Universiteit van Amsterdam

Microsimulation of Financial Markets

Joost Geerdink
Ketan Maheshwari
Dhiraj Hegde
Thomas Miedema
Stanley Jaddoe

13th June 2006

Computational Finance

Contents

CHAPTER 1

Introduction

The complex dynamics of financial markets can be characterized by some ‘stylized facts’ (e.g., non-Gaussian (fat-tailed) distribution of return and long term memory of volatility, etc.). These stylized features are common across many financial instruments, markets, and time horizons. Most of them are counter-intuitive and are contrary to the expectation of traditional financial theories.

1.1 The Cont-Bouchaud Model

Among the microscopic models, the Cont-Bouchaud (CB) model [1] is a relatively simple one, which can be solved even analytically. In this model, traders are organized in a random communication structure (i.e., a random graph), in which the nodes represent traders. The connection between traders represents the formation of a coalition of traders who follow the same strategy (buy, sell, or not to trade). This model has led to interesting conclusions regarding the microscopic origin of heavy tails in the distribution of asset price variation. Specifically, this model shows the relation between the non-Gaussian distribution of price change and the tendency of traders to imitate each other (herding).

1.2 The Cellular Automaton Model

For the same purpose (i.e., simulating the complex dynamics of financial markets), a cellular automaton (CA) model was developed [2]. Within this model, a financial market is represented by a two-dimensional lattice where each vertex stands for a trading agent. According to the typical transaction behavior in real financial markets, agents of only two types are adopted: fundamentalists and imitators. This model is based on local interactions and adopts simple rules for representing the behavior of agents and a simple rule for price updating. It is shown that the model can reproduce, in a simple and robust manner, the main characteristics observed in empirical financial time series. Fat tails due to large price variations are generated through the imitation behavior of agents, where long-range interactions are formed from local interactions. In contrast to other microscopic simulation (MS) models, results of this CA model suggest that it is not necessary to assume a certain network topology in which agents group together (e.g., a random graph or a percolation network). Volatility clustering, which also leads to fat tails, seems to be related to the combined effect of a fast process and a slow process: the evolution of the influence of news and the evolution of agents' activity, respectively. After all, the non-Gaussian distribution of return is produced by agents' behavior in response to the arrival of news, even though the influence of the news on agents' perceived value of the asset is assumed to follow a Gaussian distribution.

1.3 Project description

In this report we we discuss some results of an emperical study. We analyse the time series and try if we can observe the listed stylized facts in Chapter 2.

We will also report on a software implementation of the CA and the CB model. In Chapter 3 we show that our CA model implementation is able to reproduce the figures in [2] regarding time series of return, distributions of return, and autocorrelations of return and volatility for the models at different levels of sophistication.

Simulation results of the CB model are discussed in Chapter 4. The results of the CA and CB model will be compared and discussed in Chapter 5.

CHAPTER 2

Emperical studies of stock price data

In this chapter, the results of a small empirical study is discussed.

We have downloaded and analysed market data from Yahoo! Finance for different stocks and indices. We will try to observe the stylized facts listed in 1.

2.1 Definitions

Before we can show the results of the analysis, we will first discuss the mathematical definitions we used for the different stylized features.

2.1.1 Return and normalised return

The list of returns $r_1, r_2, \dots r_N$ is calculated using the stock prices S_t at time t, and S_{t+1}.

$$r_{t+1} = \log S_{t+1} - \log S_t \tag{2.1}$$

The return can be normalised by

$$R_t = \frac{r_t - \bar{r}}{\sigma} \tag{2.2}$$

with $\bar{r}$ the average return and

$$\sigma = \sqrt{1/N \sum_{i=1}^{N} (r_i - \bar{r})^2} \tag{2.3}$$

The (normalised) return is calculated on a daily basis (i.e., t is in days). We use the normalised return for easy comparison of different stocks with the Gaussian and Lorentzian distributions.

The normality of a distribution, (or 'fatness' of the tail) can be calculated using

$$\kappa = \frac{m_4}{(m_2)^2} \tag{2.4}$$

where m_2 and m_4 are the second and fourth central moment, respectively. If $\kappa = 3$ then the distribution is Gaussian, for $\kappa > 3$ a leptokurtic distribution with sharp peak and fat tails is obtained.

2.1.2 Autocorrelation

The autocorrelation of both the normalised return and volatility is calculated.

Given a list of normalised returns $R_1, \ldots, R_N$, then the autocorrelation τ_k with lag k is calculated using

$$\tau_k = \frac{\sum_{i=1}^{N-k}(R_i - \bar{R})(R_{i+k} - \bar{R})}{\sum_{i=1}^{N}(R_i - \bar{R})^2} \tag{2.5}$$

with $\bar{R}$ the average normalised return.

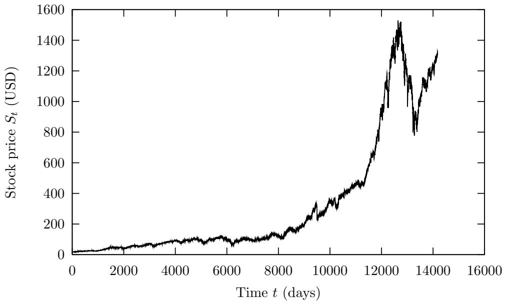
The numerator of (2.5) can be calculated as the inner product of two vectors. Using a special linear algebra package (BLAS), a very fast implementation of the autocorrelation function was written.

The volatility at t is defined as $|R_t|$. The autocorrelation of the volatility is calculated in the same way as the autocorrelation of the normalised return.

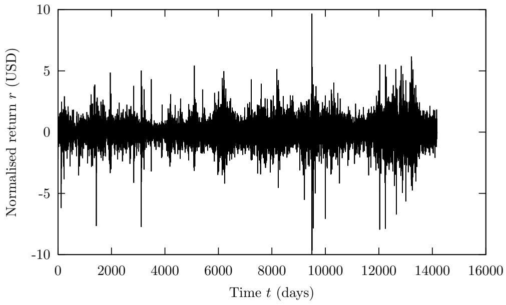
2.2 Examples

In this section, actual stock prices and index values are investigated using the methods described in the previous section. For each stock, the normalised return, return probabilities, and autocorrelations are calculated.

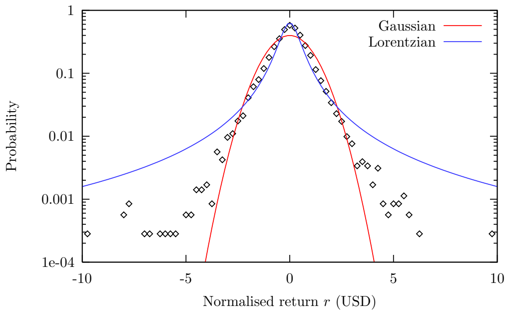
In each of the Figures 2.1, 2.2, 2.3, 2.4, and 2.5, we have observe the expected stylized facts (fat tails, volatility clustering).



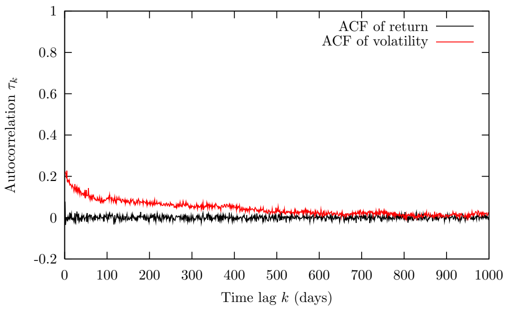
(a) Price movements as function of time in days.



(b) Normalised return as function of time in days.

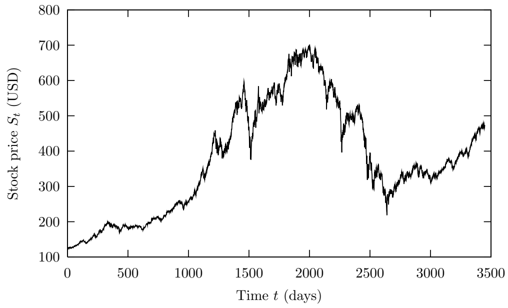


(c) Probability of normalised returns.

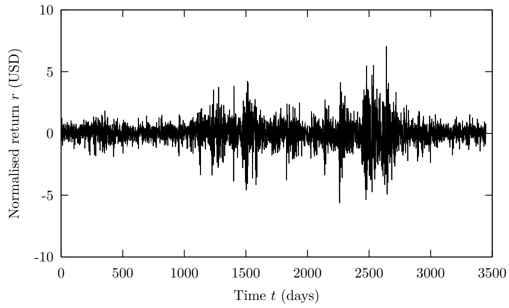


(d) Autocorrelation functions of the volatility and the normalised returns

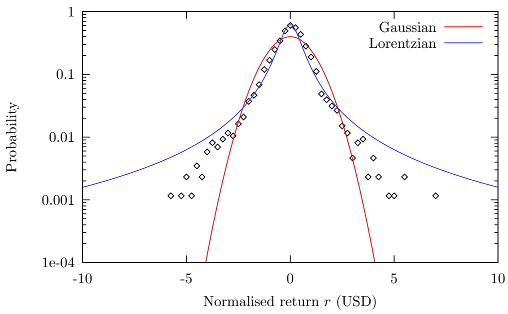
Figure 2.1: Analysis of the S&P 500 Index between January 1950 and May 2006. There is fat tail, the kurtosis is $\kappa = 38.22$. The autocorrelation of the volatility is decaying slowly, this means there is 'volatility clustering'. This is the phenomenon where high (positive or negative) returns group together.



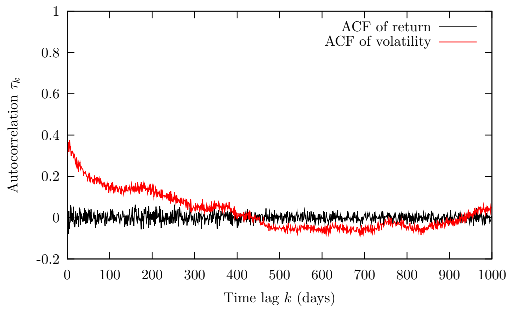
(a) Price movements as function of time in days.



(b) Normalised return as function of time in days.

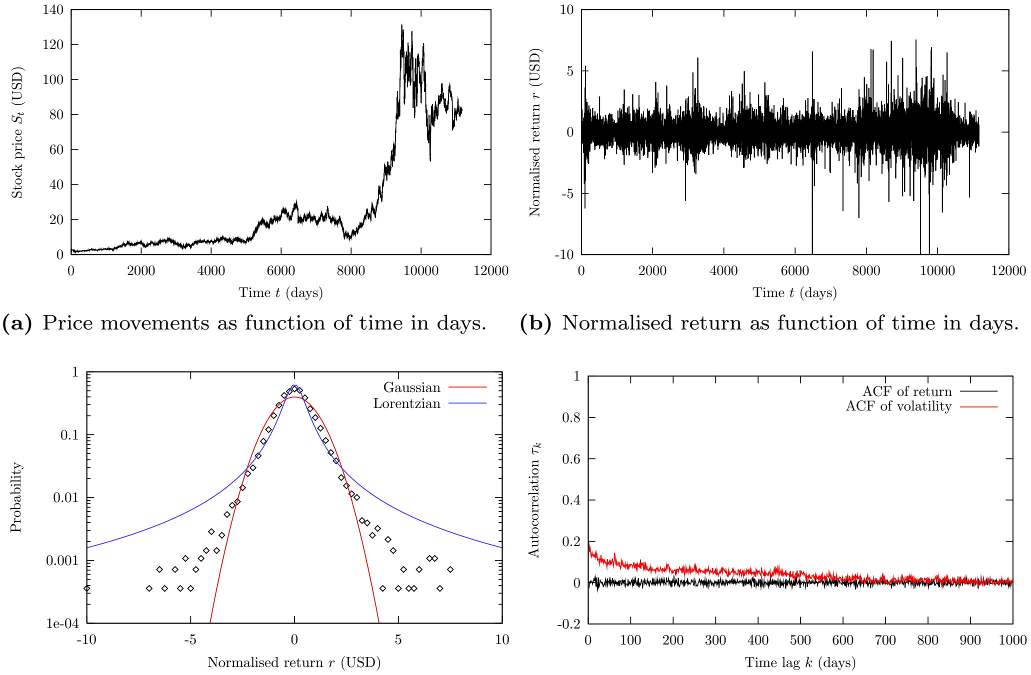


(c) Probability of normalised returns.



(d) Autocorrelation functions of the volatility and the normalised returns

Figure 2.2: Analysis of the AEX Index between October 1992 and May 2006. There is a fat tail, the kurtosis is $\kappa = 7.83$. The high autocorrelation of the volatility decays into the negative range, this means there is a lot of volatility clustering. Indeed, in (b) high positive and negative returns are grouped together.



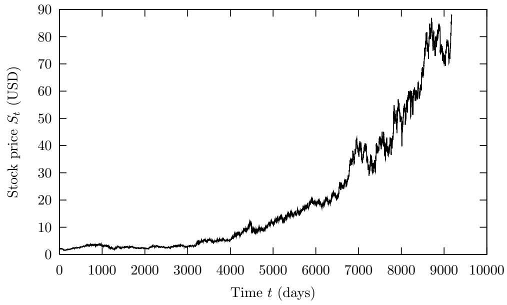
(a) Price movements as function of time in days.

(b) Normalised return as function of time in days.

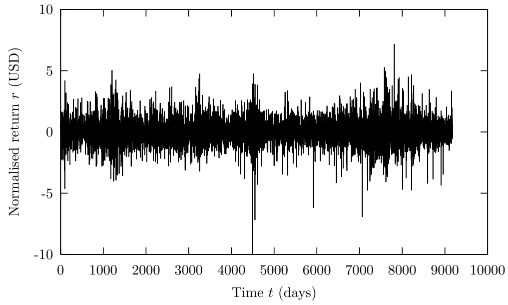
(c) Probability of normalised returns.

(d) Autocorrelation functions of the volatility and the normalised returns

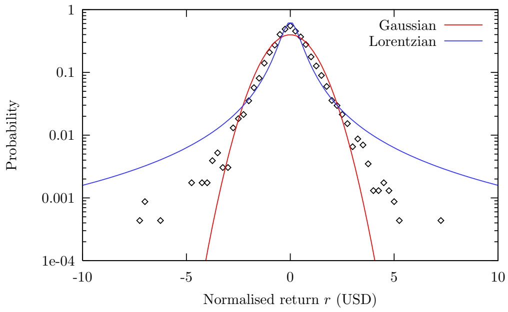
Figure 2.3: The stock price, return and autocorrelations of IBM, between January 1962 and May 2006. The slowly decaying of the ACF of the volatility indicates some volatility clustering. There is a fat tail, $\kappa = 16.14$. Also very noticable is the internet/dotcom hype, with its peak around year 2000.



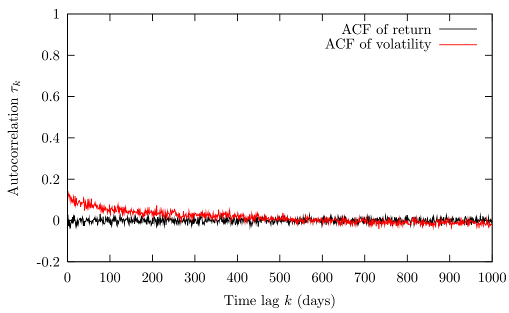
(a) Price movements as function of time in days.



(b) Normalised return as function of time in days.

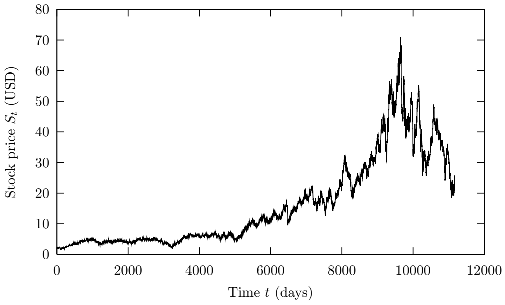


(c) Probability of normalised returns.

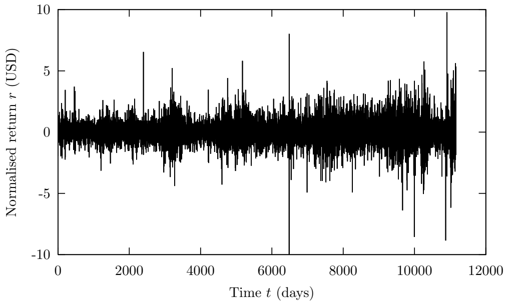


(d) Autocorrelation functions of the volatility and the normalised returns

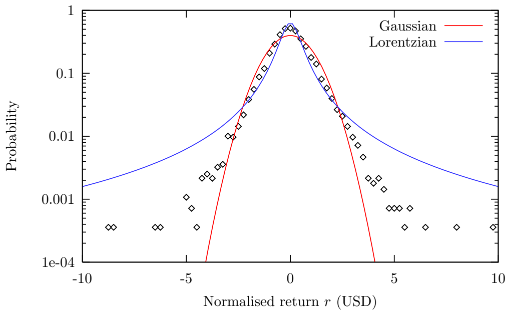
Figure 2.4: The stock price, return and autocorrelations of 3M Company, between January 1970 and May 2006. The slowly decaying of the ACF of the volatility indicated some volatility clustering. There is a fat tail, $\kappa = 12.34$. Compared to the stock of IBM in Figure 2.3, the effect of the Internet/IT hype is rather small.



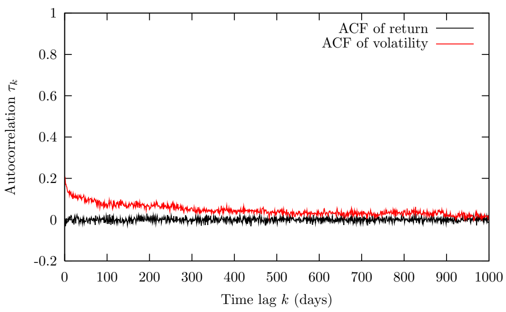
(a) Price movements as function of time in days.



(b) Normalised return as function of time in days.



(c) Probability of normalised returns.



(d) Autocorrelation functions of the volatility and the normalised returns

Figure 2.5: To compare the stock price, volatility, and return of an IT company with a non-IT company, the stock price of General Motors between January 1962 and May 2006 was analysed. The kurtosis $\kappa = 10.97$ shows there was less fat tail compared to IBM. The decaying ACF of volatility show volatility clustering.

CHAPTER 3

The Cellular Automaton Model

3.1 Overview

In this model, a financial market is represented by a two-dimensional lattice where each vertex stands for a trading agent. A trading agent can be a fundamentalist or an imitator. Fundamentalists are those agents who make the decisions based on the face value of the news they receive. Imitators are those agents who make decisions based on the decisions made by the neighboring fundamentalists, (i.e., they copy the behaviour of fundamentalists).

As a result of this, we get a long term auto-correlation of volatility, also called "volatility clustering". This can also be described by non-Gaussian or fat-tailed "stylized facts". A commonly used, although not rigorous, criterion for the normality of a distribution is its kurtosis as defined in Equation **??**. The model reflects the behaviour of microscopic components acting in a financial market. We represent a financial market as a two-dimensional $L \times L$ lattice, every vertex of which contains an agent. Within our model, agents of only two types are adopted: fundamentalists and imitators. Every agent has a radius of one (Moore) neighbourhood.

3.1.1 Level 1 Model

The Level 1 model assumes that the fundamentalists know the fundamental value of asset at any given time and buy (or sell) it in accordance with its fluctuation relative to their perceived fundamental value. This means the quantity traded by a fundamentalist at any time instance is determined by the difference real fundamental value and the known price of the asset in a previous time step. This can be expressed as

$$q_{fu}^{t+1} = e_{fu}(F - P^t) \tag{3.1}$$

where e_{fu} is a modulating constant, F is the real fundamental value, and P^t is the price at time t. The imitators in this model do not care about the fundamental price or are not very much aware of it. So, they just copy the behaviour of their neighbour. Imitators trading options are determined by their neighbour's trading options and represented as

$$q_{im}^{t+1} = \langle q_{nb}^t \rangle \tag{3.2}$$

3.1.2 Level 2 Model

In this case, we take into account the reaction of the news that comes to the market continuously. Fundamentalists perceive the value of an asset in light of the news and it usually fluctuates around the real fundamental value of the asset. This means that the perceived fundamental value at time $t+1$ is $F\eta_{fu}^{t+1}$; where η_{fu}^{t+1} is the effect of the news at time $t+1$. The value of η_{fu}^{t+1} is determined as

$$\eta_{fu}^{t} = 1 + c_{fu}\phi_{fu}^{t} \tag{3.3}$$

where ϕ is a normally distributed random variable and c_{fu} is a constant. Imitators also perceive and react to the news. So an imitator's behaviour is determined by the effect of news as well as the behaviour of its neighbours. This is represented as

$$q_{im}^{t+1} = \langle q_{nb}^{t} \rangle \eta_{im}^{t+1} \tag{3.4}$$

where η_{im}^{t} indicates the effect of the news at time t and is computed as

$$\eta_{im}^{t} = 1 + c_{im}\phi_{im}^{t} \tag{3.5}$$

3.1.3 Level 3 Model

In Level 3 of the CA model, the briskness of the trading is also taken into account. The underlying assumption is that the more brisk is the trading of an asset, more desirable that asset is. The briskness is measured as the average price fluctuation of the asset in a past interval of time. This effect is realised by defining a monotonically increasing function M^t which is represented, for simplicity, as $M^t = c_l L^t$, where L^t is a price fluctuation given by

$$L^t = \frac{1}{k} \sum_{i=t-1-k}^{t-1} \mid P^i - \overline{P} \mid \tag{3.6}$$

Considering this, the quantity of fundamentalists is updated using

$$q_{fu}^{t+1} = V_{fu}^{t+1} M^{t+1} = e_{fu}(F\eta_{im}^{t+1} - P^t) M^{t+1} \tag{3.7}$$

The quantity of imitators is updated using

$$q_{im}^{t+1} = \langle V_{nb}^{t} \rangle \eta_{im}^{t+1} M^{t+1} \tag{3.8}$$

Finally, the prices are updated according to the rule

$$P^{t+1} = P^t + \beta Q^t \tag{3.9}$$

where Q^t is the total transaction quantity (or excess demand) of the market at time t, β is a constant that indicates the sensitivity of the price to the excess demand. Because stock prices cannot be negative, the lower bound of P^t is 0.0.

3.2 Implementation details

We have implemented the CA model according to Qiu's paper. The pseudocode for the implementation is as follows:

```
1. Initialise System
 1.a. Initialise constants for fundamentalists
 1.b. Initialise constants for imitators
 1.c. Initialise the fundamentals:imitators ratio
 1.d. Initialise other constants like number-of-iterations etc.
 1.e. Initialise the matrix
2. For i=0 till number-of-iterations
 2.a. update the news_constant for fundamentalists
 2.b. update the news_constant for imitators
 2.c. Compute M
  2.c.i. Compute average price
  2.c.ii. Compute the sum of difference in price -- L
  2.c.iii. Compute average of L
  2.c.iv. Compute M=L*cl
  2.c.v. If m<0.1 then set it to 0.1
 2.d. Update Matrix
  2.d.i. Update fundamentalist nodes(according to level I, II or III)
  2.d.ii. Update imitator nodes(according to level I, II or III)
 2.e. Update pastprices with the new prices
```

3.3 Results

Figure 3.1 shows our results for Level 1 and Level 2.

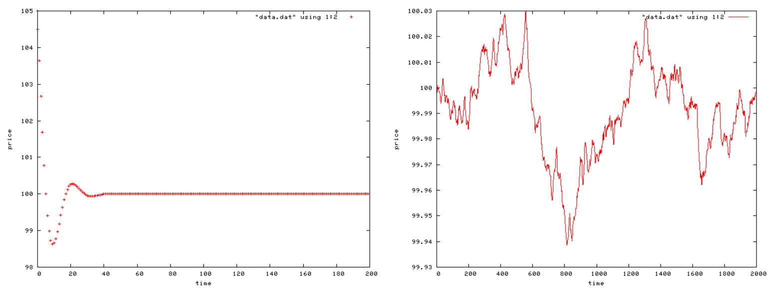


Figure 3.1: Result for Level 1 and Level 2

As Figure 3.1 shows, we get appropriate results for Level 1. As expected, the price fluctuates before getting steady at the value 100. This happens because the fundamentalists start with buying the asset that takes the price up and the same behaviour copied by imitators. Eventually fundamentalists sell the assets and the price go down until it becomes steady at its real fundamental value.

3.4 Tuning

Since the level two of the model did not work as intended, we started looking for errors in the code. This has been done very carefully, and by multiple people. It did not, however, result in any changes. The model is fairly simple and with clean code it is easy to check if the program is written as it should be.

When drawing random numbers many, many times, using certain random number generators can give errors. The one we used was the Mersenne-Twister as already mentioned, therefore this could not be the source of the failing model either.

The only option left was that the constants we used were not the right ones. As mentioned before, constants giving a fat tail for the level two of the model were stated in [2]. Because the price fluctuated only marginally (see Section **??**), we thought the problem was a too small β since this constant has a direct influence on the change in price according to the number of quantities traded. This however did not give any bigger tails: as β increased the fluctuations grew slightly, up to a point where the β was so big that the model became unstable resulting in prices going from zero to infinity. These results are not shown because it would take at least two graphs to properly show the result of one measurement, and for many runs this would mean flooding the report with pictures.

The only option left was checking the other variables. Since we were not sure which variable was the source of our problems we took four of them which would most likely give better results, and did a parameter sweep. The variables we have tested were

P_0	the starting price
η	the constant which influences the linear relation between present/fundamental price deviation and quantity traded
fraction	the relative number of imitators (rest being fundamentalists)
β	the constant for the influence of the traded quantity or excess, on price

The other constants remained unchanged. The range and stepsizes of the parameters are:

constant	minimum	maximum	stepsize	number of steps
P_0	80	120	20	3
η	0.01	0.16	factor 4	3
fraction	0	1	0.1	11
β	10^{-6}	1000	factor 10	10

The ranges were taken in order to scan around the numbers given in the paper, in a natural way. The fraction can for instance only be in the range specified so we took the whole range. The value of β was giving slightly better price fluctuations when it was increased a thousand fold and therefore we took a very large range.

Using these setting meant we had to run the simulation $3 \cdot 3 \cdot 11 \cdot 10 = 990 \approx 1000$ times. Running the simulation one time for 14000 timesteps (days if you like) on a modern desktop takes around three minutes. This means we would have to wait two whole days before we would have the data. Clearly this was not acceptable and therefore we used the Lisa cluster at SARA, an ICT service centre for High Performance Computing. This cluster has 1260 tightly coupled 3.4 GHz processors which can be used for job farming or parallel computing. The parameter sweep executed in only a few hours on this cluster. Afterwards the data was reformatted and plotted, again taking some hours, producing about 500 figures.

Scanning through these figures was done using Eye Of Gnome, manually checking each figure. The best two figures are shown in Figure **??**. The fat tails are clearly much weaker than they should be, compared to the results of the empirical study in Chapter 2. Looking for a relation between the appearance of the fat tails and the constants we found that the fraction of imitators must be very high (80 or 90%), β must be big (ranging from 10^{-1} to 10^{-5}), η should small

(0.04 or 0.01) and the start starting price must be 100. We thought the starting price would not change the result, however due to the slow change in price of this implementation a starting price of 80 or 120 would result in respectively a slow constant climbing or falling of the price to 100, generating wrong returns and therefore a wrong return distribution.

The bad fat tails may still be the result of too small fluctuations in the price of the stock since they remained in the order of cents. If we looked at the amounts of stock traded per agent this was not to be expected since they were quite large. Obviously all the selling and buying agents evened each otherout. This led us to the belief that either we used a too good random number generator, too many agents or a wrong type of news.

If the random number generator would have been flawed it would produce not truly random numbers which in turn would make the total amount traded by all agents not go to zero. Deliberately using a flawed random number generator is however unacceptable.

3.4.1 Reducing the number of agents

Using less agents increases the spread of the total amount traded since this spread converges as the square root of the number of agents (standard statistics). The same parameter sweep as before was done but now with an agent space of 50 by 50 agents, giving a total of 2500 agents instead of the normal 10000 agents.

3.4.2 Global news

In the paper of Qiu the *news* is presented as a personal impression of the fundamental value of the stock. This means that each fundamentalist has its own opinion on the value, completely independent of the beliefs of other agents. According to this opinion he trades some stock. As a result the total of stock traded by all fundamentalists fluctuates around zero. News does however not come on a per person basis, but more as something that is brought to groups of agents, in the form of newspapers, radio and TV broadcasts. This news is usually widely spread, reaching all traders, we also call this global news.

Since news in the real world comes mostly in a broadcast fashion, we changed the way news is implemented in the simulation to reflect this. The perceived fundamental value of the stock is now determined once per time step for *all* fundamentalists and once for all imitators. This means the total amount of tradings will be far from zero.

This new implementation was tested with the standard parameter sweep. The two best results are shown in Figure **??**. A decently fluctuating price can be seen, and as predicted the fat tails are now also more pronounced. These big fat tails are present in many parameter settings. It no longer matters whether the starting price is 80, 100 or 120 since now the price quickly goes to an average of 100. The range of the other constants that give good results, slightly less than the two cases shown, are $\beta \in 10^{-3}, 10^{-5}$, $\eta \in 0.04, 0.16$ and fraction $\approx 0.7 \pm 0.1$.

```
  switch on method
    case none, default: error, exit

    case plucked:
# loop over all points of the string in this process,
# excluding extra ''neighbour'' points
      for integer i = 1 , i < n + 1 , i++
        x = offset + i - 1
        if x = -1
          amplitude(point i) = 0
```

```
# check if x is left of the peak
        else if x < relative_peak_location * total_points
          amplitude(point i) = x / relative_peak_location * total_points
# x is right of peak
        else
          amplitude(point i) = 1 - (x - relative_peak_location * total_points) /
                    ( (1 - relative_peak_location) * total_points )
```

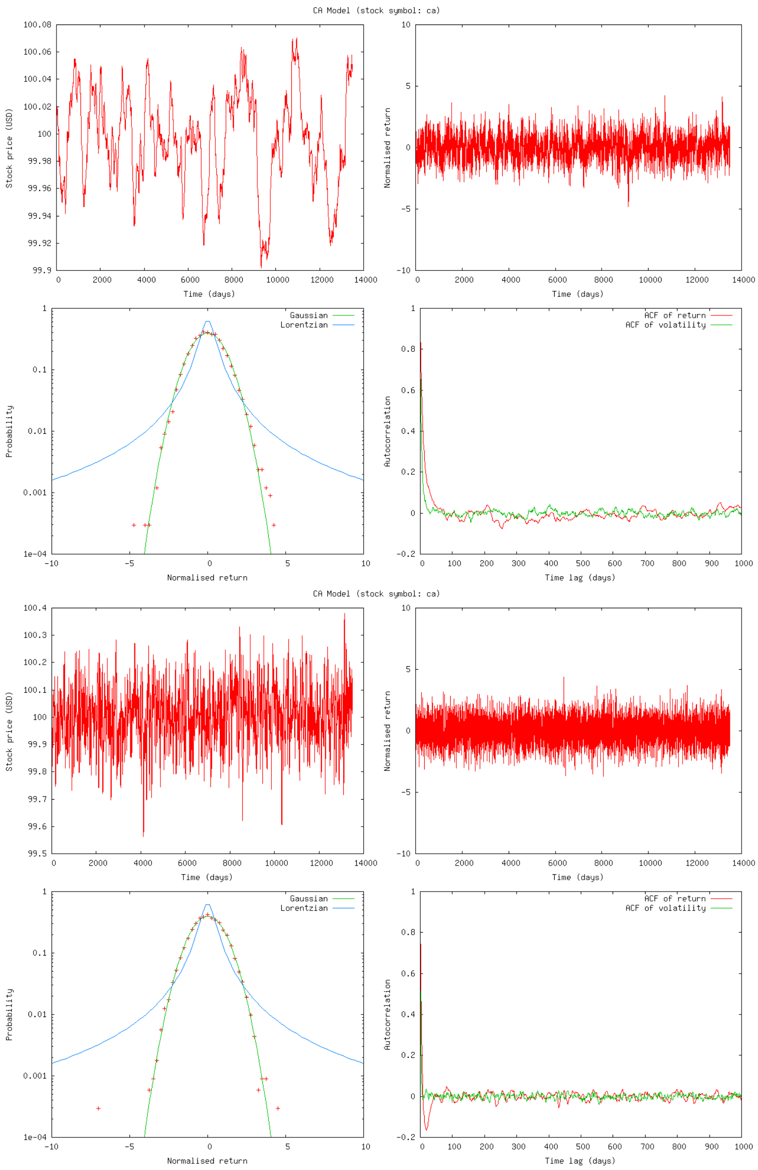


Figure 3.2: The two best fat tails from the parameter sweep of Section 3.4, on the left $P_0 = 100,\ \eta = 0.04,\ \text{fraction} = 0.9,\ \beta = 10^{-5}$, on the right $P_0 = 100,\ \eta = 0.01,\ \text{fraction} = 0.8,\ \beta = 10^{-3}$

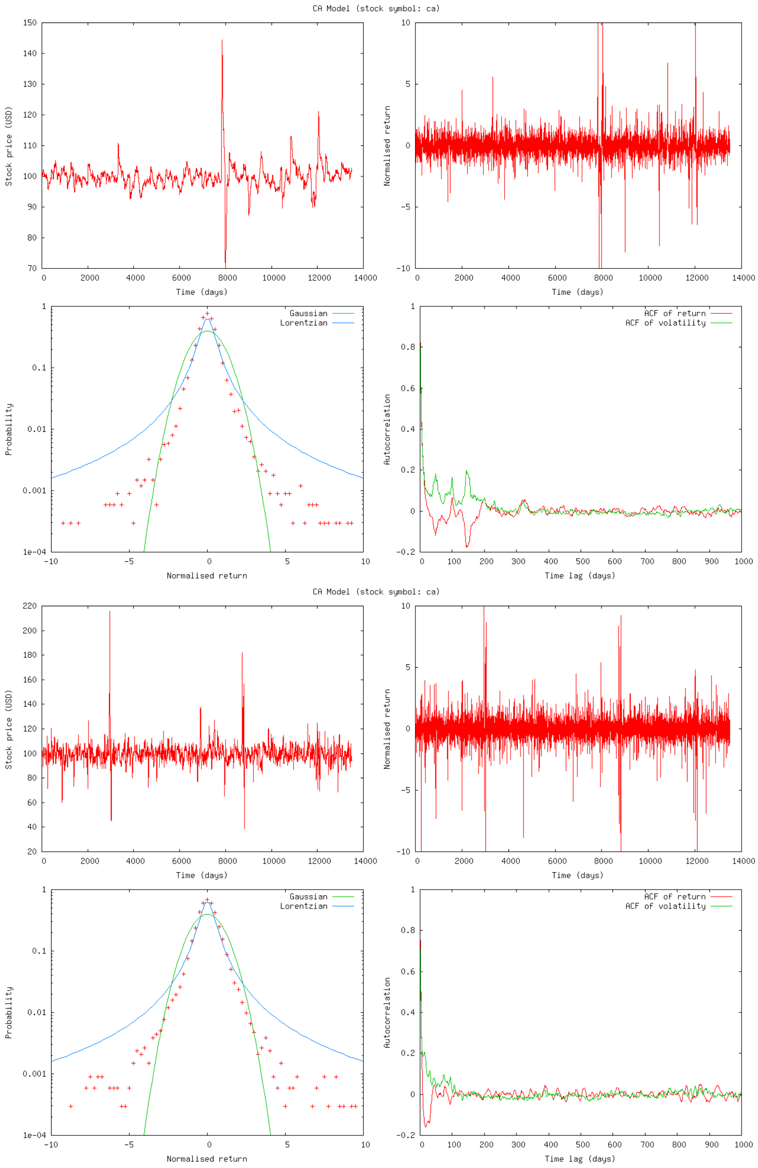


Figure 3.3: The two best fat tails from the parameter sweep of section 3.4.2, on the left $P_0 = 100,\ \eta = 0.16,\ \text{fraction} = 0.7,\ \beta = 10^{-5}$, on the right $P_0 = 100,\ \eta = 0.01,\ \text{fraction} = 0.7,\ \beta = 10^{-4}$

CHAPTER 4

The Cont-Bouchaud Model

4.1 Overview

The Cont-Bouchaud model (CB) is an agent based model, which provides a link between two well-known market phenomena: the heavy tails observed in the distribution of stock market returns on one hand, and herding behaviour in financial markets on the other hand. This model assumes agents can be grouped into clusters, wherein all the agents belonging to the same cluster, trade in the same way. We use the CB model to simulate a financial market and intend to reproduce the fat tails observed in the distribution of stockmarket returns. We also study the distribution of the size and number of the clusters formed in the CB model.

4.2 Theory

The model comprises of N agents and it is assumed that the agents organise themselves into groups by forming independent binary links between each other with probability c/N, where $0 < c < 1$ is the degree of clustering (connectivity parameter). All the agents which are connected by these binary links belong to the same cluster (an agent connected to another agent belonging to the a particular cluster, also belongs to the cluster). Thus, the market comprises of many such clusters. It is then assumed that all the agents belonging to a particular cluster trade in the same way, (i.e., all of them either buy at a particular instant of time or sell or choose not to trade). We model the decision process of agent i with the random variable ϕ_i. When $\phi_i = +1$ it indicates the agent decides to buy and $\phi_i = -1$ indicates the agent decides to sell. The marginal distribution of agent i's individual demand is assumed to be symmetric.

$$P(\phi_i = +1) = P(\phi_i = -1) = a \tag{4.1a}$$

and

$$P(\phi_i = 0) = 1 - 2a \tag{4.1b}$$

The aggregate excess demand has an impact on the price of the stock, causing it to rise if the excess demand is positive and to fall if it is negative. A common speculation, which is compatible with standard tâtonnement (groping) ideas, is to assume a proportionality between price change (or return) and excess demand:

$$\delta x = x(t+1) - x(t) = \frac{1}{\lambda}\sum_{i=1}^{N}\phi_i(t) \tag{4.2a}$$

where λ is the market depth: it is the excess demand needed to move the price by one unit. In the model since all the agents within a cluster/group trade in the same way.

$$\delta x = x(t+1) - x(t) = \frac{1}{\lambda}\sum_{i=1}^{n_c} W_i\phi_i(t) \tag{4.2b}$$

where n_c is the number of clusters and W_i is the number of agents in cluster i.

The structure that we get is a *random graph*. Hence, using results on random graphs we can show that the distribution of the cluster size is given by

$$P(W) =_{W\to\infty} \frac{A}{W^{5/2}}\exp[\frac{(1-c)W}{W_0}] \tag{4.3}$$

and the average number of clusters is given by $N(1-c/2)$.

4.3 Implementation details

The simulation of the CB model would require us to simulate the links formed between any two agents in the market (a link being formed between agents with the probability c/N). This would require at least $N(N-1)/2$ iterations. Since the results of the CB model depend just on the number of agents in a particular cluster, we can speed up the process by a constant factor, by simulating links formed between agents and clusters, rather than between agents. The average number of clusters is given by $M = N(1-c/2)$, therefore $M \leq N$. As the number of clusters M, is always less than or equal to the number of agents, the simulation process becomes faster by a constant factor.

The CB model has been implemented in C, the code can be found in Appendix **??**.

4.4 Drawbacks of the model

B model Cont-Bouchaud model. This model is able to explain heavy tailed probability distributions through a herding or group behaviour mechanism.

The drawbacks of the model are:

1. It cannot explain the phenomenon of volatility clustering
2. The model does not explain the time evolution of clusters.

4.5 Modifications

4.5.1 Changing demand

As we have seen the CB model has the drawback that it cant produce volatility clustering. Volatility clustering is the property where large changes in prices tend to occur together and small changes in prices occur together. In the CB model we see that the price change is proportional to the demand.

$$\delta x = x(t+1) - x(t) = \frac{1}{\lambda} \sum_{i=1}^{N} \phi_i(t)$$

So we can see that the clustering of large price changes is due to the occurrence of high demands and low price changes due to the occurrence of low demands. In the CB model the demand remains the same always because the probability that an agent trades remains constant. i.e.,

$$P(\phi_i = +1) = P(\phi_i = -1) = a$$

In our extended model we don't keep the probability that an agent trades constant, but varies depending on which state the market is in. When the system moves to state S_1, the probability that an agent buys increases and the probability that an agent sells decreases. When it moves to state S_3, the probability that an agent buys decreases and the probability that an agent sells increases. When it moves to state S_2, the probability that an agent sells or buys remains the same. See Figure XX The selling and buying probabilities of an agent are defined as:

$$P(\phi_i = +1) = a_t$$

and

$$P(\phi_i = -1) = b_t$$

When the system moves from any state S_i to S_1 at time $t+1$, the new probabilities are

$$a_{t+1} = a_t + \alpha$$

and

$$b_{t+1} = b_t - \alpha$$

where $0 < \alpha << a_0$

When the system moves from any state S_i to S_3 at time $t+1$, the new probabilities are

$$a_{t+1} = a_t - \alpha$$

and

$$b_{t+1} = b_t + \alpha$$

When the system moves from any state S_i to S_2 at time $t+1$, the new probabilities are

$$a_{t+1} = a_t$$

and

$$b_{t+1} = b_t$$

Let $P_{i,j}$ indicate the probability that the market moves from state i to state j. The probabilities are set such that

$$(\forall i, j) P_{i,2} > P_{i,j}$$

where

$$j \neq 2$$

Hence, whatever state the system was in at time t, it tends to remain in that state at time $t+1$. For example, if there was an increase in buying in the market, it tends to remain in that state the next time step too. Hence we can expect price changes to remain in the next time step too and hence volatility clustering too.

We can also have the system to move from one state to another based on the information that arrives at that time step.

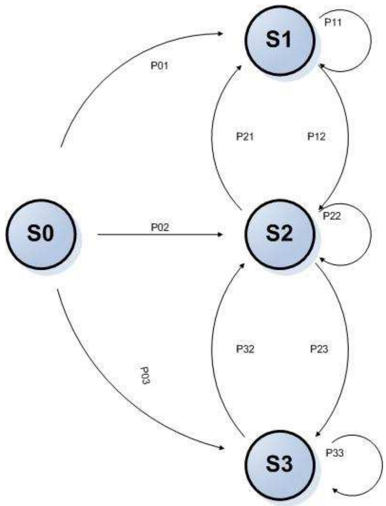


Figure 4.1: The value S_0 is the initial state; Moving to S_1 increases Buying; Moving to S_2 causes no change in buying or selling ; Moving to S_1 increases Buying; P_{ij} is the probability of going from state i to state j.

4.5.2 Dynamic clusters

In the CB model the clusters remain static, which is not what is expected of the real world. Therefore we suggest an extension where the agents reorganise themselves into different clusters with time. In the CB model all the agents belonging to a particular cluster trade in the same way, (i.e., all of them decide to buy or sell or not trade at a particular time instant). The agent is in the market to get good returns and if he/she does not profit by following the actions of the agents in the cluster to which he/she belongs to, then he/she is likely to abandon that cluster and move to a different cluster. The extended model depicts this phenomenon. We also hope that this extended model will give rise to volatility clustering.

For every agent i, we associate a parameter $b_{i,t}$ with it at time step t. This gives an indication how well the agent has been trading. The parameter is calculated as follows:

$$b_{i,t+1} = \begin{cases} 2 + \alpha \cdot b_{i,t+1} & \text{if agent i makes a profit at timestep } t \\ 1 + \alpha \cdot b_{i,t+1} & \text{if agent i doesn't trade at timestep } t \\ \alpha \cdot b_{i,t+1} & \text{if agent i makes a loss at timestep } t \end{cases}$$

where $0 \leq \alpha \leq 1$.

We assume that at every time step n_{jump} agents, where $n_{jump} << n_{order}$, jump from one cluster to another. We choose n_{jump} agents probabilistically from the total set of N agents. These agents form new connections with other agents. The probability that agent i connects to agent j at time step t is given by

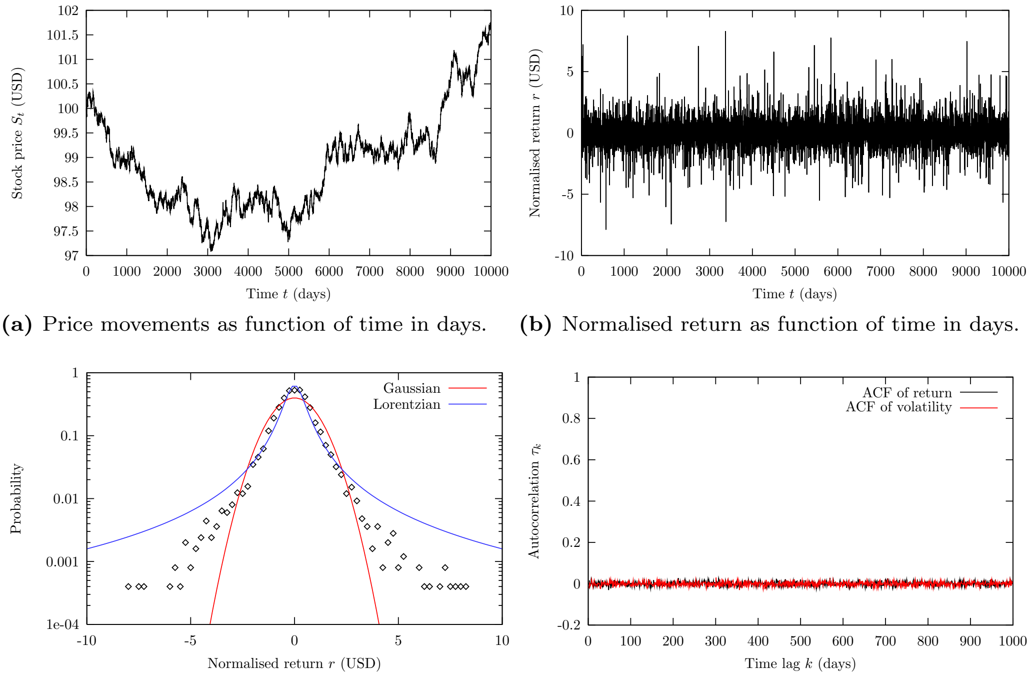
$$= \frac{c \cdot b_{j,t}}{\sum_{k=1}^{N} b_{k,t}}$$

Thus it is more probable that an agent who is leaving it's current cluster will move to a cluster which has been trading well. Here c is the order of clustering, so that the expected value of the probability remains c/N.

Things to think about :

1. Will it still remain a random graph ?
2. how do we choose the parameters α and n_{jump}

4.6 Results



(a) Price movements as function of time in days.

(b) Normalised return as function of time in days.

(c) Probability of normalised returns.

(d) Autocorrelation functions of the volatility and the normalised returns

Figure 4.2: Stylized facts after CB simulation, for $c = 0.9$ (degree of clustering), $a = 0.0005$ (probability to buy or sell), $n_{order} = 1000$ (average order flow), $\lambda = 1000$ (markt depth or sensitivity of price to fluctuations in excess demand), 100000 agents and 10000 timesteps. The calculated kurtosis is: $\kappa = 10.1$.

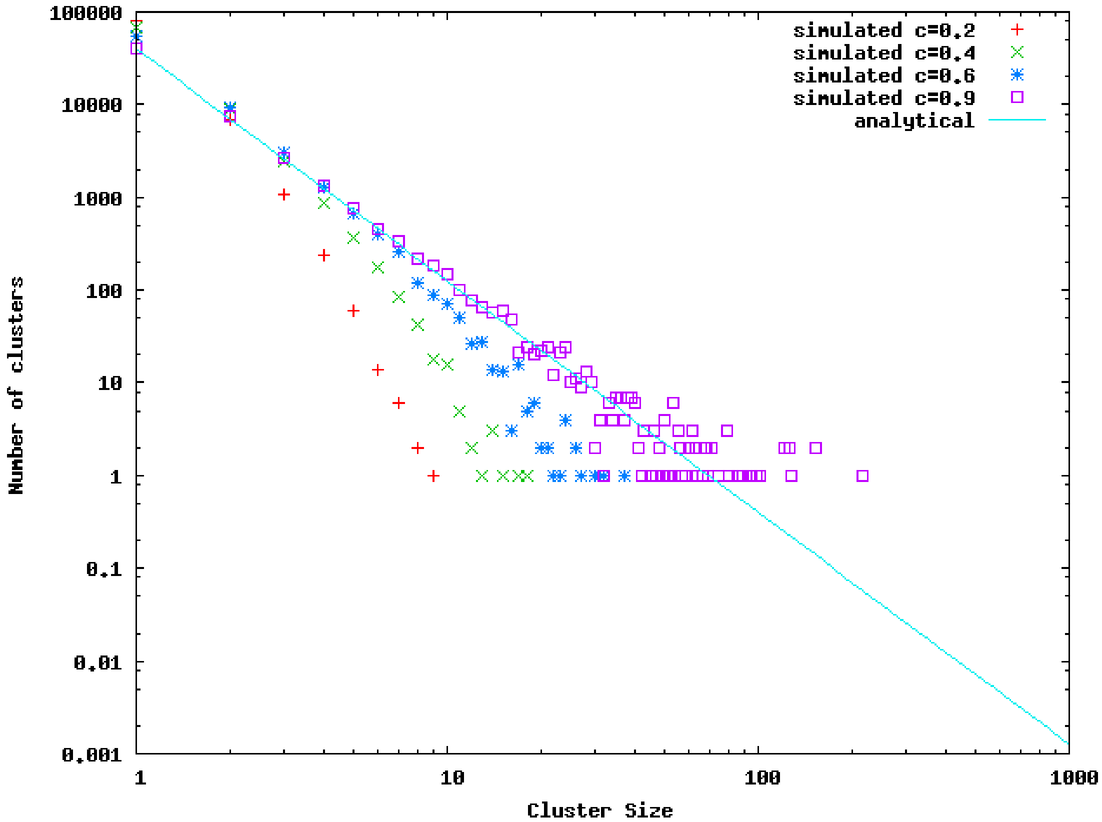


Figure 4.3: Number of clusters as a function of clustersize, for $c \in \{0.2, 0.4, 0.6, 0.9\}$. The number of agents is 100000. Comparison with the analytical function: $p(W) = W^{-5/2}exp^{(1-c)W/W_0}$, where W is the clustersize and W_0 is equal to the number of agents. Both axes are logarithmic.

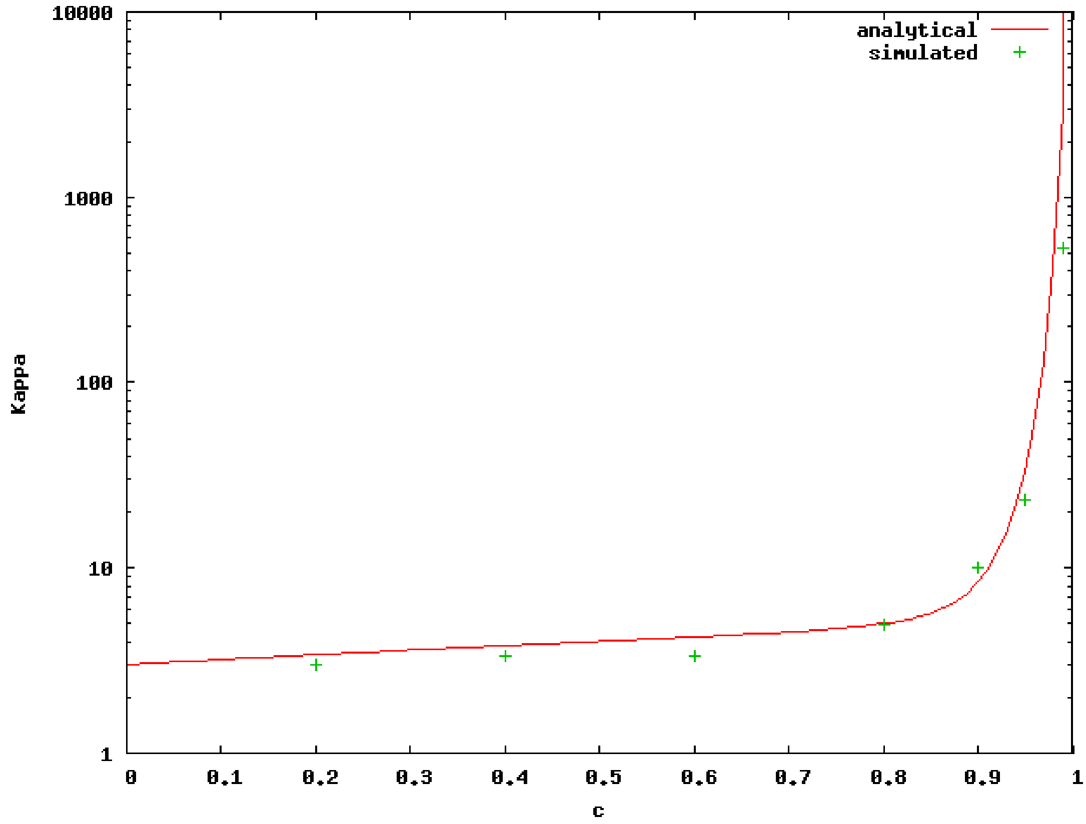


Figure 4.4: Simulated kurtosis after CB simulation, for different values of c (degree of clustering), with $a = 0.0005$ (probability to buy or sell), $n_{order} = 1000$ (average order flow) $\lambda = 1000$ (markt depth or sensitivity of price to fluctuations in excess demand), 100000 agents and 10000 timesteps. Comparison with analytical kurtosis function: $\kappa(c) = 2c + 1/(n_{order} \cdot (1 - c/2) \cdot A(c) \cdot (1 - c)^3$, where $A(c)$ is here approximated by 1/2. The κ-axis is in logarithmic scale.

CHAPTER 5

Comparisons

CHAPTER 6

Conclusions

Bibliography

[1] Rama Cont and Jean-Philipe Bouchaud. Herd behavior and aggregate fluctuations in financial markets. *Macroeconomic Dynamics*, 4(2):170–196, June 2000.

[2] G. Qiu, B.D. Kandhai, and P.M.A. Sloot. Understanding the complex dynamics of financial markets through cellular automata. Working paper of Universiteit van Amsterdam, 2006.