

Simulating Financial Markets using MASON Framework^{*}

Robert Axtell^a and CSS 739 Class Project Team^{**}

^aCenter for Social Complexity, George Mason University, USA
`{rax222}@gmu.edu`

Key words: Agent-based Modeling, Computational Social Science, Financial Markets

1 Motivation and Objectives

The purpose of this project is to create computational platform that would be used to implement agent-based models of financial markets. In particular, following functionalities are desired:

- Simulation may host heterogeneous agents that follow news processes, learn, obey budget constraints etc.;
- Many different assets traded on markets using different trading rules may be present (based on price-impact heuristics, double auction);
- Flexible GUI and reporting facilities, including ability to summarize batch results from sensitivity analysis.

This paper outlines the prototype prepared during the CSS739 class, presents three sample models and follows the steps taken to ensure correctness of our implementation.

2 Platform Architecture

The implementation took place using MASON simulation framework, see Luke et al. [2005]. The static structure of our platform is outlined on Figure 1, where UML class diagram is presented (minus classes corresponding to orders). Figure 2 presents interactions taking place during a sample iteration between two agents and an orderbook.

^{*} Prepared for the 2nd World Congress on Social Simulation, GMU, Fairfax, 14–17 July, 2008.

^{**} Programming Group: Justin Briggs, Francesco Trusiano, Maciej Łatek, Randy Marks. Data Analysis Group: Sandler Lin, Cheryl Hansen and Thomas Tzitzouris.

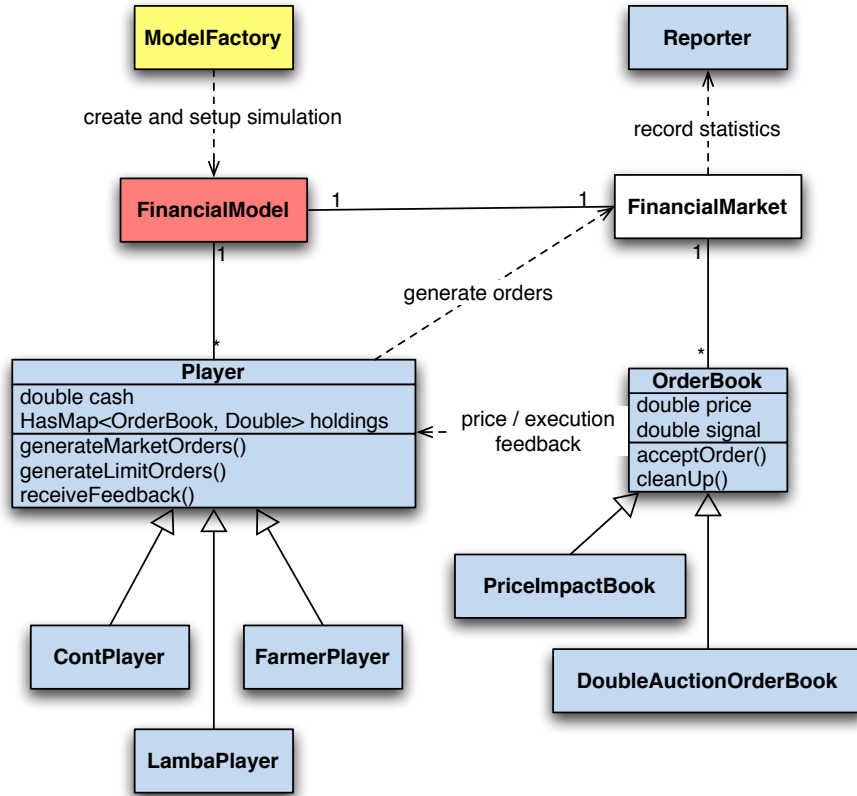


Fig. 1. High-level UML class diagram of the main components and relations in the FinancialMarketModel, including the main attributes of Players and OrderBooks. Agent classes (light blue) inherit from the MASON `Steppable` interface while the master class is implementation of MASON's `SimState`.

3 Implemented Models

3.1 Cont Model

The Cont model (Cont [2006]) is the simplest model implemented. All traders (agents) follow the same behavioral rules. They are heterogeneous in the sense that they are given independently assigned (subjective) volatility thresholds $\theta_i(t)$. Each period, all agents receive a common signal, which can be interpreted as public information or "news", in the form of IID Gaussian random variables $\epsilon_t \sim N(0, D^2)$. Each agent i responds to this signal by selling if $\epsilon_t < -\theta_i(t)$, buying if $\epsilon_t > \theta_i(t)$, and otherwise sitting out the period. The market then determines the excess demand and arrives at the market clearing price by means of a market impact function. Lastly, with probability s , agents update their threshold to

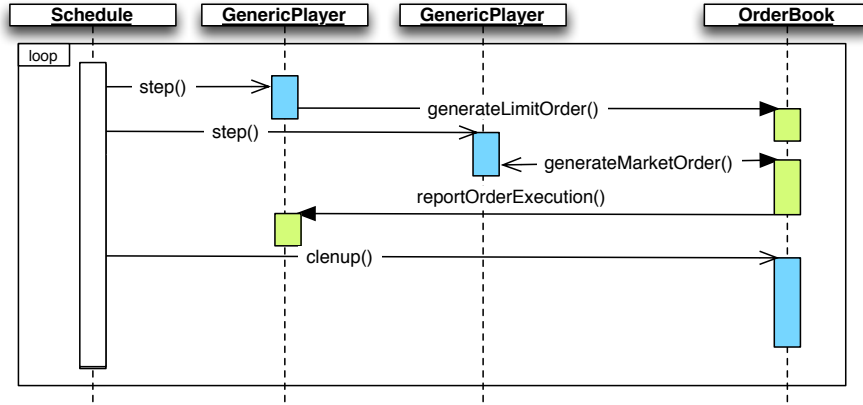


Fig. 2. High-level UML sequence diagram of the interactions between main object of the FinancialMarketModel, with messages send between two agents and a single orderbook during a single tick shown. Agents are activated by MASON simulation engine and may generate different kinds of **Market-** and **LimitOrders**, depending on their internal states and signals from environment (prices, news process). The **MarketOrder** gives an agent instantaneous feedback, whereupon effects of **LimitOrder** may be delayed in time. The **cleanup** signal causes the Orderbook to arbitrage all possible trades between **LimitOrders**, recalculate new prices and returns (if applicable) and remove expired orders.

match the absolute value of the return rate for the current period. Somewhat surprisingly, even this very simple "zero intelligence" model of agent behavior yields fairly realistic market movements, as presented in Figures 3 (for small s) and 6 (for large value of s).

One down side of the Cont model that is not noted in the papers is that there appears to be a great degree of periodic cyclicity in the volatility of the markets. This is especially apparent with higher probability s of updating thresholds. With $s = 0.1$, this is readily apparent if you look at the autocorrelation of absolute returns over longer time scales than shown in the paper. This effect can be particularly well seen when averaging over many runs, as on Figure 9. Holding s constant at 0.1 and varying D , we find the period and amplitude to be predictable.

3.2 Farmer Model

The Farmer model (Doyme Farmer et al. [2003]) has two types of agents, which trade a single asset by means of a continuous double auction similar to those normally used on stock markets. "Patient" traders place limit orders, which specify both a price and quantity they wish to buy or sell. Limit orders do not execute immediately, and may be canceled at any time. "Impatient" traders

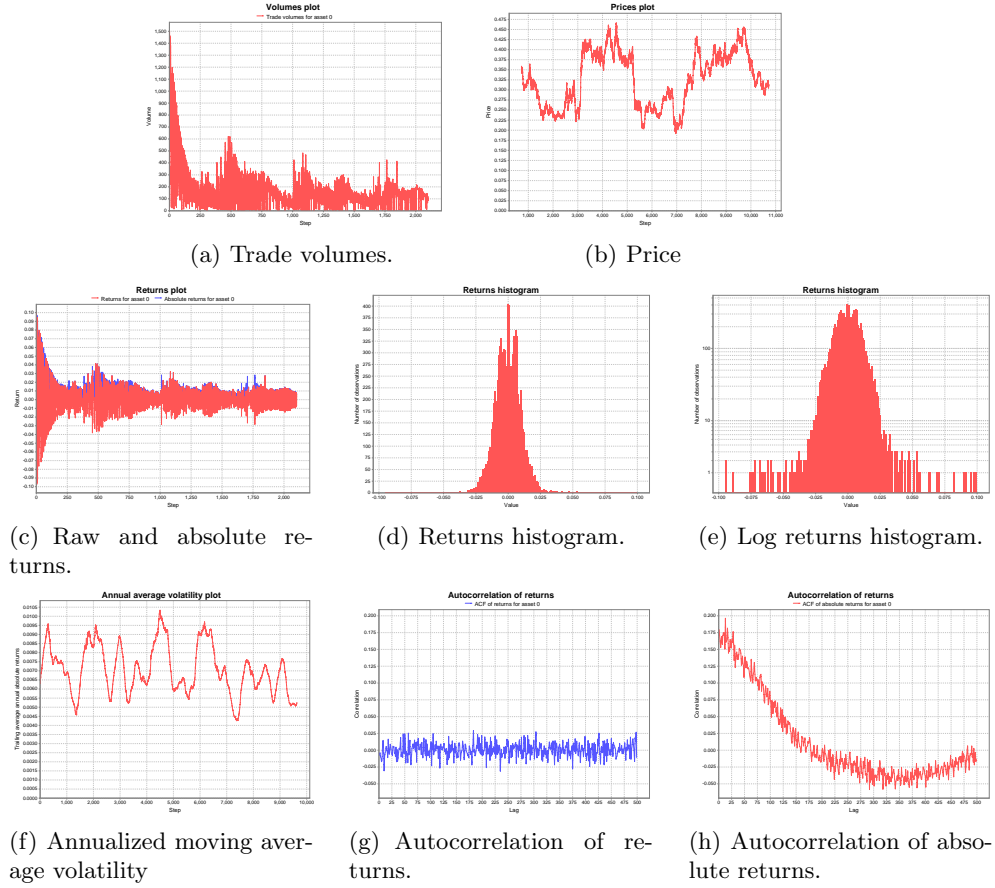
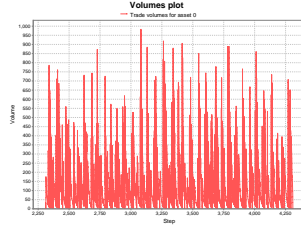


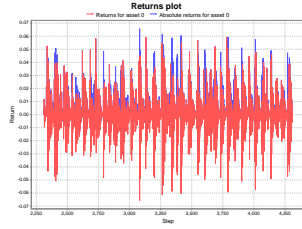
Fig. 3. Examples of outputs and statistics from a single run of the Cont FinancialModel simulation with $D = 0.001$, $s = 0.01$.



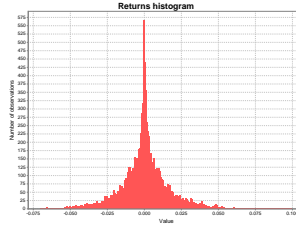
(a) Trade volumes.



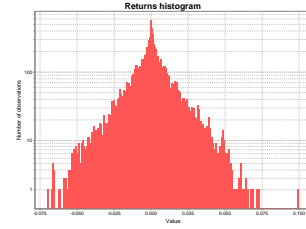
(b) Price



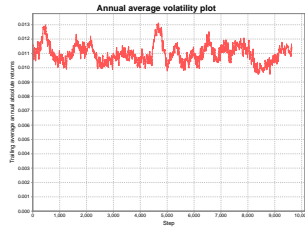
(c) Raw and absolute log returns.



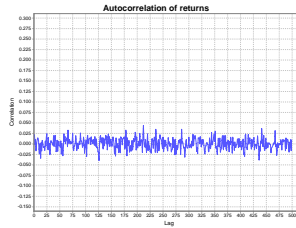
(d) Returns histogram.



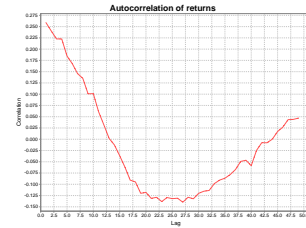
(e) Log returns histogram.



(f) Annualized moving average volatility



(g) Autocorrelation of returns.



(h) Autocorrelation of absolute returns.

Fig. 4. Examples of outputs and statistics from a single run of the Cont FinancialModel simulation with $D = 0.001$, $s = 0.1$.

place market orders, specifying only a quantity, which execute immediately at the current best price, assuming there is sufficient liquidity provided by outstanding limit orders. This model also has simple behavioral logic for traders. Behavior is regulated by:

- Patient agents place **LimitOrders** of constant size at a Poisson rate per unit time, and with a randomly generated price. Orders are equally likely to be either buy or sell orders. Buy limit orders are placed uniformly anywhere in the semiinfinite interval below the current ask price ($-\infty < p < a(t)$), where p is the *logarithm* of the price; similarly for sell orders. A uniform distribution in log prices is equivalent to an exponential distribution in prices. In addition, outstanding limit orders are canceled at a Poisson rate per unit time. All of these processes are independent.
- Impatient agents place **MarketOrders** of constant size at a certain Poisson rate per unit time.
- A double auction order book. This manages the pending limit orders and facilitates trades between market and limit orders. It also provides aggregate market information such as return rate per unit time and bid-ask spreads.

This model explains a large portion of the spread and price diffusion of actual stocks given an order flow rate, which essentially means that the double auction structure has a large impact on the nature of market movements. The model also yields market behavior qualitatively close to actual markets in many respects, see Figure 5.

3.3 Farmer-Cont Model

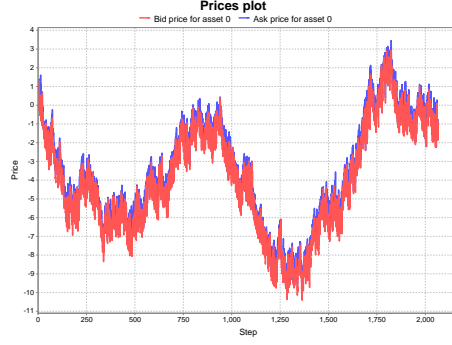
As an experiment, we constructed a model combining some aspects of the Farmer model with some aspects of the Cont model. The result is a model which is still essentially zero intelligence, but which exhibits empirically positive characteristics of both models. In this model there are two types of agents, which interact via the continuous double auction mechanism, as in Farmer. In fact, the Patient traders, which place limit orders and thus provide liquidity, remain the same. Impatient traders are modeled after Cont. They have heterogeneous volatility thresholds, and place market orders depending upon how public information in the form of a normally distributed random variable compares with these thresholds. Some fraction s of agents then adjust their volatility thresholds in light of current volatility by setting it to the trailing period's return rate.

TODO Note: liquidity issues

4 Verification of Correctness

TODO: show results side by side with original papers' figures

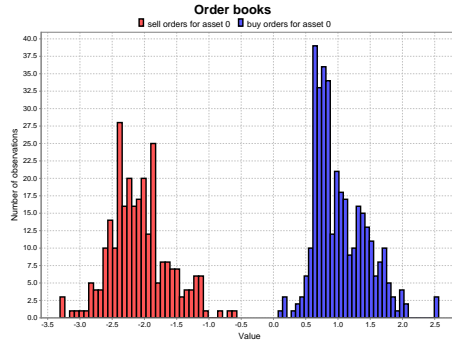
These compare well with data shown in the original paper (shown in Figures 7 and 8).



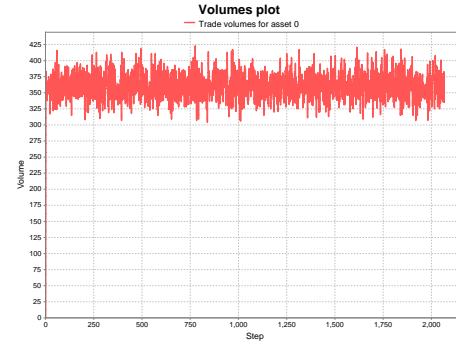
(a) Ask and bid prices.



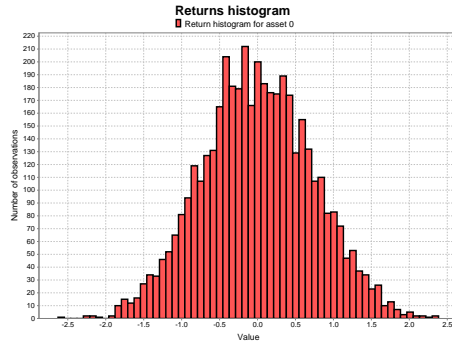
(b) Raw and absolute returns.



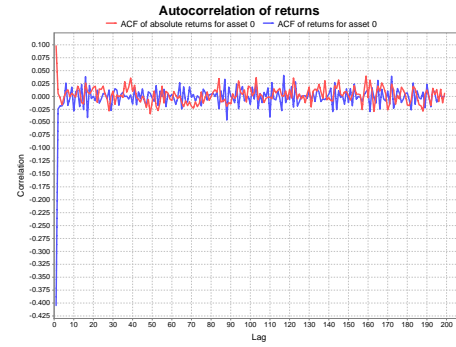
(c) Orderbook shape.



(d) Trade volumes.



(e) Returns histogram.



(f) Autocorrelation of returns.

Fig. 5. Examples of outputs and statistics from a single run of the FinancialModel simulation for default Farmer's parametrization.

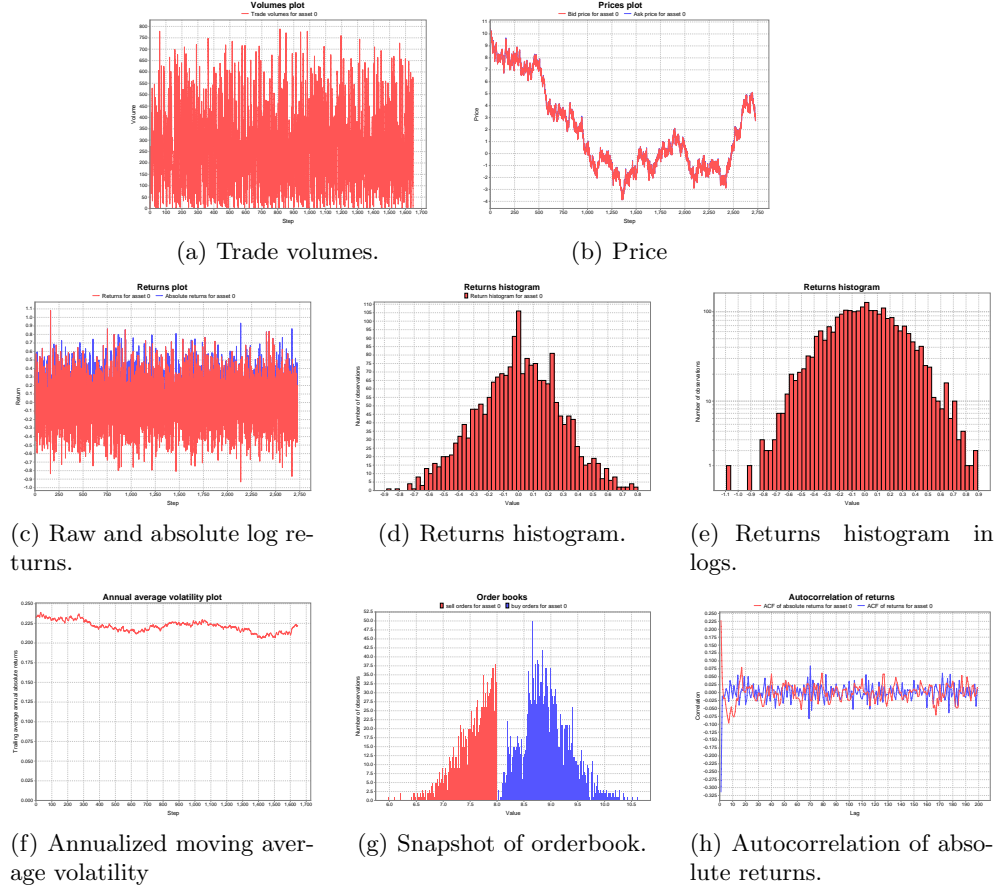


Fig. 6. Examples of outputs and statistics from a single run of the Farmer-Cont hybrid.

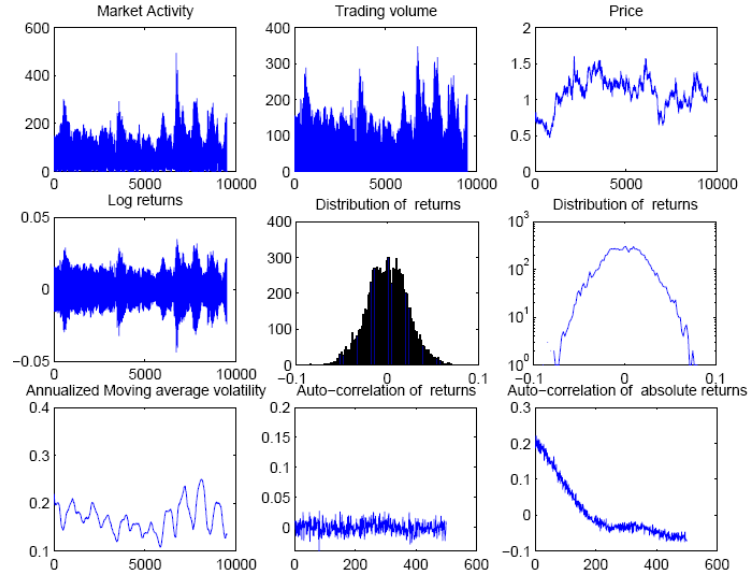


Fig. 7. Outputs from Cont's paper with $D = 0.001$, $s = 0.01$. Compare directly to Figure 3.

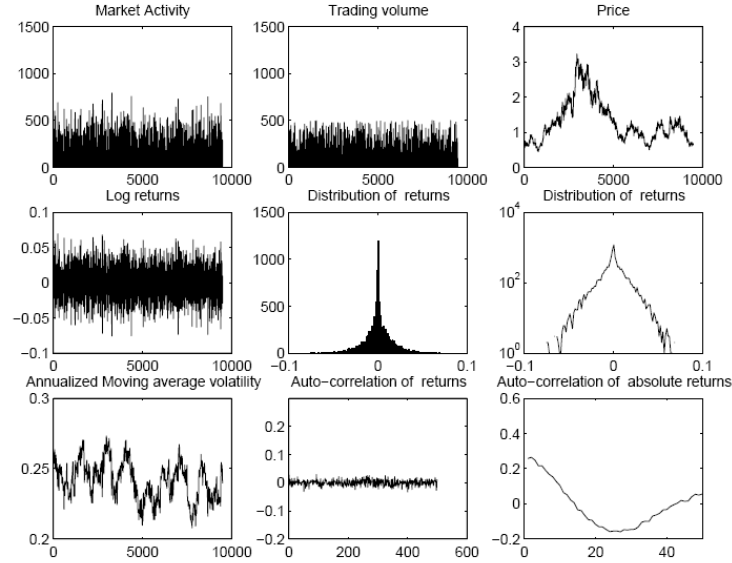


Fig. 8. Outputs from Cont's paper with $D = 0.001$, $s = 0.1$. Compare directly to Figure 6.

5 Batch Mode Instrumentation

FinancialMarketModel includes a set of wrappers that are useful for conducting multi-run batch parameter space exploration or sensitivity analysis. A simple wrapper is included in `experiments` package. Post-run analysis can be performed by using Matlab, sample scripts are included in `scripts` folder and can be used to render shape of autocorrelation curves and histograms of returns as a function of simulation settings. Figure 9 presents a sample prepared by varying the D news process variance for Cont's model.

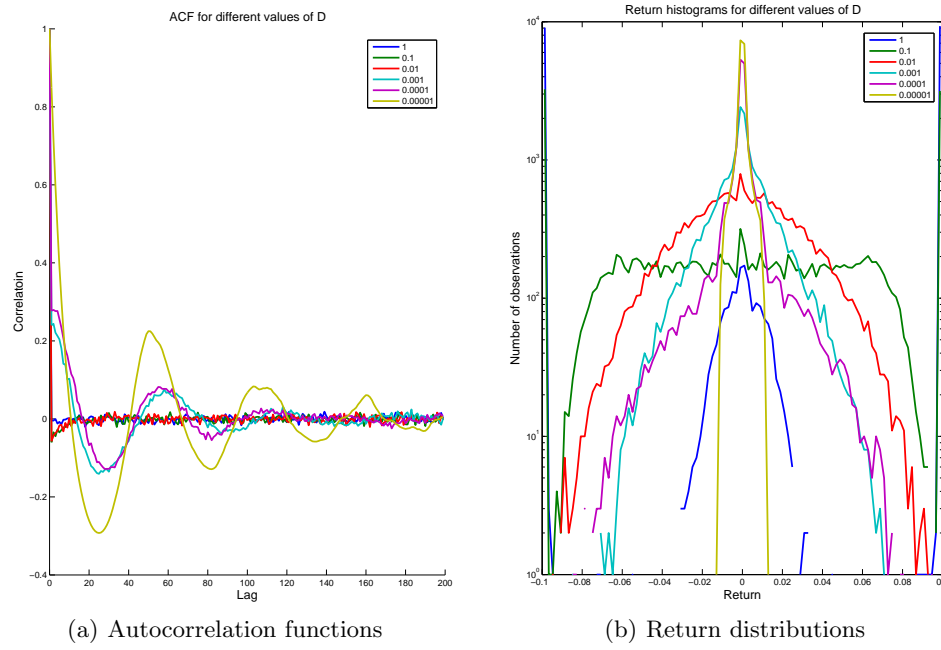


Fig. 9. Autocorrelation and distributions in Cont simulation with $s = 0.1$ and varying D , averaged over many runs.

Bibliography

- Rama Cont. *Long Memory in Economics*, chapter Volatility Clustering in Financial Markets: Empirical Facts and Agent-Based Models. Springer Berlin Heidelberg, 2006.
- J. Doyne Farmer, P. Patelli, and I. I. Zovko. The Predictive Power of Zero Intelligence in Financial Markets. *ArXiv Condensed Matter*, September 2003.
- Sean Luke, Claudio Cioffi-Revilla, Liviu Panait, Keith Sullivan, and Gabriel Balan. MASON: A Multiagent Simulation Environment. *Simulation*, 81:517–525, 2005.