IB9JH0 Programming Project

Group 1

May 2023

1. Introduction

Bid-ask spread is a tool for market makers to be implemented in order to prevent adverse selection from the buyers. The goal for this project is to find the behaviour of the convergence rate of the spread and the profit for traders corresponding to different factors under the Glosten-Milgrom (GM) model by simulating the actions and response between market participant markets in C++. The first model will be the original GM, while the other will be the extended GM with realistic assumptions.

2. Methodology

2.1 The original Glosten-Milgrom model

Suppose the value of an asset is v. At the end, the asset's value will be

$$v = \begin{cases} v_H, & \text{with probability of } 1/2 \\ v_L, & \text{with probability of } 1/2 \end{cases}$$

Therefore, the market makers estimate the initial value of asset to be the expectation of final value of asset

$$\mu_0 = \frac{(v_H + v_L)}{2}$$

Initially, the market makers will quote the first bid ask price from the μ_0, v_H and v_L. For each period, a trader will arrive at the market and decide to take action based on the current offer. There are two types of traders which are informed traders and uninformed traders. The difference between two is that informed traders could observe the final value of the stock.For informed traders, there are two strategies. If the final value takes v_H, informed traders will buy the stock. Otherwise, if the final value takes v_L, informed traders will sell the stock. In contrast, the uninformed trader will either buy or sell with equal probabilities.

Suppose π represents the probability that the next trader is an informed one. Then, the market makers will set initial ask price and bid price as

$$ask_0 = \mu_0 + \frac{\pi}{2}(v_H - v_L)$$
$$bid_0 = \mu_0 + \frac{\pi}{2}(v_H - v_L)$$

After the initial market price is quoted, a single trader will arrive and trade. As a result, The only information that the market makers know is the type of orders. If they know for certain whether this order comes from an informed trader or not, they can set the price to maximise their profit. Based on Bayes' theorem, the market maker estimated the probability of the asset go to v_H given the information from previous time is

$$\theta_t^+ = P(v = v_H|\Omega_{t-1}, q_t = 1) = \frac{\frac{1+\pi}{2}}{\pi\theta_{t-1} + \frac{1-\pi}{2}}(\theta_{t-1})$$
$$\theta_t^- = P(v = v_H|\Omega_{t-1}, q_t = -1) = \frac{\frac{1-\pi}{2}}{\pi(1-\theta_{t-1}) + \frac{1-\pi}{2}}(\theta_{t-1})$$

$$where \qquad q_t = \begin{cases} 1, & \text{if a buy order arrived at time t} \\ -1, & \text{if a sell order arrived at time t} \end{cases}$$

$$and \qquad \theta_{t-1} = \begin{cases} \theta_{t-1}^+, & \text{if} \quad q_{t-1} = 1 \\ \theta_{t-1}^-, & \text{if} \quad q_{t-1} = -1 \end{cases}$$

We are assuming that θ_0 equals to 0.5 since the market maker does not have extra information from orders on the final value of the stock. After receiving orders from the traders, the updated bid-ask price and spread will be

$$ask_t = \mu_{t-1} + \frac{\pi\theta_{t-1}(1-\theta_{t-1})}{\pi\theta_{t-1} + \frac{1-\pi}{2}}(v_H - v_L)$$

$$bid_t = \mu_{t-1} + \frac{\pi\theta_{t-1}(1-\theta_{t-1})}{\pi(1-\theta_{t-1}) + \frac{1-\pi}{2}}(v_H - v_L)$$

$$s_t = 2\pi\theta_{t-1}(1-\theta_{t-1})(\frac{1}{2\pi\theta_{t-1} + (1-\pi)} + \frac{1}{2\pi(1-\theta_{t-1}) + (1-\pi)})(v_H - v_L)$$

Since the term μ_{t-1} will cancel out when calculated spread we can assume that the parameters that affect the bid-ask spread in GM model are π, θ_{t-1}, v_H and v_L

2.2 The extension of Glosten-Milgrom model

In this part, we aim to develop the existing GM model by introducing a more realistic assumption. The original GM model assumes that the market maker can observe the last transaction in the market to calculate the bid-ask prices. However, financial markets involve numerous market makers and traders, making it impossible for a single market participant to directly observe whether the last transaction was a buy or a sell, especially for individual traders.

We propose a modification to address this limitation. Instead of relying on the quantity information to update and calculate the bid-ask prices, we introduce a new approach. Our approach involves determining the bid-ask prices based on the last observed price in the market, considering that market makers do not have direct visibility into the transaction details of other participants.

The additional assumption in the model is that a Brownian motion drives the stock price. Therefore, we applied Brownian Bridge to simulate price paths with a given initial price and a given final value. A Brownian Bridge models the continuous paths of a stochastic process. Simulating the stock price movement using a Brownian Bridge and calculating the bid-ask price based on the derived probabilities allows the model to realistically capture the interaction between the market maker and the stock price dynamics.

Given B_0 and B_T, we can calculate $B_{T/2}$ at the midpoint.

$$B_{T/2} = \frac{1}{2}(B_0 + B_T)\sqrt{\frac{T}{4}}X \qquad where \quad X \sim N(0,1)$$

Once we have computed $B_{T/2}$, we have two new sub intervals [0, $T/2$] and [$T/2$, T], with specified values at the end points of each B_0, $B_{T/2}$ and $B_{T/2}$,B_T. We can use the same procedure to fill in the midpoints of the two sub intervals.

To update the prior theta, the algorithm undergoes the following steps:

The last price is either the lower value (v_L) or the higher value (v_H). This assumption is realistic because published analysts' forecasts could be used as references into the final value. The simplification taken is that the final price will be either the highest (v_H) or the lowest (v_L) of forecasts. With the bound, using naive uniform assumption yields a relative position within the range and results in a probability denoted as p. It is ensured that the calculated probability, p, remains bounded within the interval [0.01, 0.99] to avoid calculation errors.

$$p = max(min(\frac{lastprice - v_L}{v_H - v_L}, 0.99), 0.01)$$

By taking the logarithm of the ratio of p to $1-p$, the logit function maps the probability to log odds, facilitating subsequent modelling.

$$log\ odds = log\frac{p}{1-p}$$

The product of the logit function reflects the market maker's belief of the stock price movement. When the dealer observes the price rises, they might think there is a higher probability and a higher odds that the final value is taking v_H. This intuition aligns with the logit-transformed value because the log odds become higher as the probability p increases. The higher log odds can be interpreted as higher confidence that the final value is v_H. Conversely, when the dealer observes the price decreases, the log odds become smaller, indicating a higher confidence that the final price is taking the lower value.

Subsequently, an updated estimation of the probability, denoted as θ_{prior} , is derived based on the observed log odds. The transformation of log odds to an updated probability by utilising the cumulative distribution function (CDF) of the standard normal distribution.

$$\theta_{prior} = \frac{1}{2}(1 + erf(log\ odds)) = \Phi(log\ odds)$$

where $\Phi(\cdot)$ represents the CDF of the standard normal distribution.

The rest of our derived model for market makers follows the original GM model.

On the other hand, the way they interact with the market has changed for both informed and uninformed traders. For informed traders, they are no longer only buying when they observe the final value as v_H and selling when the final value is v_L. They will compare the current bid and ask prices against the observed final value. Therefore, the informed trader will only buy if the final value is above the current ask price, selling if the final value is below the current bid price, or they will not trade. Uninformed traders, they are allowed not to trade also. For a constant probability of $\frac{1}{3}$, they will not buy and sell. The remaining $\frac{2}{3}$ probability for buying and selling is distributed according to the last observed price in the market, which is the same as how market makers calculate p.

3. Implementation

Market.cpp

The Market.cpp file implements the Market class, which represents a market where trading of securities takes place. It provides functionality to track price movements, trade matching, and access to market information. These attributes collectively store and manage various aspects of the market, including price information, quantity traded, and historical data. They enable mar-

ket participants to access relevant market information and track the progress of trading over time.

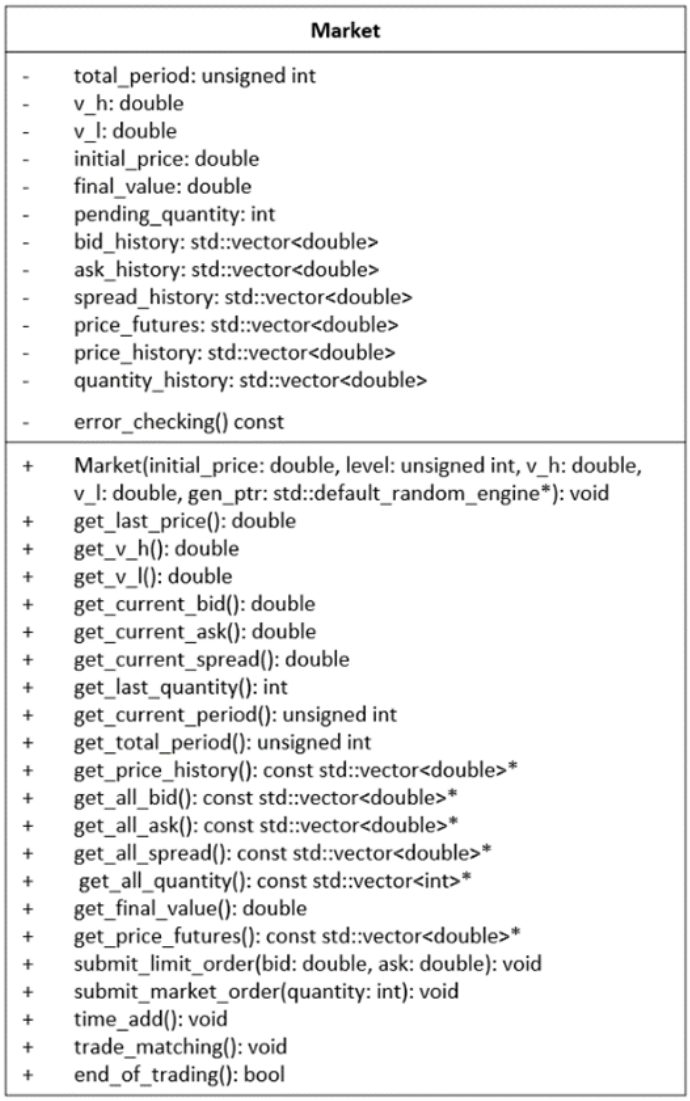

Figure 1: Attributes and functions of market

Market Participants

The MarketParticipants class represents the abstract concept of participants in a market. It serves as the base class for specific types of market participants, such as market makers and traders. It defines two pure virtual functions, read_market_info and submit_order, which are intended to be overridden by derived classes to provide specific implementation details for reading market information and submitting orders. This class establishes a common interface for different types of market participants to interact with the market.

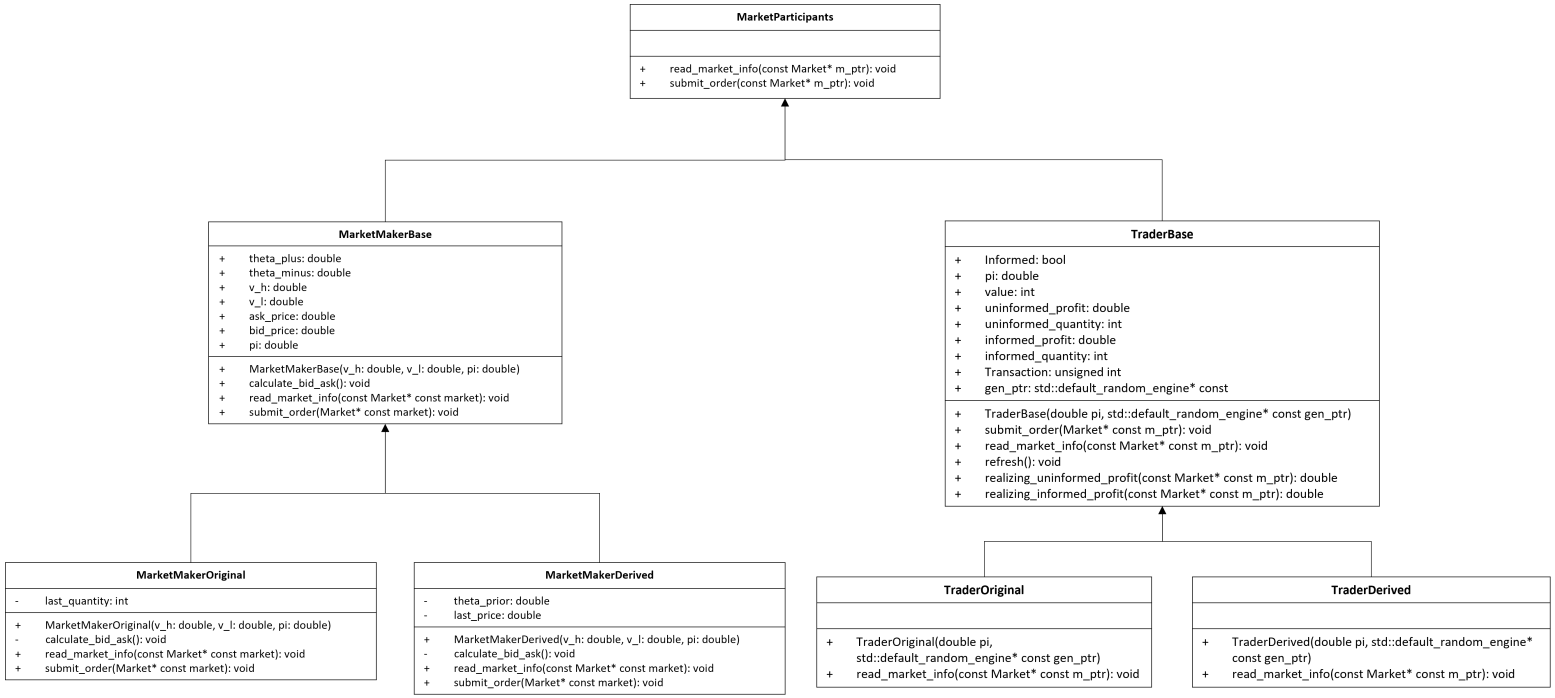

Figure 2: Market Participant class structure

Market Maker

The MarketMakerBase class, derived from MarketParticipants, represents a specific type of market participant. The MarketMakerBase class further extends the functionality defined in the base class by introducing specific attributes and methods related to market making. It further defines some member attributes and virtual functions such as theta_plus, theta_minus, and the virtual calculate_bid_ask method. Then, in each of the two models, specific classes of market makers are extended from MarketMakerBase, namely MarketMakerOriginal and MarketMakerDerived.

Trader

The TraderBase class, also derived from MarketParticipants, represents another type of market participant known as a trader. Traders engage in buying and selling financial instruments within the market. The Trader class extends the base class by introducing specific attributes and methods related to trading. It includes member variables such as informed. The class also specifies some virtual functions, for example, the virtual submit_order method to submit market orders based on the trader's trading strategy. By inheriting from MarketParticipants, the Trader class adheres to the common interface and two specific classes of traders, namely TraderOriginal and TraderDerived, are extended from TraderBase to fit into the models.

Market Simulation

The MarketSimulation class represents a simulation of market trading activity. It facilitates the interaction between market participants, such as traders and market makers, within a given market. There are two derived classes of MarketSimulation: MarketSimulationGMOriginal and MarketSimulationGMDerived, each designed for a specific model.

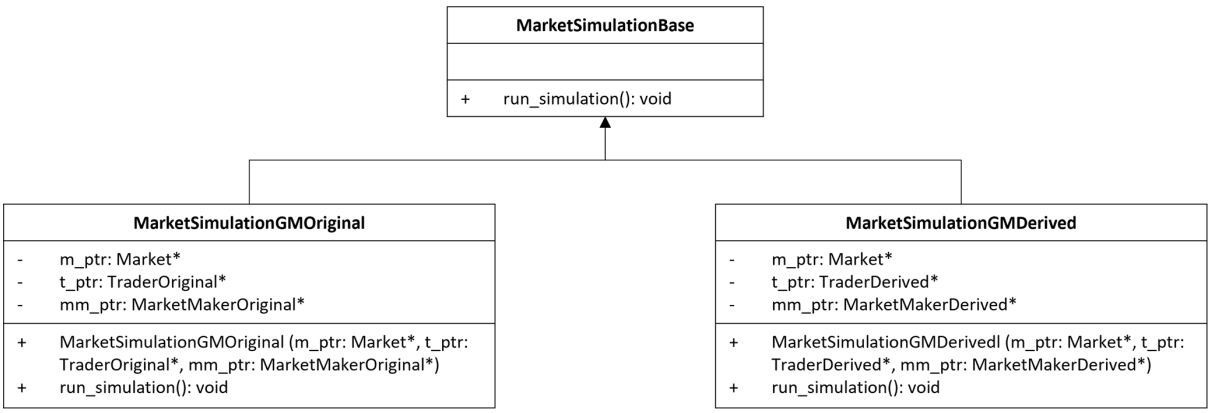

Figure 3: Market Simulation class structure

Original GM model

The MarketSimulationGMOriginal class, derived from MarketSimulation, is specifically tailored for the TraderOriginal and MarketMakerOriginal classes. It simulates the original GM model. It implements the run_simulation method, which initiates the trading simulation. Within the simulation, it iterates over the trading periods until the end of trading is reached. For each period, it performs the following actions:

1. Displays the current period information.
2. Allows the market maker (mm_ptr) to read the market information and submit a limit order.
3. Enables the trader (t_ptr) to refresh its state, read the market information, and submit a market order.
4. Executes the trade matching process in the market.
5. Prints the new spread of the market.

Derived GM model

The MarketSimulationGMDerived class, also derived from MarketSimulation, is specifically designed for the TraderDerived and MarketMakerDerived classes. It follows a similar structure to MarketSimulationGMOriginal and implements the run_simulation method. However, it includes an additional step during each trading period:

1. Displays the current period information.
2. Displays the remaining periods and the last price.
3. Allows the market maker (mm_ptr) to read the market information and submit a limit order.
4. Enables the trader (t_ptr) to refresh its state, read the market information, and submit a market order.
5. Executes the trade matching process in the market.
6. Prints the new spread of the market.

4. Analysis

We use different values of v_H, v_L and π to analyse the impacts on the convergence of bid-ask spread and the average profit for traders, under both the original and derived GM models. In the analysis, we ran 30 iterations for market simulation with 64 trading periods in each iteration. Overall, the initial bid-ask spread increases with the proportion of informed traders (π), and the volatility of the asset's value (v_H - v_L), which aligns with the GM model. Over the trading period, the bid-ask spread gradually decreases and converges

eventually, as the market makers are more certain about the final value of the assets. To analyse the convergence of the bid-ask spread, we set the threshold to 0.01 and measure the period when the current and all future bid-ask spread drops below the threshold. For the original GM model, the convergence of the bid-ask spread is faster when the volatility of the asset's value decreases. The bid-ask spreads converge even faster when the proportion of informed traders increases. For the derived GM model, the convergence does not vary much under different settings. The observed prices may contain more noise and the market makers are uncertain about the direction of the market. It takes a longer time to reveal the final value of the asset. Moreover, we analyse the average profit per transaction for informed and uninformed traders. In general, informed traders gain, while uninformed traders suffer from loss. In both the original and derived GM models, informed traders gain more and uninformed traders lose more when the volatility of the asset increases or pi decreases. For the derived GM model, the difference between the average profit for informed traders and that of uninformed traders further widens. Informed traders earn even more profits and uninformed traders suffer more losses. This is related to the slow convergence of bid-ask spread under the derived GM model. When the market makers are uncertain about the final value of the asset, they take more periods to update their value estimates and informed traders can earn more by exploiting any mispricing by market makers over the trading periods.

The analysis results are shown in the table below:

Parameters						
Number of iteration	30					
Number of trading period	6					
Threshold	0.01					
High value	150	135	120	150	135	120
Low value	50	65	80	50	65	80
Pi	0.3	0.3	0.3	0.7	0.7	0.7
Results						
Uninformed Average Profit (Original)	-3.17	-2.22	-1.27	-0.561	-0.393	-0.225
Informed Average Profit (Original)	2.92	2.04	1.17	0.517	0.362	0.207
Convergence of Spread (Original)	51.9	49.9	48.1	11.4	11.4	10.2
Uninformed Average Profit (Derived)	-8.15	-5.81	-3.28	-5.85	-4.16	-2.37
Informed Average Profit (Derived)	4.44	3.12	1.83	3.47	2.45	1.47
Convergence of Spread (Derived)	61.7	61.4	60.5	62.1	61.9	60.2

Figure 4: Result of simulation

Reference

1. Foucault, T., Pagano, M. and Röell, A.(2013). Market liquidity: theory, evidence, and policy. Oxford University Press, USA.
2. Glosten, L.R. and Milgrom, P.R. (1985). Bid, ask and transaction prices in a specialist market with heterogeneously informed traders. Journal of financial economics, 14(1), pp.71-100.