1) \\
&\geq \sum_{i=j_2}^{j_1-1} p_{bi} * q_{bi} + p_{bj_2} * \left(A_2 + \sum_{i=1}^{j_2-1} q_{bi}\right) - p_{bj_2} * \left(A_1 + \sum_{i=1}^{j_1-1} q_{bi}\right) \qquad (2.12) \\
&= p_{bj_2}(A_2 - A_1) + \sum_{i=j_2+1}^{j_1} p_{bi} * q_{bi} - p_{bj_2} \sum_{i=j_2+1}^{j_1} q_{bi} \\
&= p_{bj_2}(A_2 - A_1) + \sum_{i=j_2+1}^{j_1} (p_{bi} - p_{bj_2}) * q_{bi} \\
&\geq p_{bj_2}(A_2 - A_1). \qquad (2.13)
\end{aligned}
$$

In equation 2.12, we used $p_{bj_1} > p_{bj_2}$ and $A_1 + \sum_{i=1}^{j_1-1} q_{bi} \leq 0$. Equation 2.13 follows from from the fact that $\forall i < j : p_{bi} < p_{bj}$.

Therefore, we have $p_2(A_2 - A_1) \leq M(A_2) - M(A_1) \leq p_1(A_2 - A_1)$ when $A_1 < A_2 < 0$.

We now prove the second case. Assume $0 < A_1 < A_2$ such that $j_1 \leq j_2$ according to lemma 2.3.5. Then we compute

$$
\begin{aligned}
&M(A_2) - M(A_1) \\
&= \left(\sum_{i=1}^{j_2-1} p_{si} * q_{si} + p_{sj_2} * \left(A_2 - \sum_{i=1}^{j_2-1} q_{si}\right)\right) - \left(\sum_{i=1}^{j_1-1} p_{si} * q_{si} + p_{sj_1} * \left(A_1 - \sum_{i=1}^{j_1-1} q_{si}\right)\right) \\
&= \sum_{i=j_1}^{j_2-1} p_{si} * q_{si} + p_{sj_2} * \left(A_2 - \sum_{i=1}^{j_2-1} q_{si}\right) - p_{sj_1} * \left(A_1 - \sum_{i=1}^{j_1-1} q_{si}\right)
\end{aligned}
$$

When $j_1 = j_2$, this simplifies to

$$
\begin{aligned}
&M(A_2) - M(A_1) \\
&= \sum_{i=j_1}^{j_1-1} p_{si} * q_{si} + p_{sj_1} * \left(A_2 - \sum_{i=1}^{j_1-1} q_{si}\right) - p_{sj_1} * \left(A_1 - \sum_{i=1}^{j_1-1} q_{si}\right) \\
&= p_{sj_1} * (A_2 - A_1) \\
&= p_{sj_2} * (A_2 - A_1).
\end{aligned}
$$

When $j_1 > j_2$, we can compute

$$M(A_2) - M(A_1)$$

$$\begin{aligned}
&\leq \sum_{i=j_1}^{j_2-1} p_{si} * q_{si} + p_{sj_1} * \left(A_2 - \sum_{i=1}^{j_2-1} q_{si}\right) - p_{sj_1} * \left(A_1 - \sum_{i=1}^{j_1-1} q_{si}\right) && (2.14)\\
&= p_{sj_1}(A_2 - A_1) + \sum_{i=j_1}^{j_2-1} p_{si} * q_{si} - \sum_{i=j_1}^{j_2-1} p_{sj_1} * q_{si}\\
&= p_{sj_1}(A_2 - A_1) + \sum_{i=j_1}^{j_2-1} (p_{si} - p_{sj_1}) * q_{si}\\
&\leq p_{sj_1}(A_2 - A_1) && (2.15)
\end{aligned}$$

In equation 2.14, we used $p_{sj_1} > p_{sj_2}$ and $A_2 + \sum_{i=1}^{j_2-1} q_{si} \geq 0$. Equation 2.15 follows from from the fact that $\forall i < j : p_{si} > p_{sj}$.

Furthermore, using the same arguments as above, we can compute

$$\begin{aligned}
&M(A_2) - M(A_1)\\
&= \sum_{i=j_1+1}^{j_2} p_{si} * q_{si} + p_{sj_2} * \left(A_2 - \sum_{i=1}^{j_2} q_{si}\right) - p_{sj_1} * \left(A_1 - \sum_{i=1}^{j_1} q_{si}\right)\\
&\geq \sum_{i=j_1+1}^{j_2} p_{si} * q_{si} + p_{sj_2} * \left(A_2 - \sum_{i=1}^{j_2} q_{si}\right) - p_{sj_2} * \left(A_1 - \sum_{i=1}^{j_1} q_{si}\right) && (2.16)\\
&= p_{sj_2}(A_2 - A_1) + \sum_{i=j_1+1}^{j_2} p_{si} * q_{si} - \sum_{i=j_1+1}^{j_2} p_{sj_2} * q_{si}\\
&= p_{sj_2}(A_2 - A_1) + \sum_{i=j_1+1}^{j_2} (p_{si} - p_{sj_2}) * q_{si}\\
&\geq p_{sj_2}(A_2 - A_1). && (2.17)
\end{aligned}$$

In equation 2.16, we used $p_{sj_1} > p_{sj_2}$ and $A_1 + \sum_{i=1}^{j_1} q_{si} \leq 0$. Equation 2.17 follows from from the fact that $\forall i < j : p_{si} > p_{sj}$.

Therefore, we have $p_2(A_2 - A_1) \leq M(A_2) - M(A_1) \leq p_1(A_2 - A_1)$ when $0 < A_1 < A_2$.

So indeed, $A_1 * A_2 \geq 0 \wedge A_1 < A_2 \implies p_2(A_2 - A_1) \leq M(A_2) - M(A_1) \leq p_1(A_2 - A_1)$. □

Theorem 2.3.11. *Function 2.4 satisfies requirement 4.*

Proof. Without lose of generality, assume random variables A_1, A_2 such that $A_1 < A_2$. Furthermore, denote $A_3 = A_1 + t(A_2 - A_1)$ such that $A_1 < A_3 < A_2$, and assume $A_1 \times A_2 \geq 0$ as in requirement 4. We then know that A_1, A_2 and A_3 all have the same sign. Therefore, $p_1 > p_2 > p_3$ for both the sell and buy side. Furthermore, lemma 2.3.10 tells us that

$$\begin{aligned}
&p_3(A_3 - A_1) \leq M(A_3) - M(A_1),\\
&M(A_2) - M(A_3) \leq p_3(A_2 - A_3).
\end{aligned}$$

Rewriting, we find

$$\frac{M(A_2) - M(A_3)}{A_2 - A_3} \leq p_3 \leq \frac{M(A_3) - M(A_1)}{A_3 - A_1}$$

as required. □

2.4. Automated Market Maker

This section introduces the Automated market maker (AMM). The AMM is an upcoming market mechanism in the cryptocurrency markets [7], [8]. We first give an overview of the basic AMM and its different components in section 2.4.1 using examples of the constant-product AMM. In 2.4.2, we introduce an advanced constant-product AMM using range orders, which is implemented in the simulation. After, we will formally define the AMM in section 2.4.3. Finally, it will be shown that the AMM satisfies the requirements as formulated in section 2.1 in section 2.4.4.

2.4.1. Basic AMM

An automated market maker (AMM) is a computer program that automatically prices an asset in terms of another one based on the liquidity external parties lend it. The external parties are liquidity providers, while the liquidity they lend forms the inventory of the AMM. This inventory is then called the liquidity pool (LP) and consists of at least two kinds of assets, like Google stocks and euros. The AMM also contains a function on the LP that should always equal a constant. When a trade changes the amount of one asset in the LP, the inventory in at least one other asset should also change. We call this function the AMM curve since it generates a curve in the xy-plane when the amount of one asset is graphed against another. The AMM curve is an inherited property of the AMM and largely determines its behaviour. The price as dictated by the AMM curve will be called the base price, for an additional fee has to be paid to compensate the liquidity providers for their service. The following subsections will explain all these components of the general AMM. In the examples, it will be shown how these components actually function in the constant-product AMM used in the simulation.

The Liquidity Pool

The liquidity pool (LP) is the inventory of the AMM and thus describes its liquidity. The LP can consist of more than two assets. Each asset in the pool can then be priced in (a combination of) the others. This thesis only deals with two-dimensional pools containing an asset denoted by A while the other is a currency (euros in the examples) denoted by M. There are two reasons why we only deal with two-dimensional glsamms in this thesis. First, we want to compare the AMM to the LOB, which only deals with two assets (shares and money). More importantly, any n-dimensional AMM can be reduced to multiple 2-dimensional AMMs by splitting up a trade and adjusting fees where necessary.

The AMM curve and base price

The base price formula gives the price of one asset in terms of the other. It is derived from the AMM curve, which we defined first. The AMM curve is a function on the inventory of the LP that is required to equal a constant. By imposing this requirement on the AMM curve, any change in the inventory of one asset will dictate the (change in) inventory of the other. An example of an AMM curve is the constant product formula $A * M = L^2$ depicted in figure 2.3[6].

Example 2.1: A simple trade

In figure 2.3, the constant-product AMM curve is depicted in the case $L^2 = 40000$. Assume that the LP is in the state $(200, 200)$, and 100 assets are bought. We then require $(M + \Delta M) * (A - 100) = 20000$, and derive $\Delta M = 200$. Therefore, the LP would have a shortage of €200 and demand this from the trader. This can also be seen in figure 2.3.

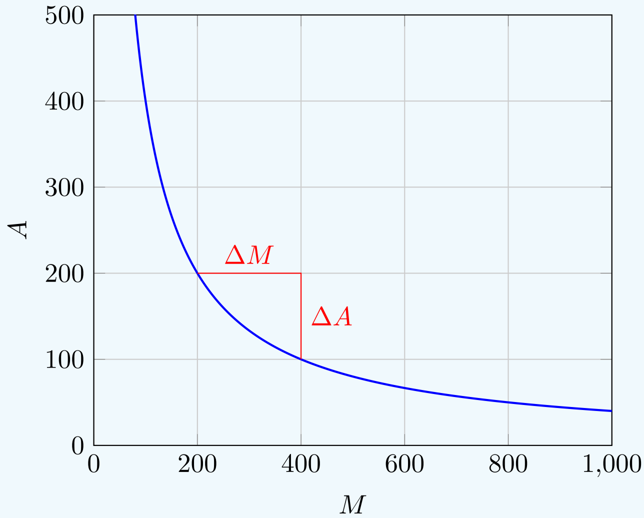


Figure 2.3.: The AMM curve of $A * M = 40000$

We turn to the actual base price formula that computes the returned assets. Assume an AMM with pool in the state (A, M) and with curve $C(A, M) = L^2$. We assume the AMM curve to have to form $C(A, M) = L^2$. Here, L^2 is a constant called the invariant. Furthermore, assume $C(A, M) = L^2$ can be rewritten as $M = f(A)$ and $M = g(A)$[7]. From this, we can derive the pricing formula for assets expressed in euros: Denote the money paid by ΔM. Then, the number of assets we receive, denoted by ΔA, is a function of both M and ΔM. The base price formula for assets is given as

$$\Delta A(M, \Delta M) = f(M) - f(M + \Delta M). \tag{2.18}$$

Similarly, using $M = g(A)$, we find the base price formula of euros to be

$$\Delta M(A, \Delta A) = g(A) - g(A + \Delta A). \tag{2.19}$$

[6]For practical reasons described in section 2.4.2, L^2 is used instead of L.

[7]In practice, the AMM curve is always a simple function such that it is easily rewritten in the form $M = f(A)$, where f is a bijection. Approximations can be used when this is not the case.

Example 2.2: Base price formula

Assume the same initial setting as in example 2.1. Then

$$C(A, M) = A * M = 40000,$$

from which we can derive that

$$A = g(M) = \frac{40000}{M}.$$

We can thus compute the price of for any change in assets as

$$\Delta A(M, \Delta M) = \frac{40000}{M} - \frac{40000}{M + \Delta M}. \tag{2.20}$$

In example 2.2, $M = 200$ and $\Delta M = 200$. As before, we find $\Delta A(200, 200) = 100$ and receive 100 A.

Equation 2.21 approximates the base price formula using Taylors approximation.

$$\begin{aligned}\Delta A(M, \Delta M) &\approx f(M) - \big(f(M) + f'(M) * \Delta M\big) \\ &= -\Delta M f'(M)\end{aligned} \tag{2.21}$$

With this approximation, the spot price can be computed. It is known that the asset price is $\frac{\Delta M}{\Delta A}$. Therefore we find that the spot price of assets is given by

$$SP_A = \frac{\Delta M}{\Delta A} = \frac{-\Delta M}{-\Delta M f'(M)} = -\frac{1}{f'(M)}. \tag{2.22}$$

We can also directly find the spot price of money as the reciprocal of the spot price of assets, thus

$$SP_M = -f'(M). \tag{2.23}$$

The AMM's market price (or midprice) is computed as SP_A. Therefore, the market price has high volatility when $f'(M)$ has a steep slope. This can be desirable when new information has entered the market but is unwanted otherwise since it makes the price unstable.

Example 2.3: Market price

Assume the same initial setting as in example 2.1. In example 2.2, we had found that $f(M) = \frac{40000}{M} = \frac{L^2}{M}$. Therefore, the derivative of f with respect to M can be found to be

$$f'(M) = -\frac{L^2}{M^2} = -\frac{A}{M}.$$

Therefore, the spot price of money is

$$SP_A = \frac{M}{A}. \tag{2.24}$$

The current market price is €1 for one asset.

The fee mechanism

Liquidity providers face several risks, including the devaluation of their assets due to transactions. In order to compensate them, each transaction includes a fee which goes to the liquidity providers. There are several ways to compute a fee, but we will state just one: A percentage of the sold funds is withheld and given to the liquidity providers. This structure's advantage is that larger orders, which impact the price more and create higher risks for the liquidity providers, are charged more. Section 2.4.1 dives into the risks the liquidity provider actually faces.

Another setting of the AMM is where the fees are collected. The fees can be added to the LP, such that the liquidity grows, or can be kept separate. The advantage of increased liquidity is that it stabilises the price, which reduces risks for the liquidity providers. However, the liquidity parameter L has to be continuously updated when fees are added to the LP. Furthermore, a liquidity provider must actively retrieve the LP fee. Following the example of Uniswap V3[28], we opt to collect the fees outside the pool for simplicity[8]

Example 2.4: Paying a fee

Assume the same initial setting as in example 2.1. Again, a trader wants to buy assets for €200. However, this time there is a fee of 2%, which means that €4 is set aside as a fee. The remaining €196 will be traded for assets. Following formula 2.20 in example 2.2, we only receive 98.99 A, which is 1.01 A less then when no fee was paid.

The liquidity mechanism

The liquidity mechanism distributes the fees among the liquidity providers and facilitates the protocol for them to enter and exit the LP. For the first task, it keeps track of each liquidity provider's shares of the LP and distributes the fees proportionally to that. The second task is to update the reserves in the LP, the shares of ownership and L.

[8] More details can be found in section 2.4.2.

When a liquidity provider exits the pool, he receives assets in the same ratio as the pool. For example, a liquidity provider that owns half the pool will receive half of A and half of M. This way, the price will not change when liquidity is added or retrieved[9]. The AMM curve, and thus the speed at which the price changes, does change.

Following the example of the three biggest exchanges[10] that implement the AMM, the liquidity provider can only enter the pool with assets in the same ratio as the pool. The reason for this is threefold: First, and most importantly, is to prohibit trading without paying a fee. When one exits the LP, he receives the assets in the same ratio as the pool. Thus, when a trader enters with a different ratio and exits immediately, he will have traded some assets for the other without paying a fee. The second reason is practical. Since some versions of AMMs, like Uniswap V3, only support a certain price range[28]. Entering the pool with another ratio will change the current ratio of the LP, which might change the price depending on the used curve. Consequently, the new price may fall outside the acceptable range. Finally, it is easy to determine who owns which part of the LP if all liquidity providers use the same ratio.

The assumption that the liquidity provider provides and receives liquidity in the same ratio as the LP does not significantly impact the options of a liquidity provider since he can always trade to obtain the ratio he would prefer.

The new state of the LP determines the new invariant. However, the used function can be changed to accommodate new circumstances.

Example 2.5: Entering and exiting a pool

Assume the initial settings as in example 2.1, with the addition of a fee of 2%. A liquidity provider adds 100 assets and €100 to the pool. The new state of the LP is $(300, 300)$ with new invariant $300 * 300 = 90000$.
A trader wants to buy 100 assets, which costs him €150, excluding the fee and the state of the LP is (200, 450). The total paid is €150/0.98 $\approx$ €153.06, €46.94 less than in example 2.1. Since liquidity is higher, there is a smaller price increase, and the trader has to pay less. When the liquidity provider pulls out his share, the AMM returns to its original curve in the state (133.33, 300). He receives 66.67 assets and €150, next to a fee of €1.02. He has essentially sold 33.33 assets for €51.02.

Theorem 2.4.1. *Assume an AMM uses the constant product AMM curve. The liquidity pool's value decreases whenever the AMM price shifts away from its original price.*

Proof. Assume a spot price $r_M = \frac{A}{M}$ per equations 2.23 and 2.24. Therefore, the LP starts in state $(r_M M, M)$ and $L^2 = r_M * M^2$.

Now assume ΔA (with fees already deducted) assets are added to the LP. When $\Delta A > 0$, assets are sold, while it is bought otherwise. We can compute the new state of the LP to be $(r_M M + \Delta A, \frac{L^2}{r_M M + \Delta A})$. The new price of an asset will be

$$r_A = \frac{\frac{L^2}{r_M M + \Delta A}}{r_M M + \Delta A} = \frac{r_M M^2}{(r_M M + \Delta A)^2}.$$

[9]For the most frequently used constant (weighted) product AMM [29].
[10]Uniswap, Curve and Balancer

We now compute whether the change in assets is worth the change in money. Therefore, we compute the value of ΔA expressed in euros, add this to the new amount of euros and see whether this is at least the old number of euros.

$$\begin{aligned}
&\frac{L^2}{r_M M + \Delta A} + \Delta A * r_A \\
&= \frac{r_M M^2}{r_M M + \Delta A} + \Delta A * \frac{r_M M^2}{(r_M M + \Delta A)^2} \\
&= M\left(\frac{r_M M}{r_M M + \Delta A} + \frac{r_M M \Delta A}{(r_M M + \Delta A)^2}\right) \\
&= M\left(\frac{r_M M * (r_M M + \Delta A)}{(r_M M + \Delta A)^2} + \frac{r_M M \Delta A}{(r_M M + \Delta A)^2}\right) \\
&= M\left(\frac{r_M^2 M^2 + 2 r_M M \Delta A}{r_M^2 M^2 + 2 r_M M \Delta A + \Delta A^2}\right) \\
&< M
\end{aligned}$$

So we find that we lose some money by restoring A to the original amount (by valuing ΔA in euros and adding it to the other side). Therefore, the value of the pool has decreased. □

The intuition behind theorem 2.4.1 explains why there is a loss of value. When the price changes from p_0 to p_1, the AMM crosses all prices in between. When assets become more valuable, the liquidity providers have sold them at lower prices than the new price. When assets become less valuable, the liquidity providers have bought assets at a range of higher prices than the new price. Therefore, there should always be a loss of value.

One might wonder why anybody would provide liquidity when a loss in value seems to be guaranteed. There are two main reasons why anybody would still choose to engage. The first is that they do not expect significant price changes over time. In that case, there would be little impermanent loss while the liquidity provider gains lots of fees. The second is that the liquidity trader might only be interested in one of the two assets in the LP. If the price shifts such that that asset of interest becomes less valuable, the value of the liquidity expressed in that asset increases. Assume a liquidity provider entered with assets and euros but is only interested in the former. The price decreases when assets are sold, and the liquidity provider buys them at increasingly lower prices. Therefore, the value of the pool expressed in assets increases.

Theorem 2.4.2. *The value of the liquidity pool expressed in an asset is at most twice the amount of that asset.*

Proof. Assume a spot price of $r_A = \frac{M}{A}$ as in equation 2.24. Thus, the LP starts in state $(\frac{M}{r_A}, M)$.

The best possible price is the spot price. Therefore, we can compute the value of the pool expressed in assets as

$$A + \frac{M}{r_A} = 2A.$$

The value of the pool expressed in money is

$$M + r_A \times A = M + \frac{M}{A} \times A = 2M.$$

□

Corollary 2.4.2.1. *If the amount of an asset in the LP increases, so does the pool's value expressed in that good and visa versa.*

Example 2.6: Changed pool value

Assume that the current state of the pool is $(A, M) = (100, 100)$ such that $L^2 = 10000$. The spot price of both assets and euros is 1. Therefore, the pool has a value of 200 assets or €200 if expressed in either asset.
Assume that 50 assets are bought. The new state of the pool is $(50, 200)$, and the spot price of assets is €4 while the spot price of euros is 0.25 assets.
If the liquidity trader wanted to regain his original number of assets, he would have to buy 50 assets. In the best case, he can buy them at spot price for €200, and he has no money left. If the trader would instead restore his money to the original amount, he would only be able to buy 25 assets. Either way, he always has lost some value.
If we value the pool in assets only, we get that it has a value of 100 assets. However, if we value the pool in euros, we find a value of €400, which is an increase compared to the original value.

2.4.2. Constant-product AMM with range orders

This section will discuss the constant-product AMM implemented in Uniswap V3 [28]. One implicit assumption in the basic AMM was that liquidity providers accepted any price and thus supplied liquidity on the whole spectrum of prices. This is both inefficient use of liquidity and deterring liquidity providers. Since the AMM will likely not obtain all these prices, liquidity at these prices is unused. Furthermore, liquidity providers cannot prevent huge losses without continuously checking the AMM to withdraw their liquidity in time. Uniswap V3 proposed a solution for this in the form of range orders. With a range order, a liquidity provider provides a range of prices that can use his provided liquidity. This decreases the risk for the liquidity provider and increases efficiency.

This section describes range orders. First, it is shown how multiple range orders integrate into one AMM curve. After, the math needed to cross borders is given, followed by an updated fee distribution mechanism. Finally, we present the math that allows liquidity traders to obtain the correct ratio.

Range order

Assume a trader has a portfolio of ΔA assets and ΔM euros. He can choose to add liquidity on the whole range of prices or a sub-range $[p_l, p_u]$. Figure 2.4 shows the difference between these two range orders. In both, ΔA assets and ΔM euros are used to cover all possible price changes in the range, but the states that can be obtained are quite different.

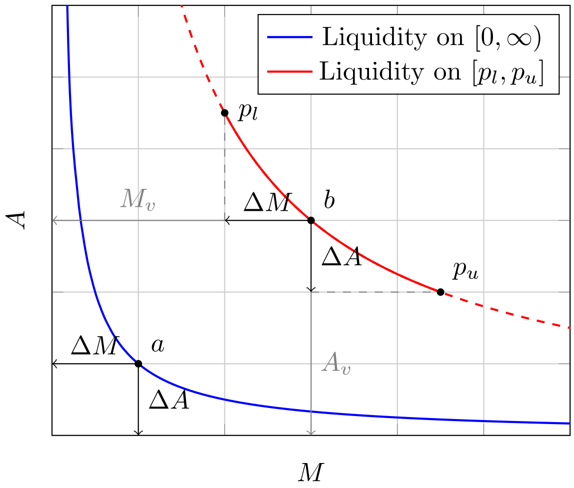


Figure 2.4.: Two range order with different ranges

Assume the pool is in the state $(\Delta A, \Delta M)$. Figure 2.4 shows that equation $A * M = L^2$ no longer holds for a range order on a part of the spectrum. Instead, it uses the curve $(\Delta A + A_v)(\Delta M + M_v) = L^2$, with A_v and M_v constant. It thus adds liquidity A_v and M_v, which it never uses. Figure 2.4 shows that A_v and M_v would have been the liquidity needed to cover the remaining price shifts.

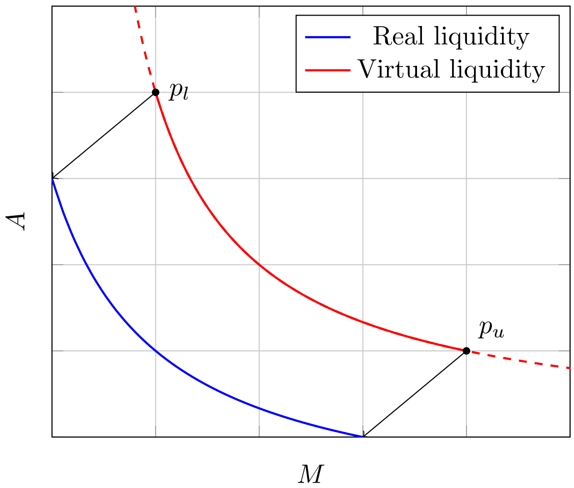


Figure 2.5.: Virtual and real liquidity

Figure 2.5 shows the difference between virtual and real liquidity. The real reserves follow the blue curve, while the total reserves (including the virtual liquidity) follow the red line. The market price equals the lower price when all money depletes, while the upper price is reached when all assets are sold. Subtracting the virtual liquidity from the red curve gives the blue curve. Therefore, the red curve follows equation 2.25.

$$(A_r + A_v)(M_r + M_v) = L^2, \tag{2.25}$$

where r and v stand for real and virtual liquidity respectively.

Assume the LP is in state (A, M) when it reaches price p_l. This is only possible when all money is depleted such that $M = M_v$. Therefore, we find

$$\begin{aligned} p_l &= \frac{M_v}{A} \\ &= \frac{M_v}{(\frac{L^2}{M_v})} \\ &= \frac{M_v^2}{L^2} \\ &\implies M_v = \sqrt{p_l} * L. \end{aligned}$$

$A = A_v$ when the price equals p_u. From this, it is deduced that

$$\begin{aligned} p_u &= \frac{M}{A_v} \\ &= \frac{(\frac{L^2}{A_v})}{A_v} \\ &= \frac{L^2}{A_v^2} \\ &\implies q_{A_v} = \frac{L}{\sqrt{p_u}}. \end{aligned}$$

We find that equation 2.25 reduces to

$$(M_r + \sqrt{p_l} * L)(A_r + \frac{L}{\sqrt{p_u}}) = L^2, \tag{2.26}$$

and L^2 can be obtained using the ABC formula. The current price p_c is computed with equation 2.27.

$$p_c = \frac{M_r + \sqrt{p_l} * L}{A_r + \frac{L}{\sqrt{p_u}}}. \tag{2.27}$$

Finally, note that equations 2.26 and 2.27 should always hold. The current reserves of the pool can thus always derive L and p_c. However, the reverse is also true. A and M can be derived when L and p_c are known. Since L and $p_s = \sqrt{p_c}$ are often used in computations, and the square root is expensive to compute, it is more practical to save these values than the reserves. Therefore, the state is sometimes also given as (L, p_s).

Interaction of range orders

This section explains how to compute the new AMM curve when two (or more) liquidity orders are combined. For this, we assume there are two range orders on ranges $[p_{li}, p_{ui}], i = 1, 2$ with $L_i, i = 1, 2$, that may overlap.

Assume that the ranges do not overlap such that $p_{l1} < p_{u1} < p_{l2} < p_{u2}$ as shown in figure 2.6. The liquidity of the second range order will never be used when $p_c < p_{u1}$, since there is no liquidity between p_{u1} and p_{l2}. The same holds for the first range order whenever $p_c > p_{l2}$. All the fees go to the holder of the used liquidity, thus either to liquidity provider one or two.

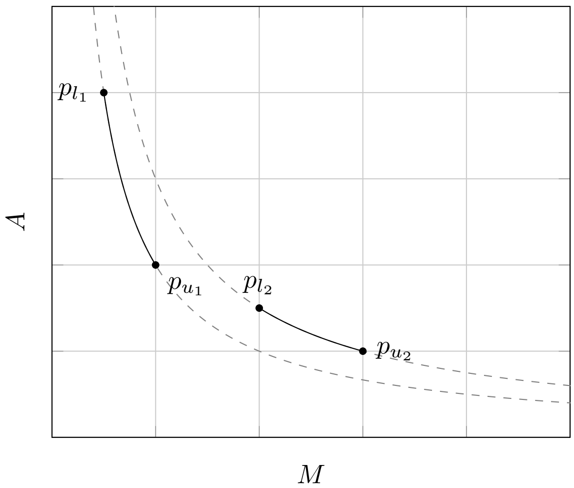


Figure 2.6.: Two distinct range orders

In the second case, the two range orders have at least one price point in common. Therefore assume $p_{l1} < p_{l2} \leq p_{u1} < p_{u2}$, as depicted in figure 2.7. The final AMM curve is drawn in red.

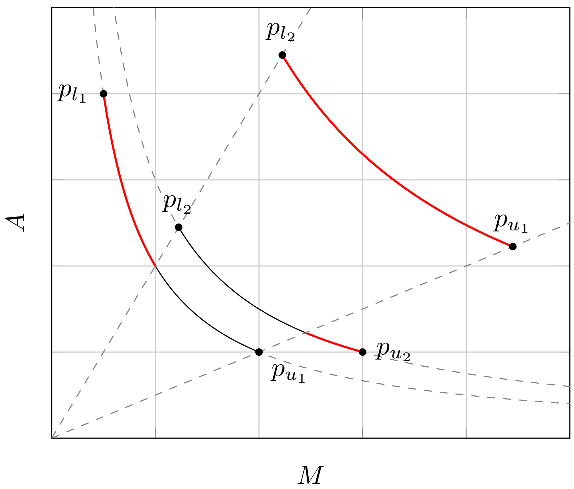


Figure 2.7.: Two overlapping range orders

It is not immediately clear how the new AMM curve is obtained. However, we will show that it has a simple solution. To do this, we first introduce two key relationships proven by Adams *et al.* [28], often used in the remaining section without reference.

$$\Delta p_s = \frac{\Delta M}{L} \iff \Delta q_M = \Delta p_s L, \tag{2.28}$$

$$\Delta \frac{1}{p_s} = \frac{\Delta A}{L} \iff \Delta A = \Delta \frac{1}{p_s} L. \tag{2.29}$$

Theorem 2.4.3. *Given two range orders $[p_{l1}, p_{u1}]$ and $[p_{l2}, p_{u2}]$ with liquidity's L_1 and L_2 such that $p_{l1} < p_{l2} \leq p_{u1} < p_{u2}$ and current price p_c. The combined AMM curve is then given by*

$$\begin{cases} (M_r + \sqrt{p_{l2}} * L_1)(A_r + \frac{L_1}{\sqrt{p_{u1}}}) = L_1^2 & & p_c \in [p_{l1}, p_{l_2}) \\ (M_r + \sqrt{p_{l2} *} * L)(A_r + \frac{L}{\sqrt{p_{u1}}}) = L^2, & L := L_1 + L_2 & p_c \in [p_{l2}, p_{u_1}] \\ (M_r + \sqrt{p_{l2}} * L_2)(A_r + \frac{L_2}{\sqrt{p_{u1}}}) = L_2^2 & & p_c \in (p_{u1}, p_{u_2}]. \end{cases} \tag{2.30}$$

Proof. When $p_c \in [p_{l1}, p_{l_2}) \cup (p_{u1}, p_{u_2}]$, there is only one active range order. Therefore, the curve that that range order dictates has to be used.

It remains to prove that $L = L_1 + L_2$ when $p_c \in [p_{l2}, p_{u_1}]$ is left to prove. To do this, we look at the liquidity at $p_c = p_{l_2}$.

At p_{l_2}, we know that we can only convert assets to euros within this segment. Therefore, $M = M_v$ and A_r is the number of assets available in both range orders to bring the price from p_{l_2} to p_{u_1}. Alternatively, A_r is computed as the number of assets sold to go from p_{u_1} to p_{l_2}.

$$\begin{aligned} A_r &= \Delta \frac{1}{p_s} L_1 + \Delta \frac{1}{p_s} L_2 \\ &= \Delta \frac{1}{p_s} \Big(L_1 + L_2 \Big) \\ &= \Big(\frac{1}{\sqrt{p_{l_2}}} - \frac{1}{\sqrt{p_{u_1}}} \Big) \Big(L_1 + L_2 \Big) \end{aligned}$$

Therefore, L can be deduced as follows:

$$\begin{aligned} L^2 &= (M_r + \sqrt{p_{l2}} * L)(A_r + \frac{L}{\sqrt{p_{u1}}}) \\ &= \sqrt{p_{l_2}} * L \Bigg(\Big(\frac{1}{\sqrt{p_{l_2}}} - \frac{1}{\sqrt{p_{u_1}}} \Big) \Big(L_1 + L_2 \Big) + \frac{L}{\sqrt{p_{u1}}} \Bigg) \\ &= \Big(L_1 + L_2 \Big) \Big(1 - \sqrt{\frac{p_{l2}}{p_{u1}}} \Big) L + \sqrt{\frac{p_{l2}}{p_{u1}}} L^2 \end{aligned}$$

$$\implies \left(L_1 + L_2\right)\left(1 - \sqrt{\frac{p_{l2}}{p_{u1}}}\right)L - \left(1 - \sqrt{\frac{p_{l2}}{p_{u1}}}\right)L^2 = 0$$
$$\implies \left(L_1 + L_2\right) - L = 0$$
$$\implies L = L_1 + L_2$$

Note that $L = 0$ is technically also a solution. However, since we know there is liquidity, this solution does not agree with the AMM. □

The three segments in theorem 2.4.3 can be seen as three contiguous range orders. Adding a third (or any number of range orders) would entail computing three new segments between each existing segment and the new range order as described above.

Trading

When a trade occurs within one continuous segment on the curve, the AMM functions like a normal constant-product AMM. However, the AMM splits the trade when it crosses a border. First, the first trade brings the price to the border. Then, the AMM updates L, and the segment across the border becomes active instead. Finally, the rest of the trade (potentially split again) is executed. Note that this might not be possible when there is no liquidity left. In that case, the AMM executes only part of the order. Since the state of the LP is now given by (L, s_p), we see that each step in a trade now only alters one variable.

When trading, the new price can be computed using equations 2.28 and 2.29. The AMM uses the same equations to derive which part of the trade can be executed within the active segment.

Example 2.7: Trading outside borders

Assume the AMM consists of two orders with ranges $[5, 10]$ and $[10, 15]$ and invariants $L^2 = 700$ and $L^2 = 1200$, respectively. Then we have the following AMM curve[a], also depicted in fig 2.8:

$$\begin{cases} (M + \sqrt{3500})(A + \sqrt{70}) = 700 & p_c \in [5, 10) \\ (M + \sqrt{12000})(A + \sqrt{80}) = 1200 & p_c \in [10, 15] \end{cases}$$

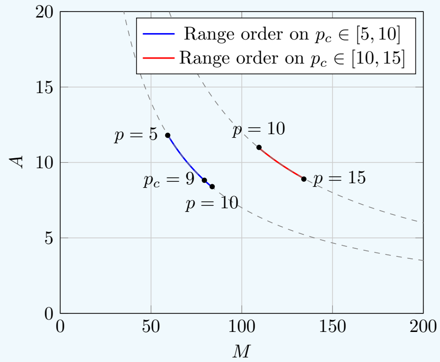


Figure 2.8.

Assume there is no fee, $p_c = 9$, and a trader wants to trade €30.
We then compute $p_s = 3$ and $\Delta p_s = \frac{\Delta M}{L} = \frac{30}{\sqrt{700}} \approx 1.13$. Therefore, the new price is $(3 + 1.13)^2 \approx 17.09$. The AMM splits the trade in two since it crosses the border. The amount that can be traded to reach the border is computed by: $\Delta M = \Delta p_s L = (\sqrt{10} - \sqrt{9})\sqrt{700} \approx 4.29$. Therefore, we first trade 4.29 M, resulting in a new price of $p_c = 10$. We can compute how many assets we get in return using equation 2.29: $\Delta A = \Delta \frac{1}{p_s} L = \left(\frac{1}{\sqrt{10}} - \frac{1}{\sqrt{9}} \right) \sqrt{700} \approx -0.45$. Therefore, we receive 0.45 assets for €4.29.
After this partial trade, we update our invariant to 1200. The trader still wants to trade €25.71, so we compute the new price: $\Delta p_s = \frac{25.71}{\sqrt{1200}} \approx 0.74$. Therefore, the new price is $(\sqrt{10} + 0.74)^2 \approx 15.24$. The AMM splits the trade in two since it crosses the border. The amount that can be traded to reach the border is computed by: $\Delta M = (\sqrt{15} - \sqrt{10})\sqrt{1200} \approx 24.62$. We can compute that we get 2.01 assets.
The remaining €1.09 cannot be traded.

[a] It can be freely chosen to which segment $p_c = 10$ belongs since there is no possible trade covering exactly one price point.

Fee computations

This section describes the fee distribution over the different range orders. Assume a trade within segment $p_c \in [p_l, p_u]$ of the AMM curve with liquidity L. If this is impossible, split the trade into multiple trades and compute the fees for the different range orders separately.

Assume there are $i = 1, .., n$ range orders active on ranges $[p_{li}, p_{ui}]$ with liquidity's L_i, such that $\forall i = 1,..,n : [p_l, p_u] \subset [p_{li}, p_{ui}]$. Note that $[p_l, p_u]$ is a subset or distinct set of $[p_{li}, p_{ui}]$ since the trade doesn't cross any borders.

The percentage of fees a liquidity provider is entitled to is computed as the portion of the money he provides compared to the whole[11]. Assume that the current price is p_u. Formula 2.29 gives the number of euros provided by liquidity provider i on the range $[p_l, p_u]$ as

$$M_i = L_i\left(\sqrt{p_c} - \sqrt{p_l}\right). \tag{2.31}$$

The percentage of the fees on range $[p_l, p_u]$ liquidity provider i is entitled to, is given by

$$f_i = \frac{L_i\left(\sqrt{p_c} - \sqrt{p_l}\right)}{\sum_{j=1}^{n} L_j\left(\sqrt{p_c} - \sqrt{p_l}\right)} = \frac{L_i}{\sum_{j=1}^{n} L_j} = \frac{L_i}{L}.$$

Therefore, a liquidity provider always earn fees proportional to the provided liquidity compared to the total liquidity in that range.

Example 2.8: Fee computations

Assume an AMM in state $(L, p_s) = (0, 3)$. Two liquidity providers add liquidity. The first provides 100 assets and €900 on the whole range such that $L_1 = 300$. The second provides ten assets and €100.30 on $[8, 10]$ such that $L_2 \approx 584.61$. It can be checked that the new state is $(884.61, 3)$. Therefore, liquidity provider two is entitled to 66.09% of the fees as long as they occur in the $[8, 10]$ range. Note that he initially provided less liquidity but used it more effectively (if the price indeed stays within the bounds).

The fee can either be stored or directly paid out. The latter induces many small transactions that come with costs for the exchange. Therefore, Uniswap V3 [28] opts to collect the fees in a contract range order contract, which can be withdrawn at any time. We will do the same but implement it slightly differently.

We make two assumptions. First, the borders of any range orders can only be integer prices since some form of discretization is needed. This assumption has no big consequences[12]. Secondly, we assume that the price always stays in the range $[p_a, p_b]$. Even though this assumption is unrealistic, we will show how the fee collection can be fixed whenever the price steps out of this interval.

We will keep two vectors, one for the assets and one for money, that keep track of the total fees accumulated per unit of L for each range's upper bound. When a liquidity provider enters the pool, he registers the total amount of fees accumulated per unit of L in his range for both A and M. When he exits (or retrieves his fees), he computes these numbers again. The difference times his liquidity is the fees he owns.

We use the vector $\mathbf{f}_A, \mathbf{f}_M \in \mathbb{Q}^{p_b - p_a}$ to keep track of the total fees earned per unit of L. Here, element $\mathbf{f}_{Ai}$ refers to price $i + p_a + 1$ (assuming a zero-indexed vector). If an added range order is outside these borders (and the price can thus reach a price beyond these borders),

[11] The portion of assets can also be used since the ratio of money and assets is the same for every provider.
[12] In theory, any step size could be chosen for the price. Uniswap V3 chooses the powers of 1.001, for example [28].

we can add entries to these vectors so that we can register the fees at those prices too. For efficiency, one could also delete indices that were not in use, but the references would be more complex.

The total accumulated fees in range $[a, b] \subset [p_a, p_b]$ can be computed as

$$f_A = \sum_{i=a-p_a+1}^{b-p_a} \mathbf{f}_{Ai}, \quad f_M = \sum_{i=a-p_a+1}^{b-p_a} \mathbf{f}_{Mi}.$$

Example 2.9: Collecting and retrieving fees

Assume that AMM is initialized on price interval $[p_a, p_b] = [1, 4]$ with $p_c = 2$ and a fee of 10%. Furthermore, one liquidity provider provides liquidity on $[1, 3]$ with $L = 100$. Since no trades have occurred yet, $\mathbf{f}_A = \mathbf{f}_M = \mathbf{0}^{p_b - p_a} = \mathbf{0}^3$ and $f_M = f_A = 0$.
A trader trades €10, of which €1 is kept aside for the liquidity providers. The new price of an asset is €2.26. The liquidity on the range is 100, such that $\mathbf{f}_M$ is updated as $[0, \frac{1}{100}, 0]^T$.
When the liquidity provider now retrieves his fees, he computes

$$\left(\sum_{i=1-p_a+1}^{3-p_a} \mathbf{f}_{M_i} - 0 \right) \times L = \left(0 + \frac{1}{100} \right) * 100 = €1.$$

$$\left(\sum_{i=1-p_a+1}^{3-p_a} \mathbf{f}_{A_i} - 0 \right) \times L = \left(0 + 0 \right) * 100 = 0 \text{ assets.}$$

Entering the pool

When a liquidity trader wants to add liquidity, his range order should imply the same price as the AMM. If not, the added liquidity will change the price since it changes the ratio of the pool, which generates an arbitrage opportunity as seen in section 2.4.1. However, it is unlikely that the liquidity provider will step in with the same ratio as the pool. Therefore, the liquidity provider should trade with the AMM to obtain the same implied price. The remainder of this section describes how such a trade is determined. Often, the AMM does this automatically. All computations shown follow Protocol [30] and are based on Uniswap V3 [28].

First, we introduce a simpler way to compute L. Assume we have a range order on $[p_l, p_u]$, with liquidity L. We have already seen that L is easily computed whenever $p_c \notin (p_l, p_u)$:

$$\begin{aligned}
p_c \leq p_L \iff & M_r = 0 \\
\iff & \left((2.26) \implies \sqrt{p_l} L \left(A_r + \frac{L}{\sqrt{p_u}} \right) = L^2 \right) \\
\iff & \sqrt{p_l} A_r L + \sqrt{\frac{p_l}{p_u}} L = L^2 \\
\iff & \sqrt{p_l} A_r L + (\sqrt{\frac{p_l}{p_u}} - 1) L^2 = 0
\end{aligned}$$

$$\iff L = 0 \quad \text{or} \quad L = \frac{\sqrt{p_l}A_r}{1 - \sqrt{\frac{p_l}{p_u}}}$$
$$\iff L = 0 \quad \text{or} \quad L = \frac{\sqrt{p_l p_u}}{\sqrt{p_u} - \sqrt{p_l}}A_r$$

$$\begin{aligned}
p_c \geq p_U &\iff A_r = 0 \\
&\iff \left((2.26) \implies \left(M_r + \sqrt{p_l}L \right) \frac{L}{\sqrt{p_u}} = L^2 \right) \\
&\iff \frac{M_r}{\sqrt{p_u}}L + \sqrt{\frac{p_l}{p_u}}L^2 = L^2 \\
&\iff \frac{M_r}{\sqrt{p_u}}L + \left(\sqrt{\frac{p_l}{p_u}} - 1 \right) L^2 = 0 \\
&\iff L = 0 \quad \text{or} \quad L = \frac{M_r}{\sqrt{p_u} - \sqrt{p_l}}
\end{aligned}$$

Since $L \neq 0$, we take the other option.

From these, we define

$$L_A = \frac{\sqrt{p_l p_u}}{\sqrt{p_u} - \sqrt{p_l}}A_r, \quad L_M = \frac{M_r}{\sqrt{p_u} - \sqrt{p_l}}. \tag{2.32}$$

We will shortly show that these represent the liquidity of the upper and lower part of a range order respectively.

When $p_c \in (p_l, p_u)$, we can split the range order in two parts, namely (p_l, p_c) and $[p_c, p_u)$. The remaining money will be used for range (p_l, p_c) while the remaining assets will be used for $[p_c, p_u)$. Both range orders should still have invariant L^2, since nothing has changed except for the artificial border at p_c. We will show this to be true.

Theorem 2.4.4. *Given range order on $[p_l, p_u]$, with invariant L^2 and current price $p_c \in (p_l, p_u)$. The two range orders formed by splitting the given one at p_c both have invariant L^2.*

Proof. A_r and M_r are used to cover the price movement to the borders, and so

$$A_r = \left(\frac{1}{\sqrt{p_c}} - \frac{1}{\sqrt{p_u}} \right) L, \quad M_r = \left(\sqrt{p_c} - \sqrt{p_l} \right) L.$$

Furthermore, we know

$$\begin{aligned}
p_c &= \frac{M_r + \sqrt{p_l}L}{A_r + \frac{L}{\sqrt{p_u}}} \\
&= \frac{\left(\sqrt{p_c} - \sqrt{p_l} \right) L + \sqrt{p_l}L}{A_r + \frac{L}{\sqrt{p_u}}} = \frac{\sqrt{p_c}L}{A_r + \frac{L}{\sqrt{p_u}}}
\end{aligned}$$

$$= \frac{M_r + \sqrt{p_l}L}{\left(\frac{1}{\sqrt{p_c}} - \frac{1}{\sqrt{p_u}}\right)L + \frac{L}{\sqrt{p_u}}} = \frac{M_r + \sqrt{p_l}L}{\frac{L}{\sqrt{p_c}}}$$

The first suborder is defined on $[p_l, p_c]$ with liquidity L_M. This range order has funds $(0, M_r)$ and so we find

$$p_c = \frac{M_r + \sqrt{p_l}L_1}{\frac{L_M}{\sqrt{p_c}}}.$$

This implies

$$\begin{aligned} \frac{M_r + \sqrt{p_l}L_M}{\frac{L_1}{\sqrt{p_c}}} &= \frac{M_r + \sqrt{p_l}L}{A_r + \frac{L}{\sqrt{p_u}}} \\ &= \frac{M_r + \sqrt{p_l}L}{\frac{L}{\sqrt{p_c}}} \\ \iff L_M = L. \end{aligned}$$

The second suborder is defined on $[p_c, p_u]$ with liquidity L_A. This range order has funds $(A_r, 0)$ and so we find

$$p_c = \frac{\sqrt{p_c}L_A}{A_r + \frac{L_A}{\sqrt{p_u}}}.$$

This implies

$$\begin{aligned} \frac{\sqrt{p_c}L_A}{A_r + \frac{L_A}{\sqrt{p_u}}} &= \frac{M_r + \sqrt{p_l}L}{A_r + \frac{L}{\sqrt{p_u}}} \\ &= \frac{\sqrt{p_c}L}{A_r + \frac{L}{\sqrt{p_u}}} \\ \iff L_A = L. \end{aligned}$$

We have proven $L = L_M = L_A$. □

Theorem 2.4.4 implies that the correct ratio has been obtained whenever $L_A = L_M$. Uniswap V3 takes $L = \min(L_A, L_M)$ as the added liquidity, and uses these formula's to see which funds cannot be added (without further trading).

From $L_A = L_M$ and edge cases $p_c \leq p_l$ and $p_c \geq p_u$, it can be derived that

$$\frac{q_{M_r}}{q_{A_r}} = \frac{\sqrt{\hat{p}} - \sqrt{p_l}}{\frac{1}{\sqrt{\hat{p}}} - \frac{1}{\sqrt{p_u}}}, \quad \text{with } \hat{p} = max(min(p_c, p_u), p_l). \tag{2.33}$$

Therefore, the liquidity provider trades assets or money until he reaches equality. However, trading changes p_c and possibly also the pool's L. To consider this, we look at how both sides of equation 2.33 change when the price goes from p_c to p_n. Using formula's 2.29 and 2.28, we define

$$R(p_n) = \begin{cases} \frac{M_r + \frac{L}{1-f}\left(\sqrt{p_c} - \sqrt{p_n}\right)}{A_r + L\left(\frac{1}{\sqrt{p_c}} - \frac{1}{\sqrt{p_n}}\right)} & p_n > p_c \\ \frac{M_r + L\left(\sqrt{p_c} - \sqrt{p_n}\right)}{A_r + \frac{L}{1-f}\left(\frac{1}{\sqrt{p_c}} - \frac{1}{\sqrt{p_n}}\right)} & p_n < p_c \end{cases}, \tag{2.34}$$

$$r(p_n) = \frac{\sqrt{\hat{p}} - \sqrt{p_l}}{\frac{1}{\sqrt{\hat{p}}} - \frac{1}{\sqrt{p_u}}}, \quad \text{with } \hat{p} = max(min(p_n, p_u), p_l). \tag{2.35}$$

Assume the AMM currently is in segment $[p_a, p_b]$.

Define p_- as the price at which the liquidity provider has traded all his assets or $p_c = p_a$. Additionally, define p_+ as the price at which the liquidity provider has traded all his money or $p_c = p_b$. Thus

$$q_+ = \min\left(p_b, \left(\sqrt{p_c} + \Delta\sqrt{p}\right)^2\right) = \min\left(p_b, \left(\sqrt{p_c} + \frac{q_{M_r}}{L}\right)^2\right), \tag{2.36}$$

$$q_- = \max\left(p_a, \left(\frac{1}{\frac{1}{\sqrt{p_c}} + \Delta\frac{1}{\sqrt{p}}}\right)^2\right) = \min\left(p_a, \left(\frac{1}{\frac{1}{\sqrt{p_c}} + \frac{q_{A_r}}{L}}\right)^2\right). \tag{2.37}$$

From this, seven cases follow:

1. $R(p_c) = r(p_c)$: The portfolio of the liquidity trader already is in the correct ratio.
2. $R(p_c) > r(p_c)$: The portfolio of the liquidity trader contains to much M, and thus swaps M for A.
   a) $R(p_+) \leq r(p_+)$: The point p_+ is still in the current segment of the AMM. Furthermore, we have passed the point since $R(p_n) = r(p_n)$ for some $p_n \in [p_c, p_+]$ since $R(p_+) \leq r(p_+)$. Therefore, the liquidity provider can trade M for A without leaving the current segment.
   b) $R(p_+) > r(p_+)$: Even when the liquidity provider trades up till the upper boundary, there is still too much M compared to A in his portfolio. The liquidity provider performs a partial trade up till the upper boundary and tries again to satisfy equation 2.33. It cannot happen that p_+ is capped by the amount M_r since this would imply that a trade within the current segment would be sufficient.
3. $R(p_c) = r(p_c)$: The portfolio of the liquidity trader contains to much A, and swaps A for M.
   a) $R(p_-) \geq r(p_-)$: The point p_- is still in the current segment of the AMM. Furthermore, we have passed the point since $R(p_n) = r(p_n)$ for some $p_n \in [p_-, p_c]$ since $R(p_-) \geq r(p_-)$. Therefore, the liquidity provider can trade M for A without leaving the current segment.

b) $R(p_-) < r(p_-)$: Even when the liquidity provider trades up till the lower boundary, there is still too much A compared to M in his portfolio. The liquidity provider performs a partial trade up till the upper boundary and tries again to satisfy equation 2.33. It cannot happen that p_- is capped by A_r since this would imply that a trade within the current segment would be sufficient.

For cases 2b and 3b, it is clear what the new price after the trade should be. However, the new price is not yet found for cases 2a and 3a. To do this, assume we have to trade up till price point p_n. Then we know $p_n \in [p_a, p_b]$. Therefore, we can fill in $\hat{p} = p_n$ in equation 2.33, and deduce p_n. Using the new price, we can find how many assets or euros we have to trade with equations 2.28 and 2.29. Keep in mind that the fee should be added to these numbers.

2.4.3. Formal definition

This section formally defines the contant-product AMM using range orders. The define the AMM as a whole, we first define its separate parts.

This section formally defines the constant-product AMM using range orders. Before we define the AMM as a whole, we first define its separate parts.

Definition 2.4.5. *A constant product AMM using range orders is a tuple $(p_a, p_b, f, p_l, p_u, \mathbf{L}, LP, \mathbf{f}_A, \mathbf{f}_M)$ such that*

- $p_a \in \mathbb{N}$ *is the minimal price the AMM can ever reach,*
- $p_b \in \mathbb{N}$ *is the maximal price the AMM can ever reach,*
- $f \in [0, 1)$ *is the fee that is withheld from the input of every trade,*
- $p_l \in \mathbb{N}$ *is the current lower bound of the segment the AMM is in,*
- $p_u \in \mathbb{N}$ *is the current upper bound of the segment the AMM is in,*
- $\mathbf{L} \in \mathbb{N}^{p_b - p_a}$ *is a vector that represents L in each price point (except for p_a),*
- *LP is a tuple $(p_s, L) \in \mathbb{R}_+ \times \mathbb{N}$, such that p_s is the square root of the current price p_c, while L is a measure of the total liquidity currently in use,*
- $\mathbf{f}_A \in \mathbb{N}^{p_b - p_a}$ *is a vector that represents the total accumulated fees in A in each price point (except for p_a), initially the zero vector,*
- $\mathbf{f}_M \in \mathbb{N}^{p_b - p_a}$ *is a vector that represents the total accumulated fees in M in each price point (except for p_a), initially the zero vector.*

The real liquidity A_r and M_r can be derived from

- $(M_r + \sqrt{p_l}L)(A_r + \frac{L}{\sqrt{p_u}}) = L^2,$
- $\frac{M_r + \sqrt{p_l}L}{A_r + \frac{L}{\sqrt{p_u}}} = p_s^2.$

L should be constant on the current segment (excluding the lower border). Therefore, the following relationships between the elements of the AMM should always hold:

- $p_c \in [p_l, p_u]$

- $\forall p \in (p_l, p_u] \cap \mathbb{N} : \mathbf{L}_{p-p_a} = L$
- $\mathbf{L}_{p_l-p_a} \neq L$
- $\mathbf{L}_{p_u-p_a+1} \neq L$

Besides the AMM, we also define a contract, which is given out to all liquidity providers.

Definition 2.4.6. *A contract C is a tuple $(p_x, p_y, L, f_A, f_M) \in \mathbb{N}^2 \times \mathbb{R}^3_+$ such that*

- *p_x is the lower bound of the range order,*
- *p_y is the upper bound of the range order,*
- *L is the liquidity that the range order adds to the AMM,*
- *f_A is the total accumulated fees of A for the range per unit of L in the AMM when the contract was created,*
- *f_M is the total accumulated fees of M for the range per unit of L in the AMM when the contract was created.*

Six basic transformations bring an AMM from one state to the other, from which all transformations described before can be derived. The first two, selling A or M, describe how assets can be traded in the AMM. The last four influence the liquidity and thus parameter L. The four are adding or retrieving liquidity and crossing the lower or upper border in the current segment. We assume that trades happen within the current segment the AMM is in. Furthermore, it is assumed that the liquidity is added and retrieved in the same ratio as in the pool. We show how each of there transformations work below, where we assume A is an AMM in state $(p_a, p_b, f, p_l, p_u, \mathbf{L}, (p_s, L), \mathbf{f}_A, \mathbf{f}_M)$.

- $T_a : \mathbb{R}_+ \times AMM \rightarrow \mathbb{R}_+ \times AMM = T_a(q_A, A) = (q_M, A')$ is the transformation in which q_A units of A are sold to the AMM. We derive the updated elements of $A' = (p_a, p_b, f, p_l, p_u, \mathbf{L}, (p'_s, L), \mathbf{f}'_A, \mathbf{f}_M)$ as follows :
  - $p'_s = \frac{1}{\frac{1}{p_s} + \frac{q_A*(1-f)}{L}}$
  - $\mathbf{f}'_{A_{p_u-p_a}} = \mathbf{f}_{A_{p_u-p_a}} + \frac{q_A*f}{L}$
  - $\forall i \neq p_u - p_a : \mathbf{f}'_{A_i} = \mathbf{f}_{A_i}$
  - $q_M = \left(\frac{1}{\frac{1}{p_s} + \frac{q_A(1-f)}{L}} - p_s \right) L$
- $T_m : \mathbb{R}_+ \times AMM \rightarrow \mathbb{R}_+ \times AMM = T_m(q_M, A) = (q_A, A')$ is the transformation in which q_M units of M are sold to the AMM. We derive the updated elements of $A' = (p_a, p_b, f, p_l, p_u, \mathbf{L}, (p'_s, L), \mathbf{f}_A, \mathbf{f}'_M)$ as follows:
  - $p'_s = p_s + \frac{q_M}{L}$
  - $\mathbf{f}'_{M_{p_u-p_a}} = \mathbf{f}_{M_{p_u-p_a}} + \frac{q_M*f}{L}$
  - $\forall i \neq p_u - p_a : \mathbf{f}'_{M_i} = \mathbf{f}_{M_i}$

 - $q_A = \left(\frac{1}{p_s + \frac{q_M(1-f)}{L}} - \frac{1}{p_s}\right)L$

- $L : \mathbb{R}_+^2 \times \mathbb{N}^2 \times AMM \to AMM \times C = L(q_A, q_M, p_x, p_y) = (A', C)$ assuming $\frac{p_s\sqrt{p_u}}{\sqrt{p_u}-p_s}q_A = \frac{q_M}{p_s-\sqrt{p_l}}$ is the transformation in which liquidity is added to the AMM (in the correct ratio). The new AMM $A' = (p_a, p_b, f, p_l', p_u', \mathbf{L}', (p_s, L'), \mathbf{f}_A, \mathbf{f}_M)$ and contract C can be derived as follows:
  - $\Delta L = \frac{q_M}{p_s-\sqrt{p_l}}$
  - $\mathbf{L}_i' = \begin{cases} \mathbf{L}_i + \Delta L & \text{if } i = p_x - p_a, p_x - p_a + 1, \dots p_y - p_a \\ \mathbf{L}_i & \text{else} \end{cases}$
  - $p_l' = \begin{cases} p_x & \text{if } p_l \leq p_x \leq p_s^2 \wedge p_y > p_s^2 \\ p_y & \text{if } p_l \leq p_y \leq p_s^2 \\ p_l & \text{else} \end{cases}$
  - $p_u' = \begin{cases} p_y & \text{if } p_s^2 \leq p_y \leq p_l \wedge p_x < p_l \\ p_x & \text{if } p_s^2 \leq p_x \leq p_u \\ p_u & \text{else} \end{cases}$
  - $L' = \begin{cases} L + \Delta L & \text{if } p_x \leq p_s^2 \leq p_y \\ L & \text{else} \end{cases}$
  - $f_A = \sum_{i=p_x-p_a+1}^{p_y-p_a} \mathbf{f}_{Ai}$
  - $f_M = \sum_{i=p_x-p_a+1}^{p_y-p_a} \mathbf{f}_{Mi}$
  - $C = (p_x, p_y, \Delta L, f_A, f_M)$
- $R : C \times AMM \to AMM \times \mathbb{R}_+^2 = R(A, C) = (A', q_A, q_M)$ is the transformation in which liquidity is retrieved from the AMM. Assume $C = (p_x, p_y, \Delta L, f_A, f_M)$. The new AMM $A' = (p_a, p_b, f, p_l', p_u', \mathbf{L}', (p_s, L'), \mathbf{f}_A, \mathbf{f}_M)$ and retrieved liquidity q_A A and q_M M can be derived as follows:
  - $\mathbf{L}_i' = \begin{cases} \mathbf{L}_i - \Delta L & \text{if } i = p_x - p_a + 1, p_x - p_a + 2, \dots p_y - p_a \\ \mathbf{L}_i & \text{else} \end{cases}$
  - $L' = \mathbf{L}'_{\lceil p_s^2 \rceil - q_a}$
  - For p_l' it holds that $\left(L' = \mathbf{L}'_{p_l'-p_a} \neq \mathbf{L}'_{p_l'-p_a-1} \vee p_l' = p_a\right) \wedge \left(\nexists q_l'' : p_c \geq q_l'' > q_l' \wedge \mathbf{L}'_{p_l''-p_a} \neq \mathbf{L}'_{p_l''-p_a-1}\right)$
  - For p_u' is holds that $\left(\mathbf{L}'_{p_u'-p_a} \neq \mathbf{L}'_{p_u'-p_a+1} = L' \vee p_u' = p_b\right) \wedge \left(\nexists q_u'' : p_c \leq q_u'' < q_u' \wedge \mathbf{L}'_{p_u''-p_a} \neq \mathbf{L}'_{p_u''-p_a+1}\right)$
  - $q_A = \Delta L\left(\sum_{i=p_x-p_a+1}^{p_y-p_a} \mathbf{f}_{Ai} - f_A\right)$

- $q_M = \Delta L\left(\sum_{i=p_x-p_a+1}^{p_y-p_a} \mathbf{f}_{Mi} - f_M\right)$

- $U : AMM \to AMM = U(A) = A'$ assuming $p_s^2 = p_u$ is the transformation in which the AMM crossed the upper border. The new AMM $A' = (p_a, p_b, f, p_l', p_u', \mathbf{L}, (p_s, L'), \mathbf{f}_A, \mathbf{f}_M)$ can be derived as follows:
  - $p_l' = p_u$
  - $L' = \mathbf{L}_{p_u - p_a + 1}$
  - For p_u' is holds that $\left(L' = \mathbf{L}_{p_u'-p_a} \neq \mathbf{L}_{p_u'-p_a+1} \vee p_u' = p_b\right) \wedge \left(\nexists q_u'' : p_c \leq q_u'' < q_u' \wedge \mathbf{L}_{p_u''-p_a} \neq \mathbf{L}_{p_u''-p_a+1} = L'\right)$

- $L : AMM \to AMM = L(A) = A'$ assuming $p_s^2 = p_l$ is the transformation in which the AMM crossed the lower border. The new AMM $A' = (p_a, p_b, f, p_l', p_u', \mathbf{L}, (p_s, L'), \mathbf{f}_A, \mathbf{f}_M)$ can be derived as follows:
  - $p_u' = p_l$
  - $L' = \mathbf{L}_{p_u'-p_a}$
  - For p_l' is holds that $\left(\mathbf{L}_{p_l'-p_a} \neq \mathbf{L}_{p_l'-p_a+1} = L' \vee p_u' = p_a\right) \wedge \left(\nexists q_l'' : p_c \geq q_l'' > q_l' \wedge \mathbf{L}_{p_l''-p_a} \neq \mathbf{L}_{p_l''-p_a+1} = L'\right)$

It should be noted that the input and output, except for the possible borders for ranges, are real numbers. In practice, this is not true since some rounding has to be done. However, since the rounding is specific to implementation and doesn't say anything about the workings of the general constant-product AMM with range orders, this has been left out.

2.4.4. Requirements

In this section, it will be shown that AMM satisfies the requirements posed on an exchange in section 2.1.

The requirement 1 is satisfied since the AMM curve and current state are published. There is no hidden liquidity like in the LOB, so everything is completely like expected.

Requirement 2 can only be enforced through legalization. However, a case can be made that the price and quantity stay the same regardless since the state of the AMM would otherwise be inconsistent.

We will now prove that the constant-product AMM with range orders as defined in 2.4.5 satisfy the third and fourth requirement.

Theorem 2.4.7. *The costs M is an increasing function in A, the number of assets we buy.*

Proof. Equation 2.26 tells us

$$(M_r + \sqrt{p_l}L)(A_r + \frac{L}{\sqrt{p_u}}) = L^2$$

holds when $p_c \in [p_l, p_u]$. When ΔA is bought, the trader has to pay ΔM and equation 2.26 becomes

$$(M_r + \Delta M + \sqrt{p_l}L)(A_r - \Delta A + \frac{L}{\sqrt{p_u}}) = L^2$$

. Therefore, $\Delta A > 0 \implies \Delta M > 0$. Furthermore, we can compute ΔM as

$$\Delta M = \frac{L^2}{A_r - \Delta A + \frac{L}{\sqrt{p_u}}} - M_r - \sqrt{p_l} * L,$$

where only ΔA is variable. Furthermore, when $\Delta A > A_r$, we split the trade into multiple parts.

Assume we split the trade in n parts of ΔAi respectively. Furthermore, assume that each part begins with the pool in state $(A_{ri}), (M_{ri})$, has invariant L_i^2 and range $[p_{li}, p_{ui}]$ respectively. Then we can compute the total costs as

$$\Delta M = \sum_{i=1}^{n} \frac{L_i^2}{A_{ri} - \Delta A_i + \frac{L_i}{\sqrt{p_{ui}}}} - M_{ri} - \sqrt{p_{li}} * L_i$$

It is now easy to see that ΔM is increasing, since either ΔA_n increases, and therefor the n'th term increases, or the number n increases, adding another positive term. □

Lemma 2.4.8. *Assume we have an AMM with LP in state (p_s, L) that wield a fee of f. When no liquidity border is crossed, the price when ΔA assets are bought for ΔM is given by $\frac{p_s\sqrt{p_n}}{1-f}$, with p_n the price in the AMM after the trade.*

Proof. Since we assume no liquidity border is crossed, we can assume C to be constant. Furthermore, define A and M as the total liquidity, including virtual liquidity, before the trade. Therefore, we have $A * M = L * 2$. Then, the price is given by

$$\begin{aligned}
p_A &= \frac{\Delta M}{\Delta A} \\
&= \frac{\frac{1}{1-f}(M - \frac{AM}{A+\Delta A})}{\Delta A} \\
&= \frac{1}{1-f}\frac{MA + M\Delta A - AM}{\Delta A * A + \Delta A^2} \\
&= \frac{1}{1-f}\frac{M\Delta A}{\Delta A * A + \Delta A^2} \\
&= \frac{1}{1-f}\frac{M\Delta A}{\Delta A * A + \Delta A^2} \\
&= \frac{1}{1-f}\frac{M}{A + \Delta A}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{1-f}\sqrt{\frac{M^2}{(A+\Delta A})^2} \\
&= \frac{1}{1-f}\sqrt{\frac{M\frac{L^2}{A}}{A+\Delta A\frac{L^2}{M+(1-f)\Delta M}}} \\
&= \frac{1}{1-f}\sqrt{\frac{M}{A}\frac{M+(1-f)\Delta M}{A+\Delta A}} \\
&= \frac{1}{1-f}p_s\sqrt{p_n}
\end{aligned}
$$

□

Lemma 2.4.9. *Assume we have an AMM with LP that wields a fee of f. Furthermore, assume that the AMM is defined on at least segments $[p_0, p_1], ..., [p_{i-1}, p_i]$ with invariants $L_1^2, \dots L_i^2$ respectively. Furthermore, assume the current price $p_c = p_0$ and the new price after the trade is $p_n = p_i$ when A assets are bought, while $p_c = p_i$ and $p_n = p_0$ it true when it is sold.*

The price of ΔA assets expressed in euros when ΔM is paid given by

$$
p_A = \frac{1}{1-f}\sum_{j=1}^{i}\frac{\Delta A_j}{\Delta A}\sqrt{p_{j-1}p_j} \tag{2.38}
$$

where

$$
\Delta A_j = \left(\frac{1}{\sqrt{p_{j-1}}} - \frac{1}{\sqrt{p_j}}\right)L_j
$$

Proof. q_{A_j} is defined as in equation 2.29, and represents the amount of A that is received from the AMM when the price changes from $\sqrt{p_{j-1}}$ to $\sqrt{p_j}$. Furthermore, we know from lemma 2.4.8 that

$$
\frac{1}{1-f}\sqrt{p_{j-1}p_j} = \frac{\Delta M_j}{\Delta A_j},
$$

where A_j is the number of assets we receive from changing the price from p_{j-1} to p_j. Therefore, equation 2.38 simplifies:

$$
\begin{aligned}
\frac{1}{1-f}\sum_{j=1}^{i}\frac{\Delta A_j}{\Delta A}\sqrt{p_{j-1}p_j} &= \sum_{j=1}^{i}\frac{\Delta A_j}{\Delta A}\frac{1}{1-f}\sqrt{p_{j-1}p_j} \\
&= \sum_{j=1}^{i}\frac{\Delta A_j}{\Delta A}\frac{\Delta M_j}{\Delta A_j} \\
&= \sum_{j=1}^{i}\frac{\Delta M_j}{\Delta A} \\
&= \frac{\Delta M}{\Delta A} \\
&= p_A
\end{aligned}
$$

□

Theorem 2.4.10. *The cost function satisfies requirement 4.*

Proof. We have to prove $\forall A_1 \times A_2 \geq 0, A_3 = A_1 - t(A_2 - A_1)$ we have

$$\frac{M(A_2) - M(A_3)}{A_2 - A_3} \leq \frac{M(A_3) - M(A_1)}{A_3 - A_1} \tag{2.39}$$

Denote $\Delta M_2 = M(A_2) - M(A_3)$, $\Delta M_1 = M(A_3) - M(A_1)$, $Delta A_2 = A_2 - A_3$ and $\Delta A_1 = A_3 - A_1$. Now note that these orders are two consecutive buy or sell orders. First ΔA_1 is traded, followed by ΔA_2, performed after a trade of A_1 assets. Lemma 2.4.9 states that there are segments $[p_0, p_1], ..., [p_{i-1}, p_i]$ with invariants $L_1^2, \ldots L_i^2$ such that

- p_0 is the price before the second trade,
- p_i is the price after the third trade,
- $p_k, 0 < k < i$ is the price after the second and before the third trade.

Furthermore, we have

$$\frac{\Delta M_1}{\Delta A_1} = \frac{1}{1-f} \sum_{j=1}^{k} \frac{\Delta A_j}{\Delta A_1} \sqrt{p_{j-1} p_j},$$

and

$$\frac{\Delta M_2}{\Delta A_2} = \frac{1}{1-f} \sum_{j=k}^{i} \frac{\Delta A_j}{\Delta A_2} \sqrt{p_{j-1} p_j}.$$

These are thus weighted averages of several price points, such that it displays the price of one asset again.

Since $p_0 < ... < p_i$, we have

$$\forall j = 0, ..., k : \sqrt{p_{j-1} p_j} \leq \sqrt{p_{k-1} p_k}$$

and so

$$\frac{\Delta M_1}{\Delta A_1} \leq \frac{\Delta M_2}{\Delta A_2}.$$

□

3. Simulation

Chapter 2 gives an exchange's requirements and desired properties. Furthermore, we have seen that both definitions of the LOB and AMM adhere to the requirements and can thus be used to facilitate trading. It is still unclear how these two kinds of exchanges compare. To measure this, we introduced metrics in chapter 2, for which data should be obtained. In this chapter, we describe the model used to generate this data[1].

In the simulation, the LOB and AMM are simulated separately in seven different scenarios. The implementation of the exchanges follows definitions 2.3.4 and 2.4.5 with few exceptions. Changes in the implementation are given in sections 3.1 and 3.2 respectively. The initialization of traders, their decisions and behaviour afterwards is given in section 3.3 for both LOB and AMM. Section 3.4 presents the settings of the simulation and how they influence the outcome. Furthermore, the seven tested scenarios are defined. Finally, section 3.5 describes how the data is extracted from the simulation.

3.1. Limit order book

This section describes the difference in the implementation and definition 2.3.4 of the LOB.

3.1.1. Data structure of buy and sell side

In definition 2.3.4, the buy and sell side are represented by two different ordered lists as depicted in figure 3.1. Assume $p_i < p_{i+1}$ and $t_i < t_{i+1}$. Figure 3.1 shows the data structure used in definitions 2.3.2 and 2.3.3 for the buy and sell side respectively.

Best ask $\longrightarrow$	(p_4, t_2)	(p_4, t_5)	(p_6, t_4)
Best bid $\longrightarrow$	(p_2, t_6)	(p_1, t_1)	(p_1, t_3)

Figure 3.1.: Ordered list of limit orders

Even though this is the most straightforward way of implementing the LOB, it does give some computational overhead. There are three actions in a LOB that modify the lists: The first places a market order, the second places a limit order, and the third cancels a limit order. The current data structure of the ordered list makes it easy to process a market order by matching it against the first limit order in the queue. That limit order is either updated

[1]The code can be found on GitHub.

when it is partially filled or deleted from the list when it is filled, whereafter the new first limit order is looked at.

However, the second and third actions have to traverse the lists. The addition of a limit order needs to find the correct place. In the worst case, the whole list has to be traversed. An example of the worst case is the addition of limit order (p_6, t_7). The same holds for the cancellation of the limit order since it must be found in the list. Using a different data structure for the two lists and saving some helper variables reduces the number of operations. We first introduce the linked list in figure 3.2.

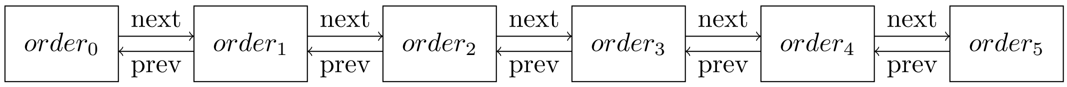
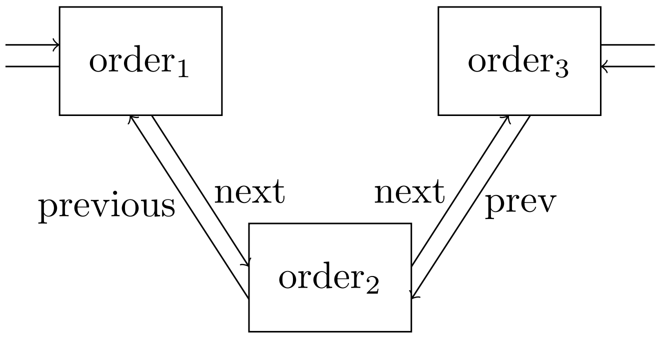
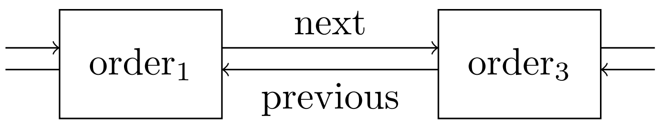


Figure 3.2.: Linked list

The linked list is not a list in the traditional sense. Instead, it consists of many separate objects, one for each limit order. This object then contains a pointer to the next and previous object in the list, which can be set to 'None' if there is no next or previous element. When we save a pointer to the best bid and best ask, it is still easy to process a market order. Furthermore, a cancelled limit order can easily access the previous and next orders and update their pointer to remove itself. See figure 3.3 for an example.



(a) Three limit orders in queue



(b) Two limit orders in queue

Figure 3.3.: Removal of limit order in linked list

The LOB still has to traverse the lists to find the place of a new limit order and update all the pointers. This would consist of going from the state in figure 3.3b to the state in figure 3.3a. The placement can be sped up if the AMM saves additional pointers to the last in order at each price point. The total data structure is given in figure 3.4.

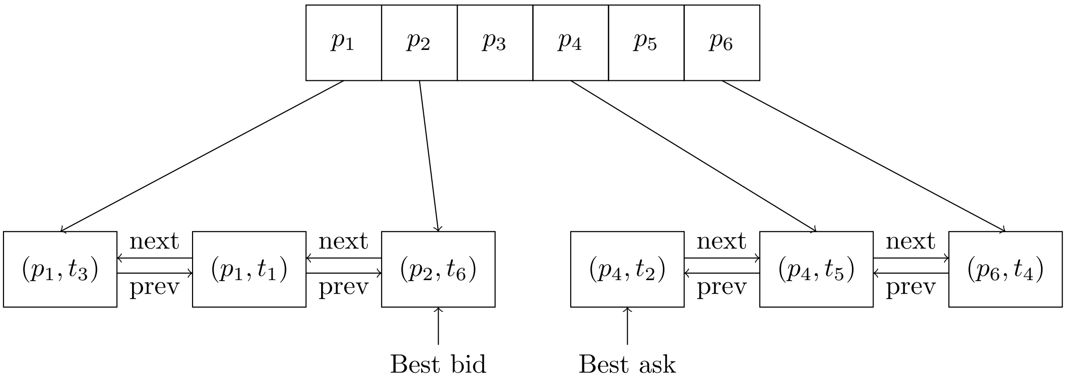


Figure 3.4.: Total structure to save ordering of limit orders

When a limit order is placed, the LOB first checks whether it can immediately match the order. If not, the LOB checks the last node at price point p_i in the list of price pointers. If there is none, which means that there is not yet a limit order at this price, we can check increasingly better prices up till the best price of the other side to find the previous in line.

Example 3.1: Adding a limit order in the middle of the queue

Assume that the LOB is in the state as depicted in figure 3.4. A liquidity trader would like to add a limit sell order at time t_7 and price p_5.
The LOB checks the best bid price first, which is p_2. The limit order cannot be directly matched against another since $p_2 < p_5$.
There is no pointer at p_5. Therefore, the new limit order will be the last limit order at this price point, and the pointer is updated. At p_4, limit order (p_4, t_5) is found. Therefore, the new limit order is placed directly behind (p_4, t_5). Therefore, $(p_4, t_5) =$ order$_1$ and the new order is $order_2$ in figure 3.3b. The LOB updates the pointes accordingly.

Example 3.2: Adding a limit order that performs an undercut

Assume that the LOB is in the state as depicted in figure 3.4. A liquidity trader would like to add a limit sell order at time t_7 and price p_3.
The LOB checks the best bid price first, which is p_2. The limit order cannot be directly matched against another since $p_2 < p_3$.
There is no pointer at p_5. Therefore, the new limit order will be the last limit order at this price point, and the pointer is updated. The LOB does not check the pointer at p_2 since this is a price point of the bid side (as is any price $\leq p_2$). The new limit order will thus be the first in line for the ask side. The best as is $order_3$ in figure 3.3b, while order$_1$ is 'None'. The LOB updates all pointers of the new limit order and best ask, followed by the pointer to the best ask.

3.1.2. Limit order object

Section 3.1.1 introduced the limit order as an object, rather than a tuple as in definitions 2.3.3 and 2.3.2. This object saves extra variables, including two pointers to the next and previous order in line. Additionally, some logic was added to aid with the pointers in case the limit order is filled or cancelled. Furthermore, the limit order fires an event automatically when the limit order is filled to notify the liquidity provider.

3.2. Automated Market maker

This section explains how the implementation of the AMM differs from definition 2.4.5. There are three small yet important differences.

The first is that the implementation does not save that p_l and p_u. Instead of the lower bound, it saves an index. Since the ith element of vectors $\mathbf{L}$, $\mathbf{f}_A$ and $\mathbf{f}_M$ actually represent the value at price point $i + p_a$, we have to subtract p_a from p_l every time to access the correct elements. The AMM keeps track of the index since it is used more frequently than the lower bound. Furthermore, p_u can easily be derived using the current price (saved in the state of the LP) and the liquidity vector $\mathbf{L}$. The AMM then finds the upper boundary p_u as the largest integer greater than the current price, such that the liquidity L is constant on $[p_c, p_u]$[2].

The second change affects how the AMM computes L. The definition wields a vector $\mathbf{L}$ that keeps track of the parameter L for all price points. However, all price points in an order's range must be updated when the order is submitted or cancelled. Therefore, instead of keeping track of L with a vector $\mathbf{L}$, we keep track of the difference in L between the price points in vector $\mathbf{dL}$. The AMM adds $\mathbf{dL}_i$ to L when the border at $p = i + p_a$ is crossed from below, while it is subtracted otherwise. This way, only the elements of $\mathbf{dL}$ that represent the lower and upper boundary of the range of the order have to be updated. However, fees are distributed over consecutive segments of range orders instead of the active one when they have the same L. To salvage this, the number of references at each price point is saved in vector $\mathbf{nR}$. An additional two elements must be updated whenever a range order is submitted or cancelled, causing a few extra computations. Furthermore, when parameter p_u is derived (or the index of the upper bound), it should be taken as the first price higher than the current price, where the number of references is nonzero.

The third change is that the AMM rounds L to the nearest integer to prevent floating point errors. A small mistake in L significantly impacts the results, whereas a small error in the current price or returned value does not since L is used to update itself and $\mathbf{dL}$. Therefore, an element of $\mathbf{dL}$ may not obtain a value of zero due to the floating point error, which causes errors like faulty borders.

We will show that the effect on the collected fees is small for several worst-case scenarios. The maximal rounding error in L is 0.5. Therefore, we maximize the relative rounding error when we minimize L. We assume the settings of the basic scenario (see section ...). L becomes smaller when the range is bigger. Therefore, we take the maximal range order, which is $[950, 1100]$ in the simulation. Furthermore, we take the smallest order possible: either 1000 assets or €$1000 * p_c$. We test the rounding error for four cases. The first is when

[2] In theory, it does not have to be the largest integer. However, the upper border is crossed less frequently when the borders are further apart, thus decreasing the number of computations that must be done around the border.

$p_c = p_l$ such that the trader's portfolio consists of assets only. The second case is when $p_c = p_u$ such that the trader's portfolio consists of euros only. In the third and fourth case, $p_c = 1000$ and $p_c = 1010$ since these are the true prices before and after a shock respectively (see section 3.4). In these cases, $(1000 - A) * p_c$[3]. The results are given in table 3.1. As we can see, even in these worst-case scenarios, the maximum error is $\mathcal{O}(10^{-6})$.

p_l	950	950	950	950
p_u	1100	1100	1100	1100
p_c	950	1100	1000	1010
A	1000	0	648	581
M	0	1100000	352000	423190
L_A	436081,4168	-	440324,438	441924,463
L_M	-	469247,6647	439611,7202	441546,3401
L	436081,4168	469247,6647	439611,7202	441546,3401
Loss (%)	1,15E-06	1,07E-06	1,14E-06	1,13E-06

Table 3.1.: Influence of rounding L in worst case scenario's.

3.3. Trading behaviour

The process of a trade goes through several steps. First, initialization and decision variables are computed as described in sections 3.3.1 and 3.3.2. After that, sections 3.3.3 and 3.3.4 describe how the traders use these variables to finalize their trading strategies in the LOB and AMM respectively.

3.3.1. Initialization variables

This section describes some variables used by the trader which determine what kind of trader he is.

A trader first checks the current liquidity in the market. It is easily computed how many A can be sold or bought at any time. If this number is below the threshold *minLiquidity*, the trader aims to fill this gab. The order in which the sell and buy-side are checked is randomized for each trader. The believed price p_b is set to the market price p_m whenever the trader becomes a so-called forced liquidity trader.

If there is no shortage of liquidity in the market, the trader will initialise several parameters like *informed*, p_b and *buy*. Other initialized parameters are used in the decision-making and thus discussed in the exchange-specific sections.

The parameter *informed* is a boolean variable that states whether the trader is informed or not. An informed trader assumes he has information that has not yet been incorporated into the market's current price. Therefore, he may believe that the asset's true price is different from the market price. An uninformed trader believes that there is no new information (or he thinks it is irrelevant to his goals) and thus that the exchange is efficient (see section 2.2). Therefore, he believes that the true and market price are equal. Assume that i per cent of

[3]Computing M this way corresponds with the simulation.

the traders is informed, and the true price of the asset is p_t while the market price is p_m. The price the trader believes to be true p_b can then be determined by lines 1-7 of algorithm 3.1.

Algorithm 3.1: Determining the initial parameters of a trader when no liquidity order is forced

1 $u_1 \sim \mathcal{U}(0,1)$;
2 $informed \leftarrow u_1 < i$;
3 **if** $informed$ **then**
4 $\quad n \sim \mathcal{N}(0,1)$;
5 $\quad p_b = p_t + \min(\max(n,-3),3)$;
6 **else**
7 $\quad p_b = p_m$
8 $u_2 \sim \mathcal{U}(0,1)$;
9 $buy \leftarrow u_2 < 1/\left(1+e^{-\frac{p_b-p_m}{2}}\right)$;

The believed price of the informed trader is not simply the true price of the asset. Since every informed trader interprets his private information differently, the believed price is slightly different for each trader. We cap the maximal difference between the believed price and true price at three for two reasons. First, negative prices are possible when n becomes very small. Furthermore, a larger difference might not be realistic since the information often points to either an increasing or decreasing price. When the market price is close to the true price, it might not be clear from the information which direction the price should go. However, when there is a large discrepancy, it should point in the correct direction. By capping the price at three, we ensure the correct direction is chosen whenever the difference is larger than three. The probability that an informed trader believes the correct direction is easily computed as

$$
\begin{aligned}
p_{rd} &= P\Big(p_b \geq p_m \iff p_t \geq p_m \Big| |p_t - p_m| = a\Big) \\
&= P\Big((p_b \geq p_m \wedge p_t \geq p_m) \vee (p_b < p_m \wedge p_t < p_m) \Big| |p_t - p_m| = a\Big) \\
&= P\Big(p_b \geq p_m \wedge p_t \geq p_m \Big| |p_t - p_m| = a\Big) + P\Big(p_b < p_m \wedge p_t < p_m \Big| |p_t - p_m| = a\Big) \\
&= P\Big(p_b \geq p_m \Big| p_t \geq p_m \wedge |p_t - p_m| = a\Big) P\Big(p_t \geq p_m \Big| |p_t - p_m| = a\Big) \\
&\quad + P\Big(p_b < p_m \Big| p_t < p_m \wedge |p_t - p_m| = a\Big) P\Big(p_t < p_m \Big| |p_t - p_m| = a\Big) \\
&= p\Big(n \geq -a\Big) P\Big(p_t \geq p_m \Big| |p_t - p_m| = a\Big) + P\Big(n < a\Big) P\Big(p_t < p_m \Big| |p_t - p_m| = a\Big) \\
&= P\Big(n < a\Big) P\Big(p_t \geq p_m \Big| |p_t - p_m| = a\Big) + P\Big(n < a\Big) P\Big(p_t < p_m \Big| |p_t - p_m| = a\Big) \\
&= P\Big(n < a\Big) \Big(P\Big(p_t \geq p_m \Big| |p_t - p_m| = a\Big) + P\Big(p_t < p_m \Big| |p_t - p_m| = a\Big)\Big) \\
&= P(n < a)
\end{aligned}
$$

The probabilities for several values of a are given in table 3.2. Finally, note that the probability $|n| > 3$ is only 0.0027, which means that the borders do not influence the believed prices too often.

$\lvert p_m - p_t \rvert$	P_{rd}
0.1	0.5398
0.5	0.6915
1	0.8413
2	0.9772
3	1

Table 3.2.: Probability that the believed price is in the correct direction for informed traders.

The believed price p_b subsequently determines whether the trader wants to buy or sell the asset. The trader has no preference when $p_b = p_m$. However, when p_b is much larger than p_m, he rather buys since he believes it to be cheap. Likewise, the trader would rather sell when p_b is much smaller than p_m. The logistic function is used to compute the probability that a trader buys assets. The logistic function with parameters $L = 1$, $k = \frac{1}{2}$, $x_0 = p_m$ and $x = p_b$ is given in equation 3.1 and depicted in figure 3.5. The parameter L is the maximum of the curve. Since the maximum probability is one, we have $L = 1$. Furthermore, k determines the logistic growth rate or steepness of the curve, currently set to 0.5[4]. Finally, x_0 is the midpoint, where the probability is 0.5.

$$P(buy) = \frac{L}{1 - e^{-k(x-x_0)}} = \frac{1}{1 - e^{-\frac{p_b - p_m}{2}}} \tag{3.1}$$

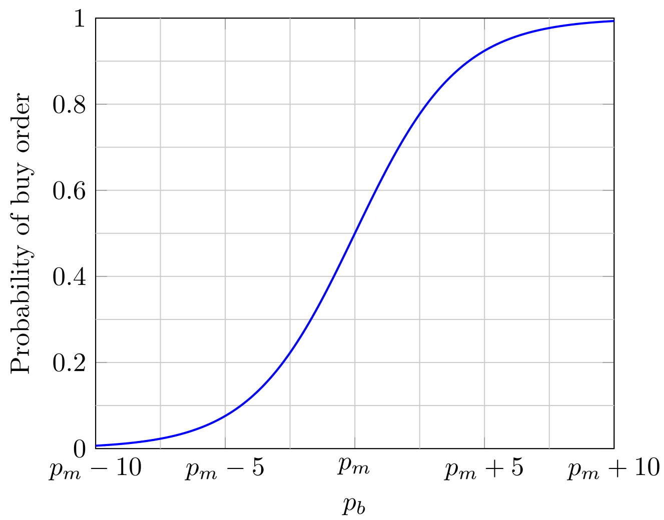


Figure 3.5.: The logistic function with $L = 1$, $k = \frac{1}{2}$, $x_0 = p_m$ and $x = p_b$.

As required, the trader is indifferent whenever $p_b = p_m$ since we have a probability of 0.5 that a buy order will occur. Furthermore, from figure 3.5, it is clear that $p_b > p_m$ returns

[4]For more information about settings, see section 3.4.

a probability larger than 0.5 that a buy order occurs and visa versa, which agrees with the intuition. Lines 8-9 of algorithm 3.1 show how to determine whether the trader places a buy or sell order.

3.3.2. Decision variables

This section describes some variables used later by the trader to determine his trading strategy. The LOB and AMM derive these almost identically.

First, we have to decide the kind of trader this liquidity trader is. For this, we introduce variable *kind*, which can be in one of three states. The first is "L", meaning the trader is a liquidity trader. This is the case, for instance, when the trader is a forced liquidity trader. The second state is "M" when the trader places a market order. The final state is "False" when the trader does not have a preference. How this variable is initialized depends on the exchange, and will be discussed in sections 3.3.3 and 3.3.4.

Lo, MacKinlay, and Zhang [31] have analyzed data from the hundred largest stocks in the S&P 500 from August 1994 to August 1995. From this, they distilled statistics about waiting times for limit orders and the number of market orders used to fill one limit order. The portfolio and maximum waiting time in the case of a liquidity provider are based on these statistics.

Two simulation settings are the minimal and maximal size of an order denoted by q_{min} and q_{max}, respectively. A trader's order size is a random draw in this range. Furthermore, Lo *et al.* [31] show that on average, 1.3662 market orders are needed to fill one limit order[5]. Generalizing this, we assume that a liquidity order is 1.3662 times as big and multiply the order size with this number in case of a liquidity order.

A market order has a maximum waiting time of zero, while a liquidity order has a positive waiting time. Lo *et al.* [31] found that the average time (in minutes) to cancellation for limit orders without any fills was distributed as $\mathcal{N}(46.92, 72.31)$ and $\mathcal{N}(34.15, 53.94)$ for buy and sell orders respectively. A draw from such a variable can thus be used as an approximation for the maximum waiting time. However, this does not take into account the maximum waiting time of (partially) filled limit orders. Therefore, the actual distribution might have a lower mean.

Finally, the portfolio has to be determined. We assume that the portfolio consists entirely of the sold funds. This means that the portfolio consists of assets when assets are sold, while it consists of euros otherwise. When the portfolio only consists of money, we assume the trader has $q * p_b$ M, where q is the order size.

Algorithm 3.2 shows how the different variables are computed. This algorithm assumes that variables q_{min} and q_{max} are predefined.

In order to estimate the probability of execution, several statistics are computed. First, the trader requests the number of assets bought and sold over the last one hundred transactions from the AMM and normalizes them to assets per second. After, he multiplies them by his waiting time and denotes the quantities by bQ and sQ respectively. Since the trader can be informed, these quantities can differ from his expectations. Therefore, he updates the quantities such that the ratio of them is closer to his expectation. Algorithm 3.3 describes

[5]See table 2 of [31], where we have taken the weighted average of the pool. Since the last column states ≥ 7, we have taken a value of 8 to compensate for higher numbers.

Algorithm 3.2: Determining portfolio and waiting time of a trader

$q \sim \mathcal{U}(q_{min}, q_{max})$;
if *buy* **then**
  $time \sim \mathcal{N}(46.92 * 60, 72.31 * 60)$;
else
  $time \sim \mathcal{N}(34.15 * 60, 53.94 * 60)$;
if *kind == "L" or time ¿ 0* **then**
  $q \leftarrow o * 1.3662$;
  while *time* ≤ 0 *and buy* **do**
    $time \sim \mathcal{N}(46.92 * 60, 72.31 * 60)$;
  while *time* ≤ 0 *and not buy* **do**
    $time \sim \mathcal{N}(34.15 * 60, 53.94 * 60)$;
else if *kind=="M"* **then**
  $time \leftarrow 0$;
if *buy* **then**
  $A \leftarrow 0$;
  $M \leftarrow q * p_b$;
else
  $A \leftarrow q$;
  $M \leftarrow 0$;

how he updates the quantities. He first computes tQ, the total amount of A that is traded. After, he computes the ratio of the believed and market price, r in line two. The trader expects the price to rise when $r > 1$ and decrease otherwise. Line three computes the weight of his expectations compared to the past. Note that the weight increases when the r diverts further from one. Finally, we update bQ as a weighted average of the current value and a value that is shifted towards the expectation of the trader. To get the second term, first notice that we have $tQ = bQ + sQ$. Since $r > 1$ implies that the price should increase, it also implies $bQ > sQ$ and visa versa. Therefore, we assume $r = \frac{bQ}{sQ}$. Solving this system of equations gives $bQ = \frac{tQ*r}{1-r}$, which is used as the updated value. Finally, in line 5, we compute the corresponding updated value of sQ.

Algorithm 3.3: Updating the buy and sell quantities

1 $tQ \leftarrow bQ + sQ$;
2 $r \leftarrow p_b / p_m$;
3 $w \leftarrow (1 + |1 - r|)/2$;
4 $bQ \leftarrow bQ * (1 - w) + tQ * r/(1 + r) * w$;
5 $sQ \leftarrow tQ - bQ$;

3.3.3. Limit Order Book

This section describes the trading behaviour when traders interact with the LOB. It is assumed that all initial variables exist.

The trader can choose what kind of order he places when there is enough liquidity in the market. In the LOB, the kind of orders depends on variables *informed*, *buy*, p_b and p_m.

When the trader is informed, there are two cases when he places a market order without additional checks. Let p_{ba} and p_{bb} denote the best-ask and best-bid price respectively. The trader buys assets via a market order when $p_b \geq p_{ba}$ since he gets immediate execution within his budget. The same s true when the trader sells assets and $p_b \leq p_{bb}$. In all other cases, the trader will try for a limit order.

An uninformed trader places a limit order with probability P_r. A larger bid-ask spread implies an increased probability of a limit order since the spread is the cost of a market order. Therefore, P_r has to (implicitly) depend on the bid-ask spread. P_r is computed using the logistic function given in equation 3.1 with $L = 1$ and $k = 1$. Furthermore, $x_0 = p_b$ and $x = p_{ba}$ when assets are bought and $x_0 = p_{bb}$ and $x = p_b$ otherwise. Therefore we see that P_r depends on the bid-ask spread, since $p_{ba} - p_b = p_b - p_{bb}$ is half the bid-ask spread.

Assume that variables q, *time*, bQ and sQ are computed using algorithms 3.2 and 3.3.

The trader immediately executes his order when it is a market order, contrary to a liquidity order since the price and execution probability are still undetermined.

There are three possibilities for the price of a limit order. In the first case, the limit order performs an undercut and $p = p_{ba} - 1$ when assets are sold while $p = p_{bb} + 1$ otherwise. The second case it that the trader gets 'in line' with $p = p_{ba}$ or p_{bb} respectively. The last case, called the overcut, has $p = p_{ba} + 1$ or $p = p_{bb} - 1$ respectively.

The probability that a trader performs an undercut is denoted by $P_u(p_b)$ and can be computed as

$$P_u(p_b) = \begin{cases} \frac{1}{1+e^{-(p_{ba}-p_b-3)}} & \text{if } \textit{informed} \text{ and } \textit{buy} \\ \frac{1}{1+e^{-(p_{ba}-p_b-6)}} & \text{if not } \textit{informed} \text{ and } \textit{buy} \\ \frac{1}{1+e^{-(p_b-p_{bb}-3)}} & \text{if } \textit{informed} \text{ and not } \textit{buy} \\ \frac{1}{1+e^{-(p_b-p_{bb}-6)}} & \text{if not } \textit{informed} \text{ and not } \textit{buy} \end{cases}. \tag{3.2}$$

Note that we have applied the logistic function once again. We have plotted the functions in figure 3.6. In these functions, an informed is more likely to do an undercut. Furthermore, the undercut is only likely with a significant difference.

The trader performs an undercut with probability P_u when it is within budget and possible. Therefore, it must hold that $p_b \geq p_{bb} + 1$ and $p_{ba} - 1 \geq p_{bb} + 1$ when assets are bought or $p_b \leq p_{ba} - 1$ and $p_{ba} - 1 \geq p_{bb} + 1$ otherwise. The limit order is placed without additional checks since an undercut has a big probability of execution.

When the trader does not perform an undercut, he first checks the overcut opportunity. It is assumed that only the excess of $sQ - bQ$ changes the liquidity of the LP since it is replenished otherwise. Assume that *liqAt* number of assets bought (sold) before the trader's assets will be bought (sold). For an overcut, we then require that $sQ - bQ \geq liqAt + amount$ ($bQ - sQ \geq liqAt + amount$).

If the overcut is unlikely to be filled, the trader checks whether an 'in-line' limit order can be filled. For this, he checks $sQ \geq liqAt + amount$ ($bQ \geq liqAt + amount$), where *liqAt* has been updated to the new price point.

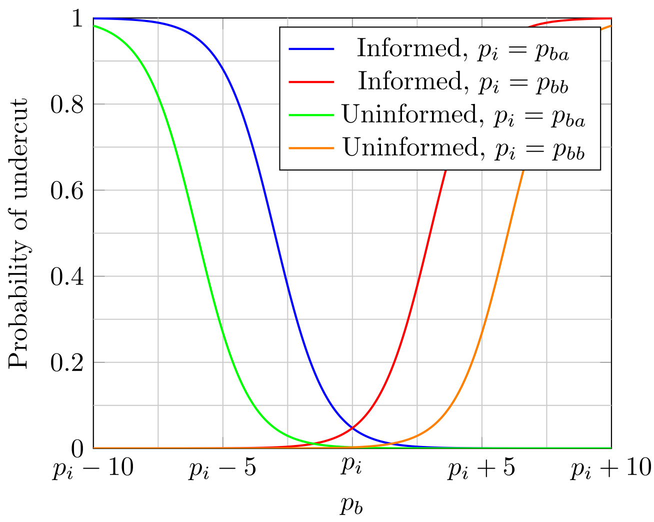


Figure 3.6.: Probability of undercut

When the trader deems execution of a limit order unlikely in all three cases, he places a market order instead. The trader cancels his limit order when it is not filled after *time* has passed and trades the remaining funds via a market order.

3.3.4. Automated Market Maker

This section describes the trading behaviour when traders interact with the AMM. It is assumed that all initial variables exist. Furthermore, we set *kind* to 'False' if there is enough liquidity and 'L' otherwise. All other decision variables can then be derived as described in section 3.3.2.

After computing bQ and sQ, the expected price p_e after *time* can be computed. For this computation, we assume that $sQ - bQ$ assets are bought (or sold when $sQ < bQ$) and compute the new price accordingly. We will use p_e extensively in further computations.

The first thing to be noted is that an undercut is impossible in the AMM since there is only one price. Even worse, it is impossible to buy (sell) assets via a range order when the price increases (decreases) since it implies that assets (euros) are converted to euros (assets)). Therefore, a range order cannot be placed for the purpose of buying (selling) whenever $p_b > p_m$ ($p_b < p_m$) and a market order will be placed instead.

For a range order, we will treat the cases of buying and selling separately, though they are mirror images of each other.

Buy range order

If the trader wants to buy assets, we assume that $p_b < p_m$. Furthermore, we compute A_n and A_w, the number of assets the trader gets if he trades right now or trades after his maximal waiting time. We then determine the range as

$$rL = \left\lfloor \max\left(p_m - 2, p_b - 1, p_5\right)\right\rfloor,$$

$$rU = \left\lceil \max(rL + 2, p_{95}, p_{nl})\right\rceil,$$

where p_5 and p_{95} are the 5 and 95 percentile of the prices of the past and p_{nl} is the first price from p_m to rL where $L = 0$.

The way rL and rU are defined, we have at least a range with width four that includes 95% of previous fees. Furthermore, it forms a 'bridge' of liquidity between our believed price and the market price since it adds liquidity where $L = 0$. We want rL as low as possible for a buy range order since we have seen that the average price paid is given by $\sqrt{rL * rU}$. Finally, p_b is also considered since the trader does not expect the price to fall below p_b.

After we have computed the range, the trader estimates the minimal fees he will collect. To do this, he computes the percentage P_{in} of orders that occur in his range. Furthermore, he computes the percentage P_{own} of liquidity he will own (L_s) in that range. Let us denote the minimal price of the range that is reached by $rL_r = \min(rL, p_e)$. The fees he will collect are then computed as $A_f = P_{in} * P_{own} * sQ$ and $M_f = P_{in} * P_{own} * sB * \sqrt{rL * rU}$ and the total number of assets received after trading with a liquidity order, denoted by A_l can be derived as

$$M_s = \left(\sqrt{rU} - \sqrt{rL_r}\right) * L_s,$$
$$A_l = A_f + \frac{M_s}{\sqrt{rL_r \times rU}} + A_s,$$

where M_s is the number of euros that were actually sold in the range order, A_s the number of assets we receive for $M_f + M - M_s$ M when $p_m = p_e$ and liquidity stayed constant.

The trader then compares A_n, A_w and A_l and goes for the strategy that returns the biggest value. Note that the trader buys assets with remaining euros when time has run out.

Sell range order

If the trader want to sell assets, we assume that $p_b > p_m$. Furthermore, we compute M_n and M_w, the number of euros the trader gets if he trades right now or after waiting his maximal waiting time. We then determine the range as

$$rU = \left\lceil \max(p_m + 2, p_b + 1, p_{95} \right\rceil,$$

$$rL = \left\lfloor p_{nl}, rU - 2, p_5 \right\rfloor,$$

where p_5 and p_{95} are the 5 and 95 percentile of the prices of the past and p_{nl} is the first price from p_m to rU where $L = 0$.

The way rU and rL are defined, we have at least a range with width four that includes 95% of previous fees. Furthermore, it forms a 'bridge' of liquidity between our believed price and the market price since it adds liquidity where $L = 0$. We want rU as big as possible for a sell range order since we have seen that the average price paid is given by $\sqrt{rL * rU}$. Finally, p_b is also considered since the trader does not expect the price to fall below p_b.

After we have computed the range, the trader estimates the minimal fees he will collect. To do this, he computes the percentage P_{in} of orders that occur in his range. Furthermore, he computes the percentage P_{own} of liquidity he will own (L_s) in that range. Let us denote the maximal price of range that is reached by $rU_r = \min(rU, p_e)$. The fees he will collect are then computed as $A_f = P_{in} * P_{own} * sQ$ and $M_f = P_{in} * P_{own} * sB * \sqrt{rL * rU_r}$ and the total number of euros received after trading with a liquidity order, denoted by M_l, can be derived as

$$A_s = \left(\frac{1}{p_m} - \frac{1}{rU_r} \right) \times L_s,$$
$$M_l = M_f + A_s \sqrt{rL \times rU_r} + M_s,$$

where A_s is the number of assets that were actually sold in the range order, M_s the number of euros the trader receives for $A_f + A - A_s$ assets when $p_m = p_e$ and liquidity stayed constant.

The trader then compares M_n, A_w and M_l and goes for the strategy that Returns the biggest value. Note that the trader sells all remaining assets when time has run out.

3.4. Settings and parameters

This section describes are parameters used in the simulation and their effect. Furthermore, we give the values they take on in the basic configuration of the simulation. We have divided the parameters into three categories. The first are the parameters used to initialize the simulation. The second set of parameters determines liquidity and its development. The third set of parameters determines how the true price changes and how the information spreads through the traders. Finally, an overview of the different tested scenarios is given.

3.4.1. Initialization settings

The true price must be set first to initialize the simulation. For this, we use variable *trueP*, initially set to 1000. Changing this price should not have a significant effect on the results. Another setting used for the AMM is the fee market takers pay.

We use variable *fee*, initially set to 0.05% per Uniswap's recommendation for stable assets [32]. Increasing the fee also increases the bid-ask spread in the AMM since the bid-ask spread depends fully on the liquidity and fee, as can be seen from formula's 2.28 and 2.29, the change in price decreases when liquidity increases. The fee is thus the main component of the bid-ask spread whenever liquidity is high.

Both exchanges need initial liquidity to kick-start the exchange. For this, variable *minLiquidity* as given in section 3.4.2, is used. Furthermore, we use variable *initialBidAsk* initially set to two. In the LOB, *minLiquidity* assets are sold at *trueP+initialBidAsk*, while *minLiquidity* assets are bought at *trueP-initialBidAsk*, creating a bid-ask spread of four. For the AMM, we create a range order with *minLiquidity* assets and €*minLiquidity*×*trueP*. Here, we set the upper boundary of the range to $trueP + initialBidAsk$. However, since we want to have a spot price of *trueP* initially, the lower boundary is derived using formulas from section 2.4 and rounded afterwards since it has to be an integer. Therefore, the initial spot price of the AMM changes slightly to 999.998 with the current settings.

Both exchanges also have a range of prices that can be obtained. This range is $[minPriceRange, maxPriceRange]$, initially set to $[950, 1100]$. This range can be changed whenever a larger (or smaller) set of prices is assumed to be obtainable. Since not all used prices are visible (limit orders may never execute, and thus the price may never be shown in the spot price graph), this range should be taken large enough. However, decreasing this range has two important effects. First, it is computationally faster to have a smaller range since smaller lists have to be traversed. However, more importantly, is that this range caps the computation of the spot price. Therefore, it might influence the expected price p_e in the trading behaviour with the AMM. However, we do not expect large effects since these are extreme cases.

Variable *days* represents the duration of the simulation, currently set to three days. A longer time frame can give insights into the stationarity of the prices, especially when there are multiple shocks. However, it negatively affects the reliability of the random walk indicators when the time between two shocks is large since these are only reliable for smaller time frames. We have set the time to three days since this gives the exchanges enough time to obtain a new equilibrium and generate enough data to test for a random walk.

The last parameter used in the initialization of the exchanges is *lookNTransactionsBack*, initially set to 100, which determines how many past market transactions are saved for statistics used by the trader for trading decisions in section 3.3. Furthermore, it is used to give an artificial past the exchanges, further explained in section 3.4.2.

3.4.2. Liquidity

This section describes all parameters used to regulate the liquidity in the simulation.

The variable *transSize* is used to determine the size of the trades in the simulation, initialized to $[1000, 5000]$. The variable itself does not influence the simulation since variable *minLiquidity*, which directly depends on *transSize* changes with it, and together they influence the amount of liquidity in the market.

The variable *minLiquidity* directly depends on *transSize*, and is set to ten average trades. The variable states the minimum liquidity on both the buy and sell sides before a liquidity order is forced. Furthermore, it is also the initial liquidity in the market. Since there will be more liquidity, price variability decreases when *minLiquidity* is a larger multiple of the average transaction size.

The speed of transactions is determined by *liqP*, initially set to one, which is a 'liquidity percentage'. From this percentage, the minimum and maximal time to a new trader *liqMin* and *liqMax* are derived as $liqMin = 4 + (1 - liqP) * 6$ and $liqMax = 8 + (1 - liqP) * 12$

respectively. When *liqP* decreases, the time between traders increases. Therefore, the change in price possible in a fixed time frame will decrease, and liquidity orders become less advantageous. This causes more market orders and, paradoxically, larger price movements. Since transactions occur more sparsely, pricing errors are fixed at a slower pace, the price is less stable, and the bid-ask spread increases.

The final liquidity parameter is variable *agressiveness*, currently set to 0.5, which determines the slope of the buy probability in equation 3.1. This variable increases the probability that a trader trades in the direction of his believed price. However, when the difference between the spot price and the believed price is small, we expect only a small preference for the direction of the believed price. Therefore, we have set the steepness of the curve relatively small.

3.4.3. Informed traders and shocks

This section describes all variables regarding (the number of) informed traders and the price shocks.

minInfP and *maxInfP* describe the range of probabilities that a trader is informed trader, while the current probability *infP* is set the *minInfP*. The informed traders push the price towards *trueP*, dampening price variability. Increasing either the boundaries or the current probability of being informed thus dampens the price volatility even further.

When the exchange is assumed to be in equilibrium, as at the simulation's start, we always have $infP = minInfP$. When a shock occurs, *infP* increases to *maxInfP* in a predetermined time, after which it decreases to *minInfP* again. Several variables regarding the shocks determine these times.

First, the variable *shocks* states the number of shocks that will occur in the simulation. It divides the total simulation time evenly among the number of shocks, saved as variable *shockTime*, which itself is divided into four parts: *shockWait*, *shockIncrTime*, *shockDecrTime* and *shockAfterTime*. These represent the time till the shock occurs, the time that *infP* increases linearly from *minInfP* to *maxInfP*, the time that in which it decreases linearly to *minInfP*, and a wait time till the next shock block. Currently, these times are set to 30%, 10%, 10% and 50% of *shockTime* respectively. Even though the length of *shockWait* and *shockAfterTime* do not really influence the simulation, *shockIncrTime* and *shockDecrTime* do. If these times are taken too short, the number of informed traders will not be big enough to cause the price to shift toward its new value. Therefore, the new information will not properly be incorporated into the price. When the times are too long, the price is kept artificially stationary after the new true price has already been reached.

3.4.4. Tested settings

To capture the influence of different parameters, we test for seven different scenarios given in table 3.3. Omitted variables are assumed to retain the value given in previous sections.

The basic scenario simulates the LOB and AMM for an asset with a reasonably stable price in a liquid market with small transaction costs. To capture the effect of the time horizon, we also simulate zero and two shocks with the same parameters. In the scenario of low liquidity, we test the exchanges when the frequency of the trades has decreased by fifty per cent. Furthermore, we also want to simulate how the metrics change when the number of informed traders increases or decreases. For this, the percentage of informed traders and the time for which there are many informed traders are updated. Finally, we compare the exchanges when the AMM fee is higher to simulate the assumption of a volatile market.

Scenario	**Basic setting**	**No shocks**	**Two shocks**	**Low liquidity**	**High fee**	**Informed market**	**Uninformed market**
Shocks	1	0	2	1	1	1	1
LiqP	1	1	1	0.5	1	1	1
Fee	0.0005	0.0005	0.0005	0.0005	0.01	0.0005	0.0005
MinInfP	0.1	0.1	0.1	0.1	0.1	0.2	0.05
MaxInfP	0.75	0.75	0.75	0.75	0.75	0.75	0.4
ShockWaitTime (as % of ShockTime)	0.3	0.3	0.3	0.3	0.3	0.3	0.3
ShockIncrTime (as % of ShockTime)	0.1	0.1	0.1	0.1	0.1	0.2	0.05
ShockDecrTime (as % of ShockTime)	0.1	0.1	0.1	0.1	0.1	0.2	0.05
ShockAfterTime (as % of ShockTime)	0.5	0.5	0.5	0.5	0.5	0.3	0.6

Table 3.3.: Seven different scenarios and the values of the parameters

3.5. Output

This section describes the two different data sources of the simulation. The first contains data about the executed orders, while the second contains the exchange state at various times.

Order data

The order data is mainly used to assess the validity of the simulation. It consists of twelve variables about each finished trade with the exchange. All but one variable are used for insight into the trader's identification and decision-making process. These eleven variables are not relevant to the actual results but give insight into anomalous transactions and indicate bugs.

The twelve columns are the name of the trader, the market price when the trader was initialized, and trade execution has finished, p_e, p_b, q, $time$, the actual time passed, A after trading, M after trading, the completion percentage and the kind of order (liquidity or market). Of all these variables, only the completion percentage is used in the results.

Exchange data

Metrics are computed on the exchange data. For each (partial) trade, the current price, time, total sell and buy volume, unit and 1000 unit bid-ask spread, $infP$ and the available number of assets and euros are saved. Note that the data excludes trades to obtain the correct ratio in the AMM. Furthermore, orders split due borders in the AMM are saved as separate orders.

4. Results

In this chapter, we present and discuss the results of the simulation. We do this in four parts. First, we validate the simulation in section 4.1. After that, results corresponding to information incorporation are presented in section 4.2, while the data regarding market liquidity is presented in section 4.3. Finally, the results concerning the completion percentages are given in section 4.4.

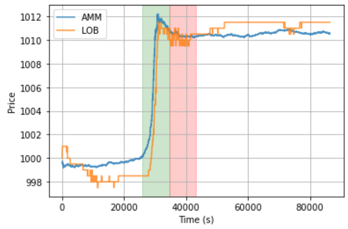
4.1. Validity

In this section, we check the validity of the simulation for each of the seven scenarios. We inspect whether the price graphs follow are realistic and explainable. Furthermore, we also check if the completion percentages of liquidity providers and market takers are realistic. Finally, we check if the parameter changes cause the predicted changes.

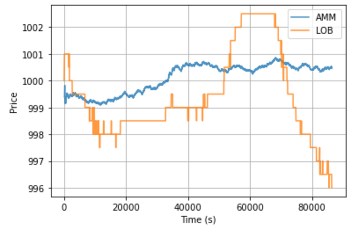
The prices are plotted in figure 4.1. We expect that the price oscillates around the true price when no new information is available. The price should move toward a new price without many regressions, while an overshoot should stay within reasonable bounds. When we look at the graphs of the scenarios, there are a few cases in which the plots seem unrealistic. In the uninformed scenario, both exchanges experience a significant overshoot of 50% before they revert to the true price. Appendix A gives the graphs for more extended time horizons with different settings. It turns out that the LOB will oscillate around the true price with increasing amplitude while the AMM remains close to the true price in 75% of the cases. We can conclude that the uninformed scenario holds less weight than other scenarios. The completion percentage underlines this conclusion for the AMM since the average completion percentage of a market taker is higher than that of the liquidity providers. We expect the reverse to be true since the liquidity providers already have additional costs of time and execution risk. Therefore, they should be favoured money-wise.

The second case in which the price series does not follow an expected curve is for the LOB in scenario 'no shock'. Appendix B gives an extended time horizon that shows that the price oscillates as expected, but the amplitude has increased to four. However, the average completion percentage of the liquidity provider lies beneath that of the market taker again is slightly lower. Therefore, we proceed with caution in this scenario but still consider it.

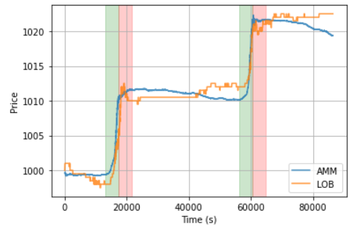
Section 3.4 gives several predictions when parameters are changed. We will check if the parameter changes have the expected effect. The first prediction was that an increased fee would cause a larger bid-ask spread in the AMM. Figures 4.6 and 4.7 show that this is indeed the case. Another prediction was that a decreased liquidity percentage would lead to increased price volatility. Figures 4.1a and 4.1d show that this is only true after a shock has taken place. However, this is to be expected since the initial liquidity restricts the price variability as long as it exists. After the price shift, at least one side of this initial liquidity has been filled, and the price depends on liquidity traders. The price variability immediately increases as expected. The third prediction was that an increased number of informed traders would decrease price variability, while a decreased number of informed traders would cause an increase. Figures 4.1, 4.1f and 4.1g show that the latter holds while the former does not. The reason for this is twofold. First, the number of informed traders remains high in the coloured areas. Therefore, the price is forced back to the true price, and every price change is inverted, increasing variability with smaller differences. The second reason may be that the additional number of informed traders do not have any effect since there already were enough informed traders in the basic scenario. We do not question the validity of the simulation since the different results can be explained. The final prediction was that increased *shockIncrTime* and *shockDecrTime* would cause forced price stability while decreasing them could cause problems incorporating all information. Figure 4.1f shows that the price is kept close to the true price by reverting the price after the information has been incorporated. Therefore, price stability is enforced. Furthermore, figure 4.1g indeed shows that information incorporation is more difficult than in the basic scenario.



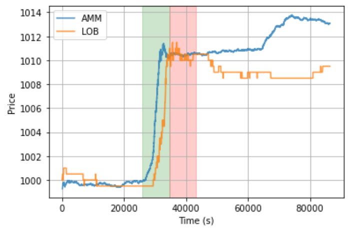
(a) Basic



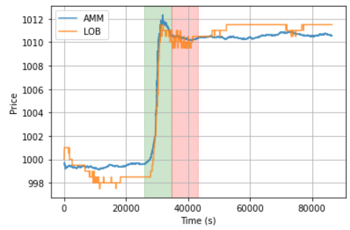
(b) No shocks



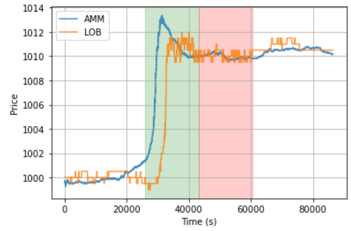
(c) Two shocks



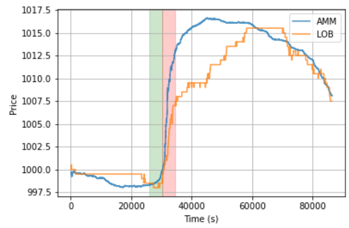
(d) Low liquidity



(e) High fee



(f) Informed



(g) Uninformed

Figure 4.1.: Spot price development over time for different scenarios

4.2. Information incorporation

In section 2.2.1, we have seen that information incorporation is measured in terms of market efficiency and price discovery, which are closely related.

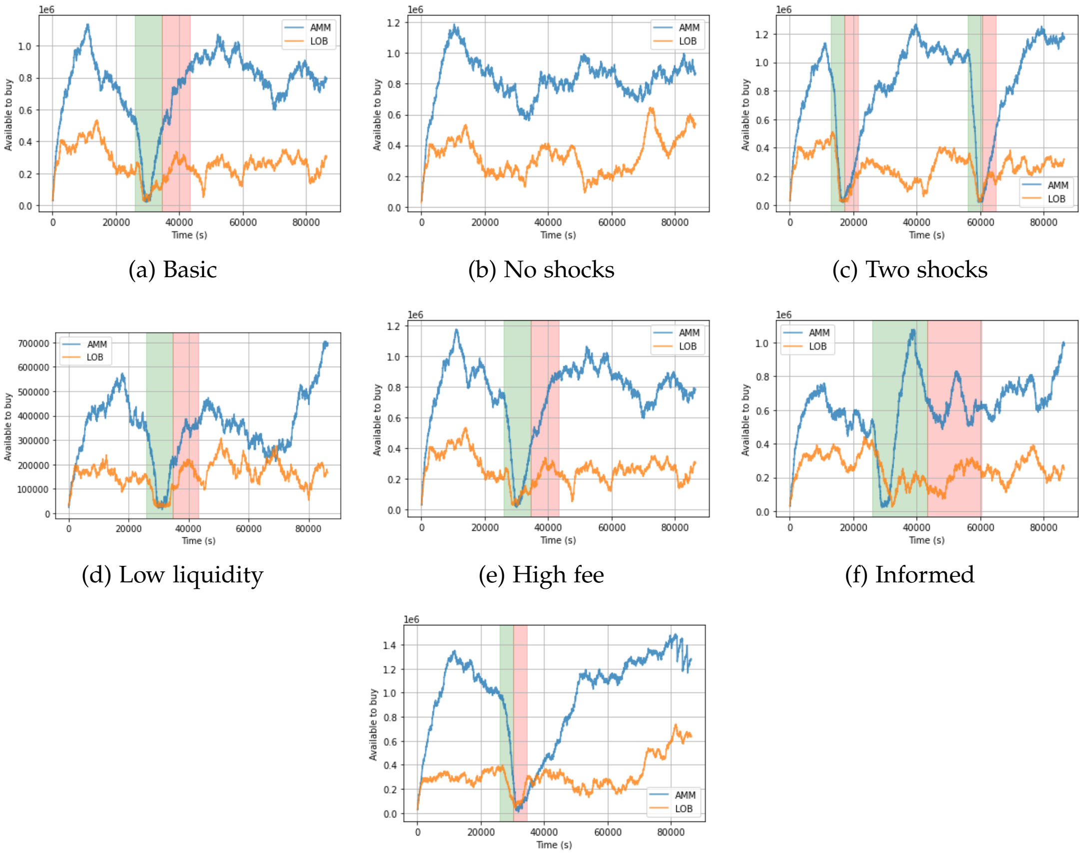
Market efficiency checks whether all information is incorporated. If it is, the price should follow a random walk. In contrast, price discovery looks at the period in which the information is not yet incorporated. Therefore, we split the simulation into multiple parts (depending on the number of shocks). When information is introduced, we compute price discovery metrics and test the EMH on the other parts. The metrics results are given in table 4.1.

			Basic	**No shock**	**2 shocks**	**Low liquidity**	**High fee**	**Informed**	**Uninformed**
LOB	**Price discovery**	Time	4827	-	5105 4256	7821	4827	6921	18996
		Trades	766	-	964 731	730	766	1137	3040
		Buy volume	1,096,429	-	1,424,573 987,229	989,532	1,096,429	1,716,834	4,027,275
		Sell volume	285,641	-	328,106 317,215	318,427	285,641	364,755	1,736,994
	Market efficiency	Adfuller p-values	0.0055 0.1654	0.9483	0.3597 0.7198 0.0415	0.7519 0.1272	0.0055 0.1654	0.0741 0.1733	0.9214 0.8624
		KPSS p-values	0.01 0.01	0.01	0.01 0.01 0.01	0.01 0.01	0.01 0.01	0.01 0.01	0.01 0.01
AMM	**Price discovery**	Time	4009	-	4211 4099	5517	4478	3720	6831
		Trades	880	-	930 906	647	931	785	1327
		Buy volume	1,632,044	-	1,858,759 1,861,457	1,259,215	1,860,471	1,490,918	2,599,939
		Sell volume	481,011	-	628,133 700,971	398,269	599,518	430,570	1,141,619
	Market efficiency	Adfuller p-values	0.9844 0.3815	0.7168	0.1654 0.9782 1.0	0.4213 0.8629	0.6378 0.4006	0.9969 0.5514	0.3021 1.0
		KPSS p-values	0.01 0.01	0.01	0.01 0.01 0.01	0.01 0.01	0.01 0.01	0.01 0.01	0.01 0.01

Table 4.1.: Results of information incorporation metrics

The data regarding market efficiency is clear. An adfuller p-value greater than 0.05 points to a random walk, whereas a KPSS p-value smaller than 0.05 does the same. We see that all cases have p-values of 0.01 for the KPSS tests. Since the build-in KPSS test in python cannot obtain a value smaller than 0.01, this is not uncommon. Furthermore, the KPSS tests indicate that the price followed a random walk for both the LOB and AMM in all scenarios. However, this is not the case for the adfuller tests where the LOB does not follow a random walk in the basic, high fee and two shock settings. It should be noted that the basic and high fee scenarios are the same for the LOB.

Rather than conclusive answering whether an exchange has good price discovery, the metrics can only be used to compare the two exchanges. We look at the time, number of trades and buy and sell volume needed to transition from the current to the new price. The AMM is always faster than the LOB, which indicates that the AMM has better price discovery. However, this conclusion does not hold for all other metrics. The number of trades favours the AMM in four of seven cases, while the buy and sell volumes are always smaller using the LOB. Therefore we can conclude that the AMM moves faster in the sense of time but also needs more trading to obtain the updated time. A limit order disappears when it is filled (or the time has run out), which is not necessarily true for the range order. Therefore, the total available liquidity remains higher in the AMM, as shown in figures 4.2 and 4.3, and more assets have to be traded in order to update the price. The total amount of assets bought and sold over time is plotted in figure 4.4. The AMM is still faster since more market orders are submitted than in the LOB. Furthermore, different strategies for liquidity providers might improve the price discovery of the AMM even further.



(a) Basic

(b) No shocks

(c) Two shocks

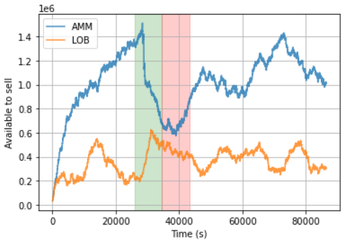
(d) Low liquidity

(e) High fee

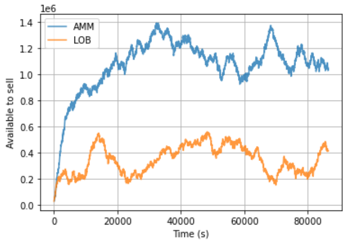
(f) Informed

(g) Uninformed

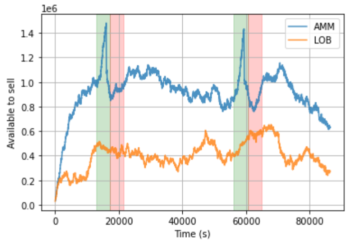
Figure 4.2.: Total buy quote over time for different scenarios



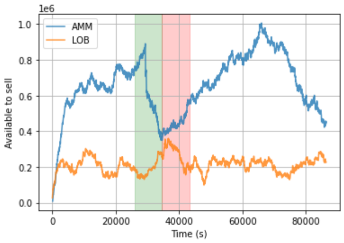
(a) Basic



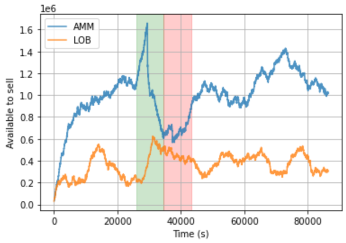
(b) No shocks



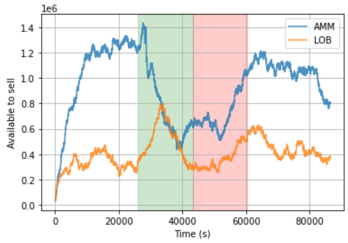
(c) Two shocks



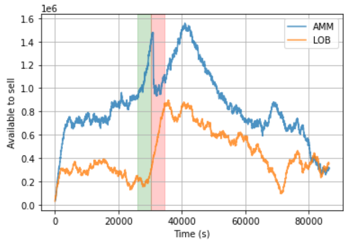
(d) Low liquidity



(e) High fee

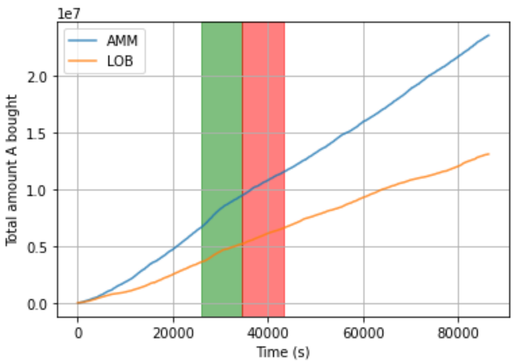


(f) Informed

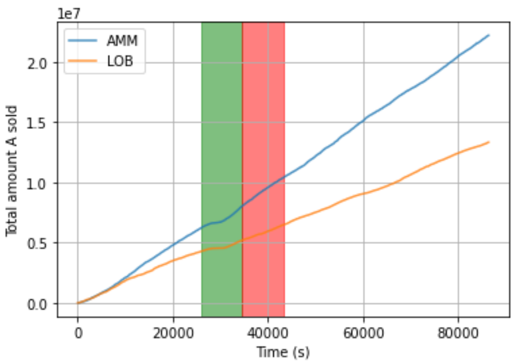


(g) Uninformed

Figure 4.3.: Total sell quote over time for different scenarios



(a) Buy



(b) Sell

Figure 4.4.: Total amount of A bought/sold over time in the basic scenario.

4.3. Market liquidity

This section discusses liquidity in the LOB and AMM. As seen in section 2.2.2, this can be measured in the bid-ask spread and unit quote sizes. We will now discuss these separately.

The bid-ask spread is a measure of the costs of transactions. We expect a small bid-ask spread in a liquid market. The unit bid-ask spreads are given in figure 4.6. There is a clear distinction between the LOB and AMM.

Liquidity traders create the bid-ask spread in the LOB. Therefore, the results are directly correlated to the implemented behaviour of the traders. The graphs show that the unit bid-ask spread generally stays in the range of one to four. The big exception is when a shock occurs, and the price of one side moves faster than the other.

In the AMM, liquidity traders only partially determine the bid-ask spread. Without liquidity at the current price, the price cannot move, and liquidity traders cannot trade or earn fees. Therefore, they are incentivized to provide liquidity between the current price and their believed price. Finally, a liquidity trader cannot provide liquidity at one price. The results can be found in figure 4.5, in which the liquidity is plotted at several times just before, during and after a shock. Liquidity is centered around the true price at $t = 25924$ and $p_t = 1000$. After this time, the true price is set to 1010 to simulate a shock in information. We see that a liquidity bridge is built from the current to the new price. Therefore, the price slippage is only a small component of the bid-ask spread in the AMM. The main component is the fee, which is 0.05% (or 1% in the case of a high fee). The bid-ask spread is €1.01 (or €20.20 in case of a 1% fee) when $p_m = 1005$ and there is no slippage. The unit bid-ask spread in figure 4.6 stays very close to these values.

However, it can be argued that the unit bid-ask spread is not a fair comparison of the LOB and AMM. In the simulation, the order sizes are in the order of thousands. Therefore, the real unit used in the simulation is one thousand assets. Since slippage becomes more evident with larger orders, this might change the conclusion. The 1000 bid-ask spread is plotted in figure 4.7. Here, we have capped the bid-ask spread at ten for all scenarios except for the high fee scenario. For the AMM, some influence of the liquidity providers can be found at the main peaks of the bid-ask spread of the LOB, but it is still very minimal and fleeting. The LOB does not change much, except for an additional peak at the beginning due to the initialization.

Therefore, we can conclude that the cost of liquidity is lower and more consistent in the AMM than the LOB, as long as the fee is appropriate.

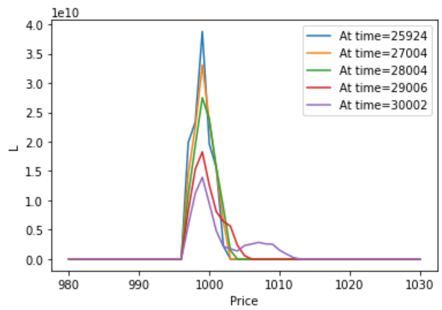
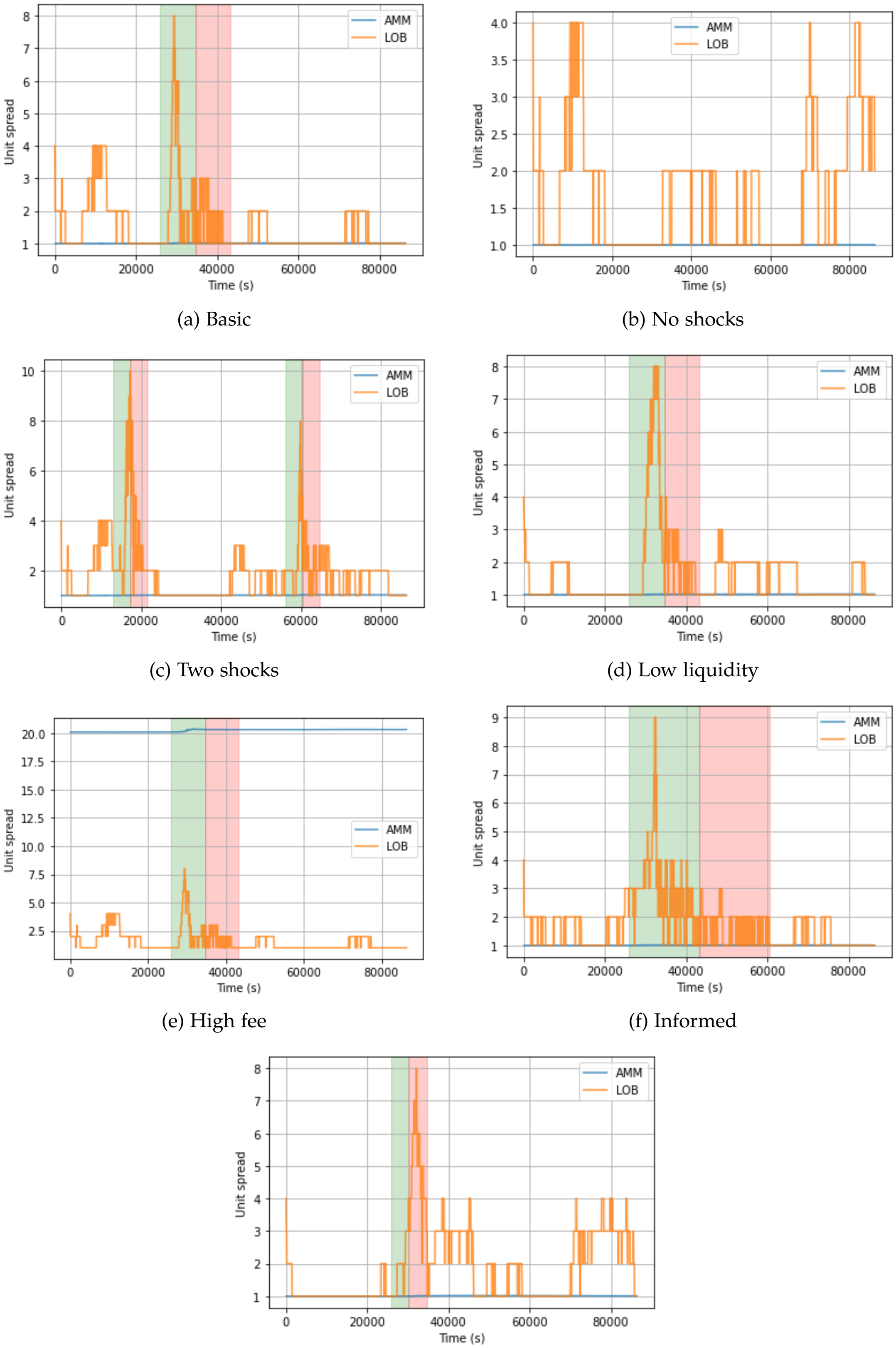
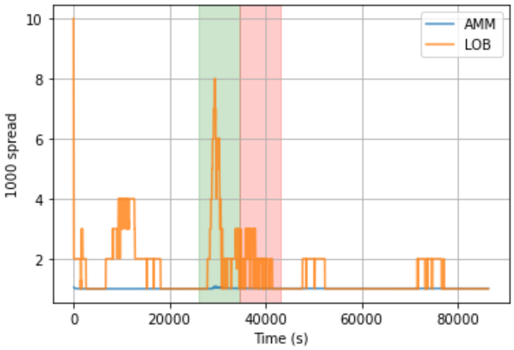


Figure 4.5.: L at different times during the shock in the basic scenario.

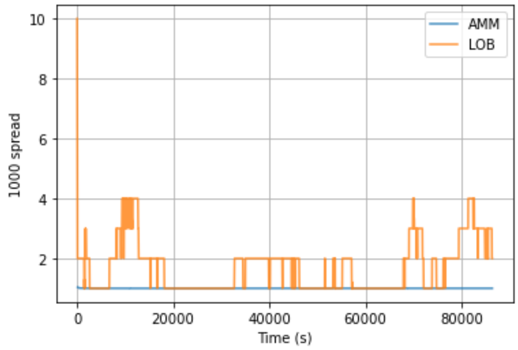


(a) Basic
(b) No shocks
(c) Two shocks
(d) Low liquidity
(e) High fee
(f) Informed
(g) Uninformed

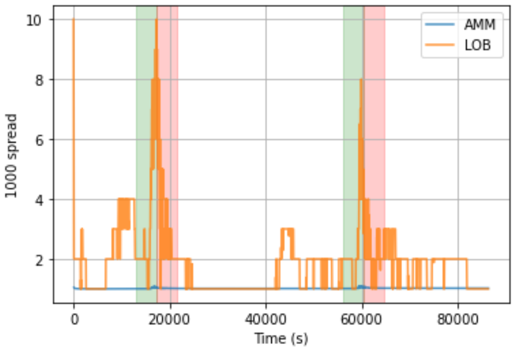
Figure 4.6.: Unit bid-ask over time for different scenarios



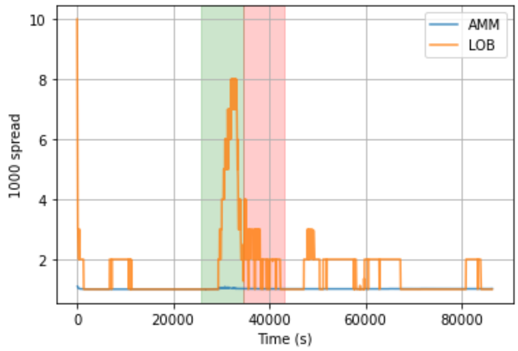
(a) Basic



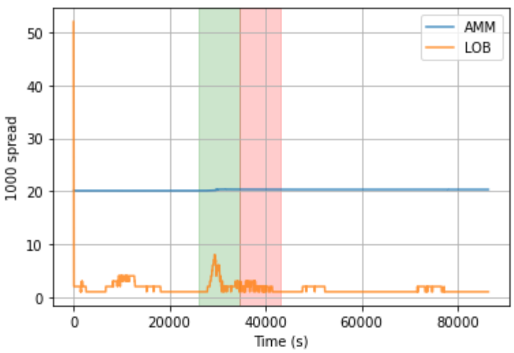
(b) No shocks



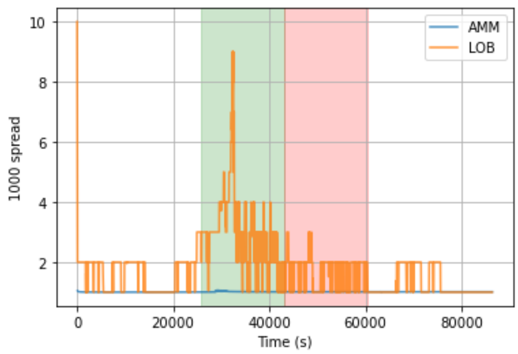
(c) Two shocks



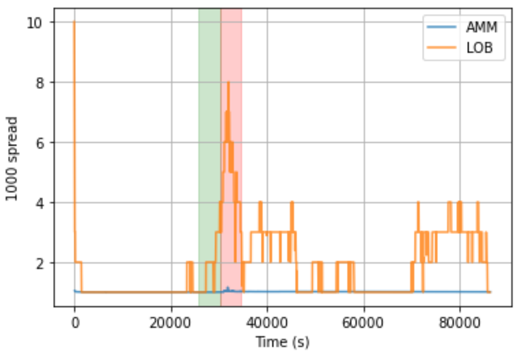
(d) Low liquidity



(e) High fee



(f) Informed

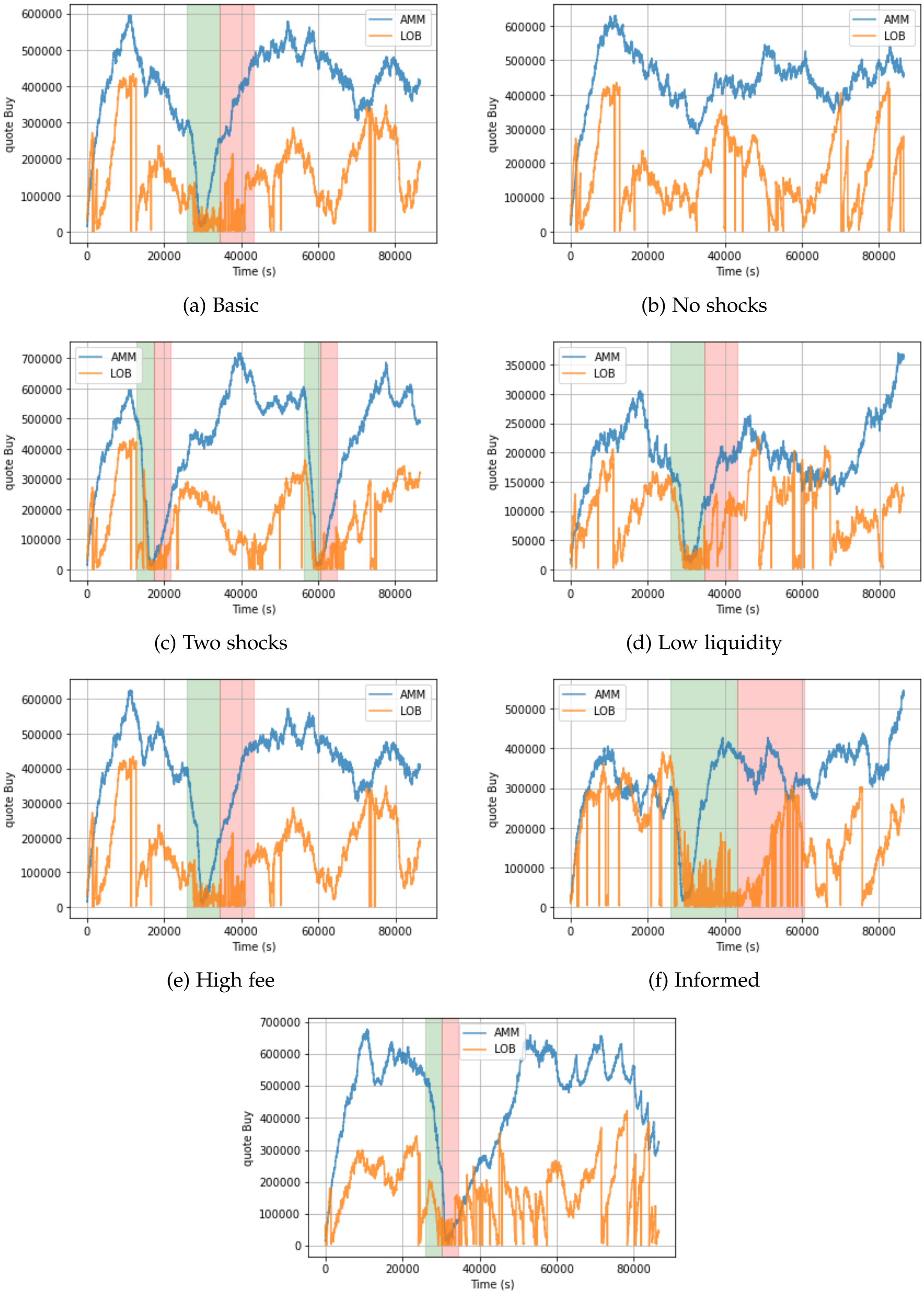


(g) Uninformed

Figure 4.7.: 1000 bid-ask over time for different scenarios

The quote size indicates price slippage and is a complementary metric for the bid-ask spread. We have plotted the unit quote sizes in figures 4.8 and 4.9. Due to frequent undercuts, a temporary drop to one is commen in the LOB. A better comparison would be to compare the upper boundary of the graph of the LOB graphs to the quote size of the AMM. There is still a clear distinction since liquidity is always higher in the AMM than the LOB.

However, the unit quote size is defined differently for the LOB and the AMM. In the LOB, the unit quote size is the number of assets that can be bought or sold at the best ask or bid. However, this change would only change the mid-price by half a currency unit. In the AMM, on the other hand, the unit quote size is defined as the number of assets that can be bought or sold to alter the price by one unit of the currency. For the market takers, there is no difference between the two exchanges since their price changes by one unit in both cases. However, the market price changes by half a unit in the LOB while it changes one unit in the AMM. Therefore, this definition of the quote size allows for more price movements in the AMM and likely more liquidity. We can also compare another definition of the quote sizes, in which the market price changes by half a unit. The half-unit quote sizes are given in figures 4.10 and 4.11. The results are not as clear as before but still favour the AMM in most cases.



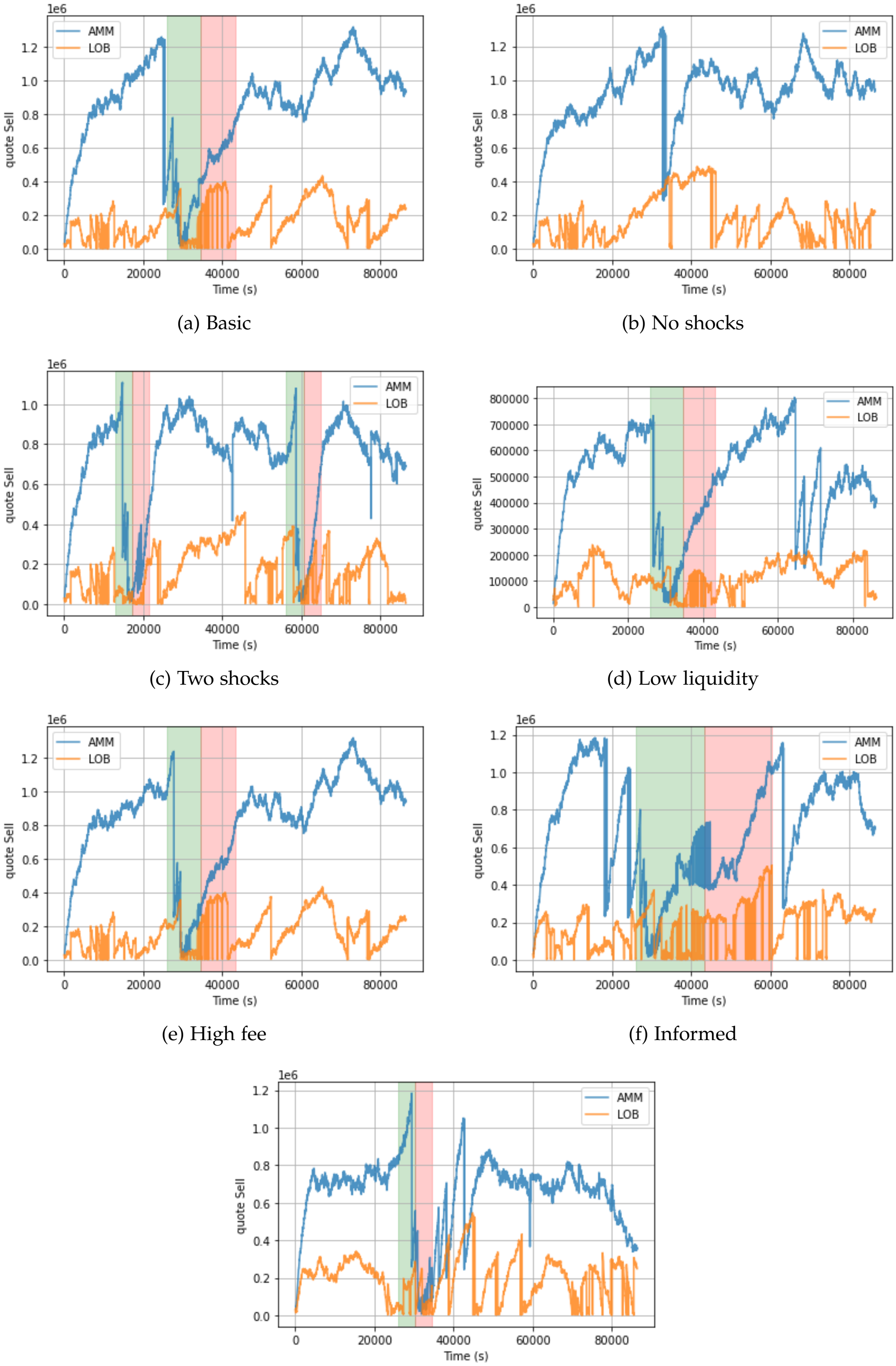
(a) Basic (b) No shocks

(c) Two shocks (d) Low liquidity

(e) High fee (f) Informed

(g) Uninformed

Figure 4.8.: Unit buy quote over time for different scenarios according to the market taker.



(a) Basic

(b) No shocks

(c) Two shocks

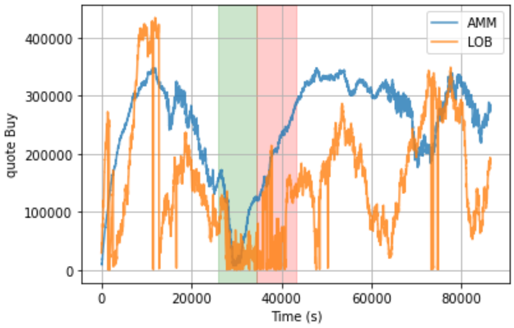
(d) Low liquidity

(e) High fee

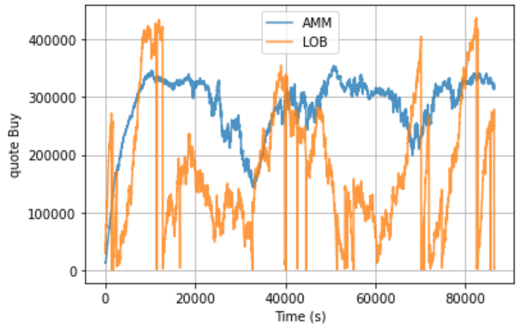
(f) Informed

(g) Uninformed

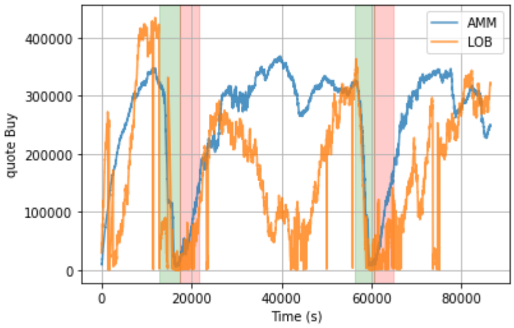
Figure 4.9.: Unit sell quote over time for different scenarios according to the market taker.



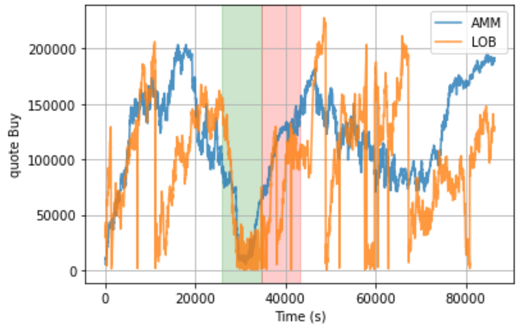
(a) Basic



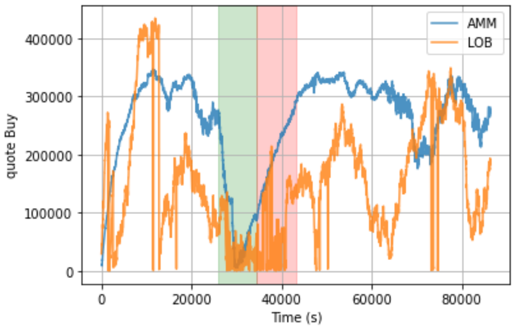
(b) No shocks



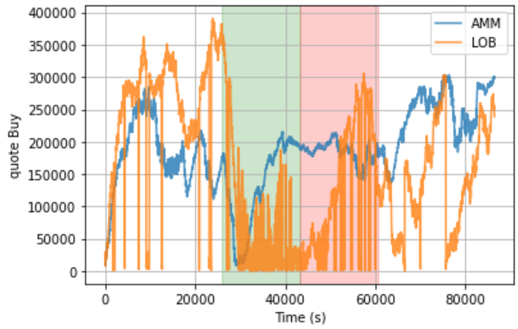
(c) Two shocks



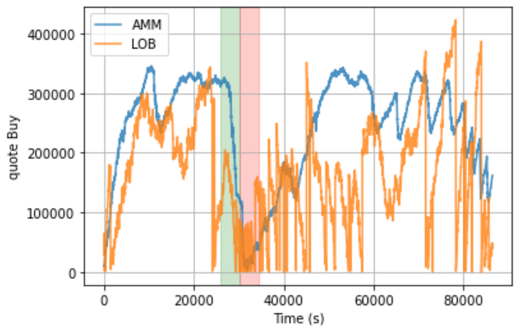
(d) Low liquidity



(e) High fee

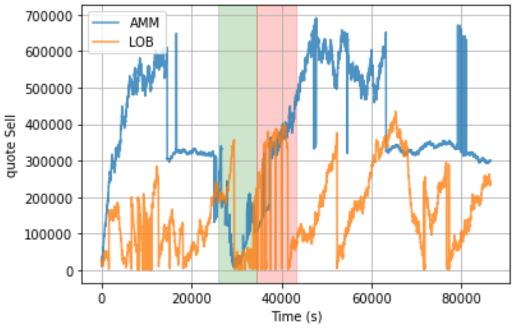


(f) Informed

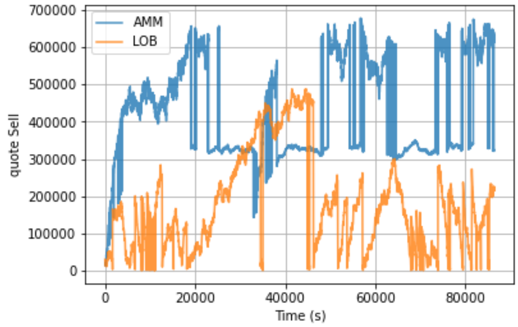


(g) Uninformed

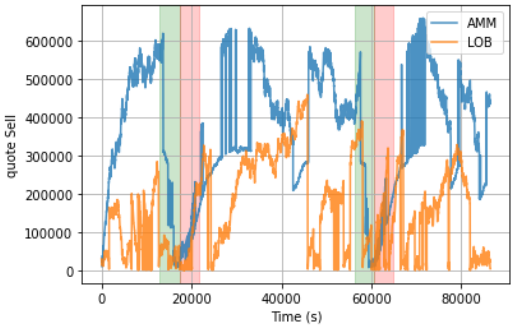
Figure 4.10.: Unit buy quote over time for different scenarios according to the spot price.



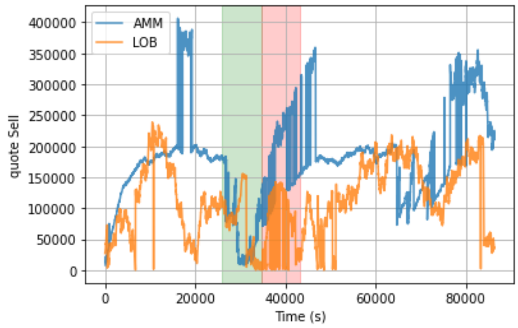
(a) Basic



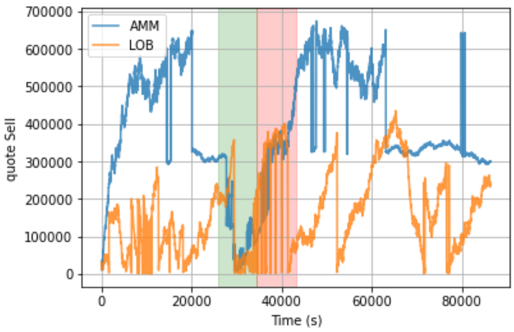
(b) No shocks



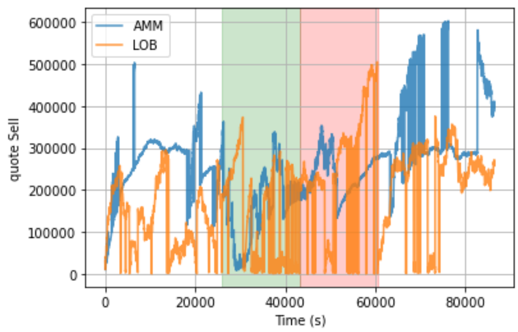
(c) Two shocks



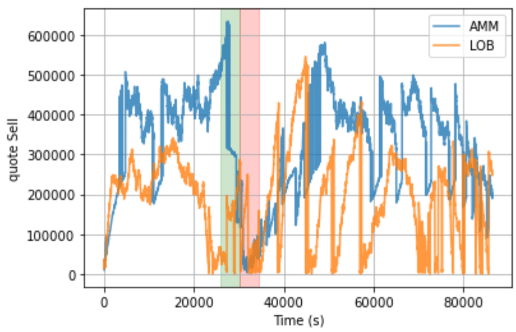
(d) Low liquidity



(e) High fee



(f) Informed



(g) Uninformed

Figure 4.11.: Unit sell quote over time for different scenarios according to the spot price.

4.4. Completion Percentage

The final metric on which we compare the LOB and AMM is the completion percentage, which measures traders' satisfaction. Table 4.2 presents the average completion percentage for all traders, liquidity providers and market takers. Additionally, the standard deviation of all traders is given. Figure 4.12 presents the histograms of the completion percentage to give insight into the distribution.

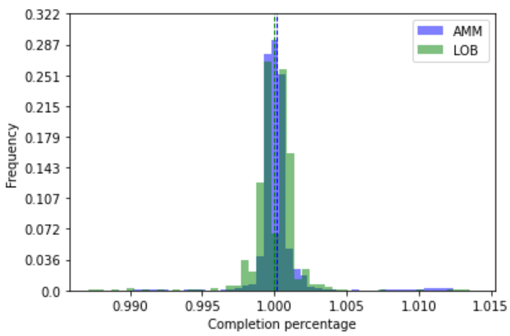
		Basic	No shock	2 shocks	Low liquidity	High fee	Informed	Uninformed
LOB	**Mean**	1.000004	0.999817	1.000075	0.999883	1.000004	1.000082	1.000031
	Std	0.001746	0.001181	0.002358	0.001710	0.001746	0.002058	0.002216
	Liquidity providers	1.000205	0.999779	1.000558	1.000136	1.000205	1.000444	1.000315
	Market takers	0.999769	0.999863	0.999489	0.999601	0.999769	0.999679	0.999677
AMM	**Mean**	1.000192	0.999996	1.000113	1.000276	0.999483	1.000239	1.000031
	Std	0.001643	0.000551	0.002536	0.001753	0.009702	0.001691	0.002433
	Liquidity providers	1.000319	0.999983	1.000478	1.000468	0.999352	1.000371	1.000127
	Market takers	0.999960	1.000025	0.999445	0.999918	0.999720	1.000053	0.999820

Table 4.2.: Completion percentage statistics

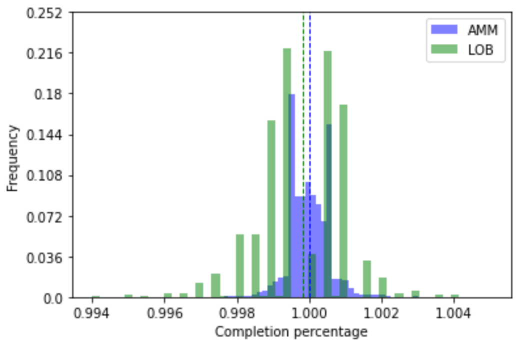
The traders are more satisfied using the AMM in all cases except for the higher fee. Furthermore, the standard deviations are very close and small except for scenarios 'no shock' and 'high fee'. In the case of the higher fee, even the liquidity providers lose since the fee is not balanced compared to the stability and price of the asset.

In four of the seven scenarios, the LOB provides better results for the liquidity providers, while the market takers are only better off in two. This suggests that the AMM has a better balance between the two different kinds of traders.

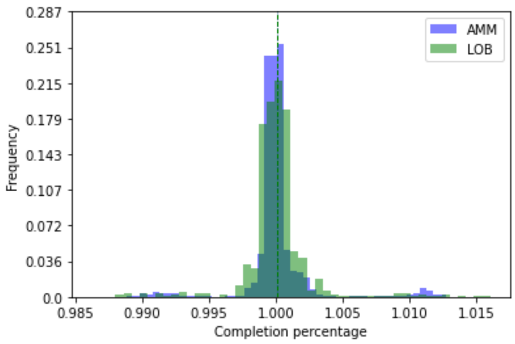
Finally, although all standard deviations are close, figure 4.12 tells a different story. In six out of seven scenarios, the lowest completion percentages are obtained using the LOB while the highest are acquired in the AMM. Furthermore, the AMM seems denser around one than the LOB is in all these six cases. Therefore, we conclude that the AMM mediates better between the bid and ask side and liquidity provider and market taker.



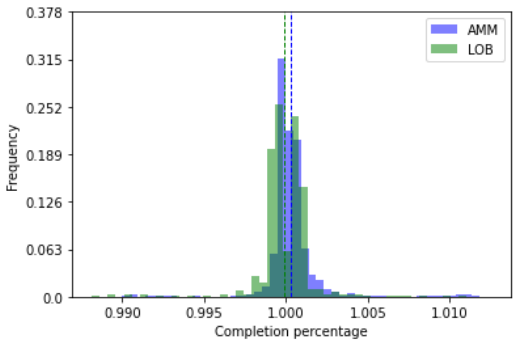
(a) Basic



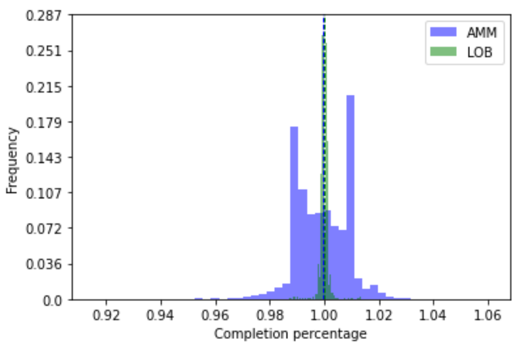
(b) No shocks



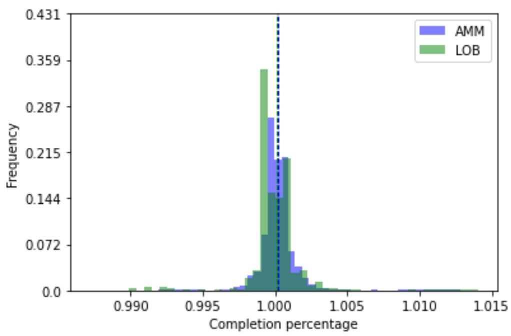
(c) Two shocks



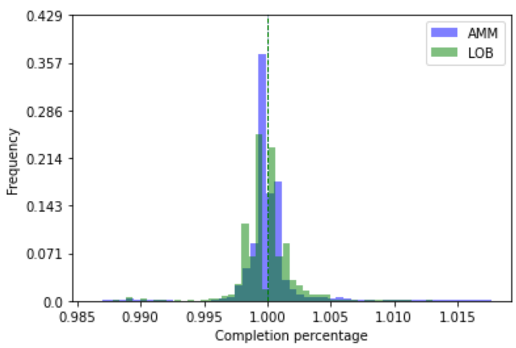
(d) Low liquidity



(e) High fee



(f) Informed



(g) Uninformed

Figure 4.12.: Completion percentage distributions for different scenarios

5. Conclusion

5.1. Summary

This thesis aimed to compare the performance of the Limit order book (LOB) and Automated market maker (AMM) in terms of market quality with the help of a simulation. We have defined the requirements and desired properties of an exchange. The requirements are needed to have a functional exchange, while the desired properties measure competitiveness. We have given a comprehensive overview of both the LOB and AMM and shown that both satisfy the minimal requirements of an exchange. To compare the LOB and AMM, we have built a simulation based on heuristics that is used to measure the three main components of market quality: information incorporation, market efficiency and satisfaction in the form of completion percentages. Seven different scenarios have been tested through different settings in the simulation.

Table 5.1 shows which exchanges outperformed the other for each metric, each part of market quality and the final winner per scenario. The AMM outperformed the LOB five out of seven times in information incorporation, six times in market liquidity and six times in the completion percentages. The LOB outperformed the AMM one, zero and one time respectively. The AMM was the overall winner in six of the seven scenarios, making it the best in this thesis.

		Basic	**No shock**	**Two shocks**	**Low liquidity**	**High fee**	**Informed**	**Uninformed**
Validity		Good	Good	Good	Good	Good	Good	Bad
Information incorporation	**Price Efficiency**	AMM	-	AMM	AMM	LOB	AMM	AMM
	Market efficiency	AMM	-	-	-	AMM	-	-
		AMM	-	AMM	AMM	LOB	AMM	AMM
Market liquidity	**Unit bid-ask**	AMM	AMM	AMM	AMM	LOB	AMM	AMM
	1000 bid-ask	AMM	AMM	AMM	AMM	LOB	AMM	AMM
	Buy quote market taker	AMM	AMM	AMM	AMM	AMM	AMM	AMM
	Sell quote market taker	AMM	AMM	AMM	AMM	AMM	AMM	AMM
	Buy quote spot price	AMM	AMM	AMM	-	AMM	LOB	-
	Sell quote spot price	AMM	AMM	AMM	AMM	AMM	-	AMM
		AMM	AMM	AMM	AMM	-	AMM	AMM
Completion percentage		AMM	AMM	AMM	AMM	LOB	AMM	AMM
		AMM	**AMM**	**AMM**	**AMM**	**LOB**	**AMM**	**AMM**

Table 5.1.: Overview of results for each metric

5.2. Strengths and weaknesses

This section describes the strengths and weaknesses of the simulation.

The simulation tests seven different scenarios that test the LOB and AMM for different varieties of price variability, liquidity, informed traders and fees. Since six out of seven scenarios favour the AMM, we confidently conclude that the AMM outperforms the LOB.

However, we have seen that the bid-ask spread depends entirely on the trading strategy of the traders in the LOB. Furthermore, we assumed rational traders who try to optimize their gains, which is not always realistic. The results may therefore change when the goals and trading behaviour of the trader change.

Another limitation of the model is the discretization, which is implemented differently than in real-life examples. A different form of rounding might impact the returned values and decisions of the traders, resulting in different trades and thus results. We have opted to keep a light discretization since this is implemented differently for real-life examples of the LOB and AMM.

However, the simulation is compartmentalized, meaning parts of the code can be easily interchanged. Therefore, other implementations of trading behaviour, metrics and exchanges can be used with little adaptation.

5.3. Future research

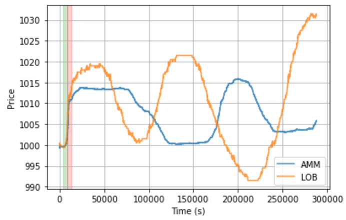
The simulation is based on heuristics instead of real-world data. When similar data for both the LOB and AMM is obtained, the same metrics can be used to derive updated conclusions. Furthermore, it can be tested whether trading data resulting from the simulation can be used as an alternative to real-world data since it is both difficult to obtain and often costly.

This thesis did not consider ease of use. Even though better results can be obtained using the AMM, this might not be possible for traders without spending several hours understanding the mechanics. Future research can indicate whether this prevents new traders from using an AMM.

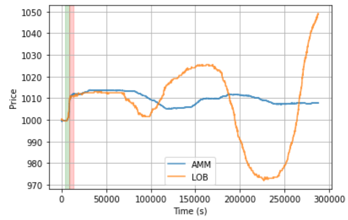
It is unclear what effect discretization has on the performance of the LOB and AMM in terms of the completion percentage. Discretization can work in favour or against a trader, which might cause more variable completion percentages. Therefore, future research is needed to find the effect on the results.

Finally, research might optimize the AMM even further. It can do so by finding optimal trading fees, possibly making them liquidity dependent. Another possible improvement of the AMM is the automatic 'fulfilment' of liquidity orders as in the LOB.

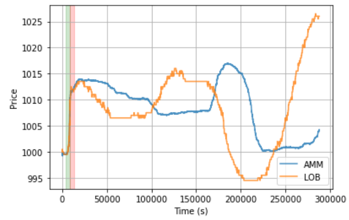
A. Price graphs for uninformed scenario



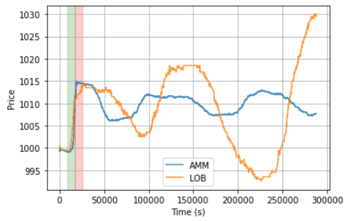
(a) $infP = 0.05$,
$maxInfP = 0.4$,
$shockIncrTime = 0.05$,
$shockDecrTime = 0.05$



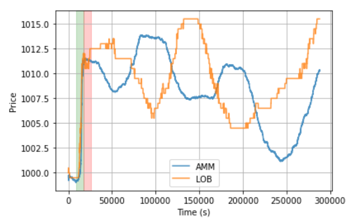
(b) $infP = 0.05$,
$maxInfP = 0.5$,
$shockIncrTime = 0.05$,
$shockDecrTime = 0.05$



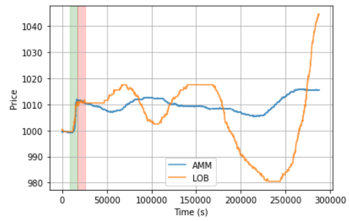
(c) $infP = 0.05$,
$maxInfP = 0.6$,
$shockIncrTime = 0.05$,
$shockDecrTime = 0.05$



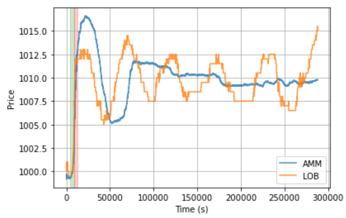
(d) $infP = 0.05$,
$maxInfP = 0.4$,
$shockIncrTime = 0.1$,
$shockDecrTime = 0.1$



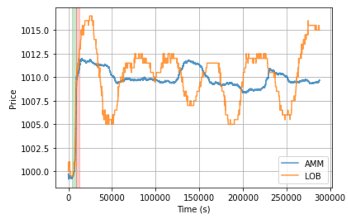
(e) $infP = 0.05$,
$maxInfP = 0.5$,
$shockIncrTime = 0.1$,
$shockDecrTime = 0.1$



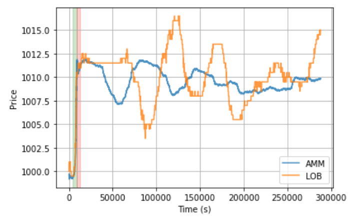
(f) $infP = 0.05$,
$maxInfP = 0.6$,
$shockIncrTime = 0.1$,
$shockDecrTime = 0.1$



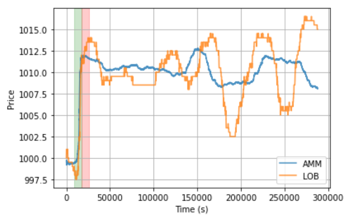
(g) $infP = 0.1$,
$maxInfP = 0.4$,
$shockIncrTime = 0.05$,
$shockDecrTime = 0.05$



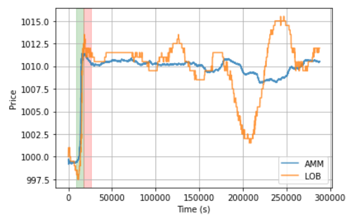
(h) $infP = 0.1$,
$maxInfP = 0.5$,
$shockIncrTime = 0.05$,
$shockDecrTime = 0.05$



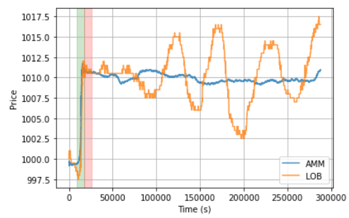
(i) $infP = 0.1$,
$maxInfP = 0.6$,
$shockIncrTime = 0.05$,
$shockDecrTime = 0.05$



(j) $infP = 0.1$,
$maxInfP = 0.4$,
$shockIncrTime = 0.1$,
$shockDecrTime = 0.1$

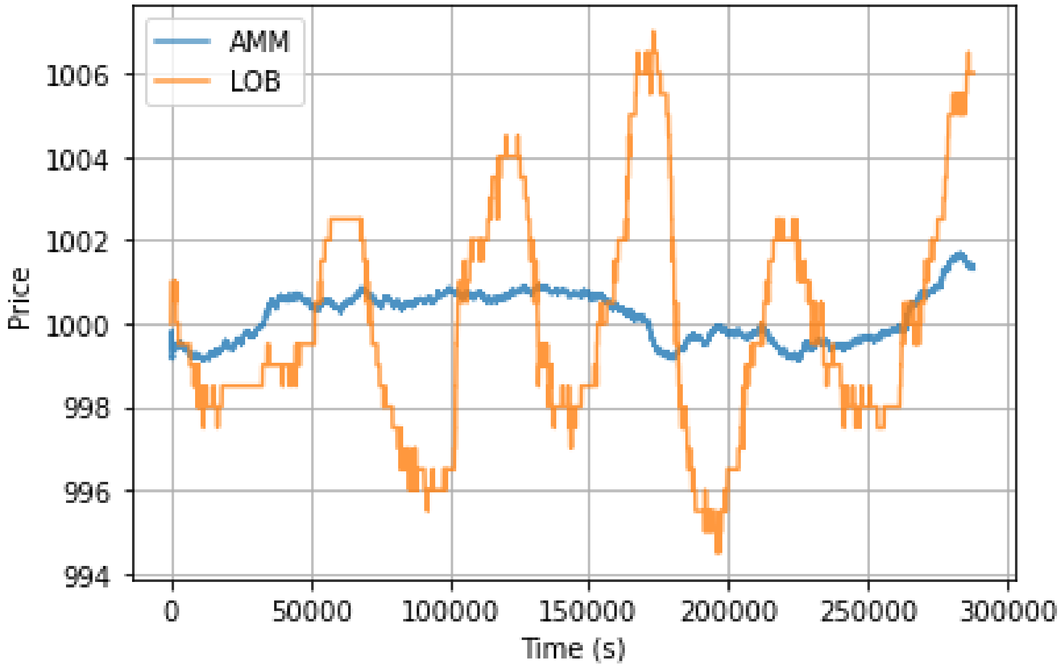


(k) $infP = 0.1$,
$maxInfP = 0.5$,
$shockIncrTime = 0.1$,
$shockDecrTime = 0.1$



(l) $infP = 0.1$,
$maxInfP = 0.6$,
$shockIncrTime = 0.1$,
$shockDecrTime = 0.1$

B. Extended graph of no shock



Bibliography

[1] "Best Execution under MiFID Peer Review Report," ESMA, Tech. Rep., Feb. 2015.

[2] C. Mack. "Assessing market quality." (Dec. 2015), [Online]. Available: https://www.nasdaq.com/articles/assessing-market-quality-2015-12-22 (visited on 03/02/2022).

[3] M. Bennouri, "Trading Mechanisms in Financial Markets: A Comparison Between Auction and Dealership Markets," 2014. DOI: 10.1007/978-3-319-05212-0{_}3.

[4] Y. C. Huang and J. H. Chou, "Order imbalance and its impact on market performance: Order-driven vs. quote-driven markets," *Journal of Business Finance and Accounting*, vol. 34, no. 9-10, pp. 1596–1614, Nov. 2007, ISSN: 0306686X. DOI: 10.1111/J.1468-5957.2007.02038.X.

[5] K. Malinova and A. Park, "Liquidity, volume and price efficiency: The impact of order vs. quote driven trading," *Journal of Financial Markets*, vol. 16, no. 1, pp. 104–126, Feb. 2013, ISSN: 1386-4181. DOI: 10.1016/J.FINMAR.2012.09.002.

[6] C. Cao, O. Hansch, and X. Wang, "The information content of an open limit-order book," *Journal of Futures Markets*, vol. 29, no. 1, pp. 16–41, Jan. 2009, ISSN: 02707314. DOI: 10.1002/FUT.20334.

[7] "Uniswap (v3) trade volume and market listings." (Feb. 2022), [Online]. Available: https://coinmarketcap.com/exchanges/uniswap-v3/ (visited on 02/15/2022).

[8] "Uniswap (v3) trade volume and market listings." (Jun. 2021), [Online]. Available: https://web.archive.org/web/20210613115431/https://coinmarketcap.com/exchanges/uniswap-v3/ (visited on 02/15/2022).

[9] CoinMarketCap. "Top defi tokens by market capitalization." (2022), [Online]. Available: https://coinmarketcap.com/view/defi/ (visited on 02/15/2022).

[10] CoinGecko. "Top decentralized exchanges on coingecko by trading volume." (2020), [Online]. Available: https://web.archive.org/web/20201124140229/https://www.coingecko.com/en/dex (visited on 12/13/2021).

[11] CoinGecko. "Top decentralized exchanges on coingecko by trading volume." (2021), [Online]. Available: https://www.coingecko.com/en/dex (visited on 12/13/2021).

[12] "Automated market maker (amm) [updated]." (Mar. 2022), [Online]. Available: https://coinmarketcap.com/alexandria/glossary/automated-market-maker-amm (visited on 03/02/2022).

[13] "The Nasdaq Stock Market Form 1 - Exhibit E," U.S. Securities and Exchange Commission., Tech. Rep., p. 7.

[14] "Serum: Return of the order book." (Dec. 2021), [Online]. Available: https://messari.io/article/serum-return-of-the-order-book (visited on 03/02/2022).

[15] E. Hu, P. Hughes, J. Ritter, P. Vegella, and H. Zhang, "Tick Size Pilot Plan and Market Quality," securities and exchange commission, Tech. Rep., 2018. [Online]. Available: https://www.sec.gov/rules/other/2014/34-72460.pdf..

[16] J. Nie, "High-Frequency Price Discovery and Price Efficiency on Interest Rate Futures," *Journal of Futures Markets*, vol. 39, no. 11, pp. 1394–1434, Nov. 2019, ISSN: 10969934. DOI: 10.1002/FUT.22016.

[17] E. F. Fama, "Efficient Capital Markets: A Review of Theory and Empirical Work," *The Journal of Finance*, vol. 25, no. 2, p. 383, May 1970, ISSN: 00221082. DOI: 10.2307/2325486.

[18] T. J. Putniņš, "What do price discovery metrics really measure?" *Journal of Empirical Finance*, vol. 23, pp. 68–83, Sep. 2013, ISSN: 0927-5398. DOI: 10.1016/J.JEMPFIN.2013.05.004.

[19] A. Gabrielsen, M. Marzo, and P. Zagaglia, "Measuring Market Liquidity: An Introductory Survey," *SSRN Electronic Journal*, Dec. 2011. DOI: 10.2139/SSRN.1976149. [Online]. Available: https://papers.ssrn.com/abstract=1976149.

[20] M. J. Fleming, "Measuring Treasury Market Liquidity," *Economic Policy Review*, vol. 9, no. 3, Sep. 2003. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=789844#.

[21] J. M. Karpoff, "The Relation Between Price Changes and Trading Volume: A Survey," *The Journal of Financial and Quantitative Analysis*, vol. 22, no. 1, p. 109, Mar. 1987, ISSN: 00221090. DOI: 10.2307/2330874.

[22] T. Marsh and K. Rock, "Exchange listing and liquidity: A comparison of the american stock exchange with the nasdaq national market system," *American stock Exchange Transaction Data Research Project*, vol. 2, 1986.

[23] P. L. Bernstein, "Liquidity, Stock Markets, and Market Makers," *Financial Management*, vol. 16, no. 2, p. 62, 1987, ISSN: 00463892. DOI: 10.2307/3666004. [Online]. Available: https://www.jstor.org/stable/3666004?casa_token=c2XoTBMyd6UAAAAA%3AxqhuoN9MnyRIlGidt8MZg8eH-JVYGK5nyMQpsnCPD41LltMSOwSKP_wO5Wm560ux3_ob4lfmchu0IZP5WXctC5Gc seq=1.

[24] F. Black, "Noise," *The Journal of Finance*, vol. 41, no. 3, pp. 528–543, Jul. 1986, ISSN: 1540-6261. DOI: 10.1111/J.1540-6261.1986.TB04513.X. [Online]. Available: https://onlinelibrary.wiley.com/doi/full/10.1111/j.1540-6261.1986.tb04513.x%20https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-6261.1986.tb04513.x%20https://onlinelibrary.wiley.com/doi/10.1111/j.1540-6261.1986.tb04513.x.

[25] D. A. Dickey and W. A. Fuller, "Likelihood Ratio Statistics for Autoregressive Time Series with a Unit Root," *Econometrica*, vol. 49, no. 4, p. 1057, Jul. 1981, ISSN: 00129682. DOI: 10.2307/1912517.

[26] D. Kwiatkowski, P. C. Phillips, P. Schmidt, and Y. Shin, "Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root?" *Journal of Econometrics*, vol. 54, no. 1-3, pp. 159–178, Oct. 1992, ISSN: 0304-4076. DOI: 10.1016/0304-4076(92)90104-Y.

[27] V. L. Tran and T. Leirvik, "A simple but powerful measure of market efficiency," *Finance Research Letters*, vol. 29, pp. 141–151, Jun. 2019, ISSN: 15446123. DOI: 10.1016/J.FRL.2019.03.004.

[28] H. Adams, N. Zinsmeister, M. Salem, R. Keefer, and D. Robinson. "Uniswap v3 core." (Mar. 2021), [Online]. Available: https://uniswap.org/whitepaper-v3.pdf (visited on 07/22/2022).

[29] B. Krishnamachari, Q. Feng, and E. Grippo, "Dynamic curves for decentralized autonomous cryptocurrency exchanges," *arXiv preprint arXiv:2101.02778*, 2021.

[30] M. Protocol. "Uniswap v3: Liquidity providing 101." (Aug. 2021), [Online]. Available: https://mellowprotocol.medium.com/uniswap-v3-liquidity-providing-101-f1db3822f16d (visited on 07/26/2022).

[31] A. W. Lo, A. C. MacKinlay, and J. Zhang, "Econometric models of limit-order executions," *Journal of Financial Economics*, vol. 65, no. 1, pp. 31–71, Jul. 2002, ISSN: 0304-405X. DOI: 10.1016/S0304-405X(02)00134-4.

[32] "Fees." (Aug. 2022), [Online]. Available: https://docs.uniswap.org/protocol/concepts/V3-overview/fees#finding-the-right-pool-fee (visited on 08/09/2022).

Colophon

This document was typeset using LaTeX, using the KOMA-Script class `scrbook`. The main font is Palatino.