Inferring the state of a double-auction market from empirical high-frequency transaction data

Minh Khoa Nguyen
CCFEA
Computer Science Department
University of Essex
mknguy@essex.ac.uk

Neil Rayner
CCFEA
Computer Science Department
University of Essex
njwray@essex.ac.uk

Steve Phelps
CCFEA
Computer Science Department
University of Essex
sphelps@essex.ac.uk

ABSTRACT

Much research in the area of multi-agent systems has been devoted to the analysis of trading agents in double-sided electronic marketplaces. However, to date there is very little *empirical* research validating these models against actual data from real markets. As a first step towards a principled approach to the calibration and validation of agent-based models of financial markets, we introduce a method for inferring the state of the order-book in a double-auction market from empirical transaction data. We use this inferred state to produce high-frequency time-series of the midpoint of the quote and the corresponding returns. We demonstrate that our model produces data that is consistent with well-known stylized facts of high-frequency financial time series data.

Categories and Subject Descriptors

I.6.4 [Simulation and Modeling]: Model Validation and Analysis

General Terms

Algorithms, Measurement, Economics, Experimentation

Keywords

Continuous double auction, trading agents, agent-based modeling, calibration.

1. INTRODUCTION & MOTIVATION

Much research in the area of multi-agent systems has been devoted to the analysis of trading agents in double-sided electronic marketplaces [10]. However, to date there is very little *empirical* research validating these models of agent-based markets against actual data from real agent-based markets. We believe that empirical validation of models against actual financial time series data represents a significant opportunity for the trading agents research community: trading in the financial markets today is almost entirely conducted electronically through automated trading agents [8], and data on every transaction in the marketplace is obtainable from the major financial exchanges [e.g. LSE[1], Eurex[2]] running to several terabytes per year per asset. Thus we have an unprecedented opportunity to calibrate and test our models against actual data from one of the largest implementations of a multi-agent system to date, viz. a financial market.

One approach to validating agent-based models is to demonstrate that they produce phenomena that are broadly consistent with those observed in reality (in the vernacular of finance we say that they reproduce the "stylized facts" of markets [3]), and that the results of this analysis are insensitive to the settings of free parameters [4]; that is, we demonstrate that the model is *robust*. However, because agent-based models typically have very many degrees of freedom, many attempts to demonstrate robustness fail when performed rigorously. However, this does not necessarily imply that we should abandon these models as unrealistic. Rather, it suggests a need to systematically calibrate the free parameters of the model based on observations of the real system.

Existing work has attempted to calibrate simple analytical models against financial time series data sampled at a daily frequency [6]. However, we are interested in building agent-based models which are able to explain phenomena such as long-memory in absolute returns that are only observable at a high sampling frequency. Many of these models take the form of agent-based simulations, and thus they cannot be straightforwardly calibrated by the standard methods since there is no closed-form likelihood function to optimise. However, there is nevertheless a scientific imperative to calibrate these models, and there is some early work on calibrating simulation models we can draw upon [17].

One way of viewing the calibration exercise is as an attempt to "reverse-engineer" a system by inferring the actual parameters of an agent-based system from observations such as historical time-series data. The challenge we face is that properties of the entities in our model are not directly observable in the historical record. For example, the raw transaction data available from the London Stock Exchange LSE shows only the details of transactions such as transaction-price and volume, and *not* details of the actual agents in the market place such as the number of agents and the strategies they are using. The (substantial) challenge is to infer the latter from the former.

[1] http://www.londonstockexchange.com/products-and-services/reference-data/trade/trade-data.htm

[2] http://deutscheboerse.com/dbag/dispatch/en/listcontent/gdb_navigation/mda/300_historical_market_data/30_eurex_hist_orderbook/Eurex_Historical_Orderbook.htm

As a first step in this direction we present work which attempts to infer the state of the auction (the "order-book"), from the historical record of transaction data in the market.

Reconstructing the order-book is fundamental to understanding the behaviour of real financial markets, since without the order-book we do not even know how prices change in between the daily closing price; the daily historical prices published in the media are low-frequency samples of a signal that contains much richer information when viewed at high-frequency. For example, the May 6, 2010 "Flash Crash[3]" (during which the Dow Jones lost 10% of its value only to recover in minutes) would remain invisible to anybody investigating historical prices sampled on a daily basis; rather a *high-frequency* analysis of prices and returns is required if we are to understand the behaviour of automated trading agents which trade on an intra-day, and sometimes intra-second basis. Our immediate focus is recovering the mid-point price from high-frequency data. Since the mid-point of the market quote depends on the state of the order-book [18] we must infer the state of the state of the order-book at previous moments in time in order to obtain the high-frequency price time-series.

The remainder of this paper is organized as following. The next section reviews related work and compares it to our provided software. Section 3 gives some background on the LSE and its electronic trading platform SETS. Section 4 describes the implementation. Section 5 is devoted to the analysis of the output of the software and section 6 concludes.

2. REVIEW

There have been various papers reporting the reconstruction of limit order books of the major exchanges. In [9] the NYSE is rebuilt, in [2] the Paris Bourse, in [7] the Australian Stock Exchange and in [5] the LSE historic order book was reconstructed.

However, to the best of our knowledge there is no open-source or publically-available commercial *software* to retrieve the historic state of the order-book. Although all major financial exchanges are double auction markets, their specific rules and order types differ considerably, requiring for every order book rebuild a specific market type data. While most exchanges provide this data, it entails some fee and usually also some legal restriction on its use, such as the dissemination of the raw data to a third party. Due this restrictions the demand of and supply of such an order book rebuild software is probably less desired. The closest software we found to a historic LSE order book rebuild is called eTradPlat[4]. It is a software designed to test trading strategies in an historic LSE environment. However, in terms of retrieving and exporting the historical order book it has some substantial drawbacks. A summary comparison with our software can be seen in Table 0.

Table 0. Software Comparison Table

Comparison	eTradePlat	Order Book Rebuilder
Programming Language	Java	Microsoft SQL Server, C#
Access to Source of Simulation	No	Yes
Access to All Market Variables	No	Yes
Explicit Printing Function	No	Yes
Open Source	No	Yes

The eTradPlat simulator [12] is not designed specifically to extract information of the evolution of the order book but rather to back-test prototype trading strategies. For this purpose the eTradPlat simulator only provides an application programming interface (API) for the user in which he or she can implement trading strategies in question. Though the API allows the user to retrieve some information from the actual order book in order to interact with the market, he cannot access all information of the market such as order details of an incoming market order, a feature that is deliberately implemented in order to ensure the realism of trading market interaction. For the latter reason the trading strategy and the actual market simulator run on different threads, meaning that they are not synchronized. Therefore although the trader can interact with the market at any time, he cannot govern the exact point in time of entry or exit.

In contrast, our software explicitly rebuilds the historical state of the order book and produces data for modeling and validating agent-based models.

3. GENERAL INFORMATION ON LSE & SETS

In this section we give a brief overview of the LSE and its electronic exchange – "Stock-exchange Electronic Trading Service" (SETS). With a history[5] of over 300 years the LSE is one of the oldest stock exchanges in the world and has become one of the most important centres of the global financial community. LSE's electronic trading service SETS is a fully electronic trading platform in which the constituents of the FTSE All Share Index, Exchange Trade Funds, Exchange Traded Commodities and other important AIM and Irish securities are traded.

The trading mechanism[6] is a double auction in which buy and sell orders are matched in an order book. Traders can submit the following type of orders (for details see Table 1):

[3] http://en.wikipedia.org/wiki/Flash_crash

[4] www.etradplat.com/

[5] http://www.londonstockexchange.com/about-the-exchange/company-overview/our-history/our-history.htm

[6] http://www.londonstockexchange.com/products-and-services/connectivity/tradelect/tradelect.htm

- Market order
- Limit order
- Day order
- Fill or kill
- All or none
- Immediate or cancel
- Stop
- Stop Limit
- Iceberg

The market itself operates in three different modes as depicted in Figure 2.

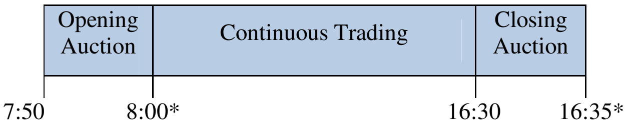

Figure 2. LSE Operation Modes

The first stage between 7:50 and a random time between 8:00 and 8:00:30 (depicted as 8:00*) constitutes an opening auction in which limit orders and market orders are entered and deleted on the order book, however no execution is carried out. The purpose of this stage is to discover an opening price for the real trading period, which commences randomly after the opening auction. The day ends with a closing auction at 16:30 and lasts for five minutes plus a random time up to 30 seconds (depicted as 16:35*) to ensure a high quality closing price of the day.

Table 1. LSE Order Types

Order Type	Description
Market order	Order to buy or sell at best available price when the order is executed. Market orders are more likely to be filled, however not a specific price
Limit order	An order that allows the trader to set the limit price. A buy limit order has a maximum price and respectively a sell limit order has the minimum price, at which the trader is willing to execute. It does not guarantee execution, but if at either pre-determined price or a better price
Day order	An order that expires if it is not executed within the given trading period
Fill or kill	An order that must be filled immediately, otherwise be cancelled instantly
All or None	An order that is either executed completely filled or not at all, in the latter however it will not be cancelled as the Fill or Kill
Immediate or Cancel	An order that requires all or part of the order to be executed immediately. The part that is not executed will be cancelled immediately
Stop	An order that becomes a market order once the security has traded through the designated stop price. Buy stops are entered above current ask price. If the price moves to or above the stop price, the order becomes a market order and will be executed at the current market price. This may be higher or lower than the stop price. Vice versa for sell stops.
Stop Limit	An order that becomes a limit order once the security trades at the designated stop price. A stop limit order will be executed at a specific price or better, but only after a given stop price has been reached or passed. It is a combination of a stop order and a limit order
Iceberg	A large single order that will be successively divided into smaller parts, usually by the use of an automated program, for the purpose of hiding the actual order quantity.

4. IMPLEMENTATION

In this section we give a brief description of the implementation. The aim of the software is to reconstruct the historical state of the auction from raw LSE transaction data in order to make inferences about market variables such as mid price, gaps, depth and level volumes or market impact of orders. These variables are defined as follows:

1. mid price: mid price between best ask and bid in the order book
2. gaps: difference of subsequent price levels in the book
3. depth: volume offered at best prices
4. level volume: volume offered at given price levels
5. market impact: mid price change caused by the execution of an order

As it can be seen from the definitions, these variables cannot be determined without knowing the actual state of the order book. However, the state of the order book is not directly observable, but must be inferred from the low-level transaction data.

Therefore the next subsection will describe the low-level transaction data before we outline the procedure of inferring the historic order book.

4.1 DATA

The LSE provides three sets of data:

- `Order Details`
- `Order History`
- `Trade Reports`

`Order Details` contains information of new orders entering the order book. The most important attributes are: price, volume, time and date, order code, buy or sell indicator. `Order History` records the history of changes of each order. There are five events that determine the history of an order:

1. the expiry of an order,
2. the deletion of an order,
3. amendments of its quantities,
4. a partial matching of an order,
5. a full matching of an order.

Once an order is matched (fully or partial) the order code of the counter order is also recorded in `Order History` and the details of the transaction are recorded in `Trade Reports` (all trades occurring during the auction process are recorded in `Trade Reports`). Note that any *non-persistent* orders, that is orders such as market orders that are never queued on the order book, are not explicitly recorded in `Order Details` or `Order History`. However, their existence can be inferred from their interaction with other orders in the book, as detailed below. For further information on LSE data see [15].

4.2 Order Book Inference

Given the nature of the LSE data we can reconstruct the state of the order book by *simulating* the auction. The general idea is to first reconstruct the sequence of events that occurred in the market from the raw transaction data, and to then run these events through a simulation of the LSE continuous-double auction in order to determine the outcome of these events on the state of the order book. However, as discussed not all events are explicitly recorded in the data and the missing events, and all non persistent orders (market orders) need to be inferred first. As set by the LSE trading rules market orders are instantly matched with orders in the book and lead to an immediate transaction. Though market orders are not explicitly recorded, the matching event (event 5 in Section 4.1) is recorded and this gives us the information on the time of the market order placement, and its order code (as the counter order of the matching also recorded in the matching event). By cross-referencing with the corresponding trade report, we can then retrieve the exact volume of the market order. This information completely describes the event of an incoming market order.

Once having all events including the arrival of non persistent orders and ordering them in the sequence of occurrence, this list should reflect the exact evolution of the order book when executed in the simulation. However in practice, one has to account additionally for data errors, such as missing partial events or conflicting variable values. E.g. occasionally one finds that an entry of a market order has led to a transaction of certain size but this change of volume for the matched order in the order book was not reported or the total transaction size associated to a certain order in the book exceeds its reported initial volume, leading to negative, post transaction volume. A solution for the former is to include the missing events, which can be detected by double checking that every trade effects two orders, resulting in two different order histories. The solution to the latter problem is to overwrite the initial order volume with the corresponding total transaction size. Including these procedures in the event completion stage, that is when retrieving market orders, ensures a complete, consistent and correct event list, which allows an accurate rebuild of the historic order book.

Overall in order to reconstruct the historic auction process using original LSE data our software performs the following steps:

1. retrieving information on all non persistent orders and missing events from the original LSE data
2. correcting conflicting variable values
3. retrieving complete historic event list, with all events sequenced by their actual occurrence
4. execution of the event list.

By following the event list step by step the software reproduces the exact evolution of the auction and allows us to retrieve the desired market variables mentioned above.

5. VALIDATION

In order to validate our approach we produce a high-frequency time-series of the quote mid-point and take this as the market price. We then calculate the corresponding time-series of returns and test for several well-known stylized facts of high-frequency financial data [3]:

1. absence of autocorrelation of returns
2. heavy tails of return distribution
3. long-memory in absolute returns
4. gain-loss asymmetry.

In particular 2 and 3 high are frequency phenomena (see [14], [1]). The absence of autocorrelation of returns reflects the random walk nature of empirical price processes where directions of price changes are not predictable. Predictable however, are the magnitude of price changes (absolute returns), which follow a long memory process where large price changes tend to follow large price changes, small price changes tend to be followed by small price changes. Large price changes itself are “frequent” events which is reflected in the heavy tails of the return distribution. The latter has been found to be left-skewed, there are more observable large draw downs in prices than upwards movements.

5.1 Data

Our price sample time series (see Figure 3) is retrieved from the order book reconstruction of a specific stock, viz: “Bluebay Asset Management”, over the period of April and May 2008. As sample interval we have chosen 5 min within the daily trading period from 8.00 AM to 4.30 PM.

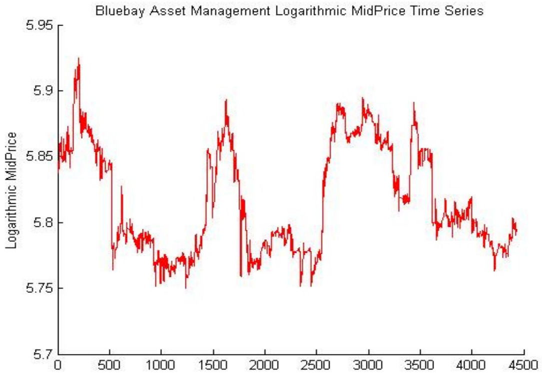

Figure 3. Price Time Series

The price series consist of logarithmic mid prices. The logarithmic mid price is defined as:

$$p(t) = \frac{1}{2}[\log(ask(t)) + \log(bid(t))]$$

,where $ask(t)$ and $bid(t)$ the best ask and bid price available in the market at time t. The logarithmic returns can be calculated as:

$$r(t) = p(t) - \ p(t-1)$$

where $p(t)$ and $p(t-1)$ are the logarithmic mid prices at time t, $t-1$. The return time series can be seen in Figure 4.

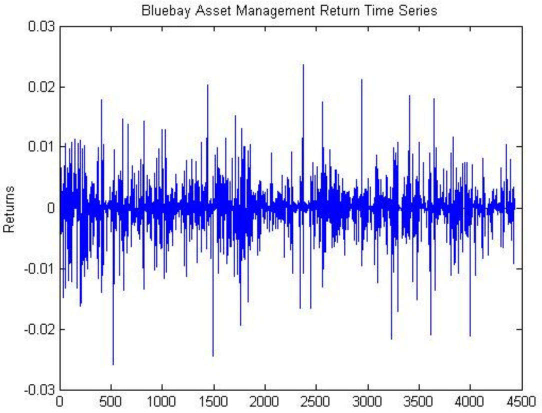

Figure 4. Return Time Series

5.2 Results

Some basic summary statistics are reported in Table 2. Of particular interest are kurtosis and skewness; the high value of the kurtosis indicates a fat tailed return distribution, whereas the negative skewness suggests an asymmetric return distribution with more weight on negative values than positive ones.

Table 2. Summary Statistics for Returns

Mean	Standard Deviation	Kurtosis	Skewness
-6.7x10^{-6}	0.0030	15.5740	-0.4966

Both fat tail return distribution and gain loss asymmetry are well known stylized facts of financial data [3].

Additionally to fat tail return distribution and gain loss asymmetry, we have also checked for i) absence of autocorrelation of returns ii) long memory in absolute returns. For the latter we use the Lo R/S statistic [11], which is a statistic specifically designed to detect long memory in time series. Figure 5 shows the autocorrelation value of returns against the lag used for calculating the autocorrelation and clearly shows the absence of autocorrelation over 20 lags.

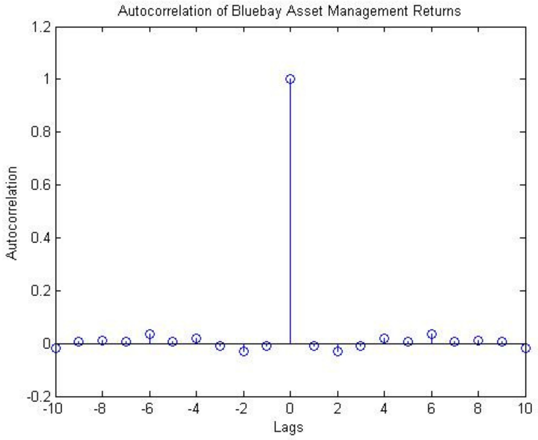

Figure 5. Autocorrelation of Returns

In the R/S statistics table (Table 3) we report the R/S statistic for absolute returns calculated at four different lags. As can be seen over a range of 50 min to 200 min (one lag corresponds to 5 min, the lag interval of the given time series) we clearly detect long memory in absolute returns.

Table 3. R/S Statistics Table

	Lags			
	10	20	30	40
R/S Statistics	2.3940	2.0755	1.9141	1.8132
Long Memory[7]	Yes	Yes	Yes	Yes

[7] If the R/S statistics exhibits the critical value of 1.747, the process exhibits long memory.

6. CONCLUSION

We have described how the state of a continuous double-auction can be inferred from empirical low-level transaction data from the London Stock Exchange. By inferring the state of the auction as it evolves over time we are able to reconstruct a high-frequency time-series of asset prices and the corresponding returns. We have validated our approach by testing that the resulting time-series data are consistent with well-known stylized facts of high-frequency financial time series.

This work is a first step towards the ambitious goal of inferring the complete state of the market in terms of an agent-based model: ultimately we hope to be able to recover properties of the *agents* in the market as well as the state of the auction process. That is, we would like to be able to *calibrate* the parameters of an agent-based by fitting it against empirical data. The work presented here is a small but fundamental step towards this goal.

7. REFERENCES

[1] Andersen,T. and Bollerslev, T. 1997. Intraday periodicity and volatility persistence in financial markets. Journal of Empirical Finance, 4(2-3):115–158.

[2] Auguy, M. and Le Saout, E. 1999. La liquidité cachée durant la séance de cotation a la Bourse deParis: une étude du règlement mensuel. Working Paper, U of Aix-Marseille III and U of Paris I.

[3] Cont, R. 2001. Empirical properties of asset returns: stylized facts andstatistical issues. Quantitative Finance, 1(2):223–236.

[4] Fagiolo, G., Moneta, A. and Windrum, P. 2007. A Critical Guide to Empirical Validation of Agent-Based Models in Economics: Methodologies, Procedures, and Open Problems. Computational Economics, 30(3):195–226.

[5] Farmer,J.D. 2005. The key role of liquidity fluctuations in determining largeprice changes. Fluctuation and Noise Letters, 5(2):209–216.

[6] Gilli, M. and Winker, P. 2003. A global optimization heuristic for estimating agent based models. Computational Statistics & Data Analysis, 42:299– 312.

[7] Hall, A. D. and Hautsch, N. 2005. Order aggressiveness and order book dynamics. Empirical Economics, 30(4):973–1005.

[8] Hendershott, T. 2003. Electronic Trading in Financial Markets. IT

[9] Kavajecz, K. A. 1999. A Specialist's Quoted Depth and the Limit Order Book. The Journal of Finance, 54: 747–771.

[10] Lebaron, B. 2006. Chapter 24 Agent-based Computational Finance. Handbook of Computational Economics, 2(05):1187–1233.

[11] Lo, A. W. 1991. Long-Term Memory in Stock Market Prices. Econometrica, 59(5):1279.

[12] Malik, A. 2007. eTradPlat. User Guide.

[13] Marks, R. E. 2007. Validating Simulation Models: A General Framework and Four Applied Examples. Computational Economics, 30(3):265–290.

[14] Müller, U., Dacorogna, M. and Pictet, O. 1996. Heavy tails in high frequency financial data.

[15] Puddick, J. 2007. Historic Order Book Rebuild Data Description and Guidance notes.

[16] Wagener, F. and Hommes, C. 2008. Complex Evolutionary Systems in Behavioral Finance. Social Science Research Network Working Paper Series.

[17] Werker, C. and Thomas, B. 2004. Empirical Calibration of Simulation Models. Eindhoven University of Technology, Eindhoven Centre for Innovation Studies, Tech. Rep.

[18] Wurman, P. R., Walsh, W. E., Wellman, M.P., Ave, B. and Arbor, A. 1998.Flexible Double Auctions for Electronic Commerce: Theory and Implementation. International Journal of Decision Support Systems, 24:17–27.