

Development and testing of an artificial stock market

Michele Marchesi*

Dipartimento di Ingegneria Elettrica ed Elettronica, Università di Cagliari
Piazza d'Armi, 09123 Cagliari, Italy.

Silvano Cincotti

Dipartimento di Ingegneria Biofisica ed Elettronica, Università di Genova
Via Opera Pia 11a, 16145 Genova, Italy.

Sergio Focardi

Intertek Group

rue De Javel 94, 75015 Paris, France

Marco Raberto

Dipartimento di Ingegneria Biofisica ed Elettronica, Università di Genova
Via Opera Pia 11a, 16145 Genova, Italy.

September 15, 2000

Abstract

In this paper, an artificial financial market based on heterogeneous agents is presented. The proposed market is composed of traders with limited amount of cash, one traded asset and a centralized mechanism, the market maker, matching buy and sell orders. The price formation process is given by the intersection of the demand and the supply curve.

The artificial financial market has been implemented using advanced software engineering techniques, in particular extreme programming and object oriented technology. The resulting system is a

*To whom correspondence should be addressed. Email: michele@diee.unica.it

powerful tool able to numerically simulate financial market operations in a realistic way. Preliminary results show that the price time series exhibits a random walk behavior with fat tails distribution of returns.

1 Introduction

In the last decade, many computer simulated artificial financial market have been proposed [1, 2, 3, 4, 5]. For a review see LeBaron [6]. Perhaps, the most famous among them is the artificial market - populated with heterogeneous agents endowed with learning and optimisation capabilities - built at Santa Fe Institute by Arthur, Holland et al. [7, 8, 9].

In this paper, we present an agent-based computer simulator of a stock market, called the Genoa artificial market, in honour of to the beautiful city where most of this work was performed. In the Middle Age Genoa was a major financial center, where the I.o.u. and the derivatives were invented [10].

The main features of the presented simulator are:

1. it has been developed using state-of-the-art programming techniques allowing an easy upgrade and modification of the system;
2. portfolio and cash of every simulated trader, as well as every order and transaction, are tracked;
3. it is endowed with a realistic price formation mechanism.

The first release of the Genoa artificial financial market is characterized by heterogeneous agents exhibiting random behavior, one traded asset and a fairly complex interaction mechanism. Agents place buy and sell orders randomly, compatibly with their cash availability and asset portfolio. Orders are then processed by a module, the market maker, which builds the well known demand and supply curves, whose intersection point determines the new asset price. Orders, whose limit prices are compatible with the value of the new price, are satisfied, while others are discarded. The traders cash and portfolio is updated and a new simulation step begins. The Genoa artificial market is stable, i.e., prices do not diverge, being constrained by the fixed amount of cash of traders. In addition, with a mechanism representing the aggregation of traders, the artificial market is able to reproduce one important feature of real markets: fat tails.

2 The Genoa market microstructure

In the Genoa artificial market, we modelled a stock market from a very realistic perspective. Each trader is modeled as an autonomous agent, and its data structure keeps track of trading operations as it could do in a real trading system. The traders are agents who are endowed with a given amount of cash and of stocks, and can issue orders to buy more stocks or to sell (a portion of) their stocks. The traders are individually tracked by the system, but do not develop individual trading strategies. They issue random orders, but their overall behavior depends on their actual state, as the quantity of stocks of the issued orders depends on their cash and stocks owned.

The price computation proceeds in unit time steps of one day. For the sake of simplicity, only one stock is traded in the market.

The system has three main degrees of freedom, i.e. three state variables: the amount of cash in the system, the number of stocks in the system and the price of the stock. Market capitalization is given by the number of stocks times their price. In addition, initial conditions and the parameters of traders need to be fixed.

The central market maker fixes a price by matching offer and demand and stands ready to satisfy all orders at these conditions. The price formation process is given by the intersection of the demand and the supply curve. Fig. 1 shows the shape of the demand and the supply curve and the resulting clearing price in a case drawn from a simulation.

Once the price is cleared, the orders compatible with that price (buy orders with maximum price lower than or equal to p^* , and selling orders with minimum price higher than or equal to p^*) are executed. Consequently, the cash and stocks amounts of traders involved in executed orders is updated. It is worth to note that the size of buy orders whose maximum price is lower than or equal to p^* may differ from the size of selling orders whose minimum price is higher or equal to p^* . If the size of compatible sell orders is larger than the size of compatible buy orders, as shown in Fig 2a, the market maker adds cash to the system of traders and subtract assets from it. Conversely, if the size of compatible sell orders is smaller than the size of compatible buy orders, as shown in Fig. 2b, the market maker adds assets to the system of traders and subtracts cash from it. In this framework, we assume an ideal market maker with an unlimited availability of cash and stocks satisfying all orders compatible with the new price p^* fixed by the intersection of the demand and the supply curve.

The orders that do not match the clearing price are discarded.

At the beginning of the simulation, the current price $p(0)$ is set in an exogenous way, and each trader is given an amount of cash and an amount of stocks. These amounts can be the same for all traders, or may differ.

The interplay of demand and supply schedule and the eventual imbalance of orders creates a rich behavior as at each step a price increase or decrease might be accompanied by either a cash increase or decrease. As a consequence the system might either evolve towards a situation of higher prices and higher cash or the lack of cash will tend to restrain the price increase. The same reasoning applies to price decrease.

The traders described so far issue buy and sell orders with the same probability and are totally independent from each other. Consequently, it should not be surprising to discover that the ensuing log-price behavior is a random walk, and that daily returns follow a normal distribution.

It is well known, however, that real markets return distributions exhibit “fat tails” following a Pareto-Levi distribution, at least for returns computed for times approximately less than one month [12, 13]. In order to make our market model more realistic, we added an aggregation mechanism based on random graphs, following recent theories [14, 15] to model the aggregate behavior of traders in real markets.

In our model, each trader is marked with a tendency to be optimist or pessimist. At the beginning of the simulation, 50% of traders are marked as optimist and 50% as pessimist. This tendency does not immediately affect the trader behavior, which remains balanced. At each time step, however, random links are added among traders sharing the same tendency, with a probability p_a . In this way, clusters of traders sharing the same opinion gradually take shape. If a link is established between two traders belonging to different clusters, these clusters merge into a bigger one.

Moreover, at each simulation step clusters of both optimist and pessimist traders are randomly chosen with probability p_c , that does not depend on the cluster size. All traders belonging to a chosen cluster receive a message to buy (if they are optimist) or to sell (if they are pessimist) as far as they can. Optimists immediately place an order to buy covering 90% of their cash, while pessimists place a selling order of 90% of their stocks. The percentage of 90% of cash was chosen because in this way if the price climbs in consequence of buy orders, it is unlikely that the trader cash becomes negative. As for sellers, the 90% percentage was chosen to keep balanced the buy and sell order flow.

After these aggregate orders are placed, all links of traders belonging to the chosen clusters are broken, and these traders change their tendency. As optimists have bought (almost) all the stocks they could, their tendency switches to pessimist, not having anymore cash for buying, but only stocks to sell. Vice-versa for pessimists.

This aggregate behavior reflects in a simplified way mechanisms of opinion formation actually in place in real markets. With this mechanism, the price behavior remains balanced and apparently does not divert from the usual random walk. One day returns, however, change their distribution, which now becomes much closer to return distribution of real data.

3 Market simulation software

The model described in the previous section was implemented in software, using advanced software engineering techniques. In this section we briefly describe implementation techniques and the architecture of the resulting system.

The simulator was implemented using object-oriented technology and Extreme Programming [16] as development process. This process is incremental and iterative. It builds a software system step by step, adding new features at each step. These features come from requirements gathering in the form of *user stories*, i.e., short sentences describing the behavior of the system and/or its interactions with the users. User stories are collected, sorted by importance and risk and then implemented. A functional system, implementing a subset of the required features, is released as soon as possible (in our case, in a few weeks) and new releases are made every week. Testing and refactoring [17] are of paramount importance in the process.

We used a revised version of XP, described in [18]. It proved effective and enabled us to successfully deliver the system in a few months. The system is divided in the following subsystems:

- Assets, whose objects hold data and implement operations pertinent to various kind of assets.
- Trading, whose objects keep track of trading orders, of actual buy and sell transactions and of single traders.
- Clusters, containing objects representing the random graphs and managing traders aggregation in clusters.

- Simulation, containing high-level objects managing the simulation, like the market maker, the stock market and the simulator itself.

An UML class diagram [19] of the “Trading” subsystem is shown in Fig. 3, showing the classes of the various objects involved, their data structure and their relationships.

Fig. 4 shows the architecture of the whole system. The user sends commands to the simulator in textual form. The simulator object manages the whole simulation, first creating the traders and the stock, and then performing the simulation steps.

At each step the cluster management module generates new links at random and chooses at random zero or more clusters, marking accordingly the traders belonging to them and breaking all their links. Then, each trader is ordered to perform trading, which happens individually and depends on the state of the trader. Each trader generates zero or more buy and/or sell orders, involving her cash and stocks, respectively. The orders are sent to the market maker, which computes the new price and satisfies compatible orders by generating buy and sell transactions.

When developing the system, we developed also an automatic test suite for all its objects. This suite is continuously increased as new objects and new features are added to the system, and is run after every modification to ensure that the modification did not break system integrity [16].

Let us note that the whole system was conceived not as a stand-alone application optimized for the presented model, but as an evolving system, able to be continuously modified and updated. For instance, the system keeps track of every trading operation. It can manage a practically unlimited number of different kinds of securities, and could be used also as an engine for a trading game, or for implementing real online trading. As another example, let us consider adding more intelligence to traders, like the capability to learn and to take decisions as in the seminal stock market model by Arthur et al. [8, 9]. This modification would amount simply to add the objects supporting this kind of computation, like Predictor, Condition, Forecasting, GeneticAlgorithm, and to modify the “Trader” object. The remaining parts of the system would remain unaffected, and trading could start immediately with the new “intelligent” traders. By the way, this upgrading of the system is planned as one of the next activities to perform.

The system is written in Smalltalk language, that proved very suited for

this kind of application.¹ Using Smalltalk, in fact, it is possible to develop complex systems and to make substantial modifications to them very quickly, not jeopardizing quality. As regards performance, though Smalltalk is an interpreted language, the simulation speed is enough for our purposes and we did not feel any need to trade Smalltalk flexibility for further speed. A simulation of 100 traders for 1000 time steps can be performed in about 4 minutes on a Pentium III 600MHz computer. Let us recall that the simulation is quite complicated, and takes into account every single trading operation. The resulting system is composed of 18 classes and 220 methods (operations). The test suite is composed of 11 classes and 112 methods.

4 Results and conclusions

Several simulations have been performed using the system. We briefly present some results of a simulation of 10,000 steps with 500 agents. In this configuration, agents are endowed with a fixed supply of cash and there are no addition or subtraction except those due to trading. Moreover, both cash and asset portfolio capitalization are initially set at the same value: the initial price of the asset is 50 \$ and every trader has a portfolio of 1,000 assets and a cash availability of 50,000 \$. The value of probabilities p_a and p_c , as defined in previous section, are respectively: $p_a = 0.09$ and $p_c = 0.05$.

A typical price path is shown in Fig. 5. The estimate of the probability density function of logarithmic returns is presented in Fig. 6, which clearly shows the presence of fat tails. Fig. 7 shows the scaling behavior of the variance of logarithmic returns versus the time horizon. The variance scales roughly linearly with exponent ~ 0.94 up to about 13 days, whereas above this time horizon there is clearly a change of scale and the variance grows more slowly with time with an exponent equal to ~ 0.59 . This fact is probably due to constraints of cash and asset availability in the case of very large orders.

The Genoa artificial stock market is being employed to investigate the asset price dynamics from a microscopic perspective and the relationship between money supply and stock price processes.

The whole system is a work in progress, as we are actively working on the following issues:

¹The first version of the Artificial Stock Market of Santa Fe Institute was written in Objective C, which is C with embedded Smalltalk instructions.

- exploring the influence of a market maker with limited resources;
- adding more intelligence to traders;
- performing simulations with cash outflows and inflows;
- performing simulation with several stocks and bonds being traded.

References

- [1] D. K. Gode and S. Sunder, “Allocative efficiency of markets with zero intelligence traders: Market as a partial substitute for individual rationality,” *J. Political Economy*, vol. 101, pp. 119–137, 1993.
- [2] L. Marengo and H. Tordjman, “Speculation, heterogeneity and learning: A model of exchange rate dynamics,” Tech. Rep. WP-95-17, International Institute for Applied Systems Analysis, Vienna, Austria, 1995.
- [3] J. Arifovic, “The behavior of the exchange rate in the genetic algorithm and experimental economies,” *J. Political Economy*, vol. 104, pp. 510–541, 1996.
- [4] K. Steiglitz, M. L. Honig, and L. M. Cohen, “A computational market model based on individual action,” in *Market-Based Control: A Paradigm for Distributed Resource Allocation*, S. H. Clearwater, Ed., New Jersey, 1996, World Scientific.
- [5] M. Lettau, “Exploring the facts with adaptive agents: The case of mutual funds flow,” *J. Economic Dynamics and Control*, vol. 21, pp. 1117–1147, 1997.
- [6] B. LeBaron, “Agent-based computational finance: Suggested readings and early research,” *J. Economics Dynamics and Control*, vol. 24, no. 5-7, pp. 679–702, June 2000.
- [7] R. G. Palmer, W. B. Arthur, J. H. Holland, B. D. LeBaron, and P. Tayler, “Artificial economic life: a simple model of a stock market,” *Physica D*, vol. 75, pp. 264–274, 1994.
- [8] W. B. Arthur, J. H. Holland, B. D. LeBaron, R. G. Palmer, and P. Tayler, “Asset pricing under endogeneous expectations in an artificial stock market,” in *The Economy as an Evolving Complex System II*, W. B. Arthur, S. N. Durlauf, and D. A. Lane, Eds. 1997, SFI Studies in the Sciences of Complexity, Addison Wesley Longman.

- [9] B. D. LeBaron, B. W. Arthur, and R. Palmer, “Time series properties of an artificial stock market,” *J. Economic Dynamics and Control*, vol. 23, no. 9-10, pp. 1487–1516, 1999.
- [10] E. Briys and F. de Varenne, “1000 years of risk management,” *Risk*, pp. 47–48, January 2000.
- [11] M. O’Hara, *Market Microstructure Theory*, Blackwell, Oxford, UK, 1995.
- [12] R. N. Mantegna and H. E. Stanley, *An Introduction to Econophysics. Correlations and Complexity in Finance*, Cambridge University Press, Cambridge, UK, 2000.
- [13] J.-P. Bouchaud and M. Potters, *Theory of Financial Risks: From Statistical Physics to Risk Management*, Cambridge University Press, Cambridge, UK, 2000.
- [14] R. Cont and J.-P. Bouchaud, “Herd behavior and aggregate fluctuations in financial markets,” *Macroeconomic Dynamics*, vol. 4, no. 2, pp. 170–196, June 2000.
- [15] S. Focardi and M. Marchesi, “Self-organization in opinion aggregation models,” in *Proceedings of Forecasting Financial Markets Conference, London, 26-28 May, 1999*.
- [16] K. Beck, *Extreme Programming Explained: Embrace Change*, Addison Wesley Longman, Reading, Massachusetts, 1999.
- [17] M. Fowler, K. Beck, J. Brant, W. Opdyke, and D. Roberts, *Refactoring: Improving the Design of Existing Code*, Addison Wesley Longman, Reading, Massachusetts, 1999.
- [18] G. Armano and M. Marchesi, “A rapid development process with UML,” *ACM Applied Computing Review*, 2000, in press.
- [19] G. Booch, I. Jacobson, J. Rumbaugh, and J. Rumbaugh, *The Unified Modeling Language User Guide*, Addison Wesley, Reading, Massachusetts, 1998.

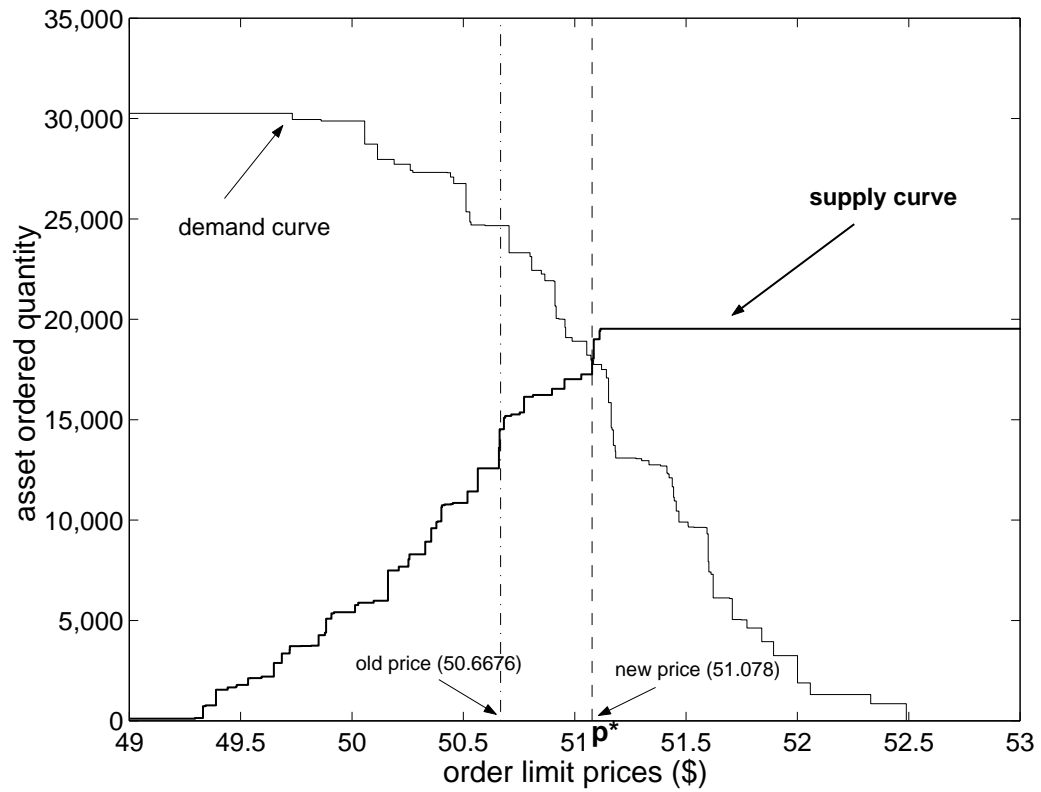


Figure 1: The new price p^* is determined by the x coordinate of the intersection point between the demand and the supply curve. The figure presented is derived from a simulation. The unbalance towards buy orders causes an increase of price.

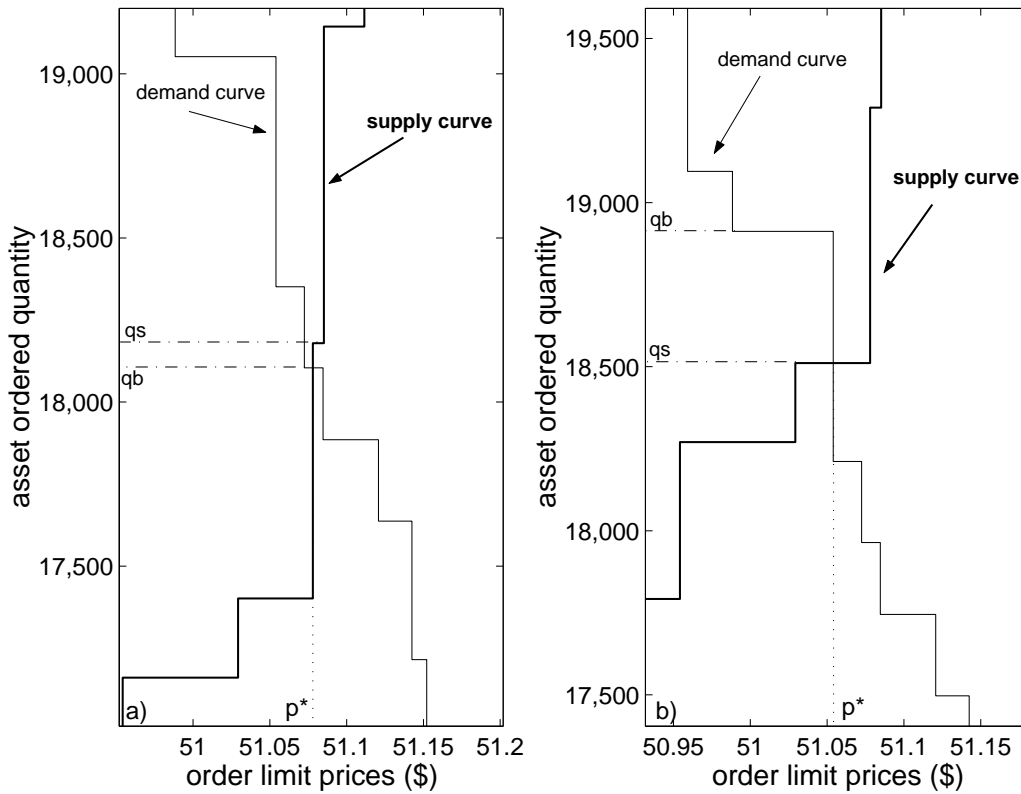


Figure 2: a) Particular of Fig. 1. The size of compatible sell orders q_s is larger than the size of compatible buy orders q_b . b) In this fictitious case, the size of compatible sell orders q_s is smaller than the size of compatible buy orders q_b .

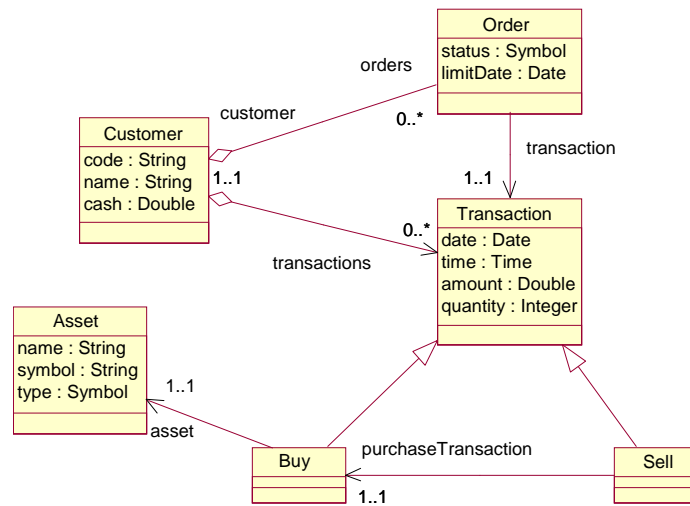


Figure 3: UML class diagram of “Trading” subsystem, whose classes represents the orders and transactions taking place in the system.

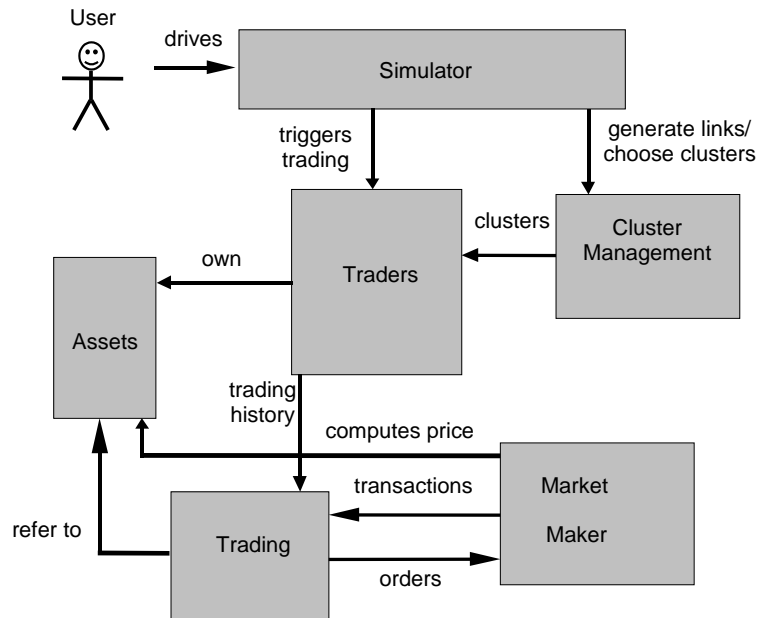


Figure 4: The overall architecture of the Genoa artificial financial market simulator.

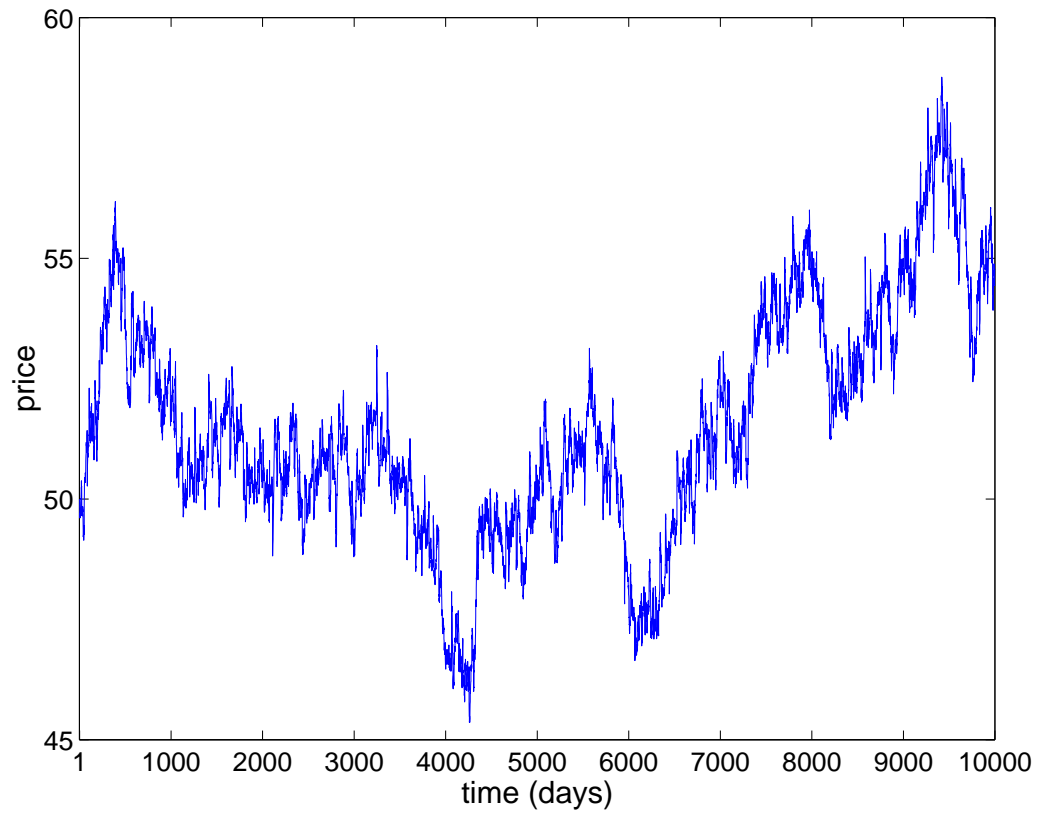


Figure 5: A typical price path.

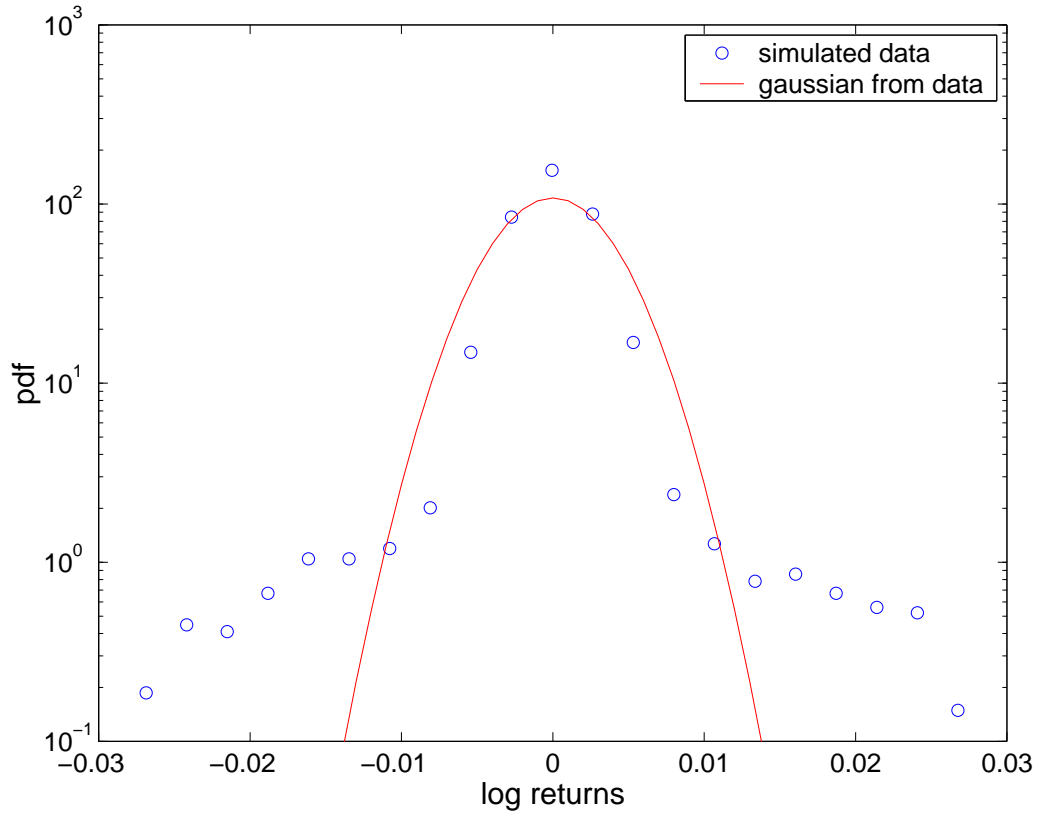


Figure 6: Estimate of the probability density function for the logarithmic returns of the simulated prices. Circles represents the histogram of simulated data, the continuous line is a Gaussian distribution whose mean and standard deviation are calculated using logarithmic returns of simulated data.

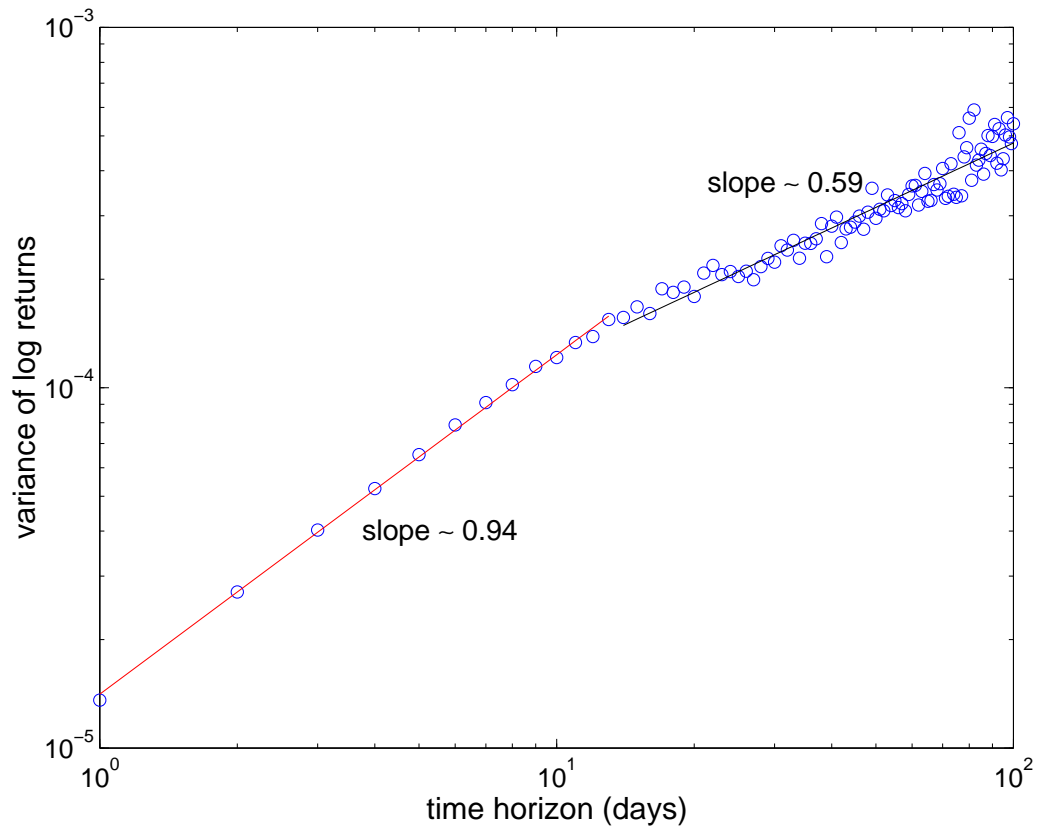


Figure 7: Scaling of the variance of logarithmic returns versus time horizon.