Software Project Report

Artemiy Astashkin

6 June 2023

Contents

Abstract

The area of our research is computer modeling of trading in financial markets by applying agent-based modeling using programming languages and algorithms to build a working simulator on several markets to study the transmission of information between two exchanges , especially rare "Black Swan" events, which in other words are called financial time series bifurcations. The previous simulators did not conduct such a research and were unable of describing of those events.

Part I

Chapter 1

Before the implementation

1.1 Introduction

The importance of understanding how the interaction between exchanges happen nowadays is crucial. The capturing of anomalies,sharp outliers and its consequences on the market it happened is the first thing , but how this situation influences and impacts the second market , where some agents from the first market are also in action is more complicated and interesting task.Our implementation of a 2-D market is concentrated not only on creating this model , but also performing the statistical research to study questions of the first and second market recoveries from this unexpected events.

1.1.1 The roles of project's members

The another project participant is Pavel Kalmanovici, 4 course student of DSBA. Our main initial task was to develop the Agents-based model architecture of several markets, then develop PNL fucntions of those traders agents that will impact the most the information transmission between markets and implement them and finally, conduct the statistical tests of several hypothesis about the simulator and agents itself. The 2-d structure simualator was implemented by me, meaning that the shock transmission can be seen by structure of several agents. Thus, the Chartist and Market Maker architectures and their PNL's is my own work.I fully developed in code Market Maker and Chartist for 2-d Market simulator Classes and changed the parent class of Trader, while Pavel did logging.py , generator.py and export.py himself without my help. The Statistical research and hypothesis testing was done by Pavel and they are not included in my report.

1.1.2 Functional and Non-Functional requirements

1. Python 3.9
2. PyCharm
3. tqdm=4.64.0,matplotlib=3.5.1,pandas=1.4.2,numpy,statsmodels =0.13.2,scipy,seaborn =0.11.2
4. Git

5. ABM theory

1.1.3 Project structure

The project is divided into 3 parts, further each of them is widened and has its own subtasks which are described in its own sections.

- ABM theory, architecture of 2-D simulator
- Code Implementation
- Empirical evaluation

The vanilla version of simulator was implemented was implemented by Niktin Bogdan in topic "Market stability simulator", as we have the same supervisor, and the construction of our 2-D simulator is based on changing the existing simulator architecture from one market to 2 market in order to conduct our research. The final outcome of our project is the developed 2-D simulator which can be simulated by different market settings and the statistical hypothesis from real life tested on these simulator in order to either reject them of accept. The possibility of failure of constructing this simulator is not valid here.

The results of our research were presented at 2 conferences. I was a reporter in MIPT 65 conference, while my supervisor was a reporter on VEGA. 1) 65th All-Russian Scientific Conference of MIPT" 2)"All-Russian seminar dedicated to the 120th anniversary of A.N. Kolmogorov"

1.1.4 Aim

The aim of Multi Agent-based model in our project is to implement **model** which will simulate and replicate the real financial exchanges. The ideal result is the golden mean between generalization of this simulator to other researches connected with ABM and plausible representation of the objective financial metrics :volatility,fundamental price,market price,volume,market states. The basic process is the formation of a price on financial market is the bid-ask prices in the orderbook,similar to supply demand in classic economics. The agents of our model are traders and they have finite number of parametres, but obligatory for everyone are cash,assets (which are also finite),other are optional and depends on agent PNl's, all of then being heterogeneous and self-organized .Also one important agent of structure is the market itself - we call it exchange agent. The aim of research is to build a 2d market , perform on it artificial inevitable price shock, transactions shock and see whether it is possible to see a shock transmission graphically between two markets and if it happens, prove it statistically in all types of shocks respectively.

1.1.5 Definitions

1. Defintions

   (a) **ABMS**-Agent-based modelling and simulation (ABMS) is an approach to modelling complex systems composed of interacting, autonomous ‘agents’. Agents have behaviours, often described by simple rules, and interactions with other agents, which in turn influence their behaviours. By modelling agents individually, the full effects of the diversity that exists among agents in their attributes(arguements)and behaviours(function of actions they perform based on situation they are in) observed as it gives rise to the behaviour of the system as a whole.To run an agent-based model is to have agents repeatedly execute their behaviours and interactions. This process often does, but is not necessarily modelled to, operate over a timeline, as in time-stepped, activity-based, or discrete-event simulation structures.(in our case , the first one, time-series are time-stepped)

(b) **Structure of an agent based model**- A typical agent-based model has three elements: set of agents, their attributes and behaviours. A set of agent relationships and methods of interaction: An underlying topology of connectedness defines how and with whom agents interact.(Basically, the scheme , or an algorithm set) The agents' environment: Agents interact with their environment in addition to other agents.In our case agents interact only with Stock Market.

(c) **Autonomous agents** -defining characteristic of an autonomous agent is its capability to act autonomously, that is, to act on its own without external direction in response to situations it encounters(in our case agents will be quite simple ans base)

(d) **Autonomous agents characteristics**-self-contained(modular,has bounary),self directive,agent has a state, it changes,agent is a social -has ineractions with ither agents.Adaptivity (learning , different patterns of behaviour.

(e) **Time series**- is a series of values describing a process running in time, measured at consecutive changeover points in time.

(f) **Stock Exchange Glass** - is the list of orders (limit and market) that a trading venue (in our case stock exchanges) uses to record the activity and interest of buyers and sellers in a particular asset of . Exchange agent uses the book to determine which orders can be fully or partially executed.

   i. bid - is an order to buy asset or assets . More concretely, maximum price that a trader-consumer is willing to pay for one asset.
   ii. ask - is an order to sell asset or assets.More concretely, minimum price that a trader-seller is willing to earn for an asset.
   iii. (spread- The difference between ask and bid , indicator of liquidity of an asset on a financial market.

(g) **limit order** - is the maximum or minimum price at which trader wants to complete the transaction, a buy or sell of concrete number of assets.

(h) **market order** - are orders which need to be executed as soon as possible at the current best bid or best ask of the market.

Part II

Chapter 2

Description of the model and agents

Simulation structure-The trading process consists of consecutive trading sessions in which traders make their actions based on predefined strategies. Each new session, traders receive publicly accessible information, including stock prices, best bid and ask prices, and upcoming dividends depending on the level of trader access to this information. This information is provided by the exchange agent. Several user-defined events occur before the trading session begins, and vital market conditions and trader statistics are collected. These statistics include important variables such as stock price, trading volume, traders' cash and assets, returns and sentiment in case of chartists and fundamentalists. Once the primary data collection is complete, traders are encouraged to begin implementing their strategies. Actions are whether to buy or sell stocks, place orders based on their strategies. As the trading session progresses and nears its conclusion, traders are paid dividends and risk-free interest payments. These dividends give agents the opportunity to review and potentially adjust their behavior,including strategy, or sentiment on the results of the session and the payouts received. Scheme of trading process of single iteration:

1. The trigger events scheduled for the session are activated at the appropriate time.
2. The statistics from both markets and information about traders are collected and documented.
3. Traders adjust their behaviour and strategies
4. Traders make actions

After that, the dividends are initialized for future payouts. The simulation stops when there are now limit orders left and market orders are the beginning the vital change of the order book reduction.This replicates the real world market crash and we do not study this situations in our work, by making the approximately equal number of limit orders to all types of traders, the overall number of orders is balanced by this approach in order to maintain stability and space for research of recovery of a flash crash, not a market crash closure. After the end of all iterations of simulation, the all information is stored in Simulator object needed for further statistical research and visualisation. To be optimally used, the SimulatorInfo object is initialized which is referenced in Simulator respectively. After that, the functions of visualisations are applied in this object after further data aggregation in specific singular module, which all use a link to the SimulatorInfo object as obligatory attribute, helping us to create a special metrics visualisations, which we want to analyse.

Exchange Agent- Here I will describe the Exchange agent principle, even though this is an agent of model, it is not trader. It is core of our simulator and matches the market itself.The Exchange Agent is the connection between the market and traders, he links the market and traders, the trading activity is transfered through him.The traders make their orders to him, updating the order book. The Exchange Agent manages the order book, determining the dynamics of stock prices during the trading session. The actions of traders are based on knowing information of best bid and best ask, though impacting the trned of stock price. The Exchange agent is firstly inizialized before the start of the whole simulation with fulfilling the order book with random orders from the initial price(the user chooses itself as parameter), fluctuating near it, it is very close to replicate a real world situation, as we don't model the opening of new market, but some intraday activity on an active one. The Exchange agent receives from traders 3 main types of orders:cancel,market and limit. Further, in the implementation section the concrete structure and mechanics will be described, as well as the specific data structures which are needed for it optimal realization.

Random Trader - random trader makes the decisions that he thinks will gain him profit , but in reality he makes random decisions. This trader replicates a person who has no any strategy in real life, but this is a huge part of traders in reality and we can not ignore it, another name for this trader in other works is Noise Trader, meaning random noise from external reality which he bases his decisions on. He has predefined constant probabilities , based upon which he makes different order type. Limit order (0.35), Market order (0.35 to 0.5), Cancel order(0.5 to 0.85) and do nothing (0.85 to 1). This constants shown good result in simulation in other works and have mechanical sense beneath it. The Limit order are not required to be executed, while market orders executed immediately. The situation when there are more market orders than limit order quickly leads to reduction of orders in order book, which is not good for stable simulation and further research and does not replicate the reality. Moreover, the chances of buying or selling sides are equal, which is obvious. The quantity for orders is always $Q \sim U\{1, 5\}$ from uniform distibution. If trader chooses limit order the price can be drawn either out of the the spread - 0.65, or in the spread - 0.35.When in the spread, price is uniformly distributed between best ask and best bid $P \sim U\{best\ bid, best\ ask\}$ Out of the spread the price is offsetted by an exponential indent from the best offer in order to state, that it differs from the situation when the trader actually captured the spread.

$$P = best\ bid/ask \pm \Delta, \text{ where } \Delta \sim exp(\lambda) \tag{2.1}$$

Chartist- The Chartist trader strategy can be described by one phrase "buy low, sell high", they focus on predicting future price trends and dependent on proportion of their type of agents on each exchange. They feel either pessimistic or optimistic about the market and based on their sentiment , what other Chartists think and market price change , they do their actions.They calculate at each iteration, how other Chartists are pessimistic or optimistic , how other Chartits changed their opinions at each market separately and behave in different patterns on different markets , calculating their own sentiment. Quantity of order function is similar to random trader and is calculated from uniform distribution $Q \sim U\{1, 5\}$. The price of a limit order, where p is market price, t is transaction cost. is by formula, :

$$P = (p \pm \Delta) * (1 \pm t), \text{ where } \Delta \sim exp(\lambda) \tag{2.2}$$

The calculation of sentiment is based on this formula:

$$\pi_{+-} = v\frac{n_c}{N}exp(U), \;\; \pi_{-+} = v\frac{n_c}{N}exp(-U)$$

$$U = \alpha_1 x + \frac{\alpha_2}{u}\frac{dp}{dt}\frac{1}{p}$$

The Chartist unlike Random trader chooses whether to buy or sell applying to sentiment. The probabilities for change in sentiment are π_{+-} - changing positive to negative, π_{-+} changing negative to positive, there are parameters v - frequency of opinion revaluation, n_c/N - the proportion of Chartist among all traders. Chartist reevaluates its sentiment on the course of trading, the factor that influences the sentiment change is U, it represents the strength of opinion propagation, the parameters α_1, α_2 - the weights the current price trend and sentiment of other traders, p - market price. They also look for opportunities on cross markets profits, knowing prices dynamics on 1 and markets and do orders based on whether it is cheaper - they buy there(i.e 1 market) and where it is more valuable and expensive (i.e 2 market) sell there.

Fundamentalist-Fundamentalist defines his decisions based on the stock's price based on its pasts dividends analysis in order to predict the future dividends and the real value of stock as well. Fundamentalist calculates stock's fundamental value comparing it with market price. The price for the limit order is the same as for the Random trader . Order volume is calculated like: $Q = min(\frac{1}{\gamma}\frac{|P_f - P_m|}{P_m}, \; 5)$. Fundamental price is calculated using Constant Dividend Model using last n known dividend.

$$P_f = known \; + \; perp$$

$$known = \sum_{i=0}^{n}\frac{Div_i}{(1+r)^i}, \;\; perp = \frac{Div_n}{r(1+r)^n}$$

The number of known dividends , defined by n, provides information asymmetry among all traders, having the "rare" limited information access about dividends future trend. If the fundamental price is greater than market price, they buy market orders or sell limit orders.The fundamentalist are the most possible profitable, yet the most risky agents, they perform only one market.

The price for limit orders is calculated by this formula:

$$P = (P_f \pm \Delta) * (1 \pm t), \text{ where } \Delta \sim exp(\lambda) \tag{2.3}$$

Market Maker- The core of traders and main shock trasmitter, performs on both markets. The market maker gains profit on spread, by setting orders every time on both sides , he byes and sells at the same time everywhere, provides liquidity for the market. To provide stability for the market , he has lower and upper limits of his inventory on both exchanges separately and the whole upper and lower limit for two markets combined. When on one market he is close to limits, he adjustes his order bigger to either sells or buys to prevent destabilizing. His Pnl is the sum of PNL in fisrt market and second market and he maximizes it. The mechanism used in our model is rather basic. Market Maker calculates the offset, and adds it to prices as follows:

$$offset = min(1, (best\ ask - best\ bid) * (assets/LL_i)$$

$$P_{bid} = best\ bid - offset,\ \ P_{ask} = best\ ask + offset$$

The volumes also depend on the current inventory state. The order volumes are the following:

$$Q_{ibid} = max\,[0,\ UL_i - 1 - assets]$$

$$Q_{iask} = max\,[0,\ assets - LL_i - 1]$$

UL_i and LL_i defines the top and bottom limits for the number of stocks in the inventory that pleases Market Maker on separate markets. If one of them is out of range, he perform market orders , making no limit orders, it is called "panic" state.

The transmission of information and shocks happens due to these constraints:

$$LL < Q_1 + Q_2 < UL$$

$$LL_1 < Q_1, UL_1 > Q_1$$

$$LL_2 < Q_2, UL_2 > Q_2$$

$$LL_1 + LL_2 < LL$$

$$UL_1 + UL_2 > UL$$

Part III

Chapter 3

Implementation

3.1 Architecture

1. **Scheme of one market Simulator-** The previous scheme looked like this.

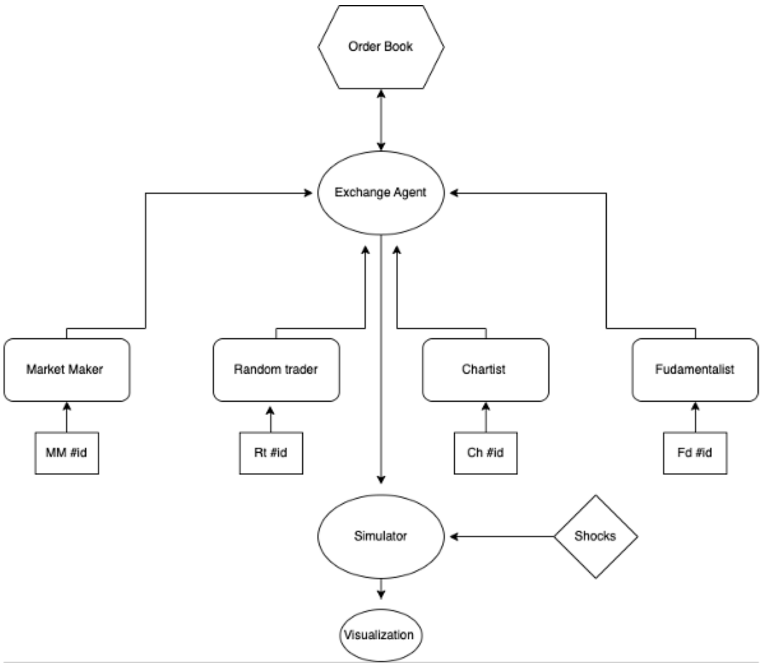

Figure 3.1: A scheme with a one market simulator

2. **Scheme of one 2d Simulator-** After our construction of the 2-d simulator, the 2 markets appear and the trasmission of shocks between them is going through the Market Makers and Chartists, as only them are present at the same time on both markets.

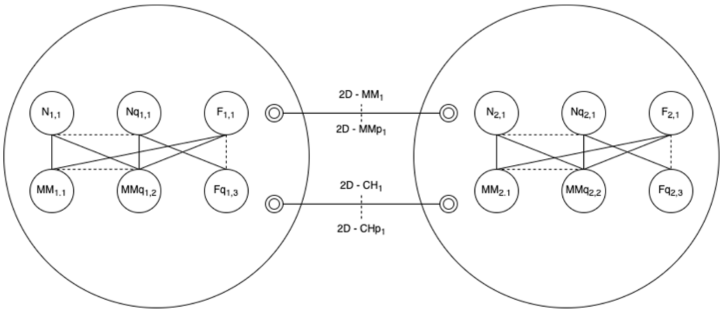

Figure 3.2: A scheme with multi market simulator

3.2 Code and data structures

3. **Data Structures** Order and OrderList data structures were created. An Order needs to contain details about the order of trader itself and own attributes. OrderList is kind of a doubly linked list structures of orders. An order has important attributes : price, quantity, order type, link to the specific trader who made this order. The OrderList, has left and right order links to keep efficient structure of updating , removing the order book.The Order has a method of comparing different orders. The OrderList uses Order options to store orders in the order book and sort them according to the best bid - best offer paradigm in efficent time and space.

   The operations we use for our virtual order book are those:

   (a) **Remove: Removes an order from the book.Limit order is done or a cancel order is done.Time complexity of O(1).**

   (b) **Append: New order at the end of the book. Time complexity of O(1).**

   (c) **Push: New order at the beginning of the book. Time complexity of O(1).**

   (d) **Insert: New order in the right place in the book to to be it sorted from best to worst order. Time complexity O(n) in worst cases**

   (e) **Fulfill: Makes market orders or limit orders that are better than the current best bid or best offer. Time complexity O(n) in worst cases**

4. **Classes of Traders**:

   (a) **Trader** - parent class of all other trader types classes. The attributes of the Trader :

      i. **Type**: type of trader.

      ii. **Name**: name of trader, unique combined with trader id.

      iii. **id**: id of trader, unique for trader in one type.

iv. **markets**: markets where trader is present, list of Exchange agents.
v. **orders**: list of orders, which trader performs.
vi. **cash**: number of money he has
vii. **assets**: number of assets he has
viii. **access**: knoeledge of some information about dividends. Used for Fundametalist and Universalist, but second trader type is not user in our 2 d simulator.

The call method in all agents apart from Trader make possible all methods of spread, order setting of different order types:limit,market,cancel. The call can be interpreted as 'strategy' of agent.

(b) **Random** - child class of the Trader class. Has initializing(init) , delta ,draw price, draw quantity and call methods. The formulas of them were presented in describing model and agents part.

(c) **Fundamentalist** - child class of the Trader class. Apart from the same parmetres from basic Trader, has gamma parametres, which is needed for calculating dividends for fundamental price evaluation. Has initializing(init) , evaluate ,draw quantity and call methods. All formulas are in Fundamentalist part of describing agents and model section. Evaluate method calculates dividend, draw quantity is for quantity and call strategy again.

(d) **Chartist** - child class of the Trader class. Has initializing(init) , call ,change sentiments method. All formulas are in Chartist part of describing agents and model section. Change sentiment method is about reevaluting chartist opinion about price trends.

(e) **Market Maker** - child class of the Trader class. Has initializing(init) , call . All formulas are in Market Maker part of describing agents and model section. Has additional attributes of lower and upper limits (soft limits with), and state attribute whether he is in panic of stable state.

3.3 link to github

Multi Agent based modeling on financial markets

Part IV

Results and research of shock transmission

Our aim was to study the transmission of shocks between markets. We estimated the shock of several types:market price shock, information shock transfering from one market to another.On the i-th iteration of simulation, shock appears. After that, Market Makers and Chartists agents adjust their strategies and we see how the shock is being transmitted and whether both markets recover or not. We got following results introducing different shocks.

(a) **Market shock-** On the 200-th iteration, market price drops by user on 20 percent using MarketPriceShock form event.py module. There are 3 types of agents on this market : market makers, random traders and fundamentalists on upper graphs and no market makers on the bottom two graphs, but the same amount of agents of types fundamentalists and market makers as on the upper ones. We seem that the shock in both cases are really transmitted. When market makers are present, the price falls quickly at first and then slowly adjusts. Market makers, who are more focused on keeping prices stable than on the real value of the asset, cause small price fluctuations, with a consecutive sudden return to the original price,having large quantites of asset, that they can control. When there are no market makers, the market price quickly returns to the fundamental value. This is because of presence of fundamentalists, who become more impactful to the market than the random traders. The same results but with a more fluctuations of market price happens, when some chartists are added.

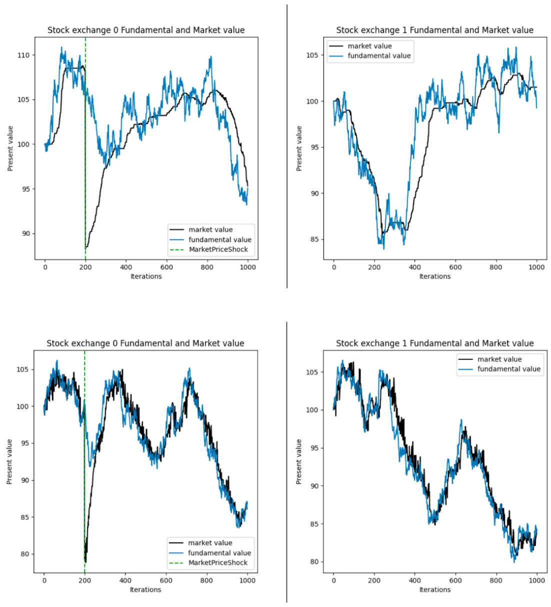

Figure 3.3: market price shock

(b) **information shock-** The same number of agents as with market price shock.Initially access parameter equals to 50, then rapidly becomes 1 because of the shock on 200-th iteration. The right graphs show difference between the fundamental value and the market price Before the shock, the market price and the fundamental value line are almost identical as fundamentalists are good at predicting the fundamental value in advance. However, after the shock, the market price lagged behind the fundamental price. This effect appears on on both markets, stronger where the shock appeared.

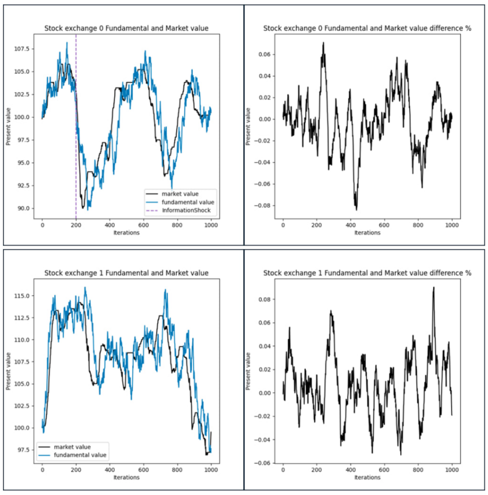

Figure 3.4: information shock

Part V

Conclusion

3.4 Outcome

We made a great job of constructing an architecture of multi agent simulator with further implementing the traders classes in code. The 2-d market maker and chartist classes were implemented and tested during the simulations. Moreover, we showed that the shocks transmission in our 2-d simulator truly is being transmitted in several types of shocks on different market settings which replicate simialr world know financial flash crahses.

3.5 Further research

However, there can be more dome even more complicated research , as create an opportunitu of more agents, which are constrcted on more complex algorithms. Furthemore, the process of multi asset trading of several assets was not implemented and should be considered as next step of our-research.

Part VI

bibliography

Bibliography

[1] Christopher D. Clack, Elias Court, Dmitrijs Zaparanuks Dynamic Coupling and Market Instability https://arxiv.org/abs/2005.13621v1

[2] Samuel Peter, 05-909-122 Agent Based Simulation on a financial market

[3] (1997) The Economy as a Complex Evolving System II in Santa Fe Institute Studies in the Sciences of Complexity, eds. Arthur, W. B., Durlauf, S. Lane, D. (Addison–Wesley, Reading, MA), Proceedings Vol. 27, pp. 15–42. Arthur, W.B. et al. (1997) 'Asset pricing under endogenous expectations in an artificial stock market' SSRN Electronic Journal. doi:10.2139/ssrn.2252

[4] M. A. King and S. Wadhwani, "Transmission of volatility between stock markets," Review of Financial Studies, vol. 3, no. 1, pp. 5–33, 1990.