Reconstruction of Order Flows using Aggregated Data

Ioane Muni Toke[1]

[1]MICS Laboratory, Chair Of Quantitative Finance, CentraleSupelec, France.
[1]ERIM, University of New Caledonia, New Caledonia.
[1]CREST, Japan Science and Technology Agency, Japan.

September 7, 2016

Abstract

We investigate TRTH tick-by-tick data on three exchanges (Paris, London and Frankfurt) and on a five-year span. A simple algorithm helps the synchronization of the trades and quotes data, enhancing the basic procedure. The analysis of the performance of this algorithm turns out to be a a forensic tool assessing the quality of the database: significant technical changes affecting the exchanges are tracked through the data. Moreover, the choices made when reconstructing order flows have consequences on the quantitative models that are calibrated afterwards on such data. Finally, this order flow reconstruction provides a refined look at the Lee-Ready procedure and its optimal lags. Findings are in line with both financial reasoning and the analysis of an illustrative Poisson model.

Keywords: Limit order book ; order flow ; trades and quotes matching ; trade signature ; Lee-Ready algorithm ; model calibration.

1 Introduction

The development of high-frequency financial markets produces a huge amount of data every day. After the studies of monthly prices of wool in the first part of the 20th century (reported by Mandelbrot (1963)), the daily indicators (open, high, low, close,...) commonly used in the 1980s and 1990s, intraday data and even (very) high-frequency data is now a common object of study for quantitative finance modelling by both academics and practitioners. However, there is currently no large public diffusion of complete and transparent data by the exchanges, and to our knowledge there is no plan for such data sharing. Because they want to protect the identity of the market participants, exchanges are reluctant to release very detailed data. Even anonymous versions of such data might not be easily found, and, in any case, even less detailed/coarse/aggregated data is often available to academics at a non-negligible price.

The basic database available to most researchers is thus the tick-by-tick "trades and quotes" database, which reports the anonymous transactions and updates of aggregated quotes as they happen during the trading days. These are (supposedly exhaustive) snapshots of the state of the limit order book, but they do not detail the messages that trigger the transitions from one state to another. Researchers have therefore devised procedures and algorithms to infer some characteristics of these messages, by tackling for example the signature problem, which aims at determining if a given transaction has been initiated by the buying or the selling party. In this matter, the seminal work by Lee & Ready (1991) is still a current object of study (see e.g. Easley et al. (2016) for a very recent contribution).

Recent developments in market microstructure however focus on a more detailed point of view, in which the characteristics of each order are to be studied in order to get a better grasp of the dynamics of the limit order books. Two examples of such works include limit order book modelling (see e.g. Cont et al. (2010), Muni Toke (2015), Huang et al. (2015) among many others) or order flow modelling with Hawkes processes (see references in Section 6).

This work deals with the reconstruction of a detailed order flow from a standard "trades" and "quotes" database. This exercise would be extremely simple and straightforward if the databases were perfect, i.e. if they never missed an update, if their timestamps were accurate to an infinite precision, and if all their sources were perfectly synchronized. As this is obviously not the case, the reconstruction of the order flow needs some adapted algorithms.

In Section 2, we describe the database available, and in Section 3 we detail the difficulties encountered when trying to synchronize the "trades" and "quotes" sources, and the consequences on the matching procedure. We propose a matching procedure that significantly improves the basic reasoning. For the purpose of illustration, this section presents results on stocks traded in Paris in 2010 and 2011, but more extensive results are provided in Section 4, in which the matching procedures are tested on a 5-year time window for three different exchanges (Paris - Euronext, London and Frankfurt - Deutsche Börse). It is shown that the proposed matching significantly improves the performance of the order flow reconstruction. Very interestingly, these tests reveals that technical changes that affected these exchanges during these periods leave profound marks on the tick-by-tick data, even after several steps of transformation and formatting by the exchanges and data vendors. In Section 5, we use our order flow reconstruction to contribute to the assessment of the Lee-Ready algorithm of trade signature. Because we have exactly matched trades to their consequences in the order book, we know their "true" sign and can assess the performance of the Lee-Ready procedure. We show that one should not recommend an absolute lag value for the Lee-Ready procedure, as the optimal lag is time-dependent. Moreover, the study of the algorithm performance is informative on some events affecting the exchanges and even a high-level operation such as trade signature on aggregated data reveals technical changes affecting data production. Finally, Section 6 illustrates how the choices made when reconstructing the order flow can potentially impact the

calibration of quantitative models on the reconstructed samples, underlining the importance of the proposed procedure.

2 Description of the data

We use the Thomson-Reuters Tick History (TRTH) database in the form available at the Chair of Quantitative Finance in CentraleSupélec, Paris, France. This database gives for a large spectrum of stocks and futures all the reported trades and all the updates of the quotes of the limit order book, up to a given number of limits (usually 5 or 10 depending on the date and stock studied). The TRTH database identifies financial products by their (unique) Reuters Instrument Code (RIC). For the sake of illustration, 'LAGA.PA' denotes the stock Lagardère traded in Paris, 'BARC.L' stands for Barclays traded in London, 'FEIZ9' is the December 2009 futures on the 3-month Euribor, etc. The version hosted by CentraleSupélec spans roughly five years of data between 2008 and 2013, with several gaps.

The basic extraction[1] from the database gives us for any given product and trading day two files[2]. The first one, which will be called "trades" file, lists all the recorded trades (price and volume) with a timestamp in milliseconds. A flag field is also available, which is used for a basic filtering of the trades : all "normal" trades are kept (the ones part of the continuous double auction), while specific reports are discarded (trades flagged as "off the book"). After this filtering step a "trades" file has thus three fields : "timestamp", "price" and "quantity". The second one, which will be called "quotes" file, lists all the updates (price and volume) of the number of shares standing in the order book at the five or ten best limits around the spread. A "quotes" file has thus five fields : "timestamp", "side", "level", "price" and "quantity".

Table 1 show the structure of the "trades" file on an actual extract of the database for the stock LAGA.PA on January 28th, 2010. The "trades" file illustrated in Table 1 is straightforward to

timestamp	price	quantity
35977.101	27.54	180
35987.244	27.545	200
35987.244	27.55	47
35987.244	27.55	129
35987.247	27.55	100

Table 1: Extract from the "trades" file for the stock LAGA.PA on January 28th, 2015

read. First line of Table 1 means that at time 35977.101 (in seconds, i.e. 9:59:37.101 a.m.), there has been a transaction of 180 shares at price 27.54 EUR. In other words, each line of the "trades"

---

[1]Basic extraction is done by a BNP Paribas-developed script which we do not control. This step does not alter the data, it only translates data from a binary format to a more readable compressed csv format.

[2]Some files can be made available to academic researchers upon request.

file can be translated into a market order that triggers a transaction. However, the "trades" file gives explicitly the price and the volume of a given market order, but does not say if it was a buy or a sell market order: we do not know the *aggressor* in microstructure terms, i.e. the last party to enter the deal and to therefore trigger a trade. This will be addressed in Section 5.

Table 2 show the structure of the "quotes" file on an actual extract of the database for the stock LAGA.PA on January 28th, 2010. The penultimate line of Table 2 means, by comparing

timestamp	side	level	price	quantity
36003.97	A	3	27.585	697
36003.97	A	3	27.585	177
36004.067	A	3	27.59	391
36004.067	A	4	27.595	311
36004.067	A	5	27.6	427
36004.067	A	6	27.605	1688
36004.067	A	7	27.61	586
36004.067	A	8	27.615	677
36004.067	A	9	27.62	1999
36004.067	A	10	27.625	568
36004.613	A	6	27.605	2315
36005.314	B	1	27.55	6829

Table 2: Extract from the "quotes" file for the stock LAGA.PA on January 28th, 2015

it to the sixth line from the top, that there has been a increase of the offered volume (new limit order) at time 10:00:04.613 at price 27.605 EUR of size 2315-1688=627. The second line of Table 2 means, by comparing to the first line of the extract, that there has been a decrease of the offered volume (cancellation of a standing limit order) at time 10:00:03.97 at price 27.585 EUR of size 697-177=520.

In short, our data is tick-by-tick data in the sense that all event modifying the order book are supposed to be individually reported, and aggregated in the sense that we do not have access to the granularity of the messages/orders but to the aggregated volume of the sizes of the orders standing in the limit order book.

3 Reconstruction of the order flow

Our goal is to reconstruct the precise *order flow*, i.e. the complete sequence of market orders, limit orders and cancellations along with their price and volume that lead to the generation of given "trades" and "quotes" files. The general algorithm for the determination of the order flow would be straightforward if the database were perfectly coherent, but inconsistencies exist, mainly because of the non-synchronization of the timestamps of the trades and quotes files, and maybe because of probable inaccuracies in the data building process.

3.1 Sample description

For the purpose of illustration while presenting the construction of the algorithm, we randomly select on our database seven liquid stocks from the CAC 40 (Air Liquide AIRP.PA, Alstom ALSO.PA, BNP Paribas BNPP.PA, Bouygues BOUY.PA, Carrefour CARR.PA, Electricité de France EDF.PA, Lagardère LAGA.PA). For each stock we extract from the database the "quotes" and "trades" files as described above from January 17th, 2011. Please note that this sample is selected for illustration only, and that Section 4 will provide more detailed results, for more stocks and an extensive range of dates.

3.2 Parsing the "quotes" file

We start with an empty structure of $2N$ pairs "price"-"quantity" representing the state of the order book, i.e. the number of share available at the N best bid and ask, and their price. ($N = 5$ or 10 limits are usually available, depending on the RIC and trading day). We scan line by line the "quotes" file. The reading of the first lines allow us to initialize the state of the order book. Each following line is an update of a given volume and price: in a first approach, each line is thus a new limit order (increase of the number of shares available at a given price), or a cancellation of an existing limit order (decrease of the number of shares available at a given price), as in the two examples provided with Table 2.

When parsing the "quotes" file, one has to be cautious of "shifts". For example, if a limit order is submitted within the spread or at a price within the current N best quotes where no previous limit order were standing, then it creates a new level, and all subsequent levels are re-numbered. This gives multiple updates messages in the "quotes" file that are not actual orders. An example is provided in Table 3. The first ten lines describe the current state of the order book. The eleventh line shows that a new ask limit order with size 66 is submitted inside the spread at price 27.52 EUR. Then the nine following lines do *not* describe any limit order, but simply update the level numbering (the new level 2 is the level 1, and so on). When parsing these lines, the available volume at level n is to be compared to the previous volume at level $n - 1$, in order to track potential changes.

Since we observe a fixed finite number of prices N (usually equal to 10), shifts may make some information disappear: in table 3 for example, the fact that 1638 shares are available at price 27.59 on the ask side is lost at the update because of a left shift of this side of the book. By a reciprocal mechanism, a right shift of the ask side (or a left shift of the bid side) may bring new information to light, that is *not* to be interpreted as a new order. In table 4 below for example, the last line stating that 598 shares are available at 27.095 on the tenth limit of the bid is not a new limit order — despite being a new information — because this price was not monitored before.

When the whole "quotes" file is parsed, we have a sequence of limit orders and cancellations — no market orders yet — with their timestamp to the millisecond, side, level, price and volume.

timestamp	side	level	price	quantity
34819.37	A	1	27.54	326
34819.37	A	2	27.545	530
34819.37	A	3	27.55	989
34819.37	A	4	27.555	318
34819.37	A	5	27.56	79
34819.37	A	6	27.565	275
34819.37	A	7	27.57	468
34819.37	A	8	27.58	100
34819.37	A	9	27.585	612
34819.37	A	10	27.59	1638
34819.37	A	1	27.52	66
34819.37	A	2	27.54	326
34819.37	A	3	27.545	530
34819.37	A	4	27.55	989
34819.37	A	5	27.555	318
34819.37	A	6	27.56	79
34819.37	A	7	27.565	275
34819.37	A	8	27.57	468
34819.37	A	9	27.58	100
34819.37	A	10	27.585	612

Table 3: "Shift" of quotes due to a limit submitted at a price where no previous limit order was standing, inside the spread in this example. Extract from the "quotes" file for LAGA.PA on January 28th, 2010.

3.3 Parsing the "trades" file in the perfect case

The second step of the order flow reconstruction deals with the parsing of the "trades" file. If everything were perfectly recorded, then each line of the "trades" file should represent a market order, and at the exact same timestamp, we should see in the "quotes" file a decrease of the quantity at the same price and with the same volume. In other words, each line of the "trades" file should match a cancel order of the same volume, at the same price and timestamp in the order flow resulting from the parsing of the "quotes" file. An example of this perfect case is documented in Table 4. After parsing the extract of the "quotes" file in Table 4, we would have the following

timestamp	price	quantity
32472.252	27.32	267

timestamp	side	level	price	quantity
32472.086	B	1	27.32	267
32472.086	B	2	27.31	500
32472.086	B	3	27.29	585
32472.086	B	4	27.285	127
32472.086	B	5	27.27	500
32472.086	B	6	27.2	300
32472.086	B	7	27.16	500
32472.086	B	8	27.155	200
32472.086	B	9	27.15	1750
32472.086	B	10	27.1	223
32472.086	B	2	27.31	710
32472.252	B	1	27.31	710
32472.252	B	2	27.29	585
32472.252	B	3	27.285	127
32472.252	B	4	27.27	500
32472.252	B	5	27.2	300
32472.252	B	6	27.16	500
32472.252	B	7	27.155	200
32472.252	B	8	27.15	1750
32472.252	B	9	27.1	223
32472.252	B	10	27.095	598

Table 4: Perfect match between the "trades" and "quotes" files. The line of the "trades" extract (top table) is perfectly matched with the same timestamp in the extract of the "quotes" file (bottom table). Stock LAGA.PA on January 28th, 2010.

order flow :

32472.086	LIMIT	B	2	27.31	210
32472.252	CANCEL	B	1	27.32	267

But after parsing the "trades" file, we conclude that the update at time 32472.252 is not a cancellation, but the result of a market order. We thus change the order flow to :

32472.086	LIMIT	B	2	27.31	210
32472.252	MARKET	B	1	27.32	267

3.4   Basic approach of trade matching in the general case

The perfect case described above is however extremely rarely observed. On our 7-stock 1-day illustrative sample, the perfect case occurs at a frequency lower than 0.001 (but one should refrain from generalizing this figure, as Section 4 will show that matching results are strongly exchange and time dependent). The cause of this situation is that the clock used to timestamp the "trades" and "quotes" files are not synchronised. Therefore, it is very rare to exactly match a corresponding market order read in the "trades" file to a cancellation read in the "quotes" file. Most of the time, the transaction is recorded in the "trades" file and the updates are published in the "quotes" file with a few milliseconds lag.

Therefore in a first approach of trade matching, when we parse the "trades" file and look for a matching cancellation in our order flow constructed from the "quotes" file, with the exact same price and volume, we look not only at the exact transaction timestamp, but within $-\delta$ and $+\delta$ seconds around this transaction timestamp. In the event of multiple matches possible, we choose the one which minimizes the timestamp difference. This procedure will called "Matching 1" from now on.

Figure 1 plots the proportion of trade left unmatched after the Matching 1 procedure has been applied to the sample. It appears that the performance of the matching is similar for all stocks. As expected, when δ increases, the matching performance improves, but unexpectedly it reaches a plateau for all stocks for $\delta \approx 150 - 200$ ms. Beyond this value, increasing δ has no effect of the performance, and the Matching 1 procedure leaves unmatched between roughly $5 - 20\%$ of the trades reported in the trades file, in this illustrative sample.

For a more precise view on this phenomenon, we plot in Figure 2 the empirical distribution of the measured time lags $\Delta\tau = \tau_q - \tau_t$ where τ_t is the time stamp of a trade in the "trades" file, and τ_q the time stamp of its matched modification in the "quotes" file. This distribution has a mainly positive support, indicating that the updates of the "quotes" files are most of the time following the report of the transaction in the "trades" file. Furthermore, all the empirical densities drop to 0 around 150 ms, indicating that it is rare to match an order more that 150 ms later in the "quotes" file.

3.5   Enhanced approach of trade matching

The previous matching procedure leaves quite a large number of unmatched transactions. It is indeed quite frequent that consecutive lines in the "trades" file lead to only one update in the
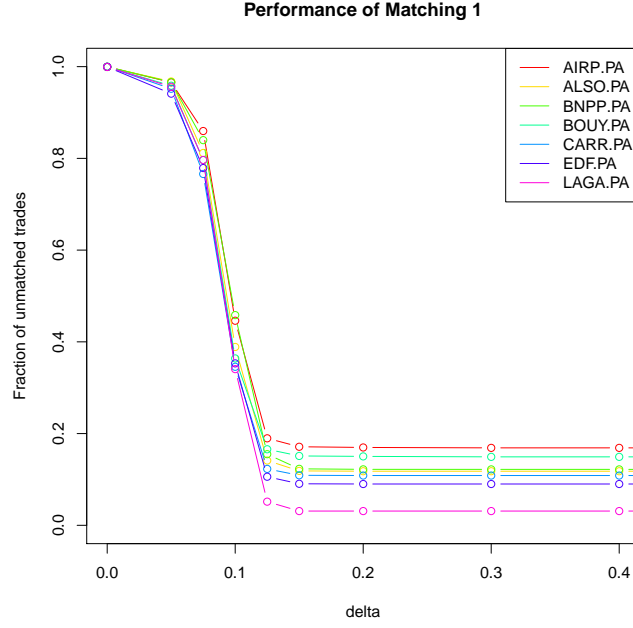
Performance of Matching 1

Figure 1: Proportion of trades left unmatched by Matching 1. All trades reported in the database on Jan. 17th, 2011 between 09:05 and 17:20 are considered for matching.
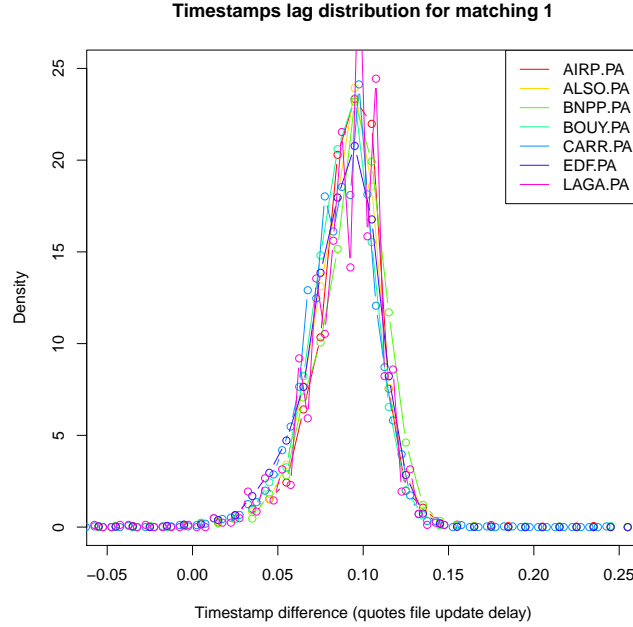

Timestamps lag distribution for matching 1

Figure 2: Empirical density of the time lag $\Delta\tau$ between a transaction recorded in the "trades" file and its matching cancellation in the "quotes" file with the "Matching 1" procedure with $\delta = 0.4$.

"quotes" files. The common explanation for this phenomenon is that the exchange issues one message for each pending limit order that is (partially) matched by an incoming market order. In this case, one has to aggregate consecutive lines of the "trades" file at the same price to find the corresponding cancellation in the "quotes" file. This is illustrated in Table 5. The result of the

timestamp	price	quantity
32951.412	27.45	300
32951.412	27.45	182

timestamp	side	level	price	quantity
32951.419	A	1	27.45	482
32951.419	A	2	27.455	730
32951.419	A	3	27.465	200
32951.419	A	4	27.47	200
32951.419	A	5	27.475	200
32951.419	A	6	27.48	279
32951.419	A	7	27.485	1813
32951.419	A	8	27.495	529
32951.419	A	9	27.5	3030
32951.419	A	10	27.505	200
32951.419	A	1	27.455	730

Table 5: Aggregation of two lines of the "trades" file (top table) to match the updates of the "quotes" file (bottom table). Stock LAGA.PA on January 28th, 2010

parsing of the extract of the "quotes" file in Table 5 is the cancel order (CANCEL, 32951.419, A, 1, 27.45, 482). After parsing the "quotes" file, we conclude that it is in fact the recording with a 7 milliseconds delay of a market order that was recorded in the "trades" file as two transactions.

The difficulty is that there is no simple rule to make this procedure automatic, since all market orders at the same timestamp do not necessarily have to be aggregated to match the "quotes" file. In order to tackle these difficulties, we enhance the matching procedure: when parsing the "trades" file, we group consecutive lines (transactions) with the same price and the same timestamp (for now ; see below) into a batch. Then for each constructed batch, we test the matching of all its possible divisions into partitions of consecutive lines, and keep as our solution the matching that matches the highest number of lines of the original "trades" file. For example, when parsing a "trades" file, if we find three transactions at the same timestamp, as follows (synthetic example, not real data):

timestamp	price	quantity
36000.000	20.00	100
36000.000	20.00	50
36000.000	20.00	200

then in addition to the matching of these three lines consecutively, we will also test the matching
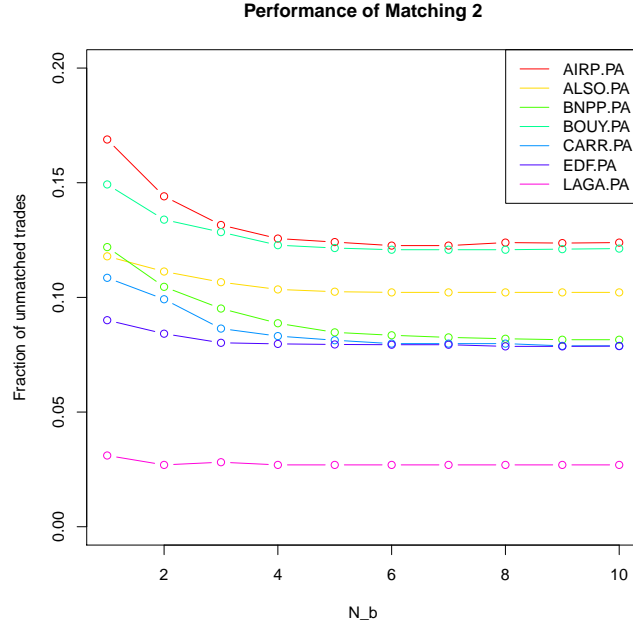
Performance of Matching 2

Figure 3: Proportion of trades left unmatched by Matching 2. All trades reported in the database on Jan. 17th, 2011 between 09:05 and 17:20 are considered for matching. In this example, $\delta = 0.4$.

the following of the arrangements of the transactions :

timestamp	price	quantity
36000.000	20.00	150
36000.000	20.00	200

timestamp	price	quantity
36000.000	20.00	100
36000.000	20.00	250

timestamp	price	quantity
36000.000	20.00	350

To illustrate the performance increase due to this enhancement, Figure 3 plots the fraction of trades left unmatched by this Matching 2 procedure as a function of N_b, the maximum size allowed for a single batch. When $N_b = 1$ (first point on the left), the matching is identical to the Matching 1 procedure ($\delta = 0.4$ in the example plotted). As expected, when N_b increases, the matching is improved : on our sample, a minimum of 12% (LAGA.PA) and up to 33% (BNPP.PA) of previously unmatched trades are now matched to the limit order book, which for BNPP.PA or

AIRP.PA represents more than 4% of the total transactions. Note however that on this sample increasing the size of the batch above 5 or 6 lines does not have any significant impact on the performance.

Among the orders that are still left unmatched by Matching 2, another difficulty is identified by a careful observation of data files : market orders that do not have exactly the same timestamp sometimes have to be aggregated to match the "quotes" file. This is illustrated in Table 6. In

timestamp	price	quantity
33095.296	27.51	100
33095.296	27.51	202
33095.296	27.51	303
33095.296	27.51	486
33095.296	27.51	334
33095.296	27.535	210
33095.299	27.535	140

timestamp	side	level	price	quantity
33046.84	A	1	27.535	3951
		[. . .]		
33085.836	B	1	27.51	2264
33095.297	B	1	27.51	839
33095.31	A	1	27.535	3601

Table 6: Aggregation of lines of the "trades" file with different time stamps (top table) to match the updates of the "quotes" file (bottom table). Stock LAGA.PA on January 28th, 2010.

Table 6, the five first line of the extract of the "trades" file have to be aggregated to match the order (33095.297, CANCEL, B, 1, 27.51, 1425), with a 1 millisecond time lag. But the last two orders, although with two different timestamps separated with 3 milliseconds have to be aggregated to match the order (CANCEL, A, A, 27.535, 350), reported with a 11 milliseconds delay from one order, and 14 milliseconds delay from the other.

In order to track these special cases, we enhance the Matching 2 procedure described above by grouping into the same batch lines that have a time stamp *close* to the first time stamp read, but not necessarily equal. The parameter measuring the maximum allowed time lag of a single batch will be denoted δ_b. To illustrate the performance increase due to this enhancement, Figure 4 plots the fraction of trades left unmatched by this Matching 3 procedure as a function of δ_b. In this sample it appears that the transactions that are to be aggregated have a very close time stamp, and that the typical time lag to be considered is 2-3 ms. For higher values of δ_b no matching improvements is observed. The improvement of Matching 3 allows to match between 3% (ALSO.PA, BOUY.PA, EDF.PA) to more that 10% of the previously unmatched trades (CARR.PA, BNPP.PA), except for LAGA.PA whose performance is not improved. However, this improvements represent in any case less than 1% of the total transactions observed.
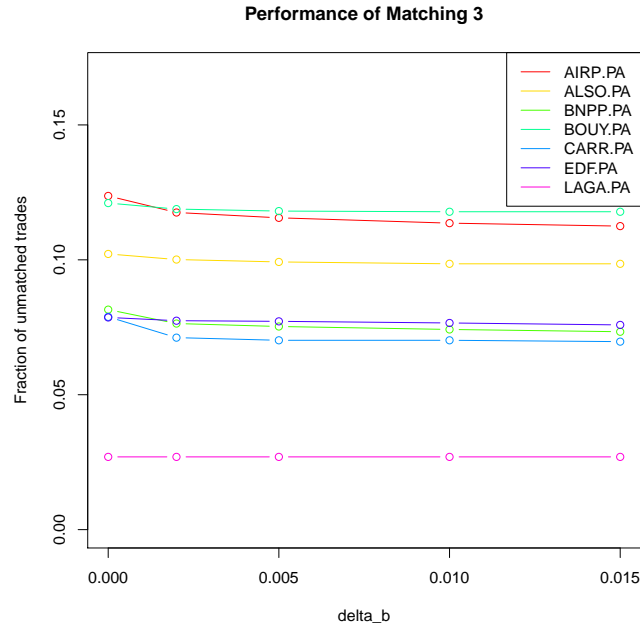
Figure 4: Proportion of trades left unmatched by Matching 3. All trades reported in the database on Jan. 17th, 2011 between 09:05 and 17:20 are considered for matching. In this example, $\delta = 0.4, N_b = 9$.

Table 7 summarizes the performances of the matching procedures described above. In short, on

	Matching 1	Matching 2	Matching 3
AIRP.PA	83.11%	87.63%	88.75%
ALSO.PA	88.21%	89.78%	90.15%
BNPP.PA	87.81%	91.85%	92.67%
BOUY.PA	85.07%	87.90%	88.22%
CARR.PA	89.15%	92.11%	93.03%
EDF.PA	90.99%	92.14%	92.41%
LAGA.PA	96.89%	97.30%	97.30%

Table 7: Percentages of transactions for the "trades" file matched by the different matching procedures described in the text

this 7-stock sample, our detailed algorithm allows in average the matching of roughly 92% of the trades reported in the "trades" file, which is an average improvement of 3% of the total transactions from the standard Matching 1 procedure. It is important to keep in mind that this small 1-day sample and the associated results are only illustrative of the method, detailed here to make the matching procedure easy to understand. Detailed results are provided in Section 4 et sq., showing that for some stocks and dates the enhanced matching (Matching 3) is crucial.

3.6   Sources of unmatched trades

Several other difficulties might be treated by further enhancing the above matching procedure, but at probable high cost with marginal enhancement. A manual observation of the files show that some of the orders that remain unmatched truly appear "out of bound", with quantities or prices that are not at all observed in the limit order book. For many unmatched orders however, only a small volume discrepancy prevents a matching: even if the price is equal to the best bid and we observe cancellations at the best bid, we cannot exactly equate their volumes, no matter how we aggregate the information. In this case, no exact matching can be obtained.

We can only speculate about the sources of these roughly 10% of trades in average that are not matched. The most obvious explanation would be that the reported transactions are executed against hidden liquidity, which is allowed by many exchanges and on Euronext Paris takes the form of "iceberg" orders. Only a fraction of the size of an iceberg order is displayed on the order book (for example assume 100 shares are displayed alone at the best quote out of a full size of 1000). When a market order occurs and eats part of this liquidity (let us say 50 shares), the quotes are updated as usual (and we observe a decrease of 50 shares, which is easily matched). But when a market order eats more than the displayed quantity (let us say 125 shares), then it is executed against the first 100 shares, which are immediately replaced by 100 previously hidden shares, and the last 25 shares are executed against the newly displayed quantity. If there is one update for this single market
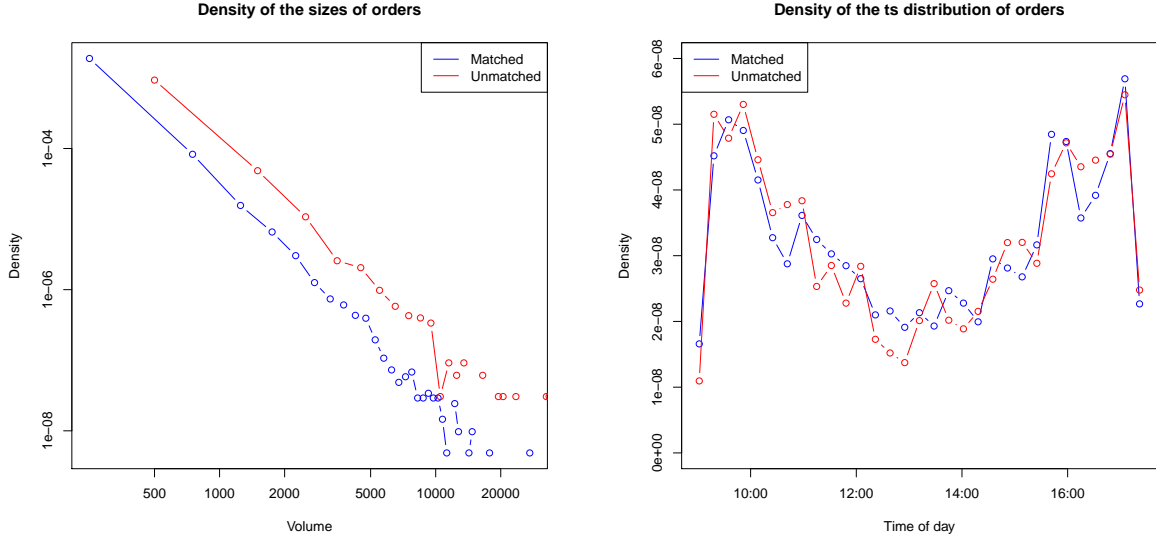
Figure 5: Empirical density distributions of the sizes (left, in log-log scale) and the timestamps (right) of the unmatched trades, compared to the matched ones. In this example, Matching 3 is used for the stock BNPP.PA on the whole month of January 2011.

order, then the final observation is an decrease of 25 shares. That would explain the fact precise quantities fail to match despite the observations of trades and corresponding cancellations at the same price. Another explanation would be that some lines report transactions that occurred outside of the limit order book and therefore they have no effect on the "quotes" file. The inaccuracies here would be in the "trades" files, since such trades are supposed to be flagged as "off the book".

For the sake of completeness, Figure 5 plots the empirical density distributions of the sizes and the timestamps of the unmatched trades, which are the only two information we have on these transactions. On these figures we observe that unmatched trades occur all day, and that the density of their timestamp is similar to the one of the unmatched trades, following a well-known U-shaped pattern indicating a daily seasonality. Therefore we cannot link unmatched trades to specific periods of the day.

However, we observe that the sizes of the unmatched orders are significantly larger than the sizes of matched orders (a Kolmogorov-Smirnov test confirms this observation). This may be in line with both previous assumptions. As for hidden liquidity, large market orders are more likely to fully eat the displayed chunk of an iceberg order, and therefore trigger liquidity that prevents the exact size matching. This is also coherent with "off the book transactions", since one reason to trade outside the book might be to trade larger volumes. Finally, this finding about size might suggest that these orders could eventually be matched by the aggregation of several quotes modifications. A manual parsing of the files does not reveal systematic examples of this behaviour, but this eventuality
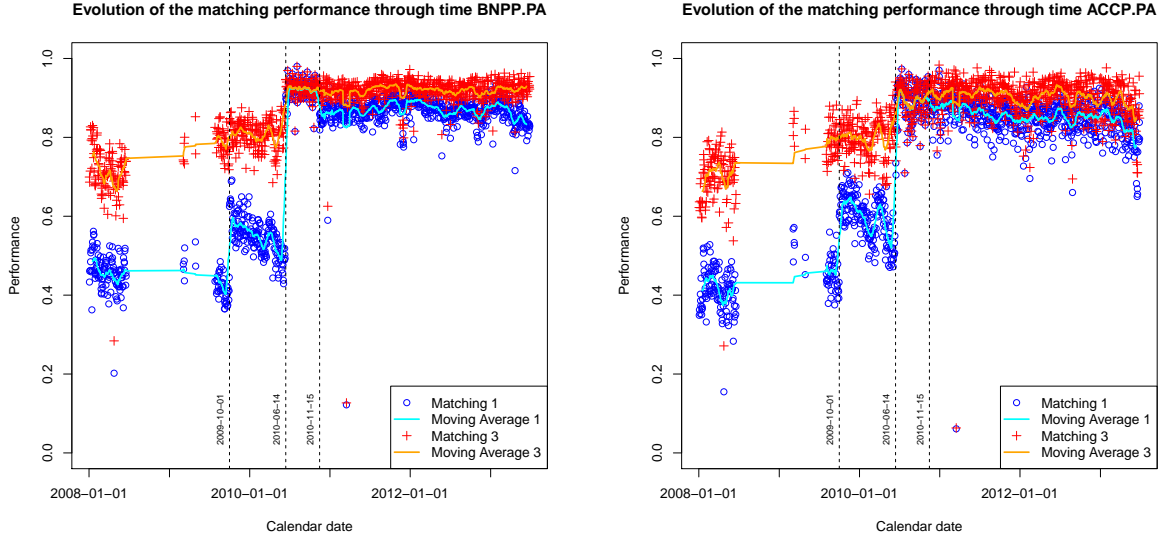
15

Figure 6: Performance of the basic and enhanced matching algorithms for the stock BNPP.PA (left) and ACCP.PA (right) from Jan, 1st, 2008 to June, 30th, 2013, with gaps.

should not be completely ruled out yet.

4 Empirical results for order flow reconstruction

In this section we present extensive results of our matching algorithm for multiple stocks and trading days.

4.1 Evolution of the performance through time – Data quality

In this section we test our algorithms on several European exchanges (Euronext - Paris Stock Exchange, London Stock Exchange, Deutsche Börse - Frankfurt Stock Exchange) on the whole range available in our database, from January 2008 to June 2013, with a large gap between June 2008 and March 2009.

Results for the stocks BNP.PA (BNP Paribas) and ACCP.PA (Accor Group), both components of the CAC 40, are presented for the basic matching (Matching 1, $\delta = 0.4$) and the enhanced matching (Matching 3, $\delta = 0.4$, $\delta_b = 0.005$, $N_b = 9$) on Figure 6. The first general observation is that on the most part of the 5-year span of the database, the improvement due to the enhanced matching procedure is very significant. Four global periods are easily identified. From Wednesday January 2nd, 2008 to Tuesday September 30th, 2009 (included), the basic matching can place roughly 45% of market orders in the book, while the performance of the enhanced matching is roughly 70%[3]. Note

---

[3]These performances might be improved by tuning the parameters, but this is not the aim of this section.

16

that our database has a large gap of missing data in this period. On Wednesday October, 1st, 2009, the performance of the basic matching suddenly jumps to roughly 55% and keeps this level until Friday June 12th, 2010 (included), while in this period the performance of the enhanced matching stays constant. On Monday June 14th, 2010, the performances of both matching algorithms jump to roughly 90%, and keep this level until Friday November 12th, 2010. During this five months, the performance of the basic algorithm is very close to the performance of the enhanced one. On Monday November 15th, 2010, however, the performance of the basic matching drops to roughly 85% while our enhanced algorithm remains at roughly 90%. This observation remains valid until the end of the sample (June 2013).

It is very interesting to observe that all the three above dates are also ruptures for the stock ACCP.PA. They are therefore to be interpreted as exchange-related events. On October 1st, 2009, Euronext made its "Universal Trading Platform - Market Data" its primary data feed for European markets[4]. We may conjecture that this change has been translated (through the data vendor building the database) into a significant improvement for the database. On June 14th, 2010, Euronext significantly improved the way it disseminates the best bid and offer quotes through its Universal Trading Platform[5], there again we may conjecture that this is the cause of a significant improvement of the quality of the historical tick data. Finally, a precise cause for the observed drop for Matching 1 on November 15th, 2010 is less clear to conjecture, but more elements will be provided in Section 5.

On Figure 7, we present a similar plot for two others exchanges, the London Stock Exchange (stock PSON.L, Pearson Group, component of the Footsie) and the Frankfurt Stock Exchange - Deutsche Börse (stock BMWG.DE, BMW AG, component of the DAX). Several observations can be made. Firstly, these plots also display obvious ruptures in the matching process, i.e. in the quality of the database. For PSON.L, the basic matching with $\delta = 0.4$ is nearly perfect for the major part of the database, starting March 2009. In 2008 however, this basic matching barely matches 1/3 of the orders even with larger parameters ($\delta = 1$). This demonstrates the usefulness of our algorithm, even for exchanges it was not designed for, since our enhanced matching is able to match roughly $90 - 95\%$ of the trades even in 2008. Due to a gap in our data, we are not able to exactly pinpoint the date of the quality change of the database, although we can conjecture, following the Euronext case above, that the "Market Data Optimisation 2009" project, that was designed in particular to improve data feeds and was precisely implemented during this period[6], is a good candidate to explain this shift in performances.

As for the stock BMWG.DE, in opposition to the previous exchanges and very counter-intuitively, we observe a sharp degradation of the matching performances through time. Until April 1st, 2010, we observe matching performances that compare to the previous exchanges, using the same param-

---

[4]Euronext Info-flash dated September 30th, 2008
[5]Euronext Info-flash dated May 18th, 2010
[6]London Stock Exchange Service Annoucements 125/08 and 130/08, December 2008.

Evolution of the matching performance through time PSON.L   **Evolution of the matching performance through time BMWG.DE**
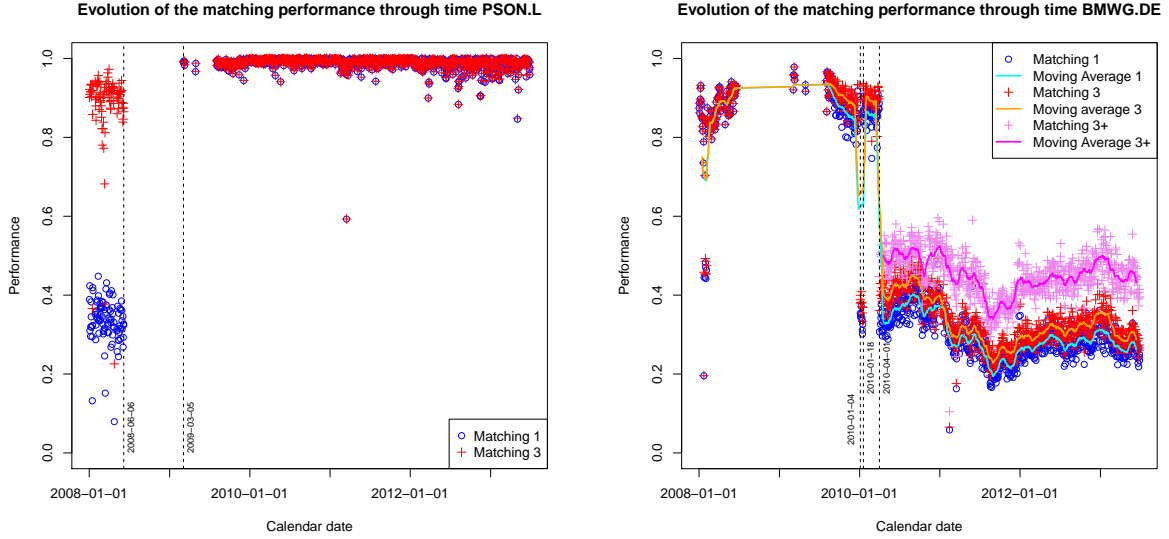
Figure 7: Performance of the basic and enhanced matching algorithms for the stock PSON.L (left) and BMWG.DE (right) from Jan, 1st, 2008 to June, 30th, 2013, with gaps. For PSON.L, $\delta = 1.0$ in 2008, 0.4 from 2009.

eters used for the other stocks : both the basic and enhanced procedures are able to exactly match around 90% of the transactions, the enhanced matching appearing slightly better starting from 2009. But on April 1st, 2010, the performance of the basic procedure suddenly drops to roughly 30% of the trades. The enhanced procedure is still better, but even with bigger parameters (Matching 3+ uses $\delta_b = 1.0$ second and $\delta = 2.0$ seconds) its performance stays in the order of magnitude of 50%. The same drop had been observed during the first two weeks of 2010. It is interesting to note that April 1st, 2010 is precisely the date Deutsche Börse implemented significant changes in its data transmission, restricted the access to un-netted market data to its "Xetra Enhanced Broadcast Solution"[7]. We can conjecture that these changes were the probable cause of the shift we observe on the quality of the database of our data vendor. Interestingly enough, the change was initially planned on January 4th[8], which reinforces our conjecture.

All in all, these contributions will hopefully be useful for researchers and practitioners dealing with the TRTH database. For example, when using TRTH data on the Paris stock exchange, one should probably not mix 2009 and 2010 data on a study that needs a precise identification of trades.

---

[7]Xetra Circular 151/09 dated December 2nd, 2009.

[8]Xetra Circular 151/09 dated December 2nd, 2009.

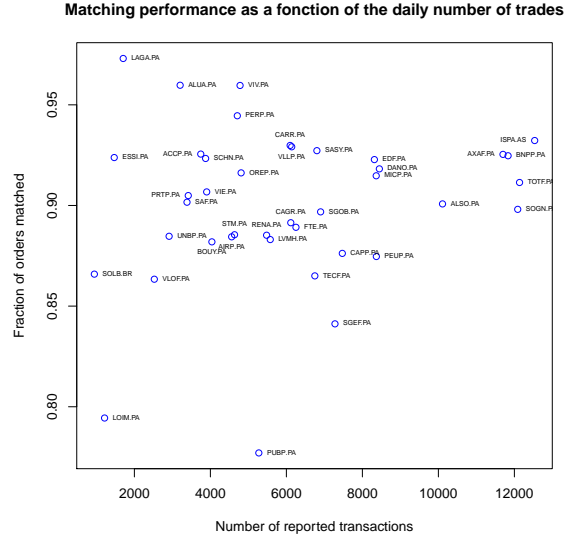**Matching performance as a fonction of the daily number of trades**

Figure 8: Proportion of matched trades as a function of the total number of reported transactions. All trades reported in the database on Jan. 17th, 2011 between 09:05 and 17:20 are considered for matching. In this example, $\delta = 0.4, N_b = 9, \delta_b = 0.005$.

4.2 Influence of the liquidity (number of transactions)

We close this section by testing the intuitive idea linking the quality of the matching to the number of trades on the studied stock: intuitively, for heavily traded stocks, overloads of the reporting system might occur, leading to potentially more inaccuracies in the "quotes" and "trades" files, so that the matching performance should decrease.

A first rough cross-stocks look at a macro level does not reveal any obvious illustration of this fact: Figure 8 plots the matching performance as a function of the number of trades reported during one trading day for 40 different stocks mostly trade on Euronext, this sample being very close, but not identical, to the composition of the CAC 40 index. Names of the stocks are provided on the plot. This scatter plot provides no obvious evidence that less traded stocks have better trades matching rates.

We can however provide a refined look at this problem by looking at matching rates for a single stock (BNPP.PA) with the following procedure : we divide the sample (full sample restricted to data post January 1st, 2011, to avoid the exchange effects observed in Section 4.1) into five-minute sub-samples, and compute the matching performance on each sub-sample. We then group the sub-samples according to the decile of the number of trades, and draw the box plots of the performance distributions for each decile on Figure 9. At this micro level, Figure 9 clearly shows a decreasing performance as the number of reported trades increases. A two-sided Kolmogorov-Smirnov test indeed rejects the null hypothesis that the empirical cumulative distribution function
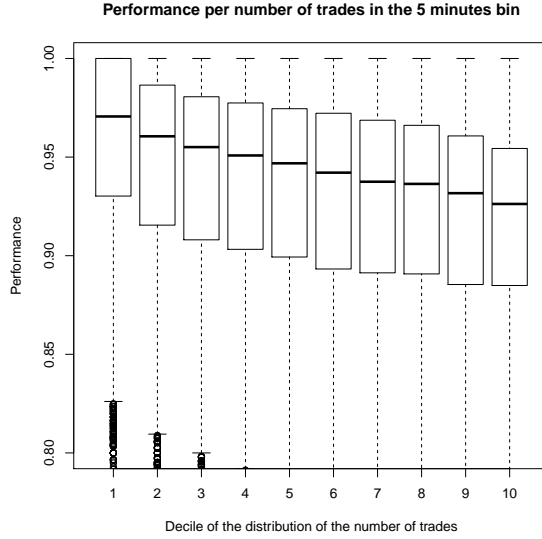
Performance per number of trades in the 5 minutes bin

Figure 9: Box plots of the empirical distribution of the matching performance as a function of the number of trades (grouped per decile, x-axis) for the stock BNPP.PA on five-minute sub-samples.

of the performance on the least active five-minute bins is equal to the one on the most active five-minute bins (p-value $< 2.2 \times 10^{-16}$). On the contrary, a one-sided Kolmogorov-Smirnov leads to the alternative hypothesis that the performance on the most active bins is significantly smaller than the one on the least active bins (p-value $< 2.2 \times 10^{-16}$). Using daily performances instead of five-minutes bins does not give an obvious decrease in the boxplots, but similar results are obtained with the Kolmogorov-Smirnov tests, although with higher p-values (1.5% and 1.2% respectively).

All in all, the naive intuition that a very high trading activity might lessen the performance of the matching algorithm is confirmed. Furthermore, this observation is stronger at a micro (minutes) level than at a daily level, which aggregates periods of different trading activities.

5 Assessment of the performance of trade signature

5.1 The Lee-Ready procedure

Signing a trade consists in assigning to each recorded transaction a "buy" or "sell" label, depending on the side of the market order that initiated this transaction. A "buy" market order is matched to liquidity (pending limit orders) on the ask side of the order book, while a "sell" market order is matched to liquidity (pending limit orders) on the bid side of the order book. One of the first notable algorithm for trade signing was proposed by Lee & Ready (1991). The motivation for this work at that time was to improve the standard signing techniques that roughly fell into two categories : the tick test, which decides whether a trade is buy or sell by comparison to adjacent

trades (i.e. without using any quotes data); and the comparison to the current quote. Lee & Ready (1991) observe that the current quote is often not synchronized to the trades, so they suggest based on this empirical observation that if a quote is less that five seconds old, the trade price should be compared to the quote 5 seconds before. This algorithm has encountered a certain success, being regularly used or revisited in the financial literature. Among others, Ellis et al. (2000) reports of roughly 80% of trades correctly classified by the Lee-Ready algorithm using 1996-1997 NYSE data (the "true" sign is determined by using several data indicators, including the identity of the trader), Theissen (2001) reports a success rate of roughly 70% using 1996 data on German DAX and MDAX stocks (the "true" sign is determined using market maker information). More recently, Chakrabarty et al. (2012) assess the performance of the Lee-Ready procedure by comparing its results to a sign provided by the exchange (i.e. not using quotes data) and find on 2005 data a misclassification by the Lee-Ready procedure of the order of magnitude of 20 to 30% (similar to Omrane & Welch (2016) on 2005 data as well). Please note these few references have been selected just as an illustration with various data (from 1996 to 2005), references in each of them should be checked for further details.

By parsing the "trades" file and matching it to the "quotes" file, our reconstruction of the order flow provides the signature of the trades as a by-product, so that we are in a position to contribute to the assessment of the Lee-Ready signing procedure. Of course, the five seconds delay recommended in 1991 is probably not relevant anymore more than twenty years later in a context of high-frequency data, but the idea behind the procedure — finding the optimal lag/quote for comparison — is still valid (see e.g. Piwowar & Wei (2006) using 1990s data or Chakrabarty et al. (2012) using 2005 data). Therefore, we implement the Lee-Ready procedure as follows : for a trade with transaction time τ_t, retrieve the best bid and ask quotes at time $\tau_q = \tau_t + \delta_{LR}$; if the trade price is strictly greater (resp. lower) than the mid-price, classify the trade as "buy" (resp. sell). If the trade is equal to the mid-price, we left its sign undecided. Of course, we could implement a tick test to decide this last case, and obtain the "full" Lee-Ready algorithm, but we coherently restrict ourselves to a quote-only procedure.

5.2 Lee-Ready performance as a function of the lag

We now compare the results of the Lee-Ready procedure to the result of our own matching procedure. Our signing is considered here the "reference" signing, since it fully uses the "quotes" data, including available liquidity volumes. Orders that are not signed by our procedure are excluded from the comparison. Obviously, and even if the use of all volume data makes our matching a very strict procedure, our reference sign may not be the "true" sign (available on detailed order data which we do not have in this case). But the interesting point is that even with this approximation, the comparison with the Lee-Ready procedure gives results in line with previous results available in the literature, and our millisecond approach allows us to provide a new refined view of the optimal

lag problem.

Figure 10 plots the fraction of the correctly signed orders by the Lee-Ready procedure as a function of the lag parameter δ_{LR} (ranging from -10 minutes to $+10$ minutes).

The first observation is that the global performance of the Lee-Ready signature compared to our reference sign is quite good, and importantly is in line with the reported performances of the Lee-Ready signature compared to "true" signs. On our 7-stock 1-day sample used above in Section 3, the Lee-Ready procedure correctly signs roughly 90% of the trades when using the current quotes or a lagged quote with a small lag (a few tens or hundreds of milliseconds). When using a larger negative lag for the signing quote, the performance drops quite fast. Between -5 and -10 seconds lag, the signature performance is roughly 80% on our sample. The 90% performance obtained in this sample is slightly better than what is usually reported in the financial literature, but note that most previous works use older data and do not go below a granularity of 1 second. It is quite intuitive that recent tick-by-tick data timestamped to the millisecond should allow for a better signature performance.

The second observation is that, at least visually on this small sample, the lag rule does not seem to be increasing the performance. The five second rule of 1991 is not relevant on 2011 high-frequency data, but even very small lags slightly depreciate the performance. The third panel of Figure 10 shows the slight performance drop as the lag goes from 0 to -200 milliseconds. This would tend to advocate for the use of contemporaneous quotes when signing trades with the Lee-Ready algorithm on *recent* data, but we will provide a more detailed view on this problem in Section 5.4.

Our third observation is that this signing performance drops dramatically for $\delta_{LR} \in [0.050, 0.150]$ before eventually increasing again for larger positive lags. In roughly 100 milliseconds, the signature performance drops from roughly 90% to roughly 60%. This allows us to make a clear link to Figure 2: the time interval $[0.050, 0.150]$ is precisely the support of the synchronization lag between the "trades" file and the "quotes" file for our 7-stock 1-day sample under scrutiny here. Therefore during that time interval the transactions are reflected on the quotes, and the prices changes occur, if any, so that the quotes after that reflect the consequences of the trades, and therefore cannot be used to sign them. In other words, Figure 10 is an illustration of the market impact of trades (see e.g. Bouchaud et al. (2009) for a review). It is particularly interesting to see that the performance of the Lee-Ready signing procedure actually reflects the instantaneous market impact of the trades: immediately after the transaction, prices may move — so that the quotes are not useful to sign the trades — but the impact gradually disappear — the performance of the signature increases slowly for large positive δ_{LR}.

5.3 A toy model for the signature performance

We now propose a toy model that account for the shape of the performance of the Lee-Ready signature as observed on Figure 10. Let m_t be the mid-price at time t. Let us assume that the
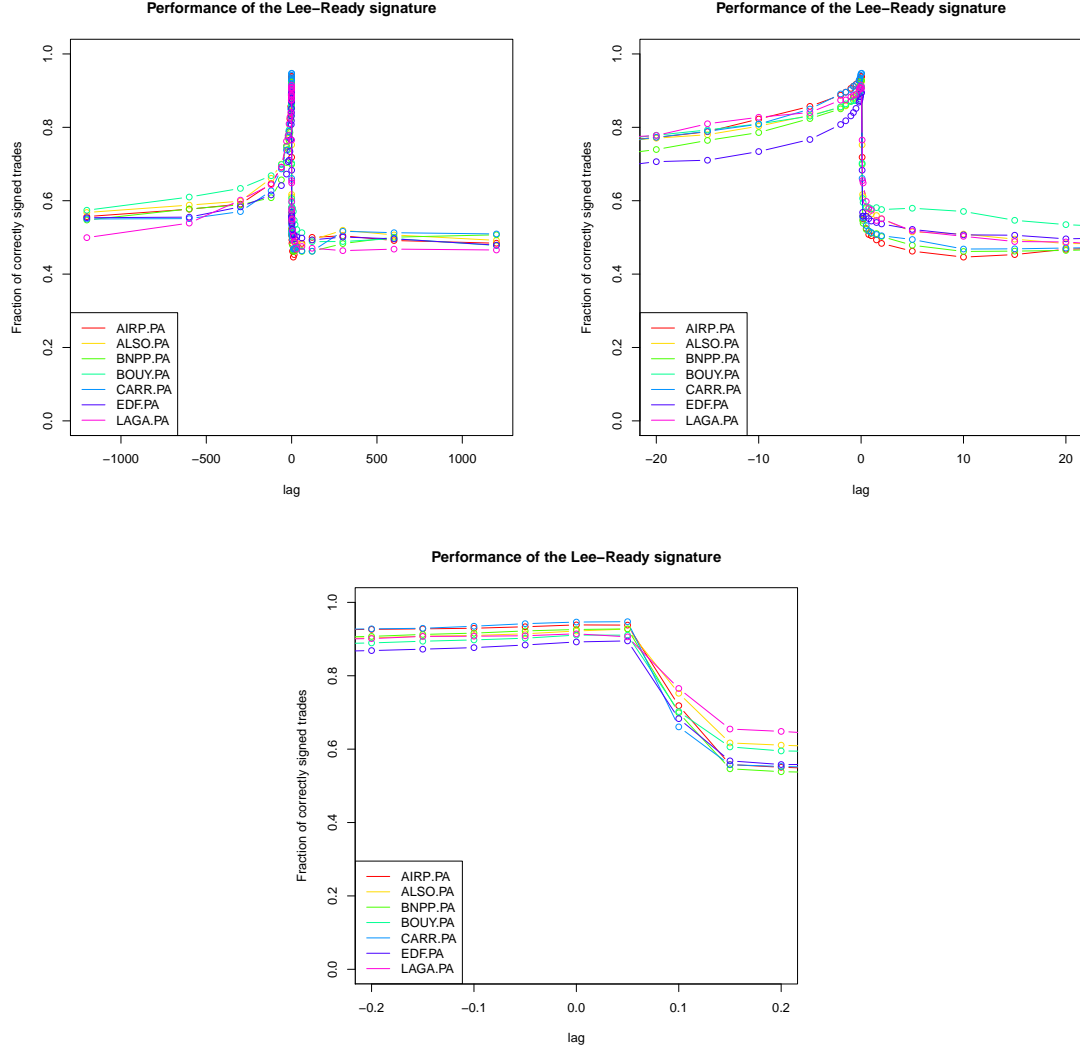
Figure 10: Performance of the Lee-Ready signing procedure (fraction of correctly signed orders) as a function of the lag parameter δ_{LR}, in seconds. All three plots represent the same data, at a different zooming level around 0.

flows of orders that moves the mid-price are homogeneous Poisson processes : let λ_+^{LC} (resp. λ_+^M) be the intensity of the limit orders and cancellations (resp. market orders) that make the mid-price move up, and λ_-^{LC} and λ_-^M be the intensities for the same events that instantaneously trigger downward moves of the mid-price. For simplicity, we assume that all movements of the best quotes have a size of one tick, and that the spread is constantly equal to one tick, so that the mid-price also moves by exactly one tick.

Let us now take a market order that is incorporated in the "quotes" file at time τ_q. This market order is (has been most of the times) reported in the "trades" files at time $\tau_t = \tau_q - \delta$ where δ is for now assumed to be deterministic. Furthermore, when attempting to sign this trade, the Lee-Ready algorithm will compare it to the mid-price at time $\tau_{LR} = \tau_t + \delta_{LR} = \tau_q + (\delta_{LR} - \delta)$.

Assume for now that $\tau_{LR} < \tau_q$, i.e. $\delta - \delta_{LR} > 0$, and that we have a buy market order. It is straightforward to see that this buy market order will be correctly signed by the Lee-Ready signature if and only if $m_{\tau_{LR}} \leq m_{\tau_q}$ i.e. if and only if $N_+ \geq N_-$ where N_+ is the number of events on the time interval (τ_{LR}, τ_q) making the mid-price move upward and N_- is the number of events making the mid-price move downward on this same time interval. Since N_+ and N_- are Poisson variables, their difference is distributed according to a Skellam (or Poisson Difference) distribution. Recall that the cumulative distribution function of the Skellam distribution with parameters (μ_1, μ_2) is written :

$$F(n; \mu_1, \mu_2) = e^{-(\mu_1 + \mu_2)} \sum_{k=-\infty}^{n} \left(\frac{\mu_1}{\mu_2} \right)^{k/2} I_k(2\sqrt{\mu_1 \mu_2}), \tag{1}$$

where I_k is the modified Bessel function of the first kind. Then the probability $p^b(\delta, \delta_{LR})$ that the buy market order considered is correctly signed by the anterior τ_{LR} mid-price is :

$$p^b_<(\delta, \delta_{LR}) = F(0; (\lambda_-^{LC} + \lambda_-^M)|\delta - \delta_{LR}|, (\lambda_+^{LC} + \lambda_+^M)|\delta - \delta_{LR}|). \tag{2}$$

In the case of a sell market order, we obviously need $N_+ - N_- \leq 0$, hence we obtain the result by exchanging the $+$ and $-$ intensities:

$$p^s_<(\delta, \delta_{LR}) = F(0; (\lambda_+^{LC} + \lambda_+^M)|\delta - \delta_{LR}|, (\lambda_-^{LC} + \lambda_-^M)|\delta - \delta_{LR}|). \tag{3}$$

Let us now turn to the case $\tau_q < \tau_{LR}$, i.e. $\delta - \delta_{LR} < 0$. Let us first assume that the market order we consider does *not* change the mid-price. In this case, with the same arguments as above, a buy (resp. sell) market order will be correctly signed if and only if $m_{\tau_q} \geq m_{\tau_{LR}}$ (resp. $m_{\tau_q} \leq m_{\tau_{LR}}$) i.e. if and only if $N_+ - N_- \leq 0$ (resp. $N_- - N_+ \leq 0$) on the time interval (τ_q, τ_{LR}). In short, we have with obvious notations the probability to correctly sign non quote-changing trades with posterior

quotes :

$$p_>^b(\delta, \delta_{LR}) = p_<^s(\delta, \delta_{LR}), \tag{4}$$

$$p_>^s(\delta, \delta_{LR}) = p_<^b(\delta, \delta_{LR}). \tag{5}$$

Now, if we consider a trade that changes the best quote (by assumption by exactly one tick), we need to correct for its instantaneous impact. Therefore, with the same arguments as above, a buy (resp. sell) market order will be correctly signed if and only if $m_{\tau_q} \geq m_{\tau_{LR}}$ (resp. $m_{\tau_q} \leq m_{\tau_{LR}}$) i.e. if and only if $(N_+ + 1) - N_- \leq 0$ (resp. $(N_- + 1) - N_+ \leq 0$) on the time interval (τ_q, τ_{LR}). Therefore, in this case :

$$p_{>,agg}^b(\delta, \delta_{LR}) = F(-1; (\lambda_+^{LC} + \lambda_+^M)|\delta - \delta_{LR}|, (\lambda_-^{LC} + \lambda_-^M)|\delta - \delta_{LR}|), \tag{6}$$

$$p_{>,agg}^s(\delta, \delta_{LR}) = F(-1; (\lambda_-^{LC} + \lambda_-^M)|\delta - \delta_{LR}|, (\lambda_+^{LC} + \lambda_+^M)|\delta - \delta_{LR}|). \tag{7}$$

Finally, let ρ_{agg} be the fraction of (aggressive) market orders (of either side) that make the price move (equivalently let us assume that each market order can make the mid-price move with probability ρ_{agg}). By basic properties of Poisson processes the fraction of buy market orders is $\rho_+ = \frac{\lambda_+^M}{\lambda_+^M + \lambda_-^M}$ since among market orders only buy (resp. sell) orders can make the price move up (resp. down). Then the probability $p(\delta, \delta_{LR})$ that a trade, reported in the quotes file with a lag δ from the "trades" file timestamp, is correctly signed by the "quotes" taken with a lag δ_{LR} from the timestamp of the "trades" file, is computed as :

$$
\begin{aligned}
p(\delta, \delta_{LR}) =& \mathbf{1}(\delta - \delta_{LR} > 0) \left[\rho_+ p_<^b(\delta, \delta_{LR}) + (1 - \rho_+) p_<^s(\delta, \delta_{LR}) \right] \\
&+ \mathbf{1}(\delta - \delta_{LR} < 0) \left[(1 - \rho_{agg}) \left[\rho_+ p_>^b(\delta, \delta_{LR}) + (1 - \rho_+) p_>^s(\delta, \delta_{LR}) \right] \right. \\
&\left. + \rho_{agg} \left[\rho_+ p_{>,agg}^b(\delta, \delta_{LR}) + (1 - \rho_+) p_{>,agg}^s(\delta, \delta_{LR}) \right] \right].
\end{aligned}
\tag{8}
$$

Finally, we can now incorporate in the model the fact that δ is a random variable with an probability density function that should resemble the empirical densities of Figure 2. If δ is assumed to be absolutely continuous with density f_δ, then the theoretical performance of the Lee-Ready signature using a δ_{LR} lagged-quote is:

$$p(\delta_{LR}) = \int p(u, \delta_{LR}) f_\delta(u) du. \tag{9}$$

Figure 11 plots examples of these theoretical performances, illustrating how Equation (9) is able to account for the general shape of the Lee-Ready performance as a function of the lag, as it is observed on our sample. As this is a toy model however, based on time-homogeneous Poisson processes that are not able to capture the features of the order flows (autocorrelation of trades signs, trades clustering, etc.), we should not expect high fitting power when calibrated on market data. The results for the fitting on our 7-stock 1-day sample gives nonetheless interesting results.
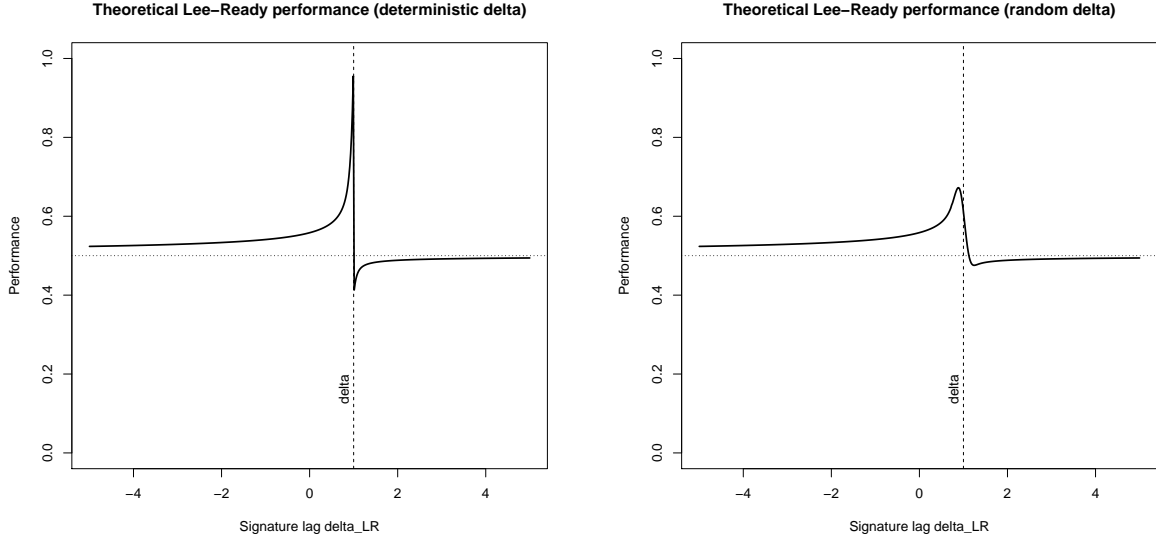
Figure 11: Theoretical performance of the Lee-Ready signature in the proposed toy model, for a deterministic δ (left) and a random δ (right). For these illustrations, $\lambda_\pm^{LC} = 5$, $\lambda_\pm^M = 1$, $\rho_{agg} = 0.6$ and $\delta = 1$ (deterministic) or δ is Gaussian with parameters $(1; 0.1)$ (random).

Figure 12 plots the results for the stock AIRP.PA (for the sake of brevity we reproduce only the graphs for the first stock of the sample, but they are all similar). Firstly, at small time scales, the model is able to satisfactorily reproduce the drop of the signature performance around $\delta_{LR} = +100$ milliseconds, translating the incorporation of the trades into the quotes. Secondly, at larger time scales, we observe that the empirical performance is globally overestimated by the model from roughly $\delta_{LR} = -1$ second and onwards. This might be a mark of the clustering of market events: trades are submitted irregularly during the trading day, with burst of activities during which all types of market events occur ; it is therefore expected that the mid-price moves more actively around the timestamps of trades submission, decreasing the actual empirical signature performance. One may also argue that the fact that the overestimation of the performance by the model is greater after the trades than before is an indication that transactions trigger market activity rather than result from it.

Remark 1. It is easy to make the model more flexible by adding a spread distribution or by distinguishing between aggressive buy and sell orders. However, the roughness of the Poisson hypothesis is such that these enhancements are not expected to be crucial.

5.4   Signature performance through time

To complement the previous results, we run the Lee-Ready procedure on the stock BNPP.PA for all the trading days of the database, with lags $\delta_{LR} \in \{-5, -2, -1.5, -1, -0.75, -0.5, -0.2,$
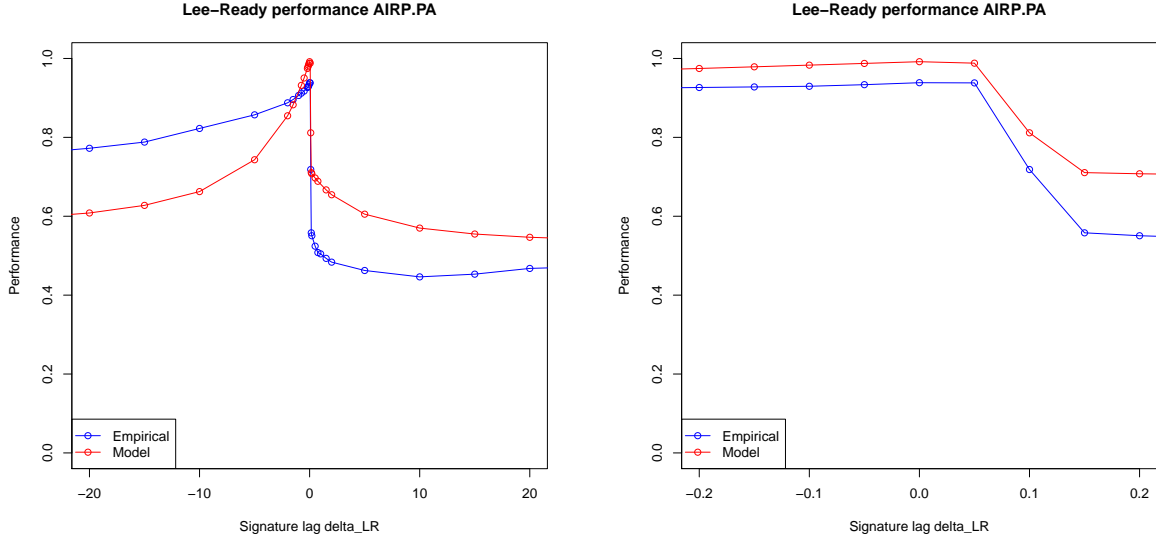
Figure 12: Empirical vs toy model performance of the Lee-Ready signature for the stock AIRP.PA. Both left and right graphs show the same data at different time scales.

$-0.15, -0.1, -0.05, 0, 0.05, 0.1\}$ seconds. On Figure 13 we plot the (*a posteriori*) optimal performance of the Lee-Ready signature as a function of the calendar dates, as well as the performance for the representative lags $-5, 0 - 1, -0.1$, and 0. On the right pannel, we plot the optimal lag (giving the optimal performance) for each of these days. The first observation is that signature performance by the Lee-Ready algorithm are roughly constant over time, with (visually) a slight downward trend of the optimal performance from 2008 to 2012 and a increasing slope for the last part of the sample, but roughly always staying around 90% and above at all time The second observation is that performances for given lags do not move in parallel. In 2008, the use of the current quote ($\delta_{LR} = 0$) and the use of a -5 seconds lagged quote would give similar performances, less that 80%, while the use of the -1 second lagged quote would improve the result (in accordance with Chakrabarty et al. (2012) on 2005 data), and the use of the 100 milliseconds lagged quote would even be better. One therefore cannot recommend absolute values for the Lee-Ready lag, since these quantities depend intrinsically on the database used, the exchange, the stock and the time-window considered. From Figure 13, we can only confirm the conjecture that the more recent the data studied, the shorter the lag should be used. On our sample, from 2011 and onwards, we even observe that a positive $\delta_{LR} = +50$ milliseconds is most of the times optimal to get the best performance of the Lee-Ready signature !

Very interestingly, the graphs allow as well for some forensic analysis of the quality of the data, as it is plausible that these variations in signature performances are marks of technical changes in the production of the data. We observe that the performance of the signature with the current
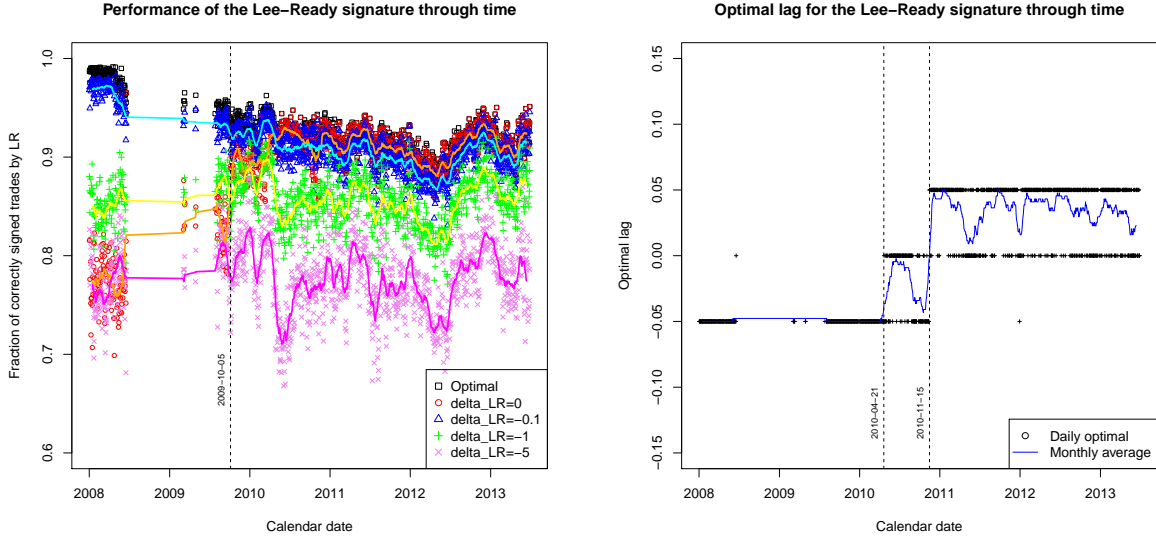
Figure 13: Performance of the Lee-Ready signature for the stock BNPP.PA (left) and optimal lag for these signatures (right) as a function of the calendar date. Continuous lines are monthly moving average of the associated quantities (± 11 trading days).

quotes ($\delta_{LR} = 0$) sharply and suddenly increases on October 5th, 2009, a date close to one we have already identified in Section 4 as a change in data feeds on Euronext. Furthermore, this analysis confirms that 2010 has been a year of significant changes in the production of the data, especially between April 21st, 2010 (the first day of the sample for which $\delta_{LR} = 0$ gives the optimal signature performance, if we except one outlier in 2008) and November 15th, 2010 (the first day of the sample for which $\delta_{LR} = +0.050$ gives the optimal signature performance). These dates are also very close to dates identified in Section 4, and interestingly, they are also the dates of the migration process conducted by Euronext to relocate its data centre to Basildon (near London). This process started in April, with the first tests in May, and the production migration was completed in October[9]. We may conjecture that such a huge technical change is the source of our observations, either directly or through changes that the data vendor (Reuters) would implement in response.

It is striking that a simple operation such as trade signature bears the marks of these technical changes.

6   Potential side effects of choices of order flow reconstruction

We end this work by an illustration of the potential side effects of choices made when reconstructing the order flow. Let us imagine the (very academic and hypothetical...) case of a trader looking to

---

[9]Euronext Info-flashes date April 21st, 2010, May 17th, 2010 and October 11th, 2010.

calibrate on market data a Hawkes process to model the clustering behaviour of trades. Hawkes models have been increasingly popular in finance and a vast literature is available on the subject (see e.g. Bowsher (2007), Large (2007) for pioneering papers and Bacry et al. (2015) for a recent review). Let us assume that our trader wants to model the counting process of market orders $(N_t^M)_{t \geq 0}$ with an intensity λ_t^M driven by the following equation :

$$\lambda_t^M = \lambda_0 + \int_0^t \alpha e^{-\beta(t-s)} dN_s^M, \tag{10}$$

i.e. to use a one-dimensional Hawkes model with a unique exponential kernel. In this model, each market order makes the intensity increase with a jump of size α, and this effect decreases exponentially fast with time with parameter β. The ratio $\frac{\alpha}{\beta}$ is interpreted as the fraction of trades triggered by previous trades in this model, in other words as a level of "endogeneity" in the market (Filimonov & Sornette 2012, Hardiman et al. 2013). We are aware that this is arguably a very poor model to try on a full trading day — a single exponential kernel, i.e. a single time scale, combined with a constant baseline intensity cannot satisfyingly model the complexity of the market, e.g. account for daily seasonality, see Lallouache & Challet (2016) —, but we will nonetheless keep this simple example for the sake of illustration. Our trader has two choices : the first one is to take the "trades" file as is, each different time stamp counting for one market order ; the second one is to precisely match each line in the "quotes" file, and in this case several lines might possibly be merged into one quote change, i.e. one transaction. In both cases, the model can then be fitted by numerical maximization of the log-likelihood of the model (see e.g. Muni Toke & Pomponio (2012) and references therein).

On Figure 14, we plot the ratio evolution $\frac{\alpha}{\beta}$ through time for the stock PSON.L in February and March 2008, i.e. two full trading months. The plot shows that the endogeneity ratio estimated with the raw "trades" file is consistently greater by roughly 15% to the endogeneity ratio estimated on a sample prepared with the matching procedure. Indeed, this result is consistent with the fact that matching procedure groups trades with close time stamps into one market order, if this is necessary to match the "quotes" file, while the calibration on the raw "trades" file considers these parts of the same trade as several closely-occurring market orders, therefore increasing the apparent clustering of transactions, hence the larger endogeneity ratio. This effect is best seen on the sample chosen since on this stock and at these dates, Matching 3 procedure groups a significant proportion of lines of the "trades" files, as is explicit on Figure 7.

The matching procedure also has an effect on the cancellations, since transactions are matched against (a priori) cancellations at the best quotes. Let us now imagine that our trader were now to model the intensity λ^C of the counting process of cancellations at the best quotes being excited

Figure 14: Hawkes ratio $\frac{\alpha}{\beta}$ calibrated for the model (10) on the stock PSON.L in January and February 2008 (left). Idem for model (11) (right).

by arrival of market orders (counting process N^M) as:

$$\lambda_t^C = \lambda_0 + \int_0^t \alpha e^{-\beta(t-s)} dN_s^M. \tag{11}$$

There again, this arguably very poor model is kept simple for the purpose of illustration. The fitted ratio $\frac{\alpha}{\beta}$ in this case is plotted on Figure 14. In this case, the sample that does not match the "trades" and "quotes" overestimates the number of cancellations at the best quotes by keeping incorrect cancellations following trades that should be identified as the incorporation of the trades effect in the quotes. There is thus an incorrectly high number of occurrences of the pattern "trade then cancellation", it is therefore not surprising to observe here again a greater ratio when fitting without matching, that when fitting with prior matching.

7   Conclusion

We have investigated in this paper the TRTH trades and quotes database. We provide an algorithm that helps synchronizing the two data sources and therefore allows for a "true" reconstruction of the flow of limit orders, market orders and cancellations, even if we have only aggregated data and not any detailed information on the trades. It turns out that the enhanced matching is very useful for older data (2008), and still significantly better even for recent data (2013). We have provided basic examples illustrating potential side effects of the reconstruction procedure on model calibration.

30

This hopefully may be useful for the community of researchers that work on high-frequency data but do not have access to detailed data down to the granularity of the messages sent to/from the exchange.

This work is also a tentative of forensic analysis of financial data in the sense that we are able to track through our tick-by-tick aggregated data fundamental technical changes that occurred between 2009 and 2011 on the way financial exchanges (Paris, London, Frankfurt are tested here) produce data. It is very interesting to observe that these technical changes still leave marks on our aggregated database, even after several steps of data transformation and formatting.

As a conclusion, this work may be useful in reminding that a database, even packaged and provided by a unique source, is not a uniform object. Therefore, despite the temptation to use always larger sets of data in a search of statistical significance, the coherence of the sample should be ensured foremost.

Acknowledgements

References

Bacry, E., Mastromatteo, I. & Muzy, J.-F. (2015), 'Hawkes processes in finance', *Market Microstructure and Liquidity* **1**(01), 1550005.

Bouchaud, J.-P., Farmer, J. D. & Lillo, F. (2009), How markets slowly digest changes in supply and demand, *in* T. H. R. Schenk-Hoppé, ed., 'Handbook of Financial Markets: Dynamics and Evolution', Handbooks in Finance, North-Holland, pp. 57 – 160.

Bowsher, C. G. (2007), 'Modelling security market events in continuous time: Intensity based, multivariate point process models', *Journal of Econometrics* **141**(2), 876–912.

Chakrabarty, B., Moulton, P. C. & Shkilko, A. (2012), 'Short sales, long sales, and the Lee–Ready trade classification algorithm revisited', *Journal of Financial Markets* **15**(4), 467–491.

Cont, R., Stoikov, S. & Talreja, R. (2010), 'A stochastic model for order book dynamics', *Operations research* **58**(3), 549–563.

Easley, D., de Prado, M. L. & O'Hara, M. (2016), 'Discerning information from trade data', *Journal of Financial Economics* .

Ellis, K., Michaely, R. & O'Hara, M. (2000), 'The accuracy of trade classification rules: Evidence from nasdaq', *Journal of Financial and Quantitative Analysis* **35**(04), 529–551.

Filimonov, V. & Sornette, D. (2012), 'Quantifying reflexivity in financial markets: Toward a prediction of flash crashes', *Physical Review E* **85**(5), 056108.

Hardiman, S. J., Bercot, N. & Bouchaud, J.-P. (2013), 'Critical reflexivity in financial markets: a hawkes process analysis', *The European Physical Journal B* **86**(10), 1–9.

Huang, W., Lehalle, C.-A. & Rosenbaum, M. (2015), 'Simulating and analyzing order book data: The queue-reactive model', *Journal of the American Statistical Association* **110**(509), 107–122.

Lallouache, M. & Challet, D. (2016), 'The limits of statistical significance of hawkes processes fitted to financial data', *Quantitative Finance* **16**(1), 1–11.

Large, J. (2007), 'Measuring the resiliency of an electronic limit order book', *Journal of Financial Markets* **10**(1), 1–25.

Lee, C. M. C. & Ready, M. J. (1991), 'Inferring trade direction from intraday data', *The Journal of Finance* **46**(2), 733–746.

Mandelbrot, B. (1963), 'The variation of certain speculative prices', *The Journal of Business* **36**(4), 394–419.

Muni Toke, I. (2015), 'The order book as a queueing system: average depth and influence of the size of limit orders', *Quantitative Finance* **15**(5), 795–808.

Muni Toke, I. & Pomponio, F. (2012), 'Modelling trades-through in a limit order book using Hawkes processes', *Economics: The Open-Access, Open-Assessment E-Journal* **6**.

Omrane, W. B. & Welch, R. (2016), 'Tick test accuracy in foreign exchange ECN markets', *Research in International Business and Finance* **37**, 135–152.

Piwowar, M. S. & Wei, L. (2006), 'The sensitivity of effective spread estimates to trade–quote matching algorithms', *Electronic Markets* **16**(2), 112–129.

Theissen, E. (2001), 'A test of the accuracy of the Lee/Ready trade classification algorithm', *Journal of International Financial Markets, Institutions and Money* **11**(2), 147–165.