Agent Based Simulation Design Principles – Applications to Stock Market

Lev Muchnik[1], Yoram Louzoun[2], Sorin Solomon[3,4]

[1] Department of Physics, Bar Ilan University, Ramat-Gan, Israel
[2] Department of Mathematics, Bar Ilan University, Ramat-Gan, Israel
[3] Racah Institute of Physics, The Hebrew University, Jerusalem 91904, Israel.
[4] Complex Multi-Agent Systems Division, ISI Torino

Summary.
We present a novel agent based simulation platform designed for general-purpose modeling in social sciences. Beyond providing convenient environment for modeling, debugging, simulation and analysis, the platform automatically enforces many of the properties inherent to the reality (such as causality and precise timing of events). A unique formalism grants agents with an unprecedented flexibility of actions simultaneously isolating researchers from most of the overhead of the virtual environment maintenance.

Key words. Agent-Based Simulation; Experimental Markets; Artificial Financial Markets; Market Microstructure.

Introduction

The classic analysis of financial phenomena is usually based on simple (often linear) *macroscopic* models, which preferably can be solved analytically. Such models can reproduce basic market macroscopic features. This type of models fails to reproduce emergent features of markets that cannot be directly deduced from the microscopic interaction producing them.

Emergent phenomena, were studied over the last couple of decades in a wide range of systems. A general approach is to model the system in question as a set of *microscopic elements* and define *microscopic interactions* between them so that the desired *macroscopic phenomenon* emerges. Being frequently and successfully exploited in physics, this method is now being applied in social sciences as well. In the specific context of the stock market, a variety of simplified *microscopic* models have been introduced over the last decade, (Bak et al. 1997, Stauffer 2000, Levy et al. 1994, Mantegna R., Stanley 1999, Maslov 2000, Solomon 2000 and many others). Most of these models focus on specific aspects of the problem: basic features of the agent's behavior or of the stock exchange procedures. They show that even a small set of simple assumptions can explain the set of ''stylized'' experimental facts characterizing generically the market (Mantegna, Stanley 1999, Cont 2001, Lux, Heitger, Takayasu): power (Pareto–Zipf) laws, fat tails (and/or

Levy-stable distributions), (multi-) fractal dynamics (Hurst exponents), long range correlations (clustered volatility), criticality (scaling exponents).

Similar models explaining stylized facts exist in practically all domains of social sciences ranging from social influence and opinion dynamics (Weidlich, Haag, 1983) to wealth distribution (Levy et. al 2000). It is obvious that in order to go beyond these generic ‘‘stylized’’ facts, one has to consider more realistic features (Solomon, 1999). In the context of the stock market, we would need to consider: detailed stock market procedures, individual trader behavior, communication lags, external events (news arrival, economic fundamentals, etc.) (Levy et al. 1994, Moss et al. 1999).

We developed a platform that simulates an arbitrary number of agents interacting with an arbitrary range of behaviors. A demand for such a tool has been present for quite some time already and several attempts were made to satisfy it (Jacobs et al. 2004, Minar et al. 1996, LeBaron 2002,2004). However, to the best of our knowledge, none of them possessed all the properties required to satisfy the growing community of researchers who could use it. We hereby introduce the basic concepts for the general simulation platform we have developed, named *NatLab* for ***N***atural ***A***synchronous-***T***ime Event-***L***ead ***A***gent-***B***ased Platform. *NatLab* is a realistic continuous time causal asynchronous event driven simulator. It is highly efficient - the cost of each event is proportional to the log of the number of candidate events. In *NatLab* the communication between agents is through a novel efficient messaging mechanism. In the next section first, the design principles of the simulation will be detailed, then a concrete application will be used to exemplify its potential.

NatLab Design Principles and Implementation Details.

NatLab deals with any arbitrary system of interacting agents unless they experience continuous interactions. As long as the microscopic inter-agent interactions can be presented as sequence of momentary “collisions”, the entire system can be simulated precisely. Each agent can engage any other agent, group of agents or the entire population. The spatial structure in *NatLab* is implemented using a fixed (or evolving) network of nodes and links, effectively introducing neighbors. The basic design principles of *NatLab* are as follow:

- Timing
  - Asynchronous update - Active agents are allowed to initiate actions as opposed to the conventional passive agents that are delayed until polled.
  - Event-Driven - The simulation engine schedules future events and processes them one by one, skipping the time in-between
  - Continues time - Unlike in conventional discrete time simulations, times of the events are precise as they are not confined to any discrete time.
- Causality - Being event-driven and executing all events at the precise time of their occurrence, the simulation does not accumulate inaccuracies as it evolves.

Moreover, the platform ensures the correct ordering of events, continuously maintaining the cause-effect principle. This allows each and every agent to respond and adjust by re-scheduling his future actions. The schedule of upcoming events is constructed on the fly, while the first event is executed.

- Realistic action cycle - Agents may exploit the asynchronous nature of the platform implementing a realistic multi-stage action cycle. Being isolated between events and exogenous to the agent messages, each agent can evolve in parallel until it either spontaneously decides to initiate some action or responds to an interaction coming from outside (either directly induced by an other agent or indirectly in response to an objective state change following some event execution). Since events are scheduled and executed when their time arrives, realistic delays in agents evolution and action time lags are naturally ensured.
- Messages - Agents interact by exchanging arbitrary delayed messages. The simulation platform engine guarantees delivery of the messages when the simulation time is promoted to the appropriate time. The same message can be delivered to a group or even all agents, which may either ignore it or respond by scheduling some future action (simulating natural delayed response).
- Efficacy -*NatLab* is optimally efficient. *NatLab* deals uniquely with events execution, wasting no resources on looping over the pool of agents, and very few resources on scheduling future events. The execution time scales as O(N*log(M)), where N is the total number of events, and M is the number of events currently in the queue. *NatLab* CPU cost does not depend on the total number of agents in the system.
- Multilevel - *NatLab* is suited to simulate not only relatively small isolated systems (such as the stock exchange), but can deal with systems of arbitrary complexity at several scales (time scales and organizational complexity). One could in principle model the entire economy– starting from individuals acting as employees through firms, banks, the stock exchanges, etc. All the interactions between them can be made as precise as required.

We will not provide all algorithmic details and the software implementation, as these are mainly technical and do not contribute to understanding the function of *NatLab*. There are, however some particulars which are essential in order to understand the general function of the simulation.

General structure: The platform is divided into three independently developed modules with a strictly defined interface and communication protocols between them. This structure allows the optimization of *NatLab*, and its implementation on very different platforms. Moreover, it allows the user to ignore the internal structure when designing agents. The basic elements are:

- *The Autonomous engine*. The highly optimized core of the platform, whose task is to automatically manipulate user-defined agents by delivering their messages.
- *Simulation User Interface* (Figure 1). This module provides the user with a convenient user interface, and is independent on of the simulation core.
- *Agents*. To reduce requirements from developers, we do not require inter-platform compatibility from each agent. However, each agent should be based

either on the template supplied with the platform, or strictly support the interface, specified by us.

Agents: *NatLab* implements a modular concept which requires each agent to be implemented and compiled as a separate binary module. These modules are automatically recognized and imported by the simulation platform as it loads or when at the configuration stage of a specific simulation run. All agents should be based upon the single basic agent class and provide the functionality required by *NatLab*. Moreover, they should all provide a common binary interface to allow the platform to operate them. The functionality includes mainly the agent ability to filter and respond to messages they receive, and to send new messages.

Messages: Messages are used for communication between agents. All messages must provide the minimal information required for their transfer and delivery: A) Issuing agent ID, B) Issue time, C) Destination agent ID, D) Delivery time, E) Message type (ENUM) and may include arbitrary additional information.

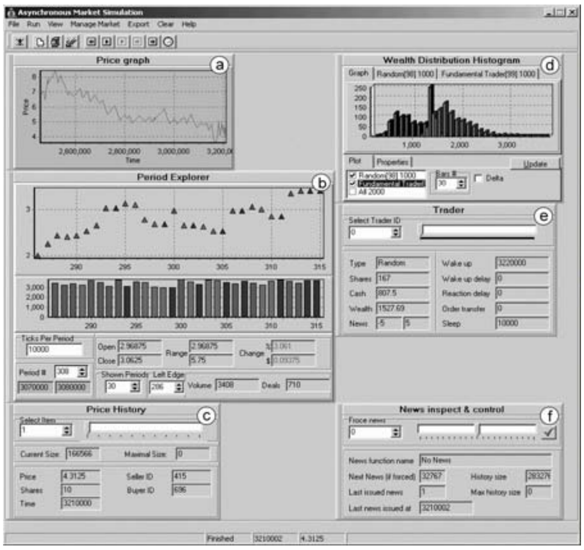

Fig. 1 GUI of NatLab. The graphical user interface of the Platform is designed to provide a comprehensive insight into the dynamical processes of any specific model at both macroscopic and microscopi levels. The windows providing textual or graphical representation of model parameters at the run time are also implemented as agents, each updating it's own content on demand. Similartly to model-specific agents, each window can be designed to update periodically or to respond to specific events. They can also allow the user to interfer into the simulation by providing the interface for setting parameters and issuing system-wide or specific messeges. In this particular example: (a)Shows the price evolution. (b)Daily price and volume. (c)Exposes details of every transaction. (d)Comparative distribution of traders' wealth for each type. (e)Provides acccess to detailed information about each agent. (f) Allows the user to inspect and issue news.

Application

Let us demonstrate the simulation platform by schematically presenting a simple stock exchange model where the spontaneous herding behavior of a tiny fraction of the traders populating it causes formation of a bubble and a crash, followed by a long recovery period. This simulation shows the huge effect of the uncoordinated action of a very small portion of traders. We do not intend to replicate or explain any of the real market properties, but to merely present an example of the platform use.

The model framework consists of a single stock traded by means of a classical orders book similar to the one employed today by most stock exchanges[1]. The market is mainly populated (98% in our runs) by one type of agents. These agents act randomly by occasionally submitting relatively small (~2% of their wealth) limit orders. The limit orders' prices are drawn from the neighborhood of the current market price. Each order has 50% probability to be a buy or a sell order. If these agents would be the only agents acting in the market, the stock value distribution would be approximately a narrow Gaussian around the value determined by the amount of money and the stock number. One could argue that the inclusion of a few uncoordinated agents should not significantly affect the market dynamics. We actually show that a small minority, trying (naively) to identify the market's behavior and to follow it can drastically change the market dynamics.

Let us, thus, assume a second small population (2% in our case, but it can be even smaller). Unlike the random traders, each of the agents belonging to the second type continuously tries to identify and exploit price trends. Those *occasional* traders represent people that do not invest in stocks unless they identify (to the best of their knowledge) a clear opportunity. In our case, we have selected a persistent positive price trend as an indicator for occasional traders to enter the market. They start with no shares at all and keep inspecting the prices until they recognize a trend. From this moment on, the agent gradually buys stocks until either its entire capital is invested or the price starts to drop. When the opposite trend is detected, the agent starts selling by submitting market orders. Note again that these minority traders (2%) are not synchronized.

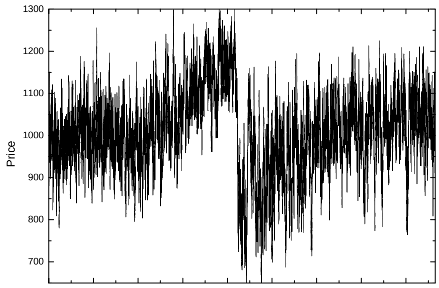

Fig. 2 The price evolution shows region of regular trade followed by gradual price raise due to massive acquisition by ocasional agents, crash and slow recovery.

In our runs executed for a population of 10,000 random traders and 200 occasional traders (Fig. 2) we observe the emergence of a herding behavior which causes a gradual increase of the price, followed by sudden crash. Each agent acts independently from the others, having personal criteria for its behavior. Occasional traders start with no stocks and watch the market sporadically identifying random fluctuation as appearing trends. In response to such an initially (erroneous) trigger the demand increases. This demand in turn actually raises the price, increasing the chance for other agents to identify it as a (real) emerging trend. Eventually, all occasional traders will interpret the raising price as an opportunity and will start competing with the rest for the available stocks (obviously raising the price even more). The process will continue until most of the occasional agents are exhausted and have no more cash to invest. At this point,

[1] www.nyse.com

the price stops rising. The traders interpret this by realization that no additional profits can be made in this situation and start realizing them as fast as they can. This, naturally enough, causes the price to crash. As soon as the majority of occasional traders manage to get rid of their stocks, the price stabilizes and even climbs gradually. The reason for the positive trend in this case is the interaction between the random agents. By definition, they do not distinguish between any of the two assets (the money and the stocks) and operate caring only for the volume of transactions. Therefore, they will tend to sell more shares at low prices, effectively causing the price to rise. This effect stabilizes the price near the price determined by the total amount of money and stocks random agents have. Note that none of the agents involved has to know what those numbers are.

References

P. Bak, M. Paczuski and M. Shubik, Price Variations in a Stock Market with Many Agents, Physica A, Dec 1997. cond-mat/9609144.

Cont R., Empirical properties of asset returns: stylized facts and statistical issues, Quantitative Finance, Volume 1, (2001) 223-236

Jacobs B. I., Levy K. N., Markowitz H.. Financial market simulations. Journal of Portfolio Management, 30th Anniversary, 2004.

LeBaron B. Building the santa fe artificial stock market. Physica A, 2002. Working Paper, Brandeis University.

LeBaron B. Agent-based computational finance. 2004. Priliminary draft.

Levy M., Levy H., Solomon S.. Microscopic Simulation of Financial Markets: From Investor Behavior to Market Phenomena. Academic Press, 2000. ISBN 0124458904

Lux T., Heitger F., Takayasu H., Micro-Simulations of Financial Markets and the Stylized Facts. The Empirical Science of Financial Fluctuations: Econophysics Approach on the Horizon. Berlin: Springer, in press.

Mantegna R. & Stanley H.E. An Introduction to Econophysics, Cambridge University Press, 1999. ISBN: 0521620082

Maslov S., Simple model of a limit order-driven market, PHYSICA A, 278, April 14, 2000, pp 571-578,cond-mat/9910502

Minar N., Burkhart R., Langton C., Askenazi M.. The Swarm Simulation System: A Toolkit for Building Multi-agent Simulations, 1996

de Oliveira S. M., de Oliveira P.M.C., Stauffer D., Evolution, Money, War, and Computers - Non-Traditional Applications of Computational Statistical Physics, Teubner, Stuttgart-Leipzig (1999), ISBN 3-519-00279-5

Solomon S. Towards behaviorly realistic simulations of the stock market. Computer Physics Communications 121 (1999) 161. http://xxx.lanl.gov/abs/adap-org/9901003

Solomon S., Generalized Lotka Volterra (GLV) Models of Stock Markets , Applications of Simulation to Social Sciences, Hermes Science Publications 2000, p 301

Stauffer S., Econophysics - a new area for computational statistical physics, International Journal of Modern Physics C, Vol. 11, No. 6 (2000) pp 1081-1087

Weidlich W., Haag G., Concepts and models of quantitative sociology. Springer, Berlin, 1983

Acknowledgement: This research was partially supported by the Israeli Science Foundation