Agent-based modeling of herd mentality in the stock market

Wilhelm Eklund

August 20, 2011

Abstract

This thesis describes agent-based modeling of a fictitious stock market. Some agents mimic real life documented trading strategies for high frequency trading and other agents use a ”herding” strategy and base their investment decisions mainly by looking at the orders placed by other agents and mimic these.

The statistical properties for this simulated market has some resemblance to historical data from the American stock market. This shows that the agent-based approach to simulate the stock market should have good potential for future development and is a realistic way of describing the interactions in the financial world.

Sammanfattning

Detta examensarbete beskriver agent-baserad modellering av en fiktiv aktiemarknad. Vissa agenter efterliknar verkliga dokumenterade aktiehandlarstrategier för högfrekvent handel och andra agenter använder ”flockstrategier” och grundar sina investeringsbeslut huvudsakligen på att oservera beställningar från andra aktörer och härma dessa.

De statistiska egenskaperna för den simulerade marknaden har vissa likheter med historiska data från den amerikanska aktiemarknaden. Detta visar att agent-baserad modellering för att simulera aktiemarknaden borde ha god potential för framtida utveckling och är ett realistisk sätt att beskriva interaktioner i den finansiella världen.

Contents

Chapter 1

Introduction

1.1 Background

For as long as there has existed market for trading stocks, numerous models for predicting and simulating the stock market has been suggested. One of the most popular methods of simulating the stock market the last decade is called agent-based modeling (ABM). In this type of simulation one create a number of autonomous agents and give them simple rules for how they trade. Then one simply observe the outcome of the market and try to draw conclusions from this.

An always recurring and highly debated subject in the financial world is the concept of herd mentality between stock brokers. People believe that a large share of the market do business purely based on the behavior of others rather than thinking by themselves. This supposedly creates instability in the financial market by amplifying the effects of financial events, thus setting the scene for financial disasters such as the global crash of 2008.

As a result of this disaster several projects has tried to answer the question of how the economic and financial system could become more stable. A European project with a price tag of 1 € billion has been proposed to make such a simulation. Decision is expected in 2012. This makes agent-based modeling of the financial market a very up-to-date subject.

It is interesting from a computer science perspective to investigate if agent-based modeling is a suitable approach for modeling the stock market. Computer systems play an essential role in todays financial market and all the large financial corporations are depending on their financial models. It is also interesting to investigate how one would model herd mentality for this scenario. To my knowledge there are no well known implementations today of models where herd mentality is modeled as a factor when trying to predict the financial market, although many articles describe that there exist herd mentality in the market.

1.2 Goals of this thesis

There are three basic goals with this thesis:

1. Test if agent-based modeling is a suitable approach to simulate the stock market, in terms of similarities with reality.

2. Test if the agent-based model is a better approach than random walk.

3. Test if herd mentality strategies for the stock market can be modeled.

1.3 Approach in this thesis

As the goal of this thesis is to make an agent-based simulation of the stock market I will perform a simulation and then compare the results with reality.

I will first go through some theoretical background about the different subjects and techniques used in this simulation. This is needed to understand what the simulation is based on.

The work performed in this thesis is primarily an extension of the existing simulation environment Java Auction Simulator API (JASA). I will go through how this environment works and then explain how the original package was extended with some new functionalities and some new kinds of trading strategies which were implemented (including strategies based on herd mentality).

Finally simulations will be performed and the results will be presented and compared to historical data.

Part I

THEORETICAL BACKGROUND

Chapter 2

ABM: Agent-Based Modeling

2.1 Background

Agent-Based Model (ABM) is a subsection in the field of artificial intelligence. It is a method of simulating the behavior of a group by only knowing how the individuals make their decisions. Knowing only the simple behavioral rules for the individual, one can create complex behavioral patterns for the group, with relatively low computational costs [1].

The agents are autonomously interacting in a shared environment.This is a ”bottom-up” approach to modeling. Small changes in the micro structure leads to large changes in the macro structure.

2.2 Applications of ABM

Agent-based modeling can be used to test how changes in individual behaviors will affect the system's emerging overall behavior. This makes it a suitable tool for modeling and explain a variety of scenarios including, but not limited to:

- social networks and social phenomena
- logistics and supply chains
- consumer behavior
- traffic jams
- the spread of epidemics
- evolutionary modeling explaining human evolution through natural and sexual selection

- movement of a herd or flock of animals
- the financial trading system
- model results on macro scale as a result of micro changes
- other patterns emerging from individual acting

In the 1980s models describing flocking were developed. These were models of biological agents and their interaction. This kind of modeling has been called "artificial life". Agent-based models has been used in economics from the 1990s.

Last year an interesting article by Philip Ball [2] was published in New Scientist regarding large scale ABMs. With the project "Eurace" a European team have created the worlds largest agent-based model of an economic system with 17 million agents trading with each other. A future project "FuturICT" is planning to create an "Earth simulator" taking many more factors into consideration. The price tag is 1 € billion and the primary objective is to simulate how changes in financial policies will impact the robustness of the financial system. Decision regarding approval of the project is expected in 2012.

2.3 Example: Conway's "Game of Life"

Game of life was one of the first models with autonomous acting entities, created by the British mathematician John Conway in 1970 [3]. It is a classic model applying cellular automata, which is a predecessor to agent-based modeling. Cellular automata is not an agent-based model since the cells can not be consided agents. They do not have any goal to pursue. They only act according to their predefined rules. Cellular automata uses principles which later have been applied in agent-based modeling. A field is divided up in a two-dimensional infinite grid and each square is an individual cell. A cell is either alive (black) or dead (white). The state depends on a set of rules and is determined by how many of the eight neighboring cells that are alive. The rules for each cell are:

1. Any live cell with fewer than two live neighbors dies, as if caused by under-population.
2. Any live cell with two or three live neighbors lives on to the next generation.

3. Any live cell with more than three live neighbors dies, as if by overcrowding.

4. Any dead cell with exactly three live neighbors becomes a live cell, as if by reproduction.

These simple rules are enough to create a complex system. On figure 2.1 a simple moving structure (known as a "glider") in Game of life has been shown.

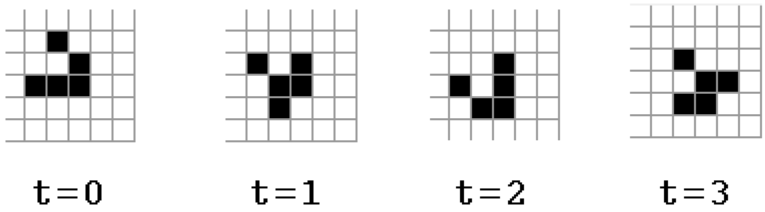

Figure 2.1: Illustration of Conway's Game of Life.

Note how every cell acts autonomously only depending of it's own neighbors.

2.4 Validation and verification of ABM

A very important part when discussing agent-based models (or any model) is to verify that the results resemble reality. In this thesis I follow the notation used in [4].

Verification is the quality control of a model, done by going through the model structure and make sure no misconceptions exist in the basic principles and that the specifications are the same as the system we want to model. This is to motivate that the model is likely to describes reality.

Validation is the quality assurance where we want to prove resemblance to reality in a desired way by test. In this thesis the test is that if the simulated results resemble some statistical properties of real financial data then the model is considered to be validated. The statistical properties of the stock market is described further in chapter 3.

In the article "A validation methodology for agent-based simulations" [4], the author proposes a validation process in four steps: face validation, sensitivity analysis, calibration and statistical validation. In this thesis the goal

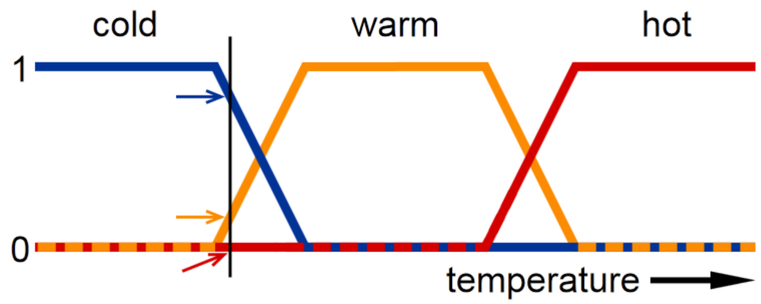

Figure 2.2: Fuzzy logic for measures of temperature

is to simulate the stock market and the validation will be performed only by face validation and statistical validation because of time constraints.

Face validation is performed by simply looking at graphical displays of the resulting time series from the simulation and some of the statistical properties of them. With the naked eye this will be compared to the corresponding measures of historical stock prices.

Statistical validation is performed by looking at statistical measures of the properties compared with historical data.

2.5 Fuzzy logic

Fuzzy logic will be briefly used in this thesis at some places. It is commonly used for many fields of artificial intelligence.

The ”ordinary” and most common type of logic is the so called ”two-valued logic”. Here a statement is either true or false and never anything else. Fuzzy logic is (among other things) a collection of different ways of handling uncertain values. This allow us to handle partial truths, which is common in the real world. Everything is not true or false. Fuzzy logic is a whole group of different kinds of logics. One of these logic types are the so called ”many-valued logic” where a logical statement can have many different values [5]. This is the kind of fuzzy logic which will be used in this thesis.

The temperature in figure 2.2 is interpreted as different levels of truth for the three measures ”cold”, ”warm” and ”hot”. At the point of the vertical line the ”cold”-value could be interpreted as ”fairly cold” [1].

[1]Example taken from http://en.wikipedia.org/wiki/Fuzzy_logic accessed on 2011-06-01.

Fuzzy logic is used when one need to know the degree of truthfulness in a statement and the two-valued value of ”true” or ”false” does not describe it enough. For instance it is used in control theory to make changes smoother than simple P-regulation (on/off).

In the case of this thesis some of the trading agents will use fuzzy logic to decide how they want to place their orders in the market.

Chapter 3

Statistical properties of stock markets

As mentioned in chapter 2 an important step when performing an agent-based model simulation is the validation of the model. This is a test performed after the simulation to verify that the results are likely to resemble reality.

In the first part of this chapter statistical definitions and tools used are described. In the second part statistical properties of historical data are presented. These will later function as reference when simulated data is compared to them.

3.1 Statistical definitions

The reader is presumed to have basic knowledge of statistics beforehand and this is only a short repetition. For a more extensive explanation of the basic measures, Blom [6] is suggested reading.

X denotes any stochastic variable.

Expected value of X: $E(X)$.

$$E(X) = \bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$$

Sometimes this is called the "first moment" of the variable and denoted as $\langle X \rangle$. Moments of higher grade are defined as below, here is the moment of grade i:

$$\langle X^i \rangle = E(X^i) = \frac{1}{n}\sum_{i=1}^{n} (X)^i$$

Standard deviation (denoted σ) is the square root of the variance for a variable.

$$\sigma_X = \sqrt{Var(X)}$$

$$Var(X) = E(X^2) - E(X)^2$$

Stock price at time t: S_t

Definition of the (log) return of the stock: $r_t = \log(S_t - S_{t-1})$.

The return of Δt time units is sometimes denoted $r(t, \Delta t)$.

$$Covariance : Cov(X,Y) = \frac{1}{n-1}\sum_{i=1}^{n}(X - \bar{X})(Y - \bar{Y})$$

$$Correlation : Corr(X,Y) = \frac{Cov(X,Y)}{\sigma_X \cdot \sigma_Y}$$

Auto Correlation Function (ACF) of the return:

$$ACF(\tau) = Corr(r(t, \Delta t), r(t + \tau, \Delta t))$$

Kurtosis is a measure of how relatively often extreme event occur for a distribution. Kurtosis will be used to measure how fat the tails of the return distributions are. Kurtosis of a distribution is defined as:

$$\kappa = \frac{< (r(t,\Delta t) - < r(t,\Delta t) >)^4 >}{\sigma(\Delta t)^4} - 3$$

As reference can be mentioned that the normal (Gaussian) distribution has a kurtosis of zero.

Brownian motion, also known as random walk, is when a stochastic variable moves in a random way and the change from one time point to another is a Gaussian variable and is proportional to the time difference Δt. This is illustrated with figure 3.1, showing five different time series with random walk.

3.2 Stylized facts about the stock market

In the article "Empirical properties of asset returns: stylized facts and statistical issues" [7] many statistical properties about the stock market are described. The term "stylized fact" refers to that the property is empirically present both in a variety of different kinds of assets and in different historical times. The name "fact" could be considered misleading but since this is the term used in this article I have chosen to use the same term in this thesis

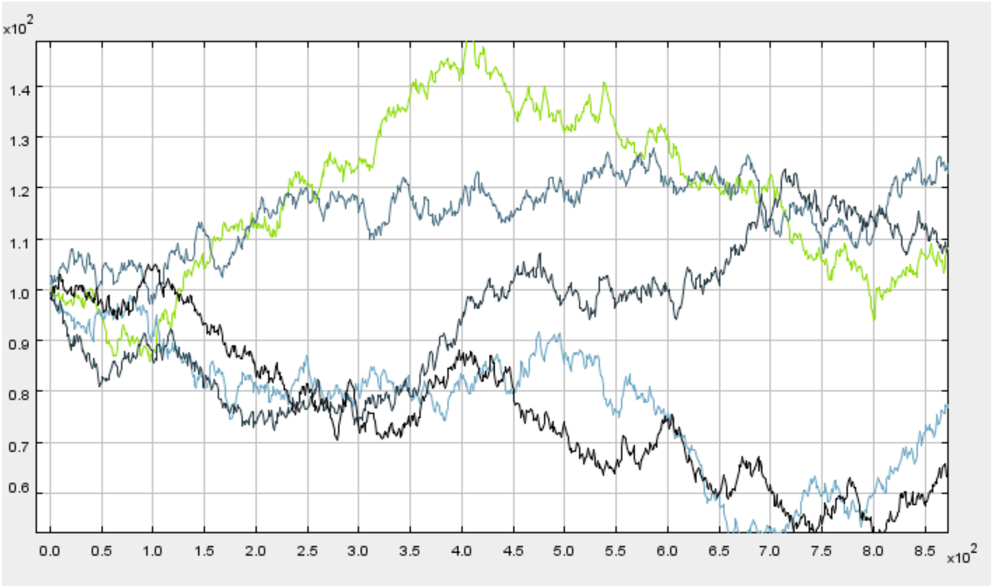

Figure 3.1: Time series with random walk.

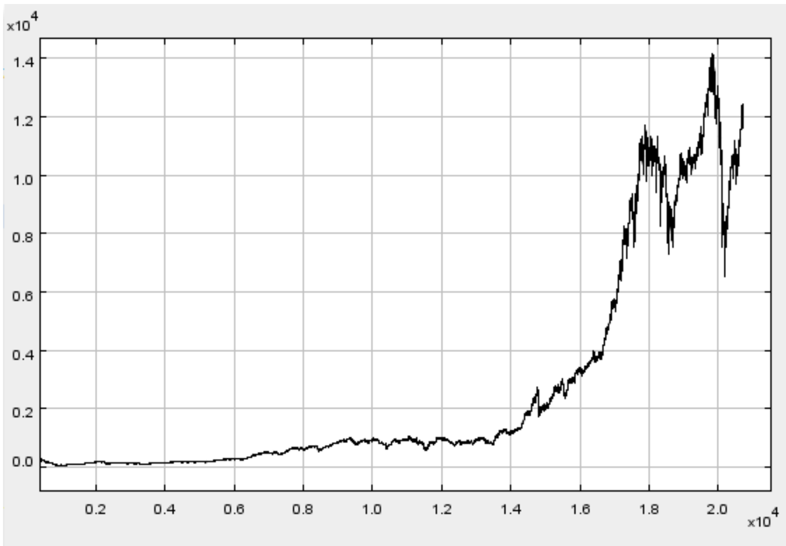

Figure 3.2: Historical prices of DJIA (Dow Jones Industrial Average) between 1901-2011

when referring to properties from this article. I have chosen to focus on some of the more basic properties which are easy to understand the meaning of.

The historical data set which I will use for reference is the American stock index "Dow Jones Industrial Average" (DJIA). This is one of the most famous indexes in the world composed from 30 of the largest American companies. I have looked at the closing prices (adjusted for dividends and splits) between 1901-2011. The time series for this time can be viewed in figure 3.2.

The stylized facts that I will investigate are:

1. **Aggregational Gaussianity:** the return distribution looks similar to the Gaussian distribution with large data sets ($N \to \infty$).

2. **Heavy tails:** the distribution of the returns displays more tail events (events far from the median or expected value) than for instance the Gaussian distribution.

   Figure 3.3 is a histogram of the returns of the historical DJIA prices compared to the Gaussian distribution. This illustrates both the aggregational Gaussianity and the heavy tails. The image is shown twice to be able to see the difference with both the Gaussian and the historical returns on top.

3. **Absence of autocorrelation in returns:** there are usually no significant autocorrelation between asset returns, except for with very short intra day times (≈ 20 minutes). In figure 3.4 this is shown for historical data together with the random walk series, which per definition is without correlation as it is a Brownian motion.

4. **Slow decay of autocorrelation in absolute returns:** the autocorrelation function for $|r_t|$ exists and decay slowly as a function of the time lag. This is shown in figure 3.5.

5. **Volatility clustering:** the volatility of the asset (standard deviation) have an autocorrelation function which is positive over several days. This shows that events cluster in time, varying between high and low volatility times. This is shown in figure 3.6.

6. **Value at Risk and Expected Shortfall larger than for random walk:** risk management and measuring risks are an important part of investment management today. Value at Risk (VaR) is the most popular risk measurement method today. It describes how much money an asset could lose in value if tomorrow is a "bad day". Usually one look at the 95% level and find the left quantile of the return distribution

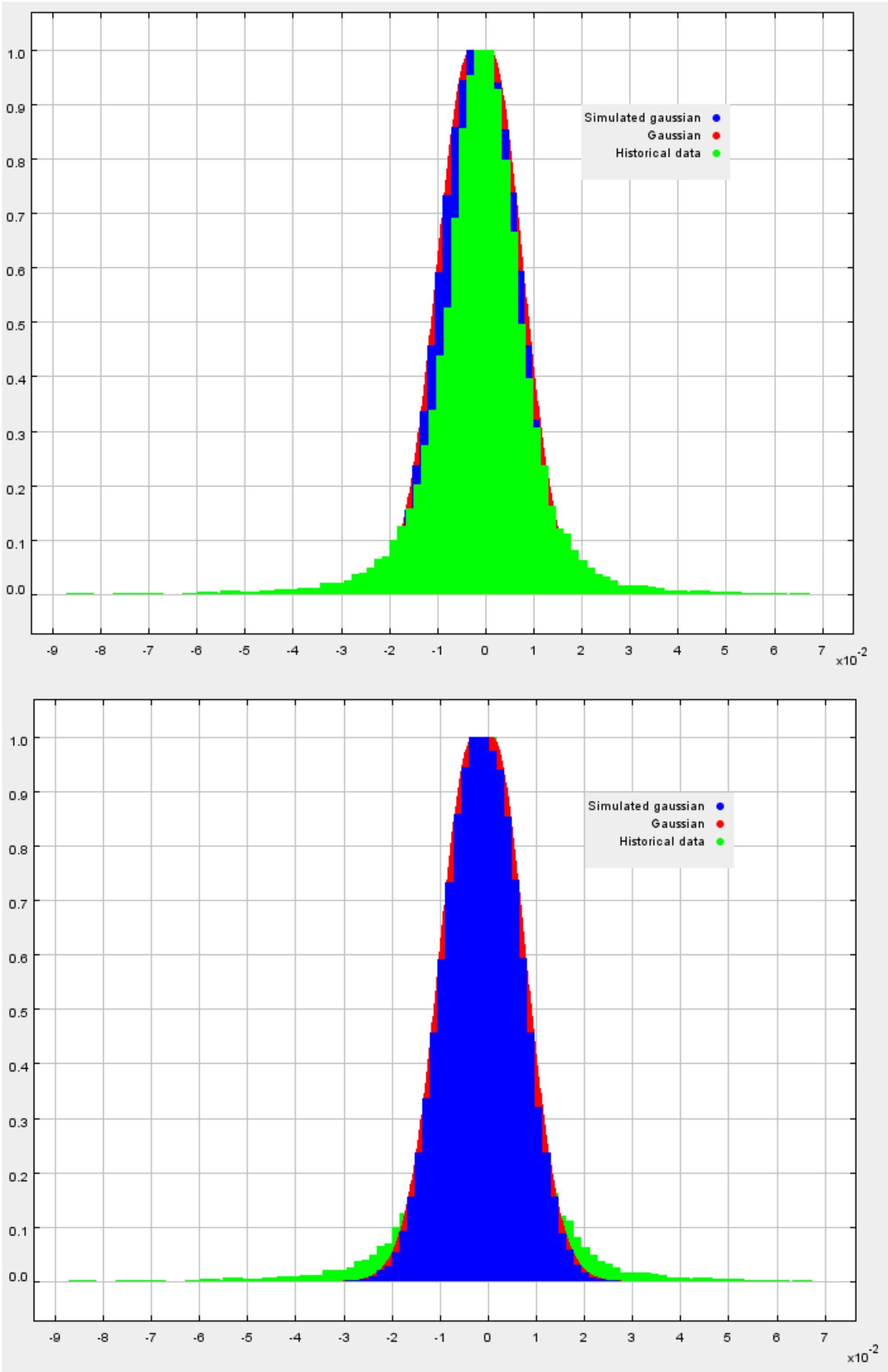

Figure 3.3: Random walk returns (blue) together with theoretical Gaussian distribution (red) and distribution of historical returns (green).

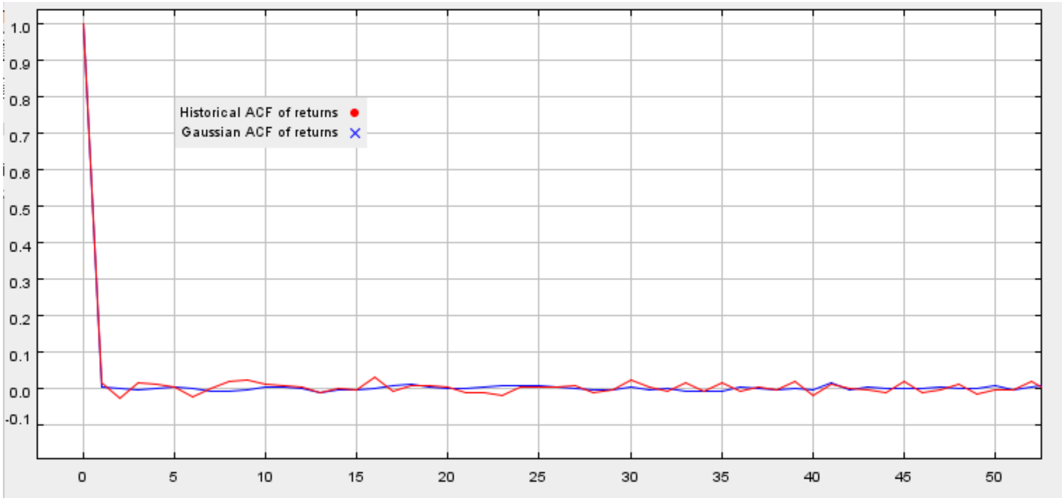

Figure 3.4: Autocorrelation function of returns.

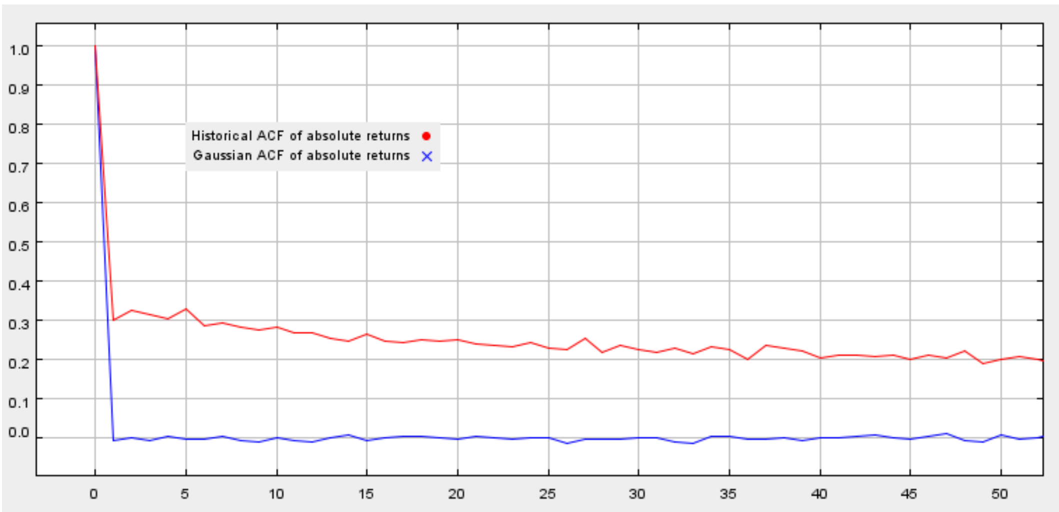

Figure 3.5: Autocorrelation function of absolute returns.

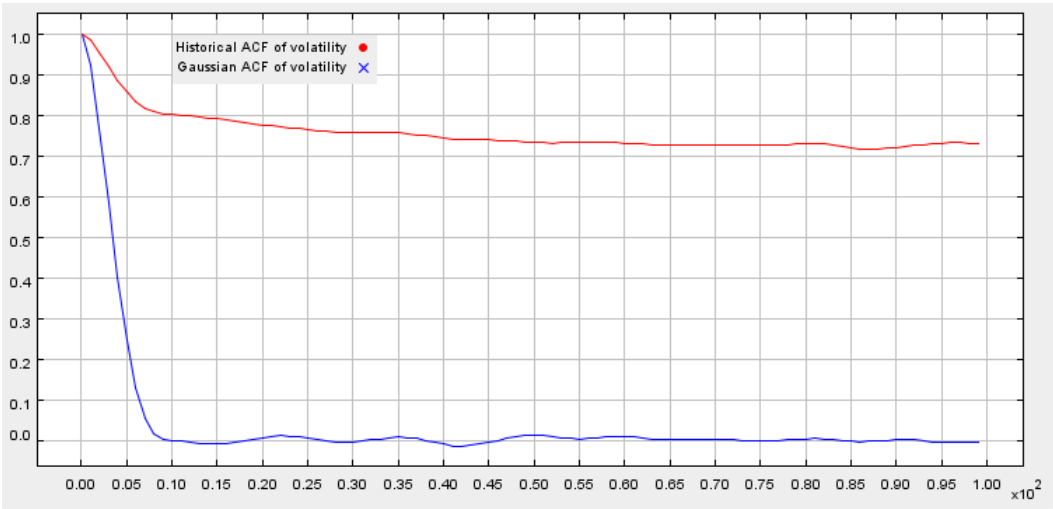

Figure 3.6: Autocorrelation function of volatility.

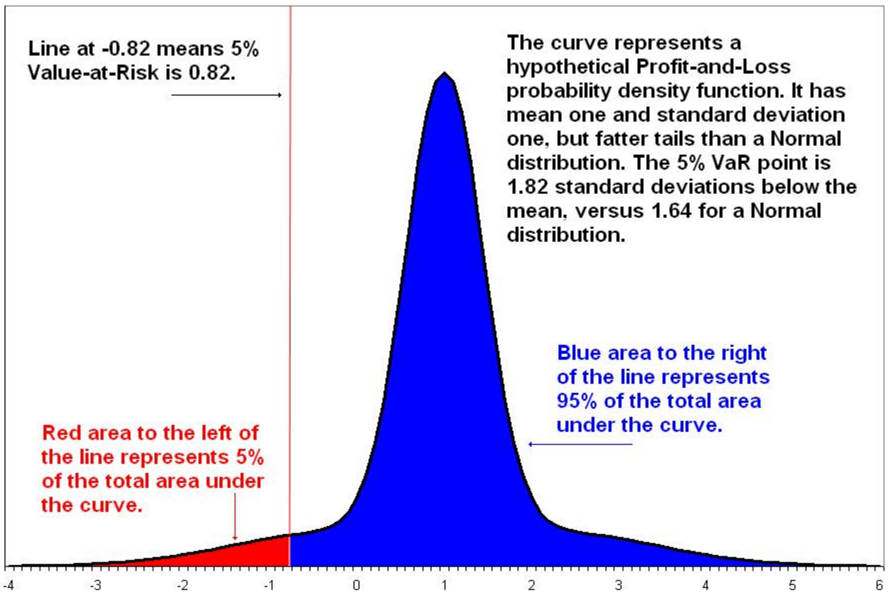

Figure 3.7: Illustration for the calculation of Value-at-Risk.

corresponding to only 5% of the returns displayed a worse result than the VaR-quantile [8]. It is interesting to look at the size of the largest negative returns in terms of percent to make different time series comparable.

VaR is illustrated in figure 3.7 [1]. The drawback of VaR is that it only describes that we are unlikely to lose more than a certain level. It does not quantify how much money we could lose if tomorrow is a bad day. Expected Shortfall (ES) can answer this and it is defined as the expected value of the losses which are worse than VaR. VaR and ES are measures of the extreme events just like the kurtosis.

[1]Image taken from [8].

Chapter 4

Herd mentality

"Herd mentality" is the phenomenon that the individual is making decisions based upon the actions of others rather than deciding by themselves. It is studied in social psychology. Herd mentality builds on the principal of social proof: "if everybody else is doing it, it must be the right thing to do". [9]

In the book "Influence" [9], the origin of herd mentality is explained from an evolutionary perspective. The summary is that the human brain has developed a set of "rules of thumb" on how to quickly make many of its decisions. With these rules of thumb only a few of the available facts and variables in the world are taken into consideration to ease the workload for the brain. In our modern society a person makes many thousands of decisions every day and if we took all the available information into account for each and every one of them we simply would not have time to get anything done. One of these convenient quick rules for making decisions is "if everybody around you is doing something which seems to work good for them, the right decision for you is most likely to do the same thing".

In the article "Consensus decision making in human crowds" [10], experiments showed that 5% of the population does 95% of the decision making. This shows that human beings are not as unlike animals as we sometimes like to believe.

4.1 Herd mentality in the stock market

Herd mentality in the stock market has been much debated by many articles. Most of them agree that herd behavior exists, but measuring the magnitude and impact of it is hard and no commonly accepted standard for this is known. One attempt of measuring herd mentality is done by Chang et al [11] based on how the equity markets from different countries covariate but

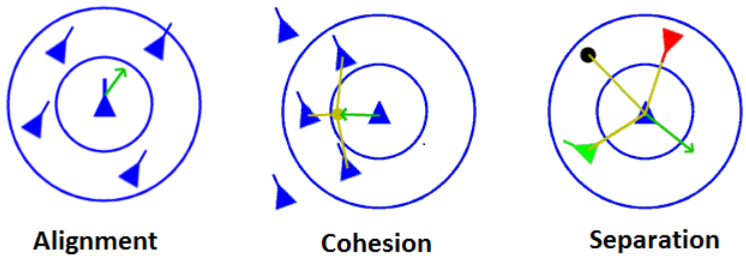

Figure 4.1: Herding rules illustrated.

they found no conclusive evidence of herding. Another article on the subject was by Grinblatt et al [12] in which they conclude that fund managers using a herding strategy perform significally better than other funds.

As the results vary vividly between different studies we can only be certain that there exists theories about herd mentality, and many of those. I have not found any financial articles describing how herd mentality can be modeled.

4.2 Mathematical rules of herd mentality

The first attempts to formulate the mathematical rules of flock behavior was done in 1987 by Reynolds [13]. His program "Boids" simulated how a bird moves in a flock. "Steering Behaviors For Autonomous Characters" [14] (also by Reynolds) describes suggested mathematical rules for flocking animals. There are basically three rules describing how animals move:

Cohesion: the animal wants to be close to the other animals and positions itself as the average of the position of its neighbors.

Alignment: the animal wants to move in the same direction as its peers.

Separation: the animal wants to avoid colliding with its neighbors and keeps a certain distance from them.

The way an animal apply the rules is illustrated in figure 4.1. These three rules have proven to be very accurate at describing the movements of animals in groups.

Chapter 5

Technical trading

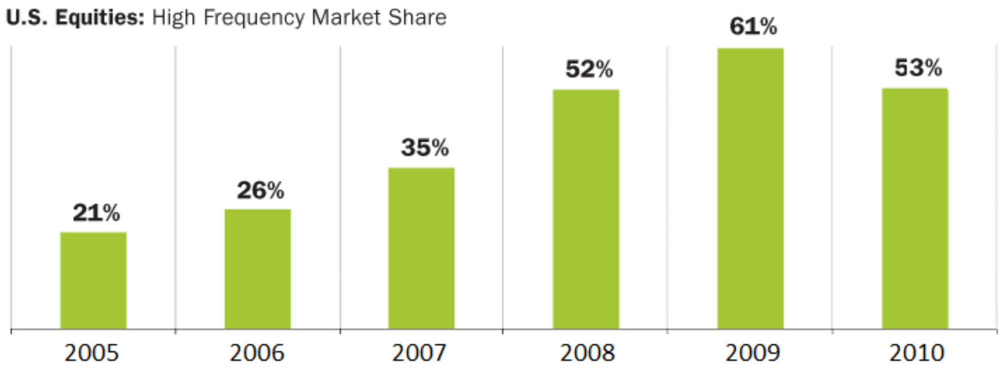

Figure 5.1: Market shares of technical trading 2005-2010.

"Technical trading" is a term for stock brokers who let a computer with a mathematic algorithm decide how they should do business and no human need to approve before the order is placed. Technical trading (also known as "Algorithmic trading") started in the 1970s and has quickly developed to become the largest field of trading in modern time, in terms of trading volumes.

"High-Frequency Trading" (HFT) is the primary user field of algorithmic trading [15]. These automatic traders do business with a time span of a few minutes or even seconds and try to analyze the trends. The trading volumes of HFT has increased massively between 2005-2009, from 21% to 61% [16]. See figure 5.1 [1].

At the so called "Flash crash" in 2010 [17], the Dow Jones Industrial Index plunged 9% in a matter of minutes. This started governmental investigations

[1]Image taken from http://www.futuresmag.com/Issues/2010/May-2010/Pages/The-lowdown-on-high-frequency-trading.aspx accessed on 2011-06-01.

about the harmful effects of high-frequency traders. Since then HFTs has decreased their trading volumes to 53% of stock-market trading volume, from 61% in 2009 [18].

5.1 Crossover strategy

The ”crossover” strategy is based on stochastic oscillation which was founded by George Lane in the 1950s [19]. It is a momentum indicator that indicates the position of the closing price compared to the high-low range within a time window. Four values are kept track of: maximum level in window, minimum level in window, three day average of maximum, three day average of minimum. These values are illustrated in figure 5.2. In the figure the moving averages are red and minimum/maximum values are green.

- When the minimum in window is higher than the three day moving average of the minimum, this is a *buy* signal.
- When the maximum in window is lower than the three day moving average of the maximum, this is a *sell* signal.

5.2 RSI strategy

One of the most famous books on technical trading is "New Concepts in Technical Trading Systems" by Wilder [20]. In this book the concept of RSI, Relative Strength Index, was introduced in 1978. RSI quickly became one of the most popular trading strategies for technical trading and it is still considered such today. The equations for calculating the RSI are as follows:

$$RS = RelativeStrength = \frac{AverageOf14DaysClosingUP}{AverageOf14DaysClosingDOWN}$$

$$RSI = RelativeStrengthIndex = 100 - \frac{100}{1 + RS}$$

When RSI is over 70 the asset is considered overbought and a sell signal is noted. When it is under 30 it is considered oversold and a buy signal is noted.

5.3 MACD strategy

MACD (Moving Average Convergence/Divergence) is a technical trading strategy invented in the late 1970’s by Gerald Appel [21]. The strategy

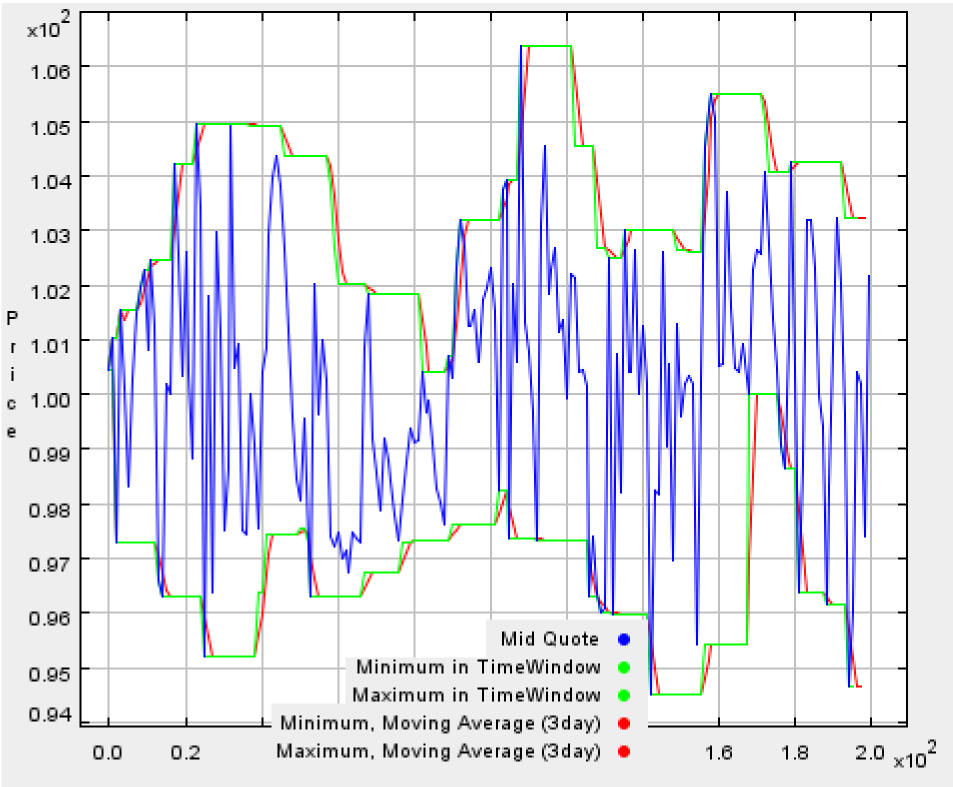

Figure 5.2: Variables used by the crossover strategy. A trading signal is received when the red line cross the green line in direction towards the center.

looks at moving averages of the asset price for different time lengths. When the curve for the shorter time moving average crosses the one for longer time a trend is spotted and a signal is noted.

Part II

METHODOLOGY

Chapter 6

Simulation environment: Java Auction Simulator API

6.1 Overview of JASA

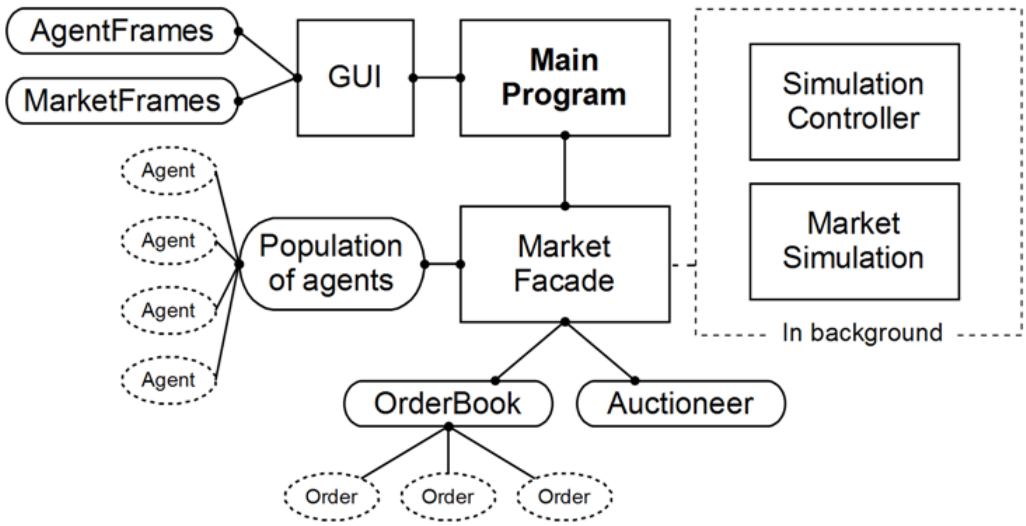

Figure 6.1: Simplified overview map of the parts in the JASA simulation environment.

Java Auction Simulator API (JASA) is an open source software created by Steve Phelps [1].

JASA consists of approximately 200 classes in 13 different packages. It is based on the simulation environment JABM (Java Agent-Based Modeling)

[1]Steve Phelps homepage: http://www.essex.ac.uk/ccfea/staff/profile.aspx?ID=205 accessed on 2011-06-01.

which in itself consists of 150 classes in 11 packages, also created by Steve Phelps.

The basic functionality of JASA is a framework for agent interaction and a market maker function. In reality the ”market maker” is the company who makes large scale trading possible by providing a market place for investors to find each other and trade. The market maker presents what assets are traded and at what price it is currently traded. When a stock broker places an order on the market it is the market maker who receives the order. The market maker also has an auctioneer which at a given time ”clears” the market. Clearing the market means it matches the buy-orders (”bids”) to the sell-orders (”asks”). If it is possible to find a matching bid at a higher price level than an ask, business can be made and the market maker acts as middle hand performing the transaction (and taking a small fee for this). Figure 6.1 show an overview of the most important parts of the JASA environment.

One important limitation of JASA is that there is only one asset on the market. This means that the agents makes an investment decision where they choose between owning the stock or owning the cash instead.

The entire runnable code is available for downloading at
http://fileadmin.cs.lth.se/ai/xj/WilhelmEklund/wilhelmeklund_thesis_code.zip.
The main program is "jasa/test/net/sourceforge/jasa/willestest/TestMarketSimulation.java".

6.2 Packages in JASA

As previously mentioned there are a lot of classes divided up into a number of packages. To understand the work performed in this thesis it is not important to go through all of them in detail. Instead here is a brief description of the most important packages and classes and their function within the JASA framework.

jasa.agent contains all different kinds of trading agents. Every agent must be given a strategy and valuation object to be able to place an order to the market.

jasa.strategy contains all the trading strategies available for the agents. This is the most important part of the agent. It decides weather the next order will be a bid or an ask and deciding the quantity of the order. The price is decided by the valuation policy (see below).

jasa.valuation contains the valuation policies. When the agent places an order at the market, the valuation policy decides what price it should have.

jasa.market contains all the classes related to the market objects. The most important classes from this package are:

- **MarketFacade** is the market maker object and the most important program for the simulation. It keeps track of (among others) the MarketSimulation, the Orderbook and the Auctioneer.
- **MarketQuote** is an object containing the current market price for the asset.
- **Orderbook** is the place where every agent places its order each round.
- **MarketSimulation** is the backbone program of JASA, handling all events occurring during simulation. Notifies classes with event listeners and keeps track of when the simulation is finished.

jasa.auctioneer contains the auctioneer classes. The auctioneer decides when to clear the order book on the market and decides the price for the transactions. In the simulations for this thesis I have chosen an auctioneer who clears the market at the end of a trading day (discrete clearing). More advanced auctioneer types can be used, for instance auctioneers that clear the market continuously when an order is placed.

jasa.event contains all the events that could happen during a simulation. The events are handled by the MarketSimulation.

jasa.report contains a large set of report classes. A report is an object that collects some kind of information about the market or the agents on it and summarizes it in a neat way. For computational reasons this saves a lot of computer power when for instance many agents want to collect the same information about something. With a report it is sufficient to collect the information once instead of everybody performing the same calculations.

jasa.view contains the Graphical User Interface (GUI) presenting all the interesting information about the market and the agents on it.

6.3 Market maker algorithm

The market maker function in the JASA framework is based upon an article by Wurman et al [22] on how a double sided auctions for electronic commerce should be implemented. The implementation is performed by an order book (where orders are placed) with four heaps:

B-in: Contains all of the bids that are in the currently matched set. The heap priority is minimal price, so that the lowest priced bid is on top.

B-out: Contains all of the bids that are not in the currently matched set. The heap priority is maximal price.

S-in: Contains all of the asks in the matched set, prioritized by maximal price.

S-out: Contains all of the asks not in the matched set, prioritized by minimal price.

The relationships between the ordering is displayed in figure 6.2 [2].

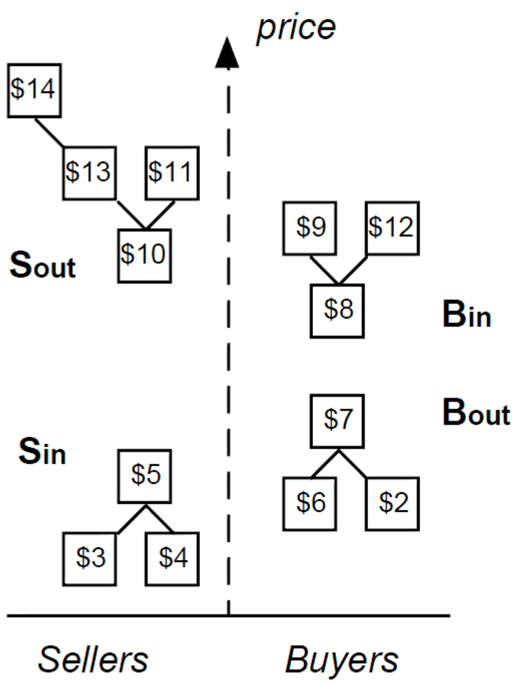

Figure 6.2: Four heap order book.

6.4 Dynamics for one day of trading

To perform the trading for one day in this simulated trading world a number of steps are taken and these are performed every day.

[2]Image taken from [22].

1. **Empty order book**. Any unmatched orders from yesterdays trading are removed from the order book since all the agents are going to want to update their orders for the new trading day.

2. **Mix agents**. The agents are organized in a list called Population. The order of the agents are mixed randomly every day to avoid that any agent gets an unfair advantage because it is placed early in the queue.

3. **Agents place orders**. Every agent is asked to place an order to the market.

4. **Clear the market**. Clearing is performed by the auctioneer.

5. **Calculate quote**. The quote price (closing price) at the end of a day is the price where no more transactions can be made with the orders in the order book. The price is calculated as the middle of the spread between the highest bid and the lowest ask that can not be matched. If all orders could be matched, the quote price is the last performed transaction price.

6. **Update reports and GUI**. All the reports that use information about the quote price are updated and the GUI is updated with the new information about the market and the state of the agents.

6.5 Newly developed classes for JASA

The largest part of my work has been in understanding the structure of JASA and developing new parts to this simulation environment. Apart from fixing some bugs in the old code some new classes were made to improve the environment. Here are some of the more important ones.

DSWReport is a report that keeps track of a lot of information about the market. The information stored here is used by almost all the trading strategies as well as the GUI. It saves away information such as closing prices, returns, moving averages of different time lengths and special time series such as those used by specialized trading strategies.

Agent base class improvements all the agents were extended with several functionalities such as calculating their fitness measure and ranking among the other agents.

OrderBookReport is a report used by the herding agents. It looks at the order book at the end of each round and calculates how the rest of the market is moving. There are two versions: one that listen to all agents and one that only listen to the top 5% performing agents (in terms of fitness measure).

AgentTypeID is a class that keeps track of the different kinds of agents, trading strategies and valuation policies that are valid on the market. This class can also easily create new trading agents if given the identity number of the strategy and valuation policy requested. This is very convenient for setting up the simulation.

DataSeriesWriter improvements . DataSeriesWriter is a class which saves a series of data results. It was extended to easily calculate statistical properties of itself and to return copies of itself that were altered in some way.

TimeSeriesProperties is a class which calculates more advanced statistical properties from a time series.

6.6 Graphical User Interface

I have developed a GUI for JASA to be able to follow the development of the market and the individual agents throughout the simulation. There are two general types of frames:

- frames watching the **market**
- frames watching the **agents**

The implemented frames are explained further in table 6.1.

Name of frame	**Market**	**Agents**	**Description**
Quote Frame	X		Follows the time series of end of day quotes for the stock. Can also visualize other series that are used by some strategies.
Auction State Frame	X		Displays the current state of the Orderbook and show supply and demand on the market.
Histogram Frame	X		Displays a histogram for the return distribution of the stock price.
Population Holding Frame		X	Displays how many stocks and how much cash each agent is currently holding. Either can have a negative value, since short selling of stocks and borrowing money to buy stocks is allowed.
Investmentgrade Fortune Frame		X	Displays the investment grade and the total fortune of each agent. Investment grade is the percentage of the agents total fortune invested in the asset. This can be negative since short selling is allowed.
Fitness Fortune Frame		X	Displays the fitness measure and the total fortune of each agent.

Table 6.1: Table describing the GUI-classes

Chapter 7

Implemented agents and trading strategies

7.1 Requirements to place an Order

Every agent has a strategy and a valuation policy to be able to place an order. To place a complete order on the market the agent needs to decide three things:

isBid decides weather to place a bid (buy-order) or ask (sell-order). This is the direction of the order.

quantity is the number of stocks to buy or sell.

price is the price level at which to place the order.

The first two are decided by the strategy and the last is set by the valuation policy as illustrated by figure 7.1.

7.2 Categories of strategies

The original JASA package contains some trading strategies and valuation policies but since the documentation is so poor for them it took a lot of effort to understand what they really did. Also once figured out most of them were very complex. Instead of using them I have implemented a bunch of my own strategies. The different categories for the implemented strategies are shown i table 7.1.

To make the performance of different agents easier to compare I have added a fitness measure to all agents. The fitness of an agent is calculated from the last months change of the fortune for the agent.

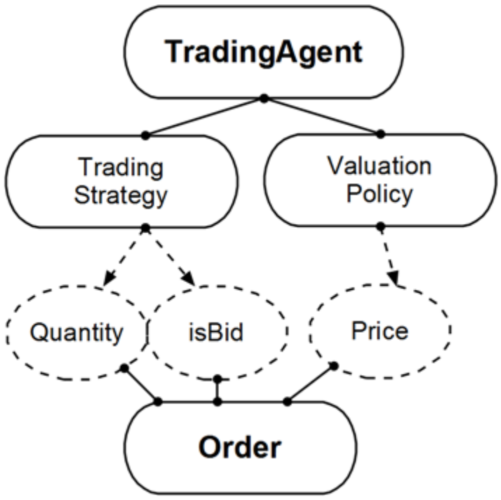

Figure 7.1: Parts of an Order and what decides what.

Strategy type	**Explanation**
Naive strategies	Very basic strategies acting on very simple rules.
Established technical trading strategies	Based on strategies found in financial articles.
Fuzzy logic trading strategies	Fuzzy logic alternatives of the other trading strategies.
Herding trading strategies	Listen to how other agents place their orders and base their own orders on this.

Table 7.1: Table of categories for implemented strategies.

$$Fortune = MoneyOnBankAccount + NumberOfStocks \cdot MarketPrice$$

7.3 Naive strategies

These strategies use very naive algorithms. Mostly used to fill out the market with some more orders.

7.3.1 FixQDir strategy

Simple strategy with a fixed quantity and a fixed direction meaning it produces the same result every day.

7.3.2 FixQVarDir strategy

Somewhat more realistic strategy where the quantity is fixed but the direction depends on weather the market price is higher or lower than the agents own valuation. If the market price is more expensive than its own valuation the asset is overpriced and it is time to sell. If the market price is cheaper it is undervalued and it is time to buy.

7.4 Established technical trading strategies

These strategies are described more extensively in the chapter on technical trading. Most of the times the rules giving them a buy/sell signal are not fulfilled and then they do not place any order to the market. To compensate this they trade with large quantities once they receive a signal. The quantity is calculated so that if the order goes through the investment grade (see table 6.1) of the agent will be either 1 or -1, depending on if it is a bid or an ask.

7.4.1 Crossover strategy

This is an implementation of the Crossover strategy.

7.4.2 RSI strategy

This is the implementation of the RSI strategy.

7.4.3 MACD strategy

This is the implementation of the MACD strategy.

7.5 Fuzzy logic trading strategies

I have implemented two trading strategies with fuzzy logic. Both of them are of the type where the direction of the order depends of the market price relatively its own valuation.

$$IF(ownvaluation > marketprice) thenBUY; elseSELL$$

The quantity of the order varies between the two. All fuzzy agents and strategies have two fuzzy variables: "confidence" of the agent and "priceworthyness" of the stock.

7.5.1 ConfQVarDir strategy

This strategy uses the confidence of the agent to determine the quantity of the orders. High confidence results in big quantity orders. "Confidense" is calculated using the ranking of the fitness of the agent compared to the rest of the population. If the agent fitness is in the top 10% of the population the quantity is five (5) stocks. If the agent is in the top 60% the quantity is two (2) stocks, otherwise the quantity is one (1) stock.

7.5.2 PriceQVarDir strategy

This strategy decides the quantity of the order based on the fuzzy variable price worthyness. Price worthyness is calculated from the market price of the stock compared to the agents own valuation. If the difference is large, in relative terms, profits from making a trade can be large since the agent considers the stock to have the wrong valuation. This leads to large quantity in the orders placed by this agent. "Price worthyness" is calculated using the relative difference of the current price of the stock and the own valuation of the agent.

$$diff = \frac{currentPrice - ownValuation}{ownValuation}$$

If $diff > 0.2$, the stock is very price worthy and the quantity is five (5). If $0.05 < diff < 0.20$, the pricerthyness of the stock is medium and the quantity is two (2). Otherwise the stock is not so price worthy and the quantity is one (1).

7.6 Herding trading strategies

The herding strategies are the most interesting for this thesis.There are two ways for them to decide what agents they should listen to: either they listen to everyone, or they only listen to the 5% of the agent population that are currently performing "best" according to the fitness measure. The 5% level is chosen since [10] suggested that 5% of the population takes 95% of the decision in moving human crowds.

It is not likely that one strategy will be successful every year on the real market. One year something works, the next year it does not. By having this more adapted herding strategy of listening to the current top 5% of the population, one imagin this agent would to be more likely to be successful many years in a row. The question of how often one should reevaluate which agents to look at is of course a matter of configuration to a specific market.

As we remember from the chapter on herd mentality there are three rules which a flocking animal follows. Table 7.2 suggests how one could transfer these rules into rules to be applied on the stock market.

	Alignment	**Cohesion**	**Separation**
Meaning for animal	· Move in direction of the others	· Place yourself as an average	· Keep some distance to the others
Translation to stock market	· They want to spot trends from other agents · Listen to the best agents	· They believe other agents have more information · Insecure about their own valuation · Don't want to be to offset from others	· They want to "beat" the market and make money

Table 7.2: Motivation for agents with herd mentality.

7.6.1 FixQHerdingDir strategy

This strategy use the same quantity every time but changes direction and follows the group they listen to.

7.6.2 HerdingQDir strategy

This strategy follows the same direction as the group they follow and they also place orders with a quantity which is the average of the quantity size for the group they listen to.

7.6.3 HerdingBeatmarket strategy

This is sort of a reversed strategy compared to the other herding strategies. It uses the same information as the previous ones but if the market indicates they want to buy, this strategy does the opposite and sell to the rest of the market at a higher price. This strategy (and it's corresponding valuation policy) is illustrated by figure 7.2.

7.7 Valuers

The valuation policy is the function that decides at what price the order should be placed. In a way this is the most important thing for the market

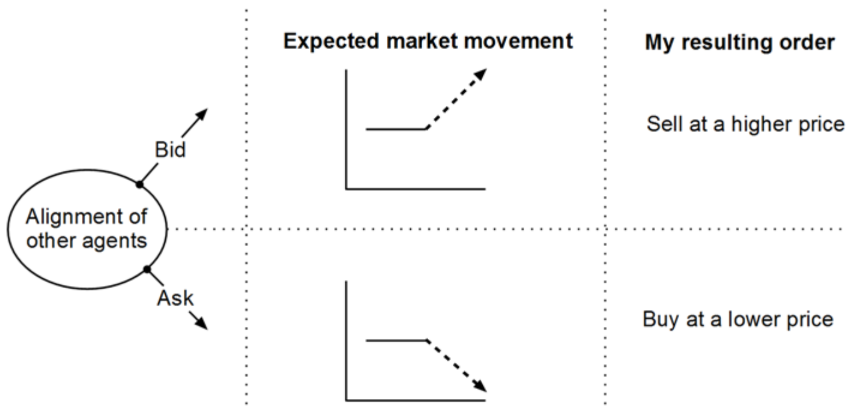

Figure 7.2: HerdingBeatmarketStrategy

price evolution. The set of valuation policies implemented for this simulation is illustrated in figure 7.3.

7.7.1 MarketpriceValuer

This simple valuation puts the price at the current market price.

7.7.2 FixedValuer

The FixedValuer returns the same value every time.

7.7.3 InflationValuer

The InflationValuer is an extension of the FixedValuer but it accounts for inflation and increases its valuation of the stock a little bit every day.

7.7.4 RandomValuer

There are two types of RandomValuers: one that changes its valuation every day but keep within a certain range. The other is a valuer which changes the value with a Gaussian distributed noise term. This is the same as the Random walk explained previously.

7.7.5 HerdingValuer

This valuation policy values the stock as an average of the price from the orders it has listened to. It can either listen to the whole population or only choose the top 5%.

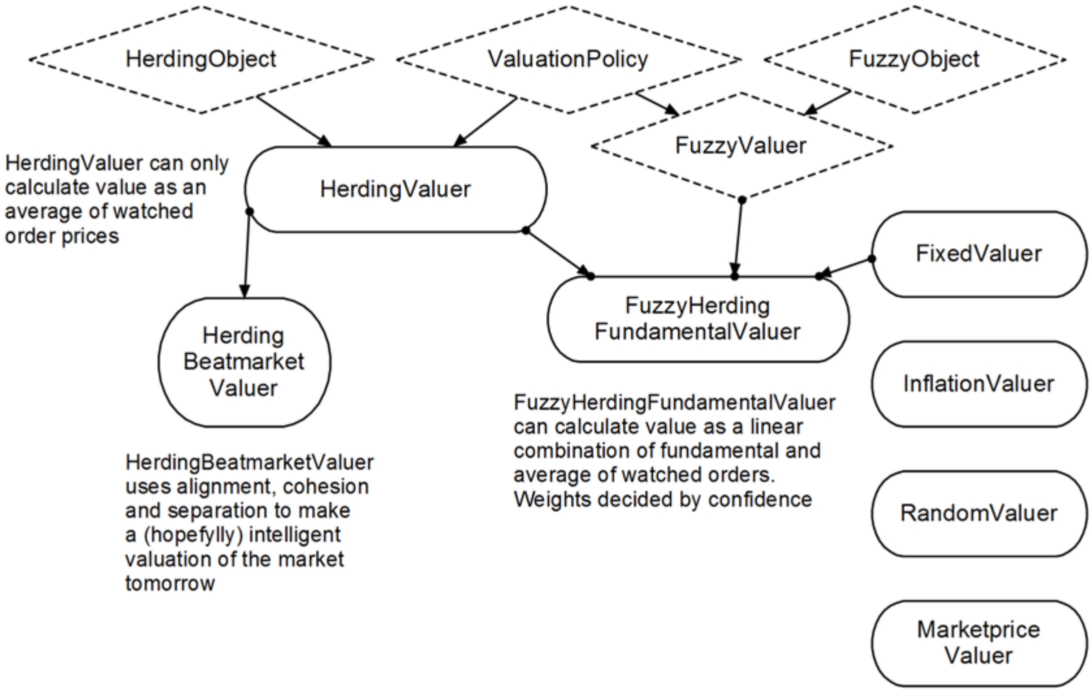

Figure 7.3: Implemented valuers

7.7.6 HerdingBeatmarketValuer

As explained by figure 7.2 this valuer goes against the herd and sell at a price slightly above the current market price if the market wants to buy, and buy at a price below the market price if the market wants to sell.

7.7.7 FuzzyHerdingFundamentalValuer

This strategy combines three other kinds of valuers: HerdingValuer, Fuzzy-Valuer and FixedValuer. The valuation is calculated as a linear combination of its fixed value and an average of the watched orders. The weights are decided by the confidence of the agent: if it has low confidence it does not believe so much in its own valuation but choses a value closer to the herding value. If on the other hand it has high confidence the value chosen is closer to its own valuation of the stock.

Chapter 8

Simulations

8.1 Setting up the simulation

Figure 6.1 has previously illustrated the anatomy of the JASA environment. The main program is the part where the setup of the simulation is performed.

1. First a MarketFacade object is created.
2. Then the rules for the simulation are decided. For instance you can decide if short selling of stocks should be allowed and if an agent can go bankrupt in case it has lost so much money that its fortune is negative.
3. The agents are created.
4. The graphical interface is started.
5. A ”run” command is sent to the MarketSimulation, initiating the simulation.
6. After the simulation has completed all interesting results are saved to file.

Since the quote price is determined by what orders are placed in the order book, and the orders placed are determined by the strategy and valuation policy for each agent, the development and outcome of the market is determined by the mix of agents and the configuration of the market. Because of this I want to variate which agent types exist on the market and how many they are between simulations. I have created a function to easily generate agents given the ID number of the strategy and valuation to make the setup part more efficient. The trading rules are also changed between different simulations, for instance boundaries for if short selling is allowed and to what level an agent can loan money to buy more stocks.

I want to keep the initial stock and cash endowments equal for all agents in every simulation to make it easier to compare the results of different strategies.

8.2 Saved results

After a simulation is completed time series about the market are saved to file together with some interesting states for each agent. They are saved in a folder named as the starting time of the simulation. The information is used in the analysis of the results.

The following time series about the market are saved for each trading day:

- Closing quote.
- Returns of closing quote.
- Standard deviation calculated for 10 and 30 days time period.
- Time series used by the crossover strategy: 14 day minimum and maximum of closing quote, and the three day moving average of minimum and maximum.
- Time series used by the MACD strategy: moving averages of the closing quote for the time periods of 3, 10, 30, 60 and 90 days.
- Time series for the RSI values.
- Total traded volumes on the market.

The following information is saved away about each agent at the end of a simulation:

- Strategy ID
- Valuation ID
- Stocks holding
- Cash holding
- Fortune at end of simulation
- Average fortune
- Fitness at end of simulation
- Average fitness

Part III

RESULTS

Chapter 9

Statistical properties of simulated market vs. real historical market

The results I have gotten from my simulated market have been compared to real historical data and to random walk simulations. As described in chapter 2 the evaluation of my results will be done both by face validation and by statistical validation.

For the validation I have used the most interesting simulations and compared them to 10 real time series of different traded stocks. The stocks used for the validation can be seen in table A.1 in the appendix. The reason why not all simulations are used in this analysis is that many of the simulations have only been done in test purposes and for short periods and do not contain very interesting results.

9.1 Face validation

9.1.1 The time series

First I compared the time series outcomes of simulations to the historical data and to time series of random walk. This can be viewed in figure 9.1. We can see that there are some similarities in the development of the time series but the simulation is a more stationary process and stays around the same value. Sudden large price jumps exist in both the simulation and the historical data.

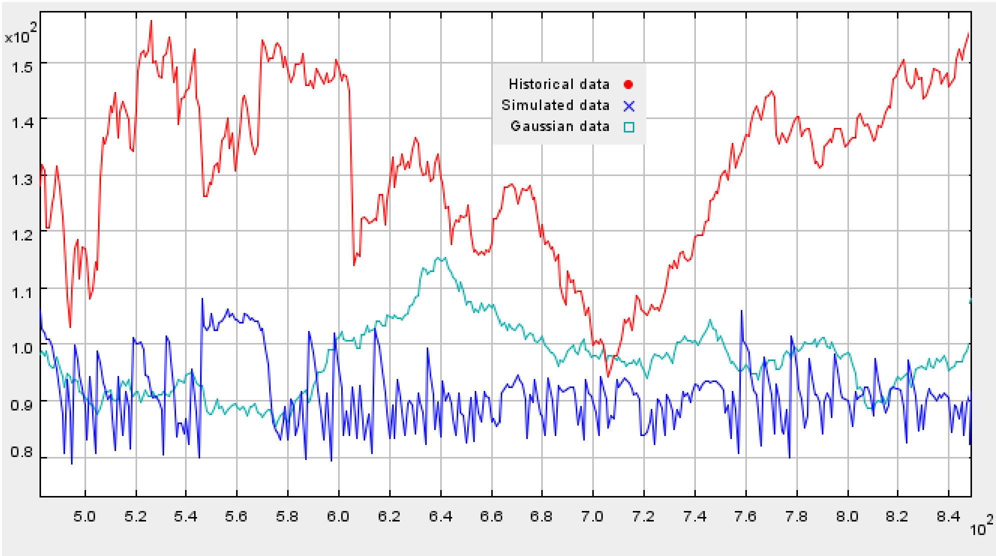

Figure 9.1: Time series of simulation (blue) compared to historical data (red) and random walk (turquoise).

9.1.2 Return distribution

It is hard to tell much about the statistical properties by only looking at the actual time series. The return distribution tell us a bigger picture of how value changes in the market appear. In figure 9.2 the histogram of the return distribution for a simulation is compared to the daily returns of the historical data. We can see that shape of the return distribution from the simulation looks vaguely similar to the historical data but the size of the returns (viewed on the X-axis) are much larger for the simulated data.

It appears that much larger changes happen on the simulated market compared to daily changes in reality. If weekly historical changes are used for comparison instead of daily, the aggregated changes should be closer to the distribution of the simulation. This result can be viewed in figure 9.3 and the similarities are surprisingly good. The shape of the main curve for the distributions are almost identical.

The tails from the simulation returns are much larger than the historical tails, especially the positive tail with extreme gains. Noteworthy is also that the positive tail is remarkably larger than the negative tail for the simulation.

The center of the historical returns appear to have a slight shift to the right compared to the simulated returns. One could believe that this would be explained by historical increase in value but this is not the case. The last value in the historical time series was 12 505 dollars year 2011 and the first was 240 dollars year 1901. This is 52.1 times its original value. Calculated for the 4 306 weeks this constitutes an average excess return of only 0.092% per week.

$$240 \cdot (1 + 0.00092)^{4306} = 12586 \approx 12505$$

This explanation does not explain the tilt since the shift in figure 9.3 is approximately 1% and 0.092% can not even be measured on this coarse scale.

A more believable theory is that the large positive gains in the simulation are compensated with many small negative losses since both the start price and the end price of the simulation are about 100 money units. This could explain why the center of the simulated distribution is shifted slightly to the left.

Because of the nice fit of the return distribution, stylized fact number 1 of aggregated Gaussianity and number 2 of heavy tails can be validated to fit to reality.

9.1.3 Autocorrelation of returns

Figure 9.4 illustrates the autocorrelation function of returns for historical data, simulations and for random walk. We can see that the simulations have captured the negative autocorrelation for the one day time lag which appears in the historical data better than the random walk. In figure 9.1 we see that the value of the simulated market almost always goes up if it went down on the previous day, and vice versa. However the oscillating effect is a little bit too frequent as the autocorrelation function does not disappear after the one day time lag. Stylized fact number 3 states absence of autocorrelation and this can not be validated.

9.1.4 Autocorrelation of absolute returns

Figure 9.5 illustrates the autocorrelation function of absolute returns for historical data, simulations and for random walk. We can see that it is hard to tell the difference between historical data and random walk as they have approximately the same shape and are in the same region of autocorrelation level.

This makes face validation for stylized fact number 4 of slow decay for the autocorrelation function in absolute returns hard and I can not validate it.

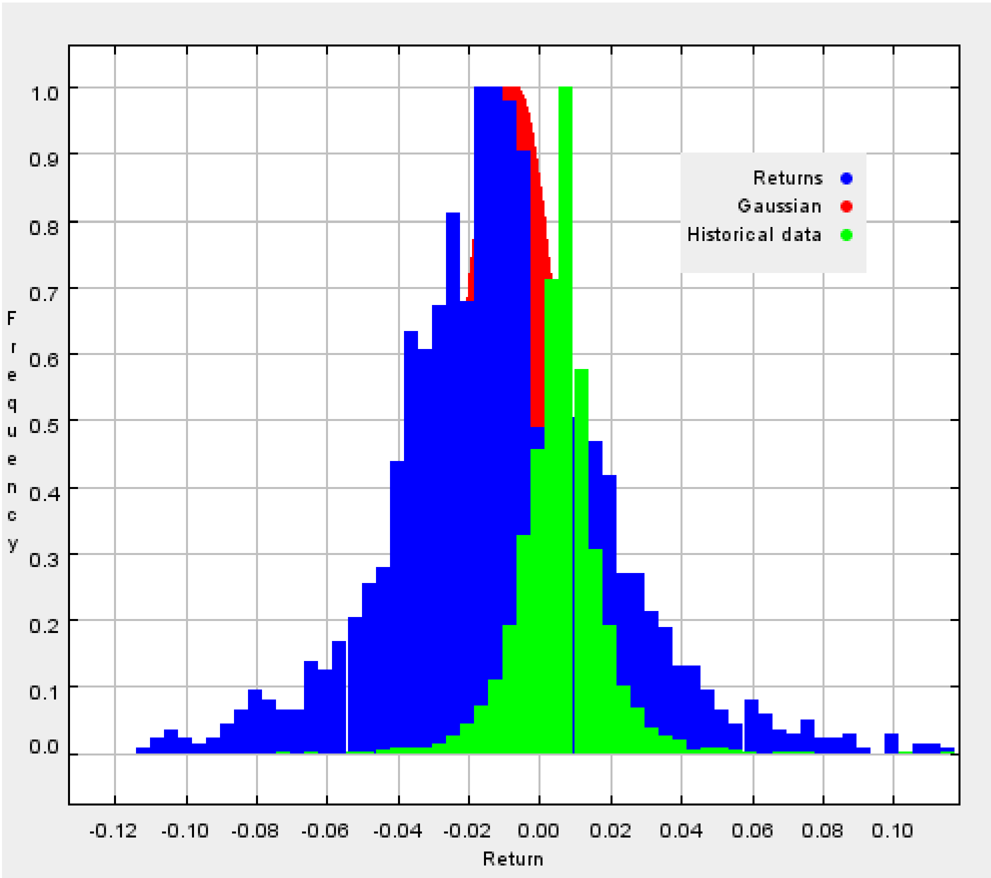

Figure 9.2: Histogram of historical daily returns (green) vs simulated returns (blue) and Gaussian for comparison (red). Simulation time used: 2011-04-22 12-26-44. See appendix B.

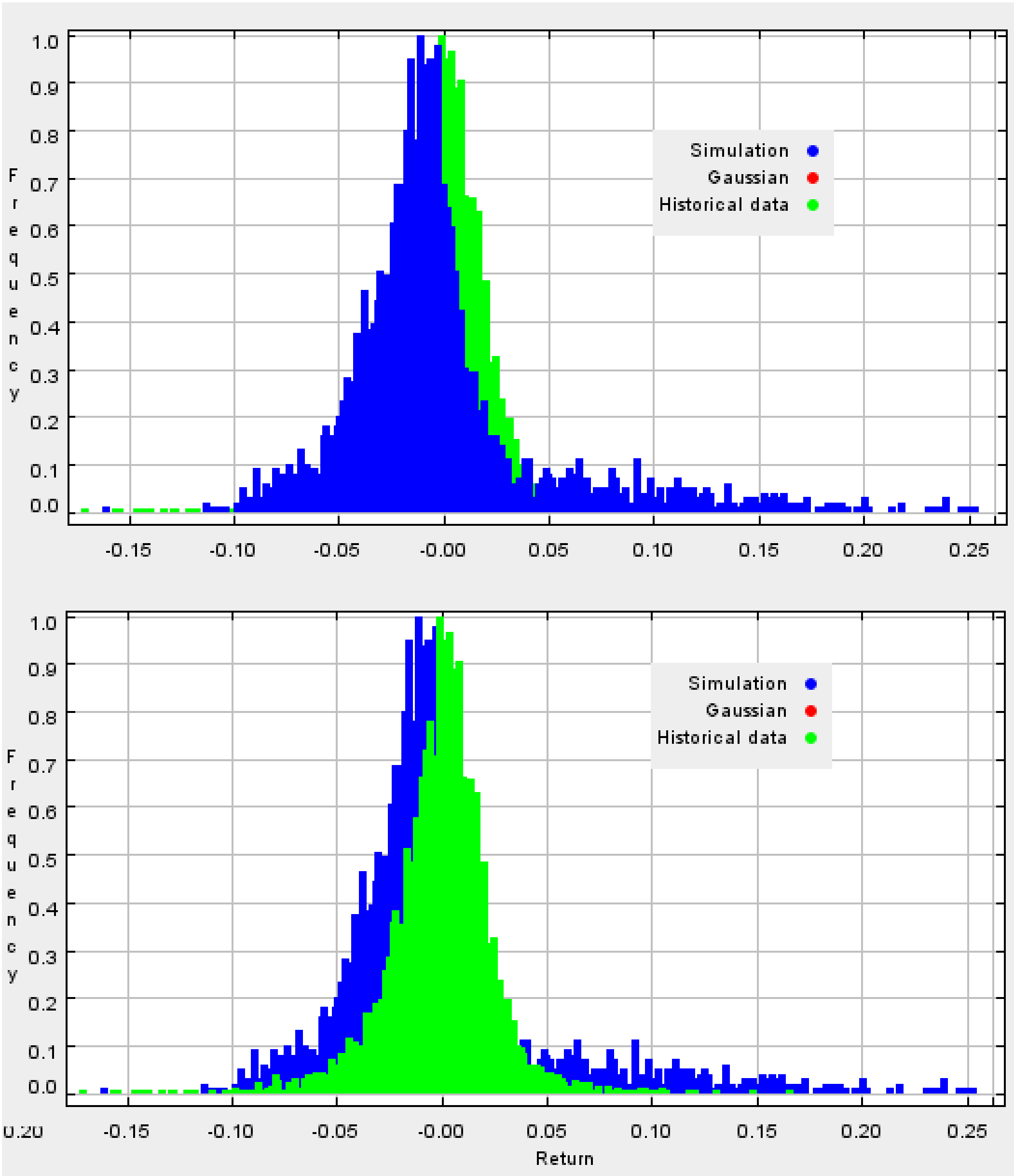

Figure 9.3: Histogram of historical weekly returns (green) vs simulated returns (blue). Simulation time used: 2011-04-22 12-32-52. See appendix B.

Figure 9.4: Autocorrelation function of returns for historical data (top), simulations (middle) and for random walk (bottom). The simulated data is the one describes in appendix B.

Figure 9.5: Autocorrelation function of absolute returns for historical data (top), simulations (middle) and for random walk (bottom). The simulated data is the one describes in appendix B.

9.1.5 Autocorrelation of volatility

Figure 9.6 illustrates the autocorrelation function of market volatility for historical data, simulations and for random walk. One can see a difference between the graphs for historical data and the random walk. The simulated data are somewhere in the middle of these two. Stylized fact number 5 states volatility clustering i.e. that times of high volatility are followed by times of low volatility. This is quantified by a slow decay of the autocorrelation function for the volatility. About half the simulations cluster their volatility as good as the historical data and half are only slightly better than random walk. This does not give support enough to validate volatility clustering.

9.2 Statistical validation

The statistical validation process contain two parts:

1. Calculation of expected value and standard deviation for the measures explained in chapter 3 which produce a single number rather than a time series.

2. Calculation of the correlation matrix between those measures.

For the statistical validation the same method as for the face validation has been used where I have compared historical data with the simulated data and for reference also random walk data.

Table 9.1 show the expected value and standard deviation for the measure of kurtosis, value at risk and expected shortfall. Here we see that the kurtosis value for the simulations is very close to historical data. This is a very interesting result since the kurtosis measure in itself remove expected value and standard deviation of the distribution and only measures how fat the tails are compared to the rest of the distribution.

The value at risk measure also exhibit the tail of the distribution, but only the negative tail. Also here the simulated data come very close to the historical data, 6.5% losses compared to 4.5% losses on average. Remember that the VaR is defined as a positive measure of how much the asset could lose if tomorrow is a bad day. An interesting detail here is that kurtosis and value at risk make different ordering of the fat tails. Kurtosis find more fat tails in historical data but value at risk consider the simulated data as more risky. This is most likely because of the asymmetric tails of the simulations.

As we remember from chapter 3, expected shortfall is calculated as the expected value of all the losses worse than VaR. This explains that the level

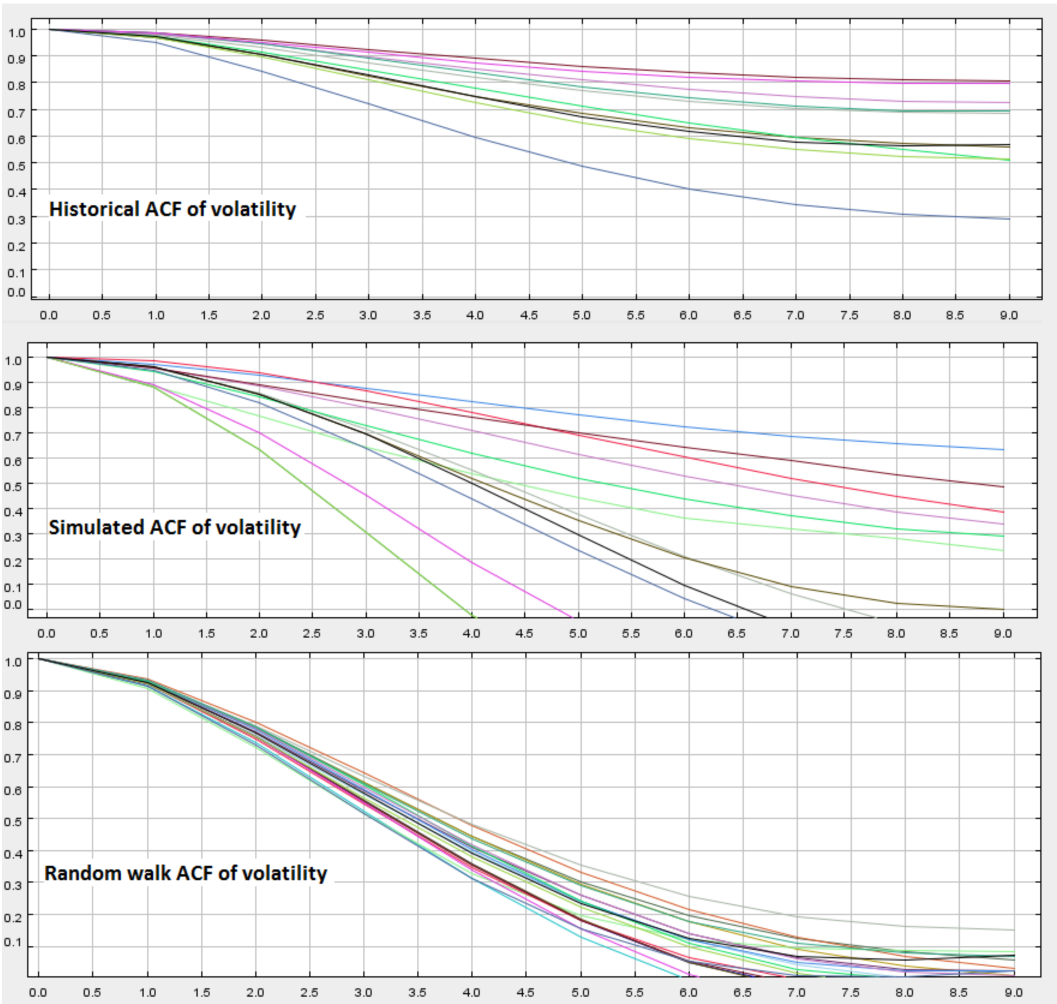

Figure 9.6: Autocorrelation function of market volatility for historical data (top), simulations (middle) and for random walk (bottom). The simulated data is the one describes in appendix B.

	Data	**nbrOfDays**	**Kurtosis**	**Value At Risk**	**Expected Shortfall**
Exp. value	*Historical*	5213.3	**15.8**	**0.045**	0.072
	ABM sim	870.7	**9.27**	**0.065**	0.113
	Random walk	903.4	0.26	0.016	0.021
Stddev	*Historical*	5472.0	13.7	0.019	0.032
	ABM sim	1061.4	13.7	0.071	0.18
	Random walk	295.5	0.62	0.0048	0.0072

Table 9.1: Expected value and standard deviation of the measures between different simulation.

of expected shortfall is also approximately the same level for the historical data and the simulations, 7.2% and 11.3% respectively. A very interesting result is the standard deviation of expected shortfall of 18%, which shows that some of the simulations had extremely fat negative tails. In figure 9.3 we saw an example with extreme positive tail.

In table 9.2 the correlation matrix for these three measures has been calculated. It is very interesting to see that the correlation between kurtosis and value at risk do not exist in the historical data but is strong in both the simulated data and the random walk. As explained in chapter 3 expected shortfall depends on the measure for value at risk so it is not surprising that all the time series has almost a perfect correlation between those two. The high correlation between kurtosis and expected shortfall is an effect of the correlations between kurtosis and value at risk for the simulated data and the random walk.

This statistical validation gave further arguments for the result found from the face validation section regarding heavy tails (stylized fact number 2). Stylized fact number 6, that value at risk and expected shortfall are larger than for random walk, could also be validated.

9.3 Some interesting simulations

Some of the simulations produced very interesting results. One of the most interesting is displayed in figure 9.7. It shows the build up of a financial bubble on this simulated market. This was caused as an effect of the crossover strategy. All the agents following this strategy has the same rules for when

	Data	**Kurtosis**	**Value At Risk**	**Expected Shortfall**
Kurtosis	*Historical*	1.0	-0.15	-0.11
	ABM sim	1.0	**0.60**	**0.69**
	Random walk	1.0	**0.82**	**0.88**
VaR	*Historical*	-0.15	1.0	**0.99**
	ABM sim	0.60	1.0	**0.98**
	Random walk	0.82	1.0	**0.98**
ES	*Historical*	-0.11	0.99	1.0
	ABM sim	0.69	0.98	1.0
	Random walk	0.88	0.98	1.0

Table 9.2: Correlation matrix between the measures.

they make business. First a buy signal is received and they start buying the stock, which increases the quote price of it. The quote price is a concave function as the daily price increments are decreasing in size. After some time the price is oscillating around a certain point and after a while the sell signal for the crossover strategy tell all the crossover agents to sell at the same time. This causes the quote price to plunge and in only a matter of days the price is back on its original level. This simulation is very interesting because these bubbles exist on the real market also, both short term and long term. One difference between this bubble and real market bubbles are that the simulated price level stay at the previous level after the plunge whereas the price would continue to drop below this price for real market. I believe this is an effect of the very simple valuation policies available to these agents. The agents using a fixed valuer always place their orders at the same price.

The performance of the agents on this market is determined by how much money they make. The fitness measure for the agents is calculated by short range fortune increments. At the end of the simulations the average fortune of an agent is calculated. With the GUI class InvestmentgradeFortuneFrame we can watch how different trading strategies perform compared to each other. Figure 9.8 show this from a simulation with four different kind of strategies. We see how agents using the same strategy are placed close to each other in most cases but not in all. In this simulation agents using the FixQVarDir strategy, represented by yellow dots, were the most successful.

In a different simulation, displayed in figure 9.9, we can see how a herd

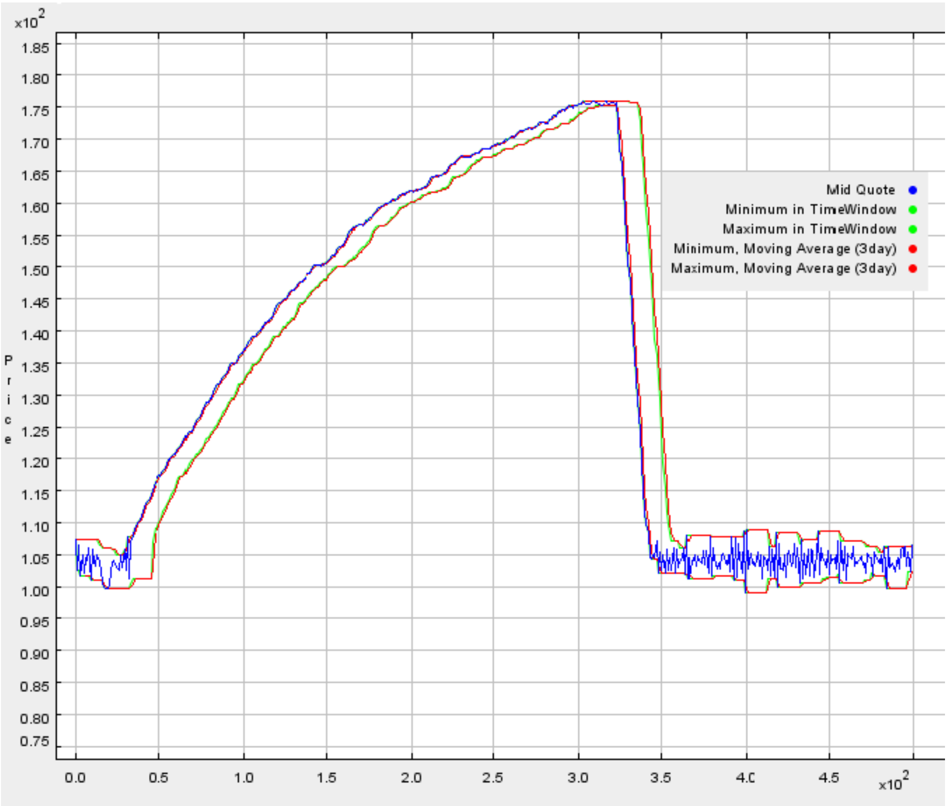

Figure 9.7: Simulation displaying a financial bubble.

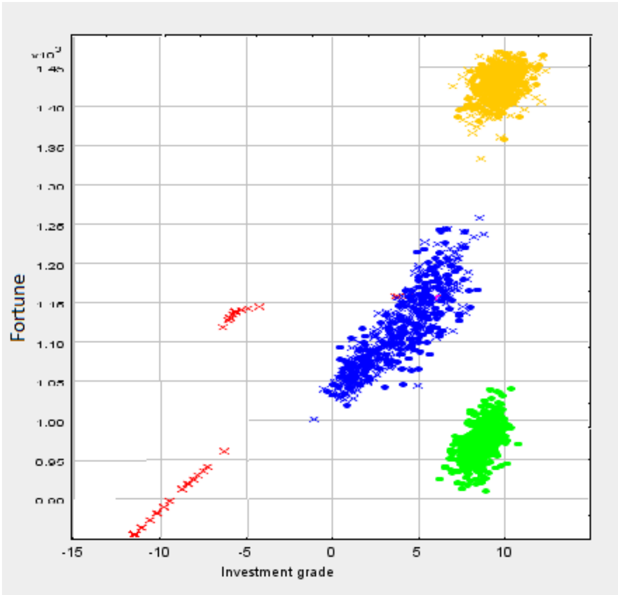

Figure 9.8: InvestmentgradeFortuneFrame showing how different strategies performance in one simulation. X-axis show investment grade, Y-axis show fortune. Green dots are FixQHerding strategy, yellow dots are FixQVarDir strategy and blue dots are FixQDir strategy.

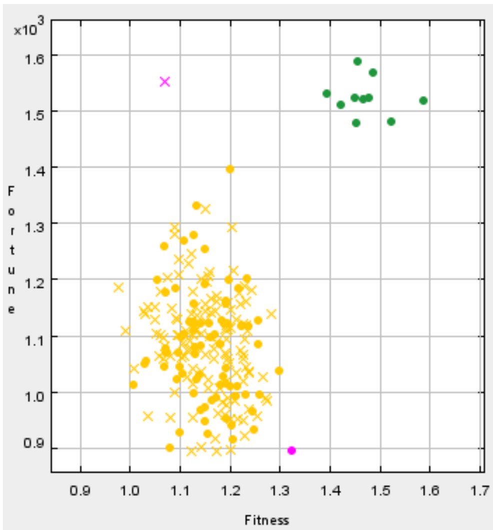

Figure 9.9: Simulation where a herd strategy (green) outperform FixQVarDir strategy (yellow).

strategy, FixQHerdingDir strategy, represented with green dots, actually outperform the FixQVarDir strategy (yellow dots). This is a very interesting result since the herd strategy only follow the majority of all the other agents. I believe this could be an effect of the negative autocorrelation function of the returns for the time lag of one day.

9.4 Interpretation of results

Two different kinds of "herd behavior" can be seen in this simulation:

1. Agents who actively look at the activities of other agents and follow.
2. Agents who has the same trading rules as many of the other agents. This can create large effects once the rule is fulfilled, as we saw with the financial bubble in figure 9.7.

The simulated "market crash" does not look like real life crashes with fast decline and then slow recovery to a more fundamental price. This is because in this simulation each agent only has **one** strategy, normally a mix of strategies is used for every broker with certain "warning bells" alerting them that something is happening. In this way, the simulated market is more stable than reality because it has "true" fundamentalists who do not change there evaluation under any circumstances.

A pattern I have noticed is that the price changes are larger on a market with few agents than it is for a market with many agents. This is a liquidity effect of having many agents trading with each other. One can imagine that the orders placed in the order book follow some distribution and with more agents the distance between two orders are more fine grained. A market with many agents is consequently more stable than one with few agents.

An important issue to mention is that the returns can easily be "scaled" by configurating the general valuations of the market to a different level. We could for instance see the same absolute price fluctuations in the market if we move the general valuation from 100 money units to 20 money units. This however will obviously have great impact on the relative returns as they are calculated as the price change in percents.

9.5 Verification of the simulation model

Some different aspects of this model are considered here.

Clearing function of the simulation : The clearing function of the market is based on the real algorithm used in the market. In my simulations I have used a discrete clearing function instead of a continuous. I choose the discrete because is was a less complex model to understand. There exist markets with bad liquidity where clearing is done in discrete time so the clearing function is a believable representation of reality.

Agent approach to model the stock market : On the real market orders are placed by brokers acting with their own strategies and valuations of the market. This is exactly what agent-based modeling describe, which makes it an ideal way of modeling the stock market. The individual strategies and valuation policies of this simulation are quite simple though and should be developed further to make the model even more believable.

Herd agent implementation : The implementation of herd agents in this simulation is based on the assumption that the market is using an open order book. This is the case in some special cases but most of the fast moving trading markets of today do not allow it.

Agent mix for the market : As mentioned earlier the mix of agents to represent the real market is not known so verifying that the model uses the right mix can not be done. This part is covered in the validation part and is different for every simulation.

In the end the most important features of reality (the way orders are placed by individual brokers and the clearing of the market) are implemented very well by this model which makes the model verified in my point of view.

Chapter 10

Conclusions

Table 10.1 show a summary of the results for the validation of the stylized facts tested in this thesis.

Three out of six stylized facts could be validated as better than the random walk approach to simulate the stock market. This is actually a good result since many of the models used in reality show approximately between two to four features. Very few models (if any) display all of these properties so, considering how simple some of the implemented strategies are, this is a pretty good result.

I can conclude that the agent-based model is a better approach than the random walk to simulate the stock market.

Strategies based on herd mentality were implemented and tested in this thesis and the results were varied. It was interesting to see that it could be modeled rather easily even though the number of markets with open order books is shrinking.

Due to the fundamental structural similarities between the agent-based approach to modeling, and the fact that the financial market is constructed from brokers (agents) trading with each other, the agent-based approach to model the stock market should be considered very good, and I believe this type of modeling will become much more popular in the future.

#	**Stylized fact**	**Result of validation**	**Comment**
1	Aggregational Gaussianity	Validated	This can be validated from all the simulations.
2	Heavy tails	Validated	The tails of many simulations are even greater than historical data in some cases. This property was validated both by face and statistical validation.
3	Absence of autocorrelation in returns	Not validated	For the time lag one day there exists a negative autocorrelation for most of the simulations, just like the historical data, but the other time lags were too correlated to support validation.
4	Slow decay of autocorrelation in absolute returns	Not validated	It was hard to tell the results from historical data from random walk data so this stylized fact was not very evident for the historical data either.
5	Volatility clustering	Not validated	This one was a close call. The results were somewhere between the historical data and random walk, most of them better than random walk.
6	Value at Risk and Expected Shortfall larger than for random walk	Validated	These properties of the simulations showed good similarities to historical data and distinct difference from random walk.

Table 10.1: Summary for the validation of the stylized facts.

Chapter 11

Future work

During the six months I have spent on this thesis, many ideas have come to me regarding what could be implemented for this financial simulation. Some of them have been implemented but not tested with data. Some of them I only have the idea for how I would like to implement them. Some I do not know how to implement but it would be interesting to see the results of. For various reasons they have not been presented in this thesis but here is a brief description of them.

- Compare agent performance on the simulated market. Investigate if any of the strategies always perform better than the other.
- Simulation against historical data. In stead of pairing together orders from the order book, the market could use historical quote prices as the threshold to whether an order will come through or not. This would allow a performance test of the strategies against real market prices and the strategies could be ranked according to how they would perform on the historical market. Worth to mention is that only mathematical strategies can be truthfully evaluated this way, since the social herd strategies depend on the market mix of the simulation.
- Multiple assets. The market should be equipped with more than one asset to give the agents more choices. If there were more assets, strategies for handling stock portfolios could be modeled.
- The simulation should use continuous clearing to make it more realistic.
- A different way for the herding agents to choose whom to listen to would be everybody within a certain "distance" in some chosen space, for instance the <Investmentgrade, Fortune> space.

- All agents should use more than just one strategy to make their decisions. Fuzzy logic would be more useful if one could classify the current market situation in some fuzzy aspects and make decisions as a linear combination of different strategy measures. The FuzzyHerdingFundamentalValuer is one step in this direction.

- The valuation policies would need some more work since they all kind of naive right now. This is obviously very important for the quote outcome.

- The technical trading strategies I have implemented are mostly used for high-frequency trading, but I have used all strategies with the same time intervals. One area of improvement would be to use more than one round each trading day and regulate how many times per day the different strategies can place orders.

- Learning algorithms should be possible to implement to reward the agents that perform well and let them know they are doing good. A weak form of this has been implemented with the FuzzyHerdingFundamentalValuer which values the asset closer to the own valuation than the market price if the agent has good confidence, i.e. the agent has performed well the last couple of days. There is no learning involved, but the behavior is altered depending on performance.

- No predictive abilities of the model has been tested, but it would of course be interesting to investigate when performing simulations using historical prices if there is any prediction ability for this model. This would require more investigating of the true agent mix on the market, which is something that can never be verified as correct.

Part IV

APPENDIX

Appendix A

Historical data used for validation of model

Name	From time	Interval	Number of values in time series
ABB	2001-2011	Daily	2546
Accenture	2001-2011	Daily	2452
Amazon.com	1997-2011	Daily	3520
Apple	1984-2011	Daily	6737
Apple	1984-2011	Weekly	1394
DJIA	1901-2011	Daily	20731
DJIA	1901-2011	Weekly	4306
DJIA	1927-1933	Daily	1064
Goldman sachs	1999-2011	Daily	3028
Microsoft	1986-2011	Daily	6355

Table A.1: Historical data used for validation of the model.

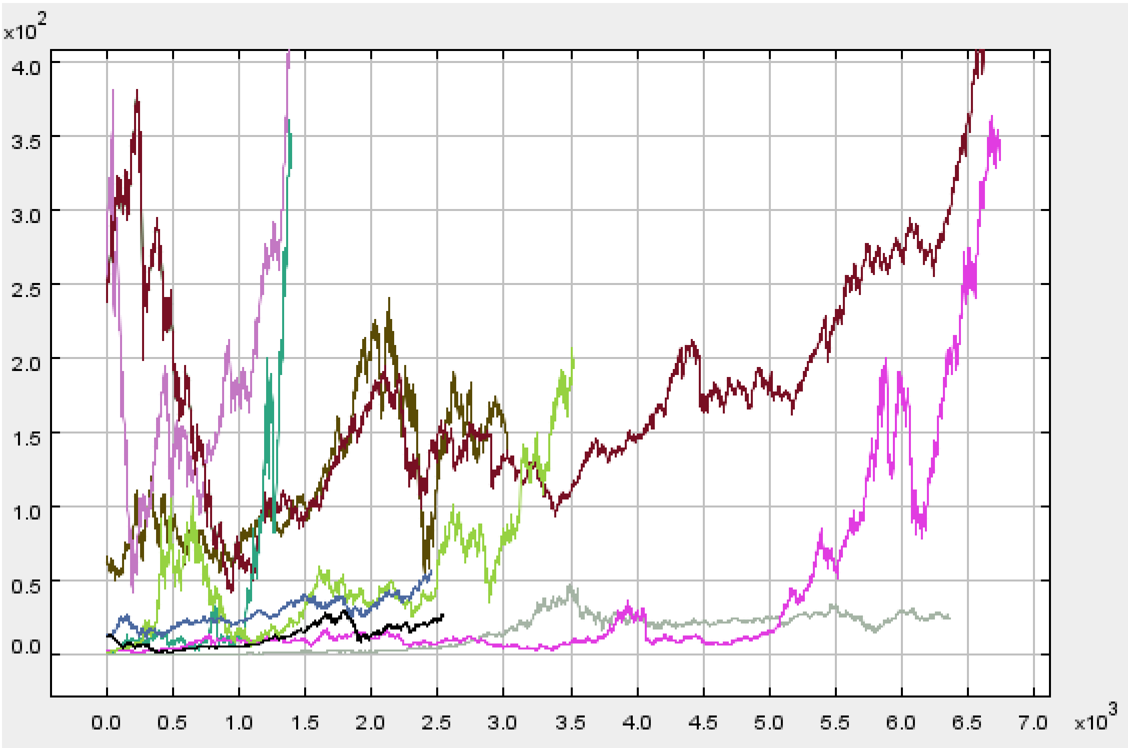

Figure A.1: Historical time series used for the validation of the model.

Appendix B

Simulated data

Simulation time
2011-04-20 – 11-27-14
2011-04-20 – 11-30-21
2011-04-20 – 19-52-46
2011-04-20 – 19-56-07
2011-04-20 – 20-07-39
2011-04-20 – 21-27-56
2011-04-22 – 12-26-44
2011-04-22 – 12-32-52
2011-04-22 – 14-24-54
2011-04-26 – 17-15-07
2011-04-26 – 17-17-06
2011-04-27 – 13-59-32
2011-04-28 – 15-04-48
2011-05-12 – 19-03-31

Table B.1: Dates of the interesting simulations used for calculating statistical properties. The actual values are found together with the runnable files at http://fileadmin.cs.lth.se/ai/xj/WilhelmEklund/wilhelmeklund_thesis_code.zip under the catalog "jasa/simulations/" and the folder named the simulation time. Simulated quote prices are found in the file "MidQuote.csv" and the file describing the agent mix is called "agentMatrix.csv".

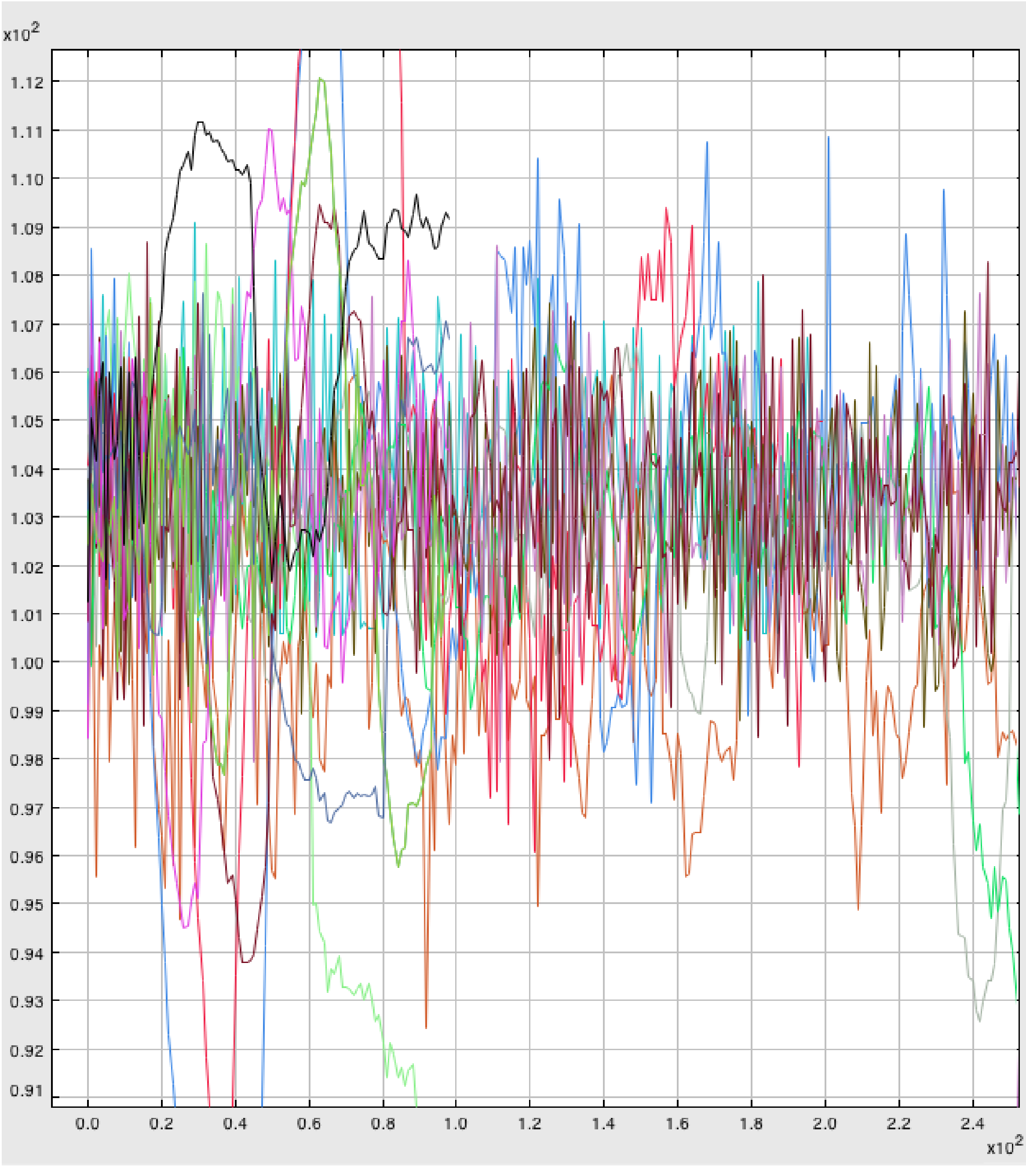

Figure B.1: Simulated time series used when calculating the statistical properties.

Bibliography

[1] Wikipedia.org *"Agent-based model"* . http://en.wikipedia.org/wiki/Agent-based_model. Accessed on 2011-06-01.

[2] Ball, P, 2010. *"The earth simulator"*. New Scientist, vol 208, p.48-51.

[3] Wikipedia.org *"Conway's Game of Life"* . http://en.wikipedia.org/wiki/Conway's_Game_of_Life. Accessed on 2011-06-01.

[4] Klügl, F. *"A validation methodology for agent-based simulations"* Proceedings of the 2008 ACM symposium on Applied computing table of contents, Fortaleza, Ceara, Brazil, Pages 39-43, 2008.

[5] Wikipedia.org *"Fuzzy logic"* . http://en.wikipedia.org/wiki/Fuzzy_logic. Accessed on 2011-06-01.

[6] Blom, G. et al. *"Sannolikhetsteori och statistikteori med tillämpningar"*. Studentlitteratur, 2005.

[7] Cont, R. *"Empirical properties of asset returns: stylized facts and statistical issues"*. Quantitative finance, Vol 1 (2001), p.223-236.

[8] Wikipedia.org *"Value-at-Risk"*. http://en.wikipedia.org/wiki/Value_at_risk. Accessed on 2011-06-01.

[9] Cialdini, R. *"Influence"*, chapter 4. HarperCollins ebooks, 2007.

[10] John R.G. Dyer et al. *"Consensus decision making in human crowds"*. Animal Behavior, vol 75, p.461-470, 2007.

[11] Chang et al. *"An examination of herd behavior in equity markets: An international perspective"*. Journal of Banking & Finance 24 (2000) p.1651-1679.

[12] Grinblatt et al. *"Momentum Investment Strategies, Portfolio Performance, and Herding: A Study of Mutual Fund Behavior"*. The American Economic Review, Vol 85, No.5 (1995), p.1088-1105.

[13] Reynolds, C. *"Flocks, herds and schools: A distributed behavioral model"*. Computer Graphics 21(4), July 1987, p.25 - 34. http://www.red3d.com/cwr/papers/1987/SIGGRAPH87.pdf. Accessed on 2011-06-01.

[14] Reynolds, C. *"Steering Behaviors For Autonomous Characters"*. http://www.red3d.com/cwr/papers/1999/gdc99steer.pdf. Accessed on 2011-06-01.

[15] Wikipedia.org *"Algorithmic trading"*. http://en.wikipedia.org/wiki/Algorithmic_trading. Accessed on 2011-06-01.

[16] Gehm, F. *"High frequency trading lowdown"*. FuturesMag, May 2010. http://www.futuresmag.com/Issues/2010/May-2010/Pages/The-lowdown-on-high-frequency-trading.aspx. Accessed on 2011-06-01.

[17] Wikipedia.org *"Flash crash"*. http://en.wikipedia.org/wiki/Flash_Crash. Accessed on 2011-06-01.

[18] Cui and Lauricella *"Mini Crashes Hit Commodity Trade"*. The Wall Street Journal, 2011-05-05. http://online.wsj.com/article/SB10001424052748704322804576303522623515478.html. Accessed on 2011-06-01.

[19] Guerillastocktrading.com *"Full Stochastic Indicator How To"*. http://www.guerillastocktrading.com/lessons/full-stochastic-indicator-how-to/. Accessed on 2011-06-01.

[20] Welles Wilder, J. *"New Concepts in Technical Trading Systems"*. Hunting Publishing Company, 1978. ISBN 0-89459-027-8.

[21] Appel, Gerald. "Technical Analysis: Power Tools for Active Investors". Financial Times Prentice Hall, 1999. ISBN 0-13147-902-4.

[22] Wurman, Walsh and Wellman, 1998. *"Flexible Double Auctions for Electronic Commerce: Theory and Implementation"*. Decision Support Systems, 24:17-27, 1998.