

A stylized model for the continuous double auction

Tijana Radivojević^{*}, Jonatha Anselmi[†], Enrico Scalas[‡]

Abstract A stylized phenomenological model for the continuous double auction is introduced. This model is equivalent to two uncoupled $M/M/1$ queues. The conditions for statistical equilibrium (ergodicity) are derived. The results of Monte Carlo simulations are presented on the behaviour of price differences and log-returns.

Keywords: Continuous double auction, high-frequency finance, high-frequency trading, jump-diffusion models.

Journal of Economic Literature Classification: G13

1 Introduction

In financial markets, when one considers tick-by-tick trades, not only price fluctuations, but also waiting times between consecutive trades vary at random. This fact is a consequence of trading rules. In many regulated financial markets trading is performed using the continuous double auction. Rule details may vary from stock exchange to stock exchange, but, essentially, things work as follows. For every stock traded in the exchange, there is a *book* where orders are registered. Traders can either place buy orders (*bids*) or sell orders (*asks*) and, for this reason, the auction is

^{*} BCAM - Basque Center for Applied Mathematics, Alameda de Mazarredo 14, 48009 Bilbao, Basque Country, Spain, e-mail: tradiojevic@bcamath.org

[†] BCAM - Basque Center for Applied Mathematics, Alameda de Mazarredo 14, 48009 Bilbao, Basque Country, Spain, e-mail: anselmi@bcamath.org

[‡] Dipartimento di Scienze e Tecnologie Avanzate, Università del Piemonte Orientale *Amedeo Avogadro*, Viale T. Michel 11, 15121 Alessandria, Italy and BCAM - Basque Center for Applied Mathematics, Alameda de Mazarredo 14, 48009 Bilbao, Basque Country, Spain, e-mail: enrico.scalas@mfn.unipmn.it

called double. When the auction is running, orders can be placed at any time, therefore the auction is called continuous. There are many different kinds of possible orders, but the typical order is the *limit order*. A bid limit order is an order to buy $q_b^{(T)}$ units of the share at a price not larger than a limit price $p_b^{(T)}$ selected by the trader, where T is a label identifying the trader. An ask limit order is an order to sell $q_a^{(T)}$ units of the share at a price not smaller than a limit price $p_a^{(T)}$ selected by the trader. The couples $(p_b^{(T)}, q_b^{(T)})$ are written in the book and ordered from the best bid to the worst bid, the best bid being $\hat{p}_b = \max_{T \in I_b} (p_b^{(T)})$, where I_b is the set of traders placing bids. The couples $(p_a^{(T)}, q_a^{(T)})$ are written in the book, as well, and ordered from the best ask to the worst ask, the best ask being $\hat{p}_a = \min_{T \in I_a} (p_a^{(T)})$, where I_a is the set of traders placing asks. At a generic time t , one always has that $\hat{p}_a(t) > \hat{p}_b(t)$. The difference

$$s(t) = \hat{p}_a(t) - \hat{p}_b(t) \quad (1)$$

is called *bid-ask spread*. Occasionally, a trader may accept an existing best bid or best ask, and the i -th trade takes place at the epoch t_i . This is called a *market order*. Market rules specify which are the priorities for limit orders placed at the same price and what to do when the quantity requested in a market order is not fully available at the best price. In the literature, the mid-price is often used, defined as

$$p_m(t) = \frac{\hat{p}_b(t) + \hat{p}_a(t)}{2}, \quad (2)$$

in order to summarize and study the above process. Both the bid-ask spread and the midprice can be represented as step functions varying at random times. Jumps in spread and midprice may occur when a better limit order enters the book or when a trade takes place. Another important process is the one of realized trades, that is the sequence of prices $p(t)$ at which a trade takes place.

Recently, Cont et al. ([1] and [2]) introduced simple models for the continuous double auction based on queuing theory. The purpose of the present paper is to further simplify the ideas and methods discussed by Cont and co-authors in order to introduce an essential and tutorial model based on queuing theory which can be easily described and understood. These models can be used to assess the relevance of statistical equilibrium in financial markets and to study the effect of market microstructure on the book and on trade prices as shown in the following.

Even if we greatly appreciate the effort made by Cont and co-authors to make their model as realistic as possible while preserving analytical tractability, we do believe that a further simplification may help in letting many scholars understand the importance and depth of this class of phenomenological models.

2 An Elementary Model for the Continuous Double Auction

2.1 Description of the model

In our simplified model, prices can assume only N discrete values from 1 to N (or from 0 to $N - 1$ without loss of generality) and every price can be seen as a *category* where incoming orders are placed. Note that this way of representing prices is a faithful representation of what happens in real auctions due to tick discretization. Further note that orders play the role of *objects*. Limit orders can be of two types: *bid orders*, that is orders to buy *one* share and *ask orders*, that is orders to sell *one* share. The distinction between objects and categories is the basis of the methods discussed in [3]. We shall assume that bid orders arrive following a Poisson process with rate λ_b whereas ask orders arrive following a Poisson process with rate λ_a . These orders are written in the book and stay there until a market order arrives. The best bid price is always strictly smaller than the best ask price. When a buy market order is sent to the market, an order belonging to the best ask price is removed, this happens again following a Poisson process with rate μ_b in the presence of ask orders. Conversely, when a sell market order arrives, an order belonging to the best bid price is removed, which is an event following a Poisson process with rate μ_a in the presence of bid orders. If no limit order is present in the book, the market orders are not executed. In order to further specify the model, we shall assume that limit ask orders can be uniformly placed in the price categories from $\hat{p}_b + 1$ to $\hat{p}_b + n$, where \hat{p}_b is the category (index) corresponding to the best bid; correspondingly, limit bid orders can be uniformly placed in the price categories from $\hat{p}_a - 1$ to $\hat{p}_a - n$, where \hat{p}_a is the category (index) corresponding to the best ask. One must always have $\hat{p}_b < \hat{p}_a$. The integer parameter n is typically such that $n \ll N$ with $N \gg 1$, but both these assumptions can be relaxed. One must also be careful when \hat{p}_a and \hat{p}_b come too close to the boundaries. This problem will be discussed in detail in the next section devoted to Monte Carlo simulations. Figure 1 shows a state of this system and illustrates the meaning of the various descriptions.

The different descriptions discussed above are useful for different purposes. For instance, individual description can be used to detect insider trading, by tracing individual orders. In this paper, as we are interested in properties such as the ergodicity of the process introduced above, we shall focus on statistical descriptions. For the sake of simplicity, we will modify the notation with respect to the caption of figure 1 and we shall denote with a_i the number of ask orders in category i (at price i) and by b_j the number of bid orders in category j (at price j).

In order to study the ergodicity of our model, we shall first consider a , the total number of ask limit orders and b , the total number of bid limit orders. One has that

$$a = \sum_{i=1}^N a_i,$$

and

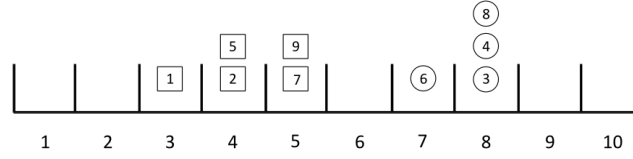


Fig. 1: A state of the system. Limit bid orders are represented by squares whereas limit ask orders are depicted by circles. Each order can be described by a label (or by a name). An individual description is a list telling us, for each limit order, whether it is a bid or and ask and to which category (price) it belongs. In this case, it is $x_1 = (3, b)$, $x_2 = (4, b)$, $x_3 = (8, a)$, $x_4 = (8, a)$, $x_5 = (4, b)$, $x_6 = (7, a)$, $x_7 = (5, b)$, $x_8 = (8, a)$, $x_9 = (5, b)$. A statistical description is a list giving us for each category (price) the number of limit orders either of bid or of ask type it contains. In this case we have $y_1^a = \dots = y_6^a = 0$, $y_7^a = 1$, $y_8^a = 3$, $y_9^a = y_{10}^a = 0$ for ask orders and $y_1^b = y_2^b = 0$, $y_3^b = 1$, $y_4^b = y_5^b = 2$, $y_6^b = \dots = y_{10}^b = 0$ for bid orders. Finally, a partition description is the number of categories (prices) with zero limit ask/bid limit orders, one ask/bid limit order, etc.. In our case we have $z_0^a = 8$, $z_1^a = 1$, $z_2^a = 0$, $z_3^a = 1$ for ask orders and $z_0^b = 7$, $z_1^b = 1$, $z_2^b = 2$, $z_3^b = 0$ for bid orders.

$$b = \sum_{i=1}^N b_i.$$

Note that the list $\mathbf{a} = (a_1, \dots, a_N)$ is such that, surely, $a_i = 0$ for $i < \hat{p}_a$ and the list $\mathbf{b} = (b_1, \dots, b_N)$ is such that, surely, $b_i = 0$ for $i > \hat{p}_b$. Note that, as in [2], one can represent the state of the auction with a single list $\mathbf{l} = (l_1, \dots, l_N)$ where $l_i \leq 0$ for $i \leq \hat{p}_b$ and $l_i \geq 0$ for $i \geq \hat{p}_a$, where negative values represent the number of bid orders at price i and positive values the number of ask orders at price i .

2.2 An exact result

Deriving analytical results is always useful. Indeed, these results can be compared with empirical data in order to verify the model. Moreover, it is possible to use analytical formulas in order to check the correctness of Monte Carlo simulations, as we will do in the following. From the description of the model and using a simple coupling argument, it turns out that $A(t)$, the total number of asks, and $B(t)$, the total number of bids, are stochastically equivalent to two independent $M/M/1$ queues with rates λ_a and μ_b , and λ_a and μ_b , respectively. Therefore, the conditions for ergodicity are given by the two following inequalities

$$\lambda_a < \mu_b \tag{3}$$

for the ask queue and

$$\lambda_b < \mu_a \quad (4)$$

for the bid queue. Moreover, the invariant measure of the process is given by two uncoupled geometric distributions

$$\mathbb{P}(A = a) = \left(\frac{\lambda_a}{\mu_b}\right)^a \left(1 - \frac{\lambda_a}{\mu_b}\right), \quad (5)$$

and

$$\mathbb{P}(B = b) = \left(\frac{\lambda_b}{\mu_a}\right)^b \left(1 - \frac{\lambda_b}{\mu_a}\right). \quad (6)$$

Formulae (5) and (6) do not take into account that orders can be cancelled. Assume that ask and bid orders are cancelled following two independent Poisson processes with rates θ_a and θ_b that are active when the number of asks or bids is non zero, respectively. Then, Formulae (5) and (6) capture the impact of cancellations if we replace μ_b with $\mu_b + \theta_b$ and μ_a with $\mu_a + \theta_a$.

Formulae (5) and (6) can be shown to hold in a much more general setting. This is outlined in the conclusions.

3 Monte Carlo Simulations

Computer simulations were run using MATLAB. Simulations were carried out in order to present and study the essential double auction trading mechanism, the dynamics of order book and the behavior of processes of interest numerically. Notice that, besides mid-price fluctuations, bid-ask spread and trade prices as a function of time, we can access many other quantities. Among these, we can mention the probabilities and expected values of the total number of ask/bid orders, the number of ask/bid orders at each price level, the positions of best ask and best bid prices, the number of ask orders conditional on the best bid and the number of bid orders conditional on the best ask. These quantities will be analyzed in detail in the future. In this paper, we decided to focus on the behaviour of tick-by-tick price differences and tick-by-tick log-returns.

The code used in this paper is given in the Appendix. The simulation setup was as follows. Prices were set in categories from 1 to $N = 500$. A trading day consisted of 10^7 events corresponding to placing one type of order at a time. Limit bid orders arrive with rate $\lambda_b = 2.4s^{-1}$ at price category $\hat{p}_a - n'$, where n' is uniformly distributed from $\{1, 2, \dots, 20\}$, i.e. $n = 20$. The problem of finding the best ask too close to the boundary, mentioned in section 2, is solved by decreasing n to a value such that the boundary is not crossed while preserving the uniform distribution. If there are no ask orders, a new bid order is placed at a category equal or slightly below the current price. Arrival of a sell market order corresponds to departure of a limit bid order at the best bid price. This occurs with rate $\mu_a = 3.0s^{-1}$, but only if

there are bid orders. If this is not the case, the market order is not executed. Since we do not assume asymmetry, rates λ_a and μ_b are equal to λ_b and μ_a , respectively. Therefore, waiting times between events within one type of orders are driven from exponential distributions with specified rates. The procedure for limit ask and buy market orders is analogous to the case of limit bid and sell market orders described above.

The initial price as well as initial best bid and best ask are assigned at the beginning of simulation and are taken from the interval of width n in the middle of the price grid, following the rule that the best bid price is always smaller than the best ask price. During time evolution, prices are registered and their changes are due to changes of best ask and best bid prices and to the arrival of market orders.

Figure 2 presents time averaged frequencies for the total number of limit ask orders, as well as for the total number of limit bid orders compared to probabilities taken from the geometric distribution of equations (5) and (6). The parameter q of the geometric distribution, which is the ratio between limit order and market order arrival rates, takes different values (0.8, 0.5 and 0.2). Apart from the expected deviations for small probabilities, the match with analytical result is almost perfect, thus confirming the correctness of the simulation presented here.

Figure 3 shows the time series of trade prices in our simulation and one can see that it resembles a Brownian motion reflected at the boundaries. However, this hypothesis is falsified by a closer look at the data.

The descriptive statistics for tick-by-tick price increments ($\Delta p_i = p_{i+1} - p_i$) and tick-by-tick log-returns ($r_i = \log(p_{i+1}/p_i)$, where i is tick index) as outcomes of the simulation are shown in table 1. One can see that both price increments and log-returns show excess kurtosis, a feature that is common in real market data and is sufficient to exclude normal returns.

	Δp	r
mean	$1.19 \cdot 10^{-4}$	$4.22 \cdot 10^{-7}$
stand. deviation	9.97	0.20
skewness	0.004	0.34
kurtosis	12.95	94.08

Table 1: Descriptive statistics of price increments and log-returns made on $5.56 \cdot 10^5$ trading prices.

The empirical autocorrelation functions (ACF) of price increments and log-returns are presented in Figure 4. They decay very fast and negative values at first lags are due to the bid-ask bounce. However, autocorrelation functions of absolute price increments and absolute log-returns persist, showing long-range memory.

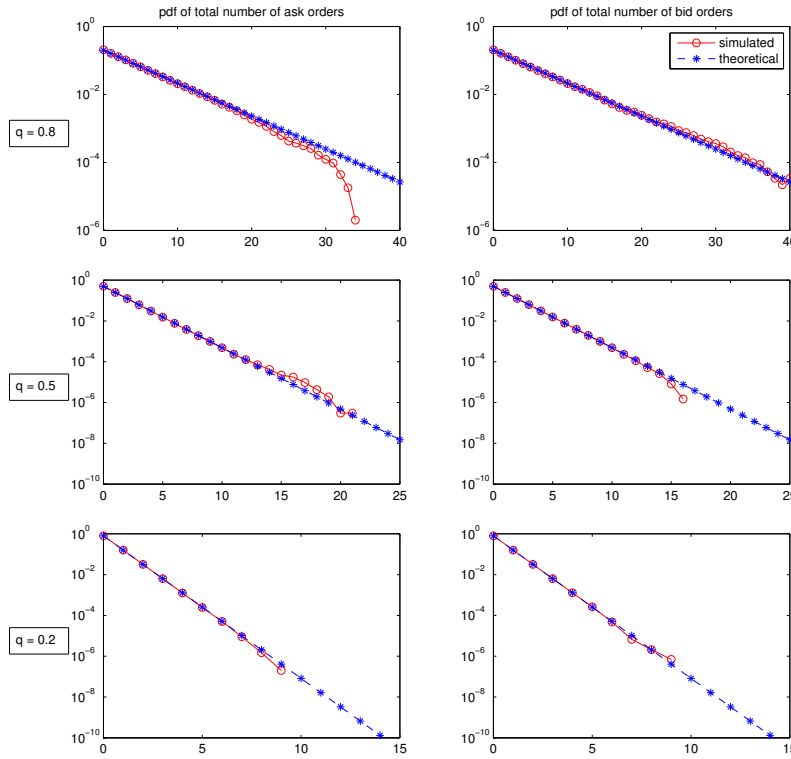


Fig. 2: Distributions of total number of ask (left) and bid (right) orders (circles), compared to the expected geometric distribution (stars) with parameter $q = 0.8$, $q = 0.5$ and $q = 0.2$, where $q = \lambda_a / \mu_b = \lambda_b / \mu_a$.

4 Summary

In this paper, we have presented a simple model for the double auction inspired by [1, 2]. The model reproduces the behaviour of zero-intelligence agent. Orders are placed at random and best bids and best asks are accepted at random as well. For what concerns the number of agents (orders), the model is equivalent to two independent $M/M/1$ queues. Some of the results discussed above are more general than it seems. For instance, this is the case of equations (5) and (6) that hold true even more generally. The coupling argument used to establish the stochastic equivalence to two independent $M/M/1$ queues does not depend on the way in which orders, upon their arrivals, distribute among prices. Therefore, (5) and (6) hold true even when orders distribute among prices according to any probabilistic or deterministic rule.

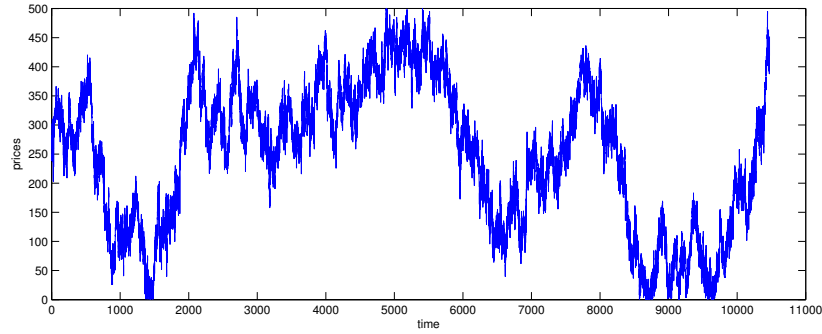


Fig. 3: Price series obtained in simulation with price grid $\{1, \dots, 500\}$, arrival rates of limit orders $\lambda_a = \lambda_b = 2.4$ and arrival rates of market orders $\mu_a = \mu_b = 3$.

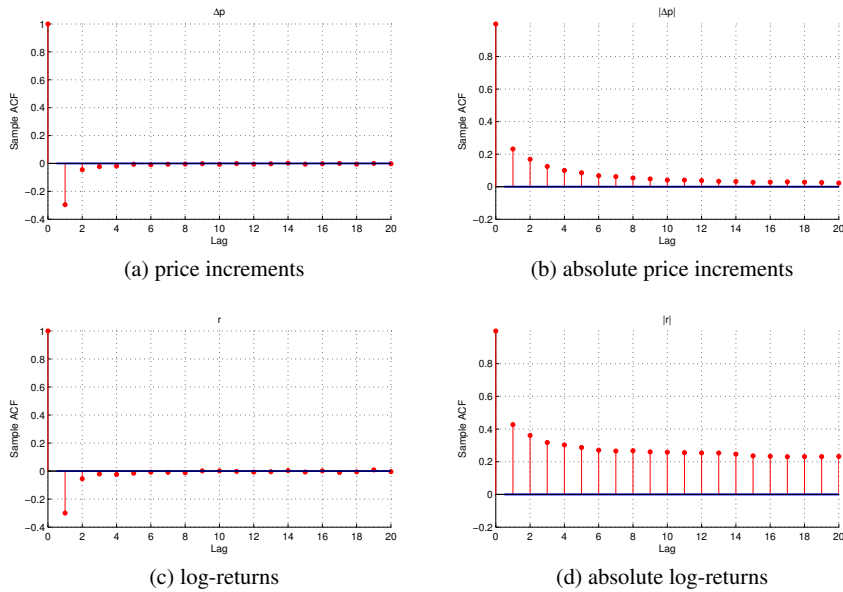


Fig. 4: Estimates of ACFs from a set of $5.56 \cdot 10^5$ trading prices.

We present the results of a Monte Carlo simulation on trade prices. Tick-by-tick price differences and tick-by-tick log-returns display excess kurtosis, microstructural noise (bid-ask bounce) and heteroscedasticity.

Acknowledgement

This paper was inspired by discussions with Mauro Politi. It was partially funded by the Italian grant PRIN 2009 2009H8WPX5_002.

References

1. Rama Cont and Adrien De Larrard (2011). *Price Dynamics in a Markovian Limit Order Market*. Working Paper Series 1735338, <http://ssrn.com/abstract=1735338>
2. Rama Cont, Sasha Stojkov and Rishi Taireja (2008). *A Stochastic Model for Order Book Dynamics*. Working Paper Series 1273160, <http://ssrn.com/abstract=1273160>
3. Ubaldo Garibaldi and Enrico Scalas (2010). *Finitary Probabilistic Methods in Econophysics*. Cambridge UK: Cambridge University Press.

5 Appendix

The MATLAB program used for generating price fluctuations and computing quantities of interest is listed below.

```
ma = 3; % market sell order rate
mb = 3; % market buy

ratio = 0.8;
lb = ratio*ma;
la = ratio*mb;

% price range
N = 500;
p_g = 1:1:N;

nevents = 10^7;

% interval for limit orders
int = 20; int1 = 2;

% state of the order book
X = zeros(N,1);

% initialization
price = floor(mean(p_g));
bestbid = price - randi(int/2);
bestask = price + randi(int/2);
P = [0 price];
midprice = (bestask + bestbid)/2;
M = [0 midprice];
```

```

X(bestbid) = X(bestbid) - 1;
X(bestask) = X(bestask) + 1;

k = 0;
T = 0;

while k < nevents
    k = k + 1

    [event, t] = order(la,lb,ma,mb); % chose type of order

    if event == 1 % limit order bid
        if bestask==0
            if price == 1
                p = price;
            elseif price <= int1 && price > 1
                p = price - (randi(price) - 1);
            else
                p = price - (randi(int1+1) - 1);
            end
        else
            if bestask < p_g(1) + int
                p = bestask - randi(bestask) + 1;
            else
                p = bestask - randi(int);
            end
        end
        X(p) = X(p) - 1; % new bid
    elseif event == 2 % limit order ask
        if bestbid==0
            if price == p_g(end)
                p = price;
            elseif price > p_g(end) - int1 && price < p_g(end)
                p = price + randi(p_g(end)-price+1) - 1;
            else
                p = price + randi(int1+1) - 1;
            end
        end
        if bestbid > p_g(end) - int
            p = bestbid + randi(p_g(end)-bestbid+1) - 1;
        else
            p = bestbid + randi(int);
        end
        X(p) = X(p) + 1; % new ask
    elseif event == 3
        if bestask~=0
            X(bestask) = X(bestask) - 1; % market buy
            price = bestask;
        end
    else
        if bestbid~=0
            X(bestbid) = X(bestbid) + 1 ; % market sell
            price = bestbid;
        end
    end
end

```

```

        end
    end

    T = T + t;

    if event==3 | event==4
        P = [P; T price];
    end

    maxnor = max(abs(X));

    % update bestask, bestbid
    bestask = find(X>0,1);
    if isempty(bestask)==1
        bestask = 0;
    end

    bestbid = find(X<0,1,'last');
    if isempty(bestbid)==1
        bestbid = 0;
    end

    midprice = (bestask + bestbid)/2;
    M = [M; T midprice];

end

```

The additional function `order` called by the main program is the following.

```

function [event, t] = order(la,lb,ma,mb)

c = la + lb + ma + mb;
x = rand(1);

if x < lb/c
    event = 1; % limit bid
    t = -log(rand(1))/c;
elseif x < (lb + la)/c
    event = 2; % limit ask
    t = -log(rand(1))/c;
elseif x < (lb+la+mb)/c
    event = 3; % market buy
    t = -log(rand(1))/c;
else
    event = 4; % market sell
    t = -log(rand(1))/c;
end

```