UNIVERSITY OF CAPE TOWN
IYUNIVESITHI YASEKAPA • UNIVERSITEIT VAN KAAPSTAD

UNIVERSITY OF CAPE TOWN

HONOUR'S PROJECT REPORT

An Exercise in R: High Frequency Covariance estimation using Malliavin-Mancino and Hayashi-Yoshida estimators

---

Author:
Patrick CHANG & Roger BUKURU

Supervisor:
Assoc. Prof. Tim GEBBIE

A report submitted in fulfilment of the requirements for the degree of Honours in Statistical Science

in the

Department of Statistical Sciences

November 12, 2019

Declaration of Authorship

We, Patrick CHANG & Roger BUKURU , declare that this report titled, "An Exercise in R: High Frequency Covariance estimation using Malliavin-Mancino and Hayashi-Yoshida estimators" and the work presented in it are our own. We confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where we have consulted the published work of others, this is always clearly attributed.
- Where we have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely our own work.
- We have acknowledged all main sources of help.
- Where the thesis is based on work done by ourselves jointly with others, we have made clear exactly what was done by others and what we have contributed ourselves.

Signed:

Date:

UNIVERSITY OF CAPE TOWN

Abstract

Science

Department of Statistical Sciences

Honours in Statistical Science

An Exercise in R: High Frequency Covariance estimation using Malliavin-Mancino and Hayashi-Yoshida estimators

by Patrick CHANG & Roger BUKURU

We revisit and demonstrate two well-known non-parametric estimators; the Malliavin-Mancino (MM) [1], [2] and the Hayashi-Yoshida (HY) [3] estimator. Both address the issue of covariance estimation for high-frequency asynchronous time-series data. The first by embracing a Fourier perspective and the latter using averaging over discrete windows. The aim of the work here aims to provide an easily re-used practical tutorial to argue for the efficacy of the MM estimator in high-frequency finance applications. Towards this end, we conduct Monte Carlo experiments to demonstrate that the two estimators differ only under asynchronous observations, where the MM estimator has lower correlation estimates compared to the HY estimator. Unsurprisingly, we attribute this difference to the Epps effect [4]. However, as a novel application, we show the existence of the Epps effect in the top 10 stocks from the Johannesburg Stock Exchange (JSE) by various methods of aggregating Trade and Quote (TAQ) data. Specifically, by comparing calendar time based sampling with volume time sampling methods. We argue that the MM estimator is more representative of trade-time reality. The world of high-frequency finance is not a missing data problem as sampled from some underlying synchronous continuous stochastic process, but rather it is a world of truly disconnected, inter-related, discrete and asynchronous events where the relationship between events are the fundamental entities and measurables. We argue the MM estimator is a more faithful representation of the underlying data as it does not over-estimate short-term correlations in such an asynchronous event driven world.

Acknowledgements

We would firstly like to thank our supervisor Tim Gebbie for guiding us through the various difficulties of this project and never revealing the answer straight away, giving us the satisfaction of solving the problem ourselves. Effectively guiding us to ensure we were on time for the deadlines and always finding the time to answer our questions.

We would secondly like to thank Etienne Pienaar and Melusi Mavuso for their input and assistance. We would also like to thank Diane Wilcox, Chanel Malherbe and Dieter Hendricks who provided us with resources to fast track our progress.

The financial assistance of the National Research Foundation (NRF) towards this research is hereby acknowledged. Opinions expressed and conclusions arrived at, are those of the author and are not necessarily to be attributed to the NRF.

Thank You

Patrick CHANG & Roger BUKURU

Contents

List of Figures

List of Tables

List of Abbreviations

API	**A**pplication **P**rogramming **I**nterface
FFT	**F**ast **F**ourier **T**ransform
GARCH	**G**eneralized **A**utoregressive **C**onditional **H**eteroskedasticity
GBM	**G**eometric **B**rownian **M**otion
HY	**H**ayashi **Y**oshida
JSE	**J**ohannesburg **S**tock **E**xchange
MM	**M**alliavin **M**ancino
NaN	**N**ot a **N**umber
NUFFT	**N**on **U**niform **F**ast **F**ourier **T**ransform
OHCLV	**O**pen **H**igh **L**ow **C**lose **V**olume
OU	**O**rnstein **U**hlenbeck
RV	**R**ealised **V**olatility
SDE	**S**tochastic **D**ifferential **E**quation
TAQ	**T**rade **A**nd **Q**oute
VG	**V**ariance **G**amma
VWAP	**V**olume **W**eighted **A**verage **P**rice

Chapter 1

Introduction

Covariation is a key parameter in finance with traditional applications in portfolio optimisation and more recent applications in unsupervised state discovery to discern changes in the system [5]. The availability of high-frequency financial data has allowed the miss-estimation of large portfolio correlation measures to be elevated by removing the problem of scarcity of data [6]. However, having high-frequency financial data comes with its caveats, specifically estimating the correlation becomes a much harder task due to the asynchrony arising from having tick-by-tick trade data. Thus the popular approach using the realised volatility estimator is problematic under high-frequency asynchronous data, as it requires a choice of synchronisation and data interpolation which leads to biases induced in the estimate [3], [7].

In this report, we present two non-parametric estimators designed specifically to deal with the asynchrony. The first estimator proposed by Malliavin and Mancino [1], [2], [8] adopts a Fourier approach and the second estimator proposed by Hayashi and Yoshida [3] uses the contributions from overlapping intervals to overcome the problem faced by the traditional realised volatility.

We will investigate the correlation under high-frequency finance using the aforementioned estimators in a data-informed approach. This novel approach is due to how the price process is generated in the financial market - we argue that the price generation is better represented as discrete, asynchronous events rather than samples from an underlying continuous stochastic process; thus we abstain from taking the more popular market microstructure noise framework [9]–[11]. A natural consequence of studying high-frequency financial data is the need to address the Epps effect [4] which is the drop in correlation associated with smaller sampling intervals. We demonstrate the Epps effect arising from asynchrony, specifically the relation this has with the level of asynchrony and the sampling interval under consideration [6], [12], [13], furthermore we will investigate the Epps effect from a new perspective, specifically by aggregating Trade and Quote (TAQ) data with different techniques.

To this end, the report is structured as follows: in Section 2 we will cover the

brief derivation and implementation algorithms for the MM and HY estimators. Section 3 we conduct Monte Carlo experiments in an attempt to identify how, when and where the two estimators differ. Therefore we begin the comparison by assessing how the two estimators perform under asynchrony - specifically asynchrony induced from a missing data manner. We then compare the two estimators with various stochastic processes. Finally, we recover the results from [12] and adapt his experiment to highlight the differences in our data-informed approach compared to the market microstructure noise approach. Section 4 outlines the data collection process and the various algorithms employed to aggregate TAQ data from different perspectives, specifically we consider Closing and Volume Weighted Average Price (VWAP) bars from the Calendar time approach. In addition, we consider an Intrinsic time approach for which we will employ the framework provided by Derman [14] and our method of aggregation in Intrinsic time. Section 5 combines the various aggregation techniques from Section 4 along with the estimators to study the Epps effect in the Johannesburg Stock Exchange (JSE). Finally, we end off with Section 6 where we discuss some future topics of investigation and we highlight the aspects of what we could and could not achieve with this study.

Chapter 2

The Theory

2.1 Trigonometric Fourier Transform

2.1.1 Derivation

We present the derivation proposed by Malliavin and Mancino [1] which is an estimator that is constructed in the frequency domain. The only assumption required for the derivation is the Bachelier hypothesis. Which states that all measurable economic data p^* are driven by semi martingales which can be decomposed into a drift term with bounded variation paths and a local martingale [1], [15]. Thus their Itô stochastic differential equation given by

$$dp^j = \sum_{i=1}^{d} \sigma_i^j dW^i + \beta^j dt. \tag{2.1.1}$$

If we denote $S_i(t)$ to be the generic asset price at time t, we will set $p_i(t) = \ln(S_i(t))$. It can be shown that the covariance matrix of diffusion processes given by (2.1.1) can be presented as

$$\Sigma^{j,k}(t) = \sum_{i=1}^{d} \sigma_i^j(t)\sigma_i^k(t). \tag{2.1.2}$$

For simplification purposes, we can always reduce a semi-martingale on a fixed time window to the case where the window is $[0, 2\pi]$ by the change of origin and re-scaling the unit of time [1]. We now define the Fourier coefficients of dp^j as

$$\begin{aligned} a_0(dp^j) &= \frac{1}{2\pi}\int_0^{2\pi} dp^j(t), \\ a_k(dp^j) &= \frac{1}{\pi}\int_0^{2\pi} \cos(kt)dp^j(t), \\ b_k(dp^j) &= \frac{1}{\pi}\int_0^{2\pi} \sin(kt)dp^j(t), \end{aligned} \tag{2.1.3}$$

$\forall k \geq 1$. Similarly, the Fourier coefficients of the volatility is defined as

$$
\begin{aligned}
a_0(\Sigma) &= \frac{1}{2\pi}\int_0^{2\pi} \Sigma(t)dt, \\
a_k(\Sigma) &= \frac{1}{\pi}\int_0^{2\pi} \cos(kt)\Sigma(t)dt, \\
b_k(\Sigma) &= \frac{1}{\pi}\int_0^{2\pi} \sin(kt)\Sigma(t)dt.
\end{aligned} \tag{2.1.4}
$$

The main idea behind this method is to find a mathematical expression of the Fourier coefficients of Σ using the Fourier coefficients of dp^j [1]. This leads to Theorem 3.1 in [1]:

Theorem 2.1.1 *Fix an integer $n_0 > 0$, then the Fourier coefficients of the volatility are given by the following formulae:*

$$
\begin{aligned}
a_0(\Sigma) &= \lim_{N\to\infty} \frac{\pi}{N+1-n_0} \sum_{s=n_0}^{N} \frac{1}{2}(a_s^2(dp^j) + b_s^2(dp^j)), \\
a_q(\Sigma) &= \lim_{N\to\infty} \frac{2\pi}{N+1-n_0} \sum_{s=n_0}^{N} (a_s(dp^j)a_{s+q}(dp^j)), \\
b_q(\Sigma) &= \lim_{N\to\infty} \frac{2\pi}{N+1-n_0} \sum_{s=n_0}^{N} (a_s(dp^j)b_{s+q}(dp^j)),
\end{aligned} \tag{2.1.5}
$$

$\forall q > 0$ for $a_q(\Sigma)$ and $\forall q \geq 0$ for $b_q(\Sigma)$.

By polarisation of the univariate case, the Fourier coefficients can be extended to the multivariate case, given by Theorem 3.2 in [1]:

Theorem 2.1.2 *Fix an integer $n_0 > 0$, then the Fourier coefficients of the volatility are given by the following formulae:*

$$
\begin{aligned}
a_0(\Sigma^{i,j}) &= \lim_{N\to\infty} \frac{\pi}{N+1-n_0} \sum_{s=n_0}^{N} \frac{1}{2}(a_s(dp^i)a_s(dp^j) + b_s(dp^i)b_s(dp^j)), \\
a_q(\Sigma^{i,j}) &= \lim_{N\to\infty} \frac{2\pi}{N+1-n_0} \sum_{s=n_0}^{N} \frac{1}{2}(a_s(dp^i)a_{s+q}(dp^j) + a_s(dp^j)a_{s+q}(dp^i)), \\
b_q(\Sigma^{i,j}) &= \lim_{N\to\infty} \frac{2\pi}{N+1-n_0} \sum_{s=n_0}^{N} \frac{1}{2}(a_s(dp^i)b_{s+q}(dp^j) + a_s(dp^j)b_{s+q}(dp^i)),
\end{aligned} \tag{2.1.6}
$$

$\forall q > 0$ for $a_q(\Sigma^{ij})$ and $\forall q \geq 0$ for $b_q(\Sigma^{ij})$.

Remark 2.1.1 *From deriving Theorem 2.1.1 and 2.1.2, we note that the scaling factors are different to that of Theorem 3.1 and 3.2 in [1] but the same as [16].*

Once the Fourier coefficients of the volatility matrix have been computed, results from Fourier theory allows the reconstruction of Σ from its Fourier coefficients [1]. Using the Fourier-Féjer inversion formula to reconstruct Σ, we get

$$\Sigma(t) = \lim_{N\to\infty} \sum_{k=0}^{N} \left(1 - \frac{k}{N}\right) \left(a_k(\Sigma)\cos(kt) + b_k(\Sigma)\sin(kt)\right), \quad \forall t \in (0, 2\pi). \tag{2.1.7}$$

The Féjer inversion formula has the advantage that if Σ is a positive function, then the approximation (2.1.7) will again be positive [1]. However, we are more interested in the integrated volatility defined as

$$\hat{\sigma}_{ij}^2 = \int_0^{2\pi} \Sigma^{i,j}(t)dt. \tag{2.1.8}$$

Which can be recovered by adjusting (2.1.5) to be

$$\hat{\sigma}_{ij}^2 = 2\pi a_o(\Sigma^{i,j}). \tag{2.1.9}$$

2.1.2 Numerical Implementation

To implement this procedure, the first thing to do is to re-scale the irregularly spaced observations $[t_1, ..., t_n]$ into the interval $[0, 2\pi]$ [15] using the formula

$$\tau_j = \frac{2\pi(t_j - t_1)}{(t_n - t_1)}, \quad j = 1, ..., n. \tag{2.1.10}$$

The integrals for the Fourier coefficients of dp^j can be computed using integration by parts [15]. Resulting in

$$\begin{aligned} a_k(dp^j) &= \frac{p^j(2\pi) - p^j(0)}{\pi} + \frac{k}{\pi}\int_0^{2\pi} \sin(kt)p^j(t)dt, \\ b_k(dp^j) &= -\frac{k}{\pi}\int_0^{2\pi} \cos(kt)p^j(t)dt. \end{aligned} \tag{2.1.11}$$

We note that (2.1.11) is numerically stable, because it does not involve the differentiation of p^j [1]. Furthermore, since the data gathered from financial markets are discrete and therefore finite, we need an assumption of how the data points are connected in order to compute (2.1.11). Malliavin and Mancino assume that $p^j(t)$ is equal to $p^j(t_i)$ in the interval $[t_i, t_{i+1}]$ [1], also known as the previous-tick interpolation [12]. Resulting in

$$\begin{aligned} a_k(dp^j) &\approx \frac{p^j(2\pi) - p^j(0)}{\pi} + \frac{1}{\pi}\sum_{i=1}^{N-1} [\cos(kt_i) - \cos(kt_{i+1})]p(t_i), \\ b_k(dp^j) &\approx \frac{1}{\pi}\sum_{i=1}^{N-1} [\sin(kt_i) - \sin(kt_{i+1})]p(t_i). \end{aligned} \tag{2.1.12}$$

We further note that the choice of interpolation is important. Barucci and Reno [7] showed that linear interpolation of prices between the interval $[t_i, t_{i+1}]$ resulted in a downward bias in the estimator. Malherbe further points out this is because linear interpolation induces spurious auto-correlation [15].

Algorithm 1 Trigonometric Fourier Transform

Require:

1. (n x m) matrix P of asynchronously sampled price
2. (n x m) matrix T of asynchronously sampled times

Re-scale the time $[t_{min}, t_{max}] \rightarrow [0, 2\pi]$
I. Extract trading times and prices
for i = 1 to m **do**
  I.1. Slice the non-uniformly re-scaled sampled times
  for the i^{th} object
  $\tilde{\tau} \leftarrow \tau(i)$
  I.2. Slice the sampled data indexing the times
  for the i^{th} object
  $\tilde{\varphi} \leftarrow \ln(p(\tilde{\tau}))$
  II. Compute Fourier coefficients for all values of k
  $\tilde{a} \leftarrow \frac{\tilde{\varphi}(2\pi)-\tilde{\varphi}(0)}{\pi} + \frac{1}{\pi}\sum_{j=1}^{N-1}[\cos(\tilde{k}\tilde{\tau}_j) - \cos(\tilde{k}\tilde{\tau}_{j+1})]\tilde{\varphi}(\tilde{\tau}_j)$
  $\tilde{b} \leftarrow \frac{1}{\pi}\sum_{j=1}^{N-1}[\sin(\tilde{k}\tilde{\tau}_j) - \sin(\tilde{k}\tilde{\tau}_{j+1})]\tilde{\varphi}(\tilde{\tau}_j)$
  I.3. Gather Fourier coefficients for i^{th} object
  for all values values of k
  $a(i) \leftarrow \tilde{a}$
  $b(i) \leftarrow \tilde{b}$
end for
Compute the integrated volatility and co-volatility over
the time window for objects i and j
$\Sigma_{ij} \leftarrow \frac{\pi^2}{|K|}\sum_{k\in K}[a_k(i)a_k(j) + b_k(i)b_k(j)]$
$R_{ij} \leftarrow \frac{\Sigma_{ij}}{\sqrt{\Sigma_{ii}}\sqrt{\Sigma_{jj}}}$
return (Σ, R)

Algorithm 1 [1] was provided by [17], [18] and [15].

2.2 Complex Exponential Fourier Transform

2.2.1 Derivation

We present the second derivation proposed by Malliavin and Mancino [2], which uses a different Fourier approach to solve the same problem. The assumptions required for the derivation is that $p^j(t)$ is a continuous semi-martingale satisfying the stochastic differential equation

[1] The pair-wise implementation can be found in ftcorr.R.

$$dp^j = \sum_{i=1}^{d} \sigma_i^j dW^i + b^j dt, \quad j = 1, ..., n, \tag{A-I}$$

where $W = (W^1, ..., W^d)$ are independent Brownian motions on a filtered probability space, σ_*^* and b^* are adapted stochastic processes satisfying

$$\begin{aligned} E\Big[\int_0^T (b^j(t))^2 dt\Big] &< \infty, \\ E\Big[\int_0^T (\sigma_i^j(t))^4 dt\Big] &< \infty, \\ i = 1, ..., d;\ j &= 1, ..., n. \end{aligned} \tag{A-II}$$

The main idea behind this derivation is the same as section 2.1. We want to establish a connection between the Fourier transform of the volatility process (2.1.2) and the Fourier transform of the price process [9].

We first re-scale the time window from $[0, T]$ to $[0, 2\pi]$. We then define the Fourier transform of dp^j as

$$\mathcal{F}(dp^j)(k) := \frac{1}{2\pi} \int_{[0,2\pi]} \exp(-ikt) dp^j(t), \tag{2.2.1}$$

and the Bohr convolution product between two functions Φ, Ψ on the integers $\boldsymbol{Z}$ as

$$(\Phi *_B \Psi)(k) := \lim_{N\to\infty} \frac{1}{2N+1} \sum_{s=-N}^{N} \Phi(s)\Psi(k-s). \tag{2.2.2}$$

This leads to Theorem 2.1 in [2]:

Theorem 2.2.1 *Consider a process p satisfying the assumption* (A-II). *Then we have for i,j = 1,2:*

$$\frac{1}{2\pi}\mathcal{F}(\Sigma^{ij})(k) = \mathcal{F}(dp^i) *_B \mathcal{F}(dp^j)(k), \forall k \in \boldsymbol{Z}. \tag{2.2.3}$$

The equality (2.2.3) *is attained in probability, which means the limit in the convolution product exists in probability*

Using Theorem 2.2.1, we get that

$$\mathcal{F}(\Sigma^{ij})(k) = \lim_{N\to\infty} \frac{2\pi}{2N+1} \sum_{|s|\le N} \mathcal{F}(dp^i)(s)\mathcal{F}(dp^j)(k-s), \quad \forall k \in \boldsymbol{Z}. \tag{2.2.4}$$

Now that we have an expression for the Fourier coefficients of the volatility process, we can reconstruct $\Sigma(t)$ using the Féjer inversion formula. Yielding

$$\Sigma^{ij}(t) = \lim_{N\to\infty} \sum_{|k|\leq N} \Big(1 - \frac{|k|}{N+1}\Big) \mathcal{F}(\Sigma^{ij})(k) \exp(ikt). \tag{2.2.5}$$

Having an expression for the Fourier coefficients of the volatility process allows for the computation of the integrated volatility as

$$\begin{aligned} \hat{\sigma}^2_{ij} &= \int_0^{2\pi} \Sigma^{ij}(t)dt = 2\pi \mathcal{F}(\Sigma^{ij})(0) \\ &= (2\pi)^2 (\mathcal{F}(dp^i) *_B \mathcal{F}(dp^j))(0). \end{aligned} \tag{2.2.6}$$

2.2.2 Numerical Implementation

We first re-scale the irregularly spaced observations $[t_1, ..., t_n]$ to the interval $[0, 2\pi]$ using (2.1.10). We then require an interpolation method for the discrete observations for the price process. Malliavin and Mancino [2] use the previous-tick interpolation and get

$$\begin{aligned} p^1_n(t) &:= \sum_{i=1}^{n-1} p^1(t^1_i) I_{[t^1_i, t^1_{i+1})}(t), \\ p^2_n(t) &:= \sum_{j=1}^{n-1} p^2(t^2_j) I_{[t^2_j, t^2_{j+1})}(t). \end{aligned} \tag{2.2.7}$$

Malliavin and Mancino [2] defines $I^1_i := [t^1_i, t^1_{i+1})$ and $J^1_j := [t^2_j, t^2_{j+1})$ and the returns by $\delta_{I^1_i}(p^1) := p^1(t^1_{i+1}) - p^1(t^1_i)$ and $\delta_{J^2_j}(p^2) := p^2(t^2_{j+1}) - p^2(t^2_j)$. Then the Fourier coefficients of the price process through use of a simple function approximation becomes

$$\begin{aligned} \mathcal{F}(dp^1_n)(k) &\approx \frac{1}{2\pi} \sum_{i=1}^{n-1} \exp(-ikt^1_i) \delta_{I^1_i}(p^1), \\ \mathcal{F}(dp^2_n)(k) &\approx \frac{1}{2\pi} \sum_{j=1}^{n-1} \exp(-ikt^2_j) \delta_{I^2_j}(p^2). \end{aligned} \tag{2.2.8}$$

By combining (2.2.2), (2.2.6) and (2.2.8) together, we get

$$\int_0^{2\pi} \Sigma^{ij}(t)dt = \frac{1}{2N+1} \sum_{|s|\leq N} \sum_{j=1}^{n-1} \sum_{i=1}^{n-1} e^{is(t^1_i - t^2_j)} \delta_{I^1_i}(p^1) \delta_{I^2_j}(p^2) \tag{2.2.9}$$

for large n, N.

Algorithm 2 Complex Exponential Fourier Transform

Require:

1. (n x m) matrix P of asynchronously sampled price
2. (n x m) matrix T of asynchronously sampled times

Re-scale the time $[t_{min}, t_{max}] \rightarrow [0, 2\pi]$

I. Extract trading times and prices

for i = 1 to m **do**

I.1. Slice the non-uniformly re-scaled sampled times for the i^{th} object

$\tilde{\tau} \leftarrow \tau(i)$

I.2. Slice the sampled data indexing the times for the i^{th} object

$\tilde{\varphi} \leftarrow \ln(p(\tilde{\tau}))$

I.3. Compute the returns

$\delta_j \leftarrow \tilde{\varphi}(\tilde{\tau}_{j+1}) - \tilde{\varphi}(\tilde{\tau}_j)$

II. Compute Fourier coefficients for all values of k

$\tilde{c}_k^+ \leftarrow \sum_{j=1}^{n-1} e^{ik\tilde{\tau}_j}\delta_j$

$\tilde{c}_k^- \leftarrow \sum_{j=1}^{n-1} e^{-ik\tilde{\tau}_j}\delta_j$

I.3. Gather Fourier coefficients for i^{th} object for all values values of k

$c^+(i) \leftarrow \tilde{c}^+$

$c^-(i) \leftarrow \tilde{c}^-$

end for

Compute the integrated volatility and co-volatility over the time window for objects i and j

$\Sigma_{ii} \leftarrow \frac{1}{|K|}\sum_{k\in K}[c_k^+(i)c_k^-(i)]$

$\Sigma_{ij} \leftarrow \frac{1}{|K|}\sum_{k\in K}[c_k^+(i)c_k^-(j)]$

$\Sigma_{ji} \leftarrow \Sigma_{ij}$

$R_{ij} \leftarrow \frac{\Sigma_{ij}}{\sqrt{\Sigma_{ii}}\sqrt{\Sigma_{jj}}}$

return (Σ, R)

Algorithm 2 [2] was provided by [17] and [18].

Remark 2.2.1 *The denominator of the scaling factors for $a_0(\Sigma)$ and $\mathcal{F}(\Sigma)(0)$ depends on how many Fourier coefficients we are summing over. For the Trig Fourier transform, we sum from $\sum_{s=n_0}^{N}$ therefore has $N + 1 - n_0$ while the Fourier transform, we sum from $\sum_{|s|\leq N}$ therefore has $2N + 1$.*

[2] The pair-wise implementation can be found in ftcorr.R.

2.3 Hayashi-Yoshida Estimator

2.3.1 Derivation

We present the estimator proposed by Hayashi and Yoshida [3]. This estimator has the ability to deal with asynchronous observations without the need to synchronize the data beforehand. The estimator is an adjustment on the well known realized covariance estimator defined as

$$V := \sum_{i=1}^{n-1} (P^1_{t_{i+1}} - P^1_{t_i})(P^2_{t_{i+1}} - P^2_{t_i}). \tag{2.3.1}$$

The realized covariance estimator has the well known property that as the Mesh tends towards 0 (i.e. $max_{1\leq i\leq n-1}|t_{i+1} - t_i| \to 0$), then $V \to \int_0^T \Sigma^{ij}(t)dt$ in probability. As Hayashi and Yoshida point out [3], there are two crucial issues regarding the implementation of the realized covariance estimator. The first is that actual transaction data is asynchronous. Secondly, due to the asynchrony, a significant portion of the original data set will be missing at pre-specified grid points. Therefore, in order to use (2.3.1), we must choose a common interval h first, and impute or interpolate the missing observations in some way [3] - this is commonly referred to as synchronizing the data. The first thing to notice is that the estimate V heavily depends on the value of h we pick. Additionally, as we have mentioned before Barucci and Renò [7] found that linear interpolation induces a bias. Malherbe [15] points out that the common intervals h need not be the same length, however we note that it is important for the intervals to be common. Otherwise the contribution in (2.3.1) will be 0.

Hayashi and Yoshida proposed a cumulative covariance estimator which is free from the need to synchronize the data beforehand. The assumptions required are that the price process follows the one-dimensional Itô process

$$dp^l = \mu^l dt + \sigma^l dW^l, \quad l = 1, 2, \tag{A-III}$$

with $d\langle W^1, W^2\rangle_t = \rho dt$ where $\rho \in (-1, 1)$ is an unknown deterministic function, $p^l(0) > 0$ is a constant, μ^l is a progressively measurable function and $\sigma^l > 0$ is a deterministic and bounded function [3]. Furthermore, for the sampling times. Let $T \in (0, \infty)$ be an arbitrary terminal time for observing the price processes. Let $\Pi^1 := (I^i)_{i=1,2,\dots}$ and $\Pi^2 := (J^j)_{j=1,2,\dots}$ be the sets of random intervals which partition $(0, T)$ for price process 1 and 2 respectively. Let $T^{1,i} := \inf\{t \in I^{i+1}\}$ represent the i^{th} observation time of P^1 and $T^{2,j} := \inf\{t \in J^{j+1}\}$ be that of P^2. Let n be the size of Π^1 and Π^2. We assume that the sampling intervals $\Pi := (\Pi^1, \Pi^2)$ satisfy the following

$$\begin{aligned} &(i).\ (I^i) \text{ and } (J^i) \text{ are independent of } P^1 \text{ and } P^2. \\ &(ii).\ As\ n \to \infty,\ \max_i |I^i| \vee \max_j |J^j| \to 0. \end{aligned} \tag{A-IV}$$

where $|I|$ is the length of an interval I [3]. Hayashi and Yoshida define the cumulative covariance estimator as

$$U_n := \sum_{i=1}^{n}\sum_{j=1}^{n}\Delta P^1(I^i)\Delta P^2(J^j)1_{\{I^i\cap J^j\neq\emptyset\}}. \tag{2.3.2}$$

This leads to Theorem 3.1 in [3]:

Theorem 2.3.1 *Suppose assumption* (A-IV) *holds*

(a) If $sup_{0\leq t\leq T}|\mu_t^l| \in L^4$, $l = 1, 2$, *then* $U_n \to \theta$ *as* $n \to \infty$.
(b) If $sup_{0\leq t\leq T}|\mu_t^l| < \infty$ *almost surely.* $l = 1, 2$, *then* U_n *is consistent for* θ, *that is,* $U_n \to \theta$ *in probability as* $n \to \infty$.

Where $\theta := \int_0^T \sigma^1\sigma^2\rho dt = \langle P^1, P^2\rangle_T$.

Remark 2.3.1 *We note that* (2.3.2) *has* n. *However the size of* Π^i *and* Π^j *need not be equal. We could have set* N_i *and* N_j *to be the size of* Π^i *and* Π^j *respectively [16].*

2.3.2 Numerical Implementation

For the case when $i = j$ the computation simply becomes the quadratic variation [16]. Therefore,

$$\int_0^T \Sigma^{ii}(t)dt = \sum_{i=1}^{N_i}[\Delta P^1(I^i)]^2. \tag{2.3.3}$$

For the case when $i \neq j$, we use Kanatani's weighted realized volatility [19] defined as

$$\begin{aligned}\int_0^T \Sigma^{ij}(t)dt &= \Delta P^{i'} W \Delta P^j \\ &= \sum_{k=1}^{N_i}\sum_{l=1}^{N_j} w_{kl}\Delta P^i(I^k)\Delta P^j(J^l),\end{aligned} \tag{2.3.4}$$

where

$$\Delta P^i = \begin{bmatrix} P^i(t_1^i) - P^i(t_0^i) \\ \vdots \\ P^i(t_{N_i}^i) - P^i(t_{N_i-1}^i)\end{bmatrix}, \; W = \begin{bmatrix} w_{11} & \dots & w_{1N_j} \\ \vdots & \ddots & \vdots \\ w_{N_i1} & \dots & w_{N_iN_j}\end{bmatrix}.$$

The weights for Hayashi Yoshida are given by

$$w_{kl} = \begin{cases} 1 & if\ (t_{k-1}^i, t_k^i]\cap(t_{l-1}^j, t_l^j] \neq \emptyset \\ 0 & otherwise.\end{cases} \tag{2.3.5}$$

Remark 2.3.2 *Kanatani's weighted realized volatility can also be used for Malliavin and Mancino's Fourier estimator. See [19] or [16].*

Algorithm 3 Hayashi Yoshida

Require:

1. (n x m) matrix P of asynchronously sampled price
2. (n x m) matrix T of asynchronously sampled times

Loop through every element of Σ [m x m]
for i = 1 to m **do**
  I.1. Compute the returns
  $\delta_i \leftarrow \ln(p^i(t_k)) - \ln(p^i(t_{k-1}))$
  for j = 1 to m **do**
    $\delta_j \leftarrow \ln(p^j(t_k)) - \ln(p^j(t_{k-1}))$
    I.2. Compute Kanatani's weight matrix for the i-th and j-th stock
    $W \leftarrow$ Kanatani weight for HY
    I.3. Compute Σ_{ij}
    $\Sigma_{ij} \leftarrow \delta_i' W \delta_j$
  end for
end for
II. Compute the correlation matrix
$R_{ij} \leftarrow \frac{\Sigma_{ij}}{\sqrt{\Sigma_{ii}}\sqrt{\Sigma_{jj}}}$
return (Σ, R)

Algorithm 3 [3] was provided by [18].

[3] The pair-wise implementation can be found in ftcorr.R.

Chapter 3

Monte Carlo Experiments

Monte Carlo experiments are conducted to identify the differences between the two estimator in hope to determine which estimator is better between the two.

3.1 Effect of missing observations

The first experiment focuses on how the two estimators differ when asynchronicity is induced by down-sampling the price path. The experiment is conducted by simulating 10,000 seconds from a bivariate Geometric Brownian motion with daily parameters $\mu_1 = 0.01, \mu_2 = 0.01, \sigma_1^2 = 0.1, \sigma_2^2 = 0.2, \rho_{12}$ ranging from $(-1, 1)$ and a starting price of R100. The non-synchronicity is achieved by randomly sampling a percentage of the observations from each sample path and removing them. The Geometric Brownian motion satisfies the following system of SDEs

$$\frac{dS_i(t)}{S_i(t)} = \mu_i dt + \sigma_i dW_i(t), \quad i = 1, 2. \tag{3.1.1}$$

Figure 3.1 [1] (a), we see that both MM (blue dotted line) and HY (red dotted line) perfectly recover the induced correlation (black dotted line) for the synchronous case. From figure 3.1 (b) through to (d), it is clear that as the level of asynchrony increases, MM appears to have a downward bias towards zero which [12] attributes to the Epps effect [4] while HY recovers the induced correlation regardless of the level of asynchrony.

Hayashi and Yoshida claim that the Epps effect is a bias that arises from the estimator for which their estimator is immune to [3]. Looking at figure 3.1, this seems to be the case. However, this goes against the findings of [6], [12], [13], [20]. The current literature has identified the main sources for the Epps Effect to be: smaller sampling intervals [4], [20], lead-lag [6], [12] and asynchronicity [12], [13]. Closed-form expressions recovering the Epps Effect can be found in [6], [20] - indicating that the Epps Effect is not a bias from the

[1] Figure 3.1 can be reproduced using MissingData.R.

estimator. This, in turn, means that it is the HY that is upward biased even though it recovers the induced correlation.

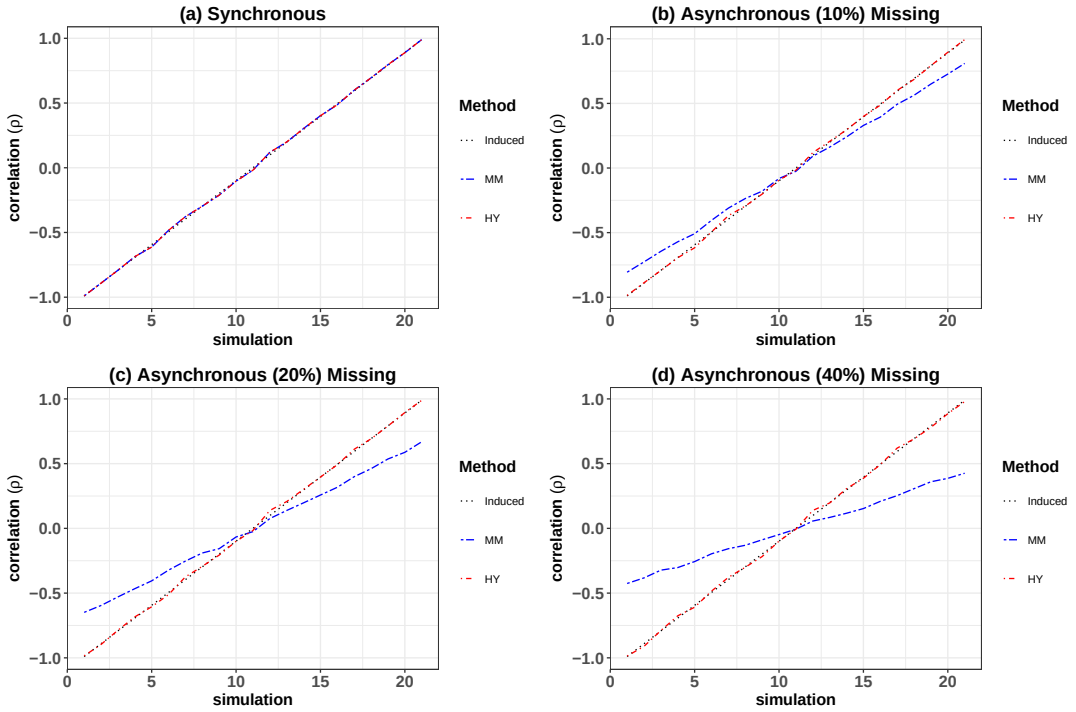


FIGURE 3.1: Comparing different levels of missing data to demonstrate the bias between the MM and HY estimators. Concretely, (a) through to (d) show 0%, 10%, 20% and 40% of each sample path replaced by missing data to replicate asynchrony. As per the figure legend, the blue dotted line is the MM estimator, estimated using Algorithm 2; the red dotted line is the HY estimator, estimated using Algorithm 3 and the black dotted line is the induced correlation of the GBM. Each simulation with varying correlation is done by simulating 10,000 seconds from a bivariate GBM satisfying (3.1.1) and numerically simulated using Algorithm 10. The figure shows that if the data is discrete and asynchronous then the MM estimator is most appropriate, however, if the data is sampled discretely from an underlying continuous-time GBM then the HY estimator is more appropriate. We argue that the Epps Effect from asynchrony is real and as such due to the discrete nature of financial market trades one should use the MM estimator rather than the HY estimator.

This experiment although recovers the Epps effect arising from asynchrony, is not an experiment conducted on a truly asynchronous process. It is rather a missing observation experiment. Therefore, the argument is that the HY estimator is the better estimator of the two if one believes that the observed prices in the market are discrete samples of an underlying continuous stochastic process and that asynchrony is a missing data problem. Then the HY estimator will be able to reproduce the true underlying correlation between

the assets by allowing multiple contributions to the estimator, which would imply the MM estimator has a downward bias attached to it. On the other hand, the argument is that the MM estimator is the better estimator of the two if one is of the belief that the world is not a missing data problem from an underlying synchronous continuous stochastic process; but is rather disconnected, discrete and asynchronous where the events and their relationships to each other are the fundamental entities and measurables of the finance world. Then the MM estimator will produce the true correlation in the system as it is lossless interpolation between the events. This would imply that the HY estimator will have an upward bias that is caused by the multiple contributions [17].

We argue the Epps effect is a fundamental property of financial market data which is not picked up by the HY estimator, and furthermore, TAQ data is truly discontinuous [21], discrete and asynchronous events which although does not fit into the framework of the two estimators; MM is the best tool we have for studying co-movements between discrete events due to the lossless interpolation. Therefore we argue MM is the preferred estimator of the two when it comes to studying high-frequency data.

Additional issues regarding the HY estimator is pointed out in [10], [11]. The first issue with the HY estimator is that when the processes are highly asynchronous, the HY estimator deletes observations through its multiple contributions (e.g. Fig. 1 in [11]), therefore it does not utilise all available observations. Furthermore, a critical assumption underlying the HY estimator is that the correlation between two assets does not extend beyond the intervals where returns fully or partially overlap. Meaning that information regarding the correlation is fully accounted for when a price update arrives. This assumption does not hold in practice and causes the HY estimator to be biased as shown by [10].

3.2 Effect of the SDE

The first experiment found that the two estimators differ under asynchrony, and the level of asynchrony determines how different the estimators behave. To further gain insight into the two estimators, various stochastic processes are studied to identify alternative situations where these two estimators differ.

The second experiment focuses on if alternative stochastic processes will cause the two estimators to differ. To this end, the Merton model, Variance Gamma, GARCH (1,1) and Ornstein Uhlenbeck will be used to compare the two estimators.

The bivariate Merton model satisfies the following system of SDEs

$$\frac{dS_i(t)}{S_i(t-)} = \mu_i dt + \sigma_i dW_i(t) + dJ_i(t), \quad i = 1, 2. \tag{3.2.1}$$

where $\text{corr}(dW_1, dW_2) = \rho$. The J_i are independent of the W_i with piece-wise constant paths [22]. J is defined as

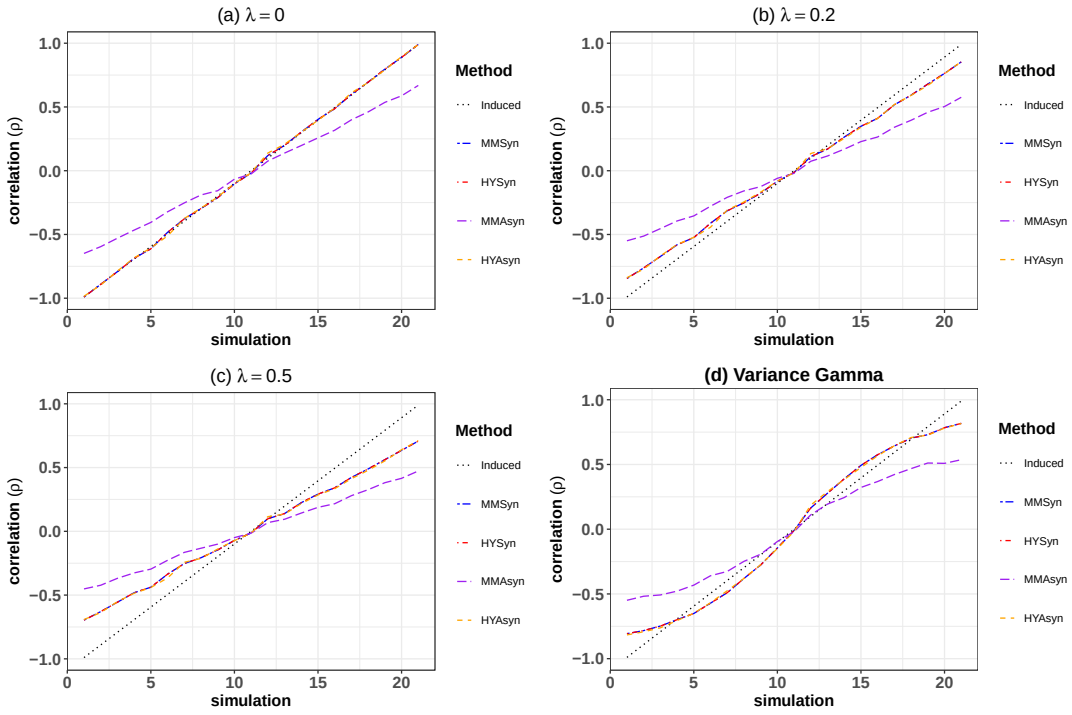


FIGURE 3.2: Comparing various stochastic processes to demonstrate the effect on the MM and HY estimators. Concretely, (a) shows a pure diffusion process, (b) and (c) show a jump-diffusion process and (d) shows a pure jump process. As per the figure legend, the blue and purple dotted lines are the MM estimator estimated using Algorithm 2 under synchronous and asynchronous observations respectively. The red and orange dotted line is the HY estimator estimated using Algorithm 3 under synchronous and asynchronous observations respectively. The black dotted line is the induced correlation from the various stochastic processes. (a) through to (c) is 10,000 seconds simulated from a bivariate Merton model satisfying (3.2.1) and numerically simulated using Algorithm 11. (d) is from a bivariate Variance Gamma model satisfying (3.2.3) and numerically simulated using Algorithm 13. The asynchrony is induced by down-sampling 20% of the observations from each sample path. The figure shows that both MM and HY produce the same estimates regardless of the underlying process and that it seems the difference between them arises due to asynchrony. Under asynchrony, HY recovers the synchronous estimates while the correlation for MM drops.

$$J_i(t) = \sum_{j=1}^{N(t)} (Y_j - 1), \tag{3.2.2}$$

where $N(t)$ is a Poisson process with $Y_j \sim LN(a, b)$ i.i.d and also independent of $N(t)$.

The daily parameters used for the Merton model are $\mu_1 = 0.01$, $\mu_2 = 0.01$, $\sigma_1^2 = 0.1$, $\sigma_2^2 = 0.2$, ρ_{12} ranging from $(-1, 1)$, $a_1 = 0$, $a_2 = 0$, $b_1 = 100$, $b_2 = 100$ and varying λ to move from a pure diffusion process to a jump-diffusion process. A sample path of 10,000 seconds is simulated starting at R100. This model will determine if the two estimators differ due to the effect of jumps.

The bivariate Variance Gamma (VG) process satisfies the following SDEs:

$$S_i(t) = U(t) - D(t), \quad i = 1, 2, \tag{3.2.3}$$

with U and D being independent gamma processes satisfying

$$\begin{aligned} U(t_{i+1}) - U(t_i) &\sim \text{Gamma}(\alpha(t_{i+1} - t_i), \beta), \\ D(t_{i+1}) - D(t_i) &\sim \text{Gamma}(\alpha(t_{i+1} - t_i), \beta). \end{aligned} \tag{3.2.4}$$

U and D are limited to have the same shape and scale parameters allowing an alternative representation $W(G(t))$ where W is a standard Brownian motion, G a gamma process [22] and $\text{corr}(dW_1, dW_2) = \rho$ ranging from (-1, 1). A sample path of 10,000 seconds is simulated starting at R100 with daily parameters $\mu_1 = \mu_2 = 0.01$, $\sigma_1^2 = 0.1$, $\sigma_2^2 = 0.2$ and $\beta_1 = \beta_2 = 1$. This model will determine if a pure jump process will cause the estimators to differ.

The bivariate GARCH (1,1) model satisfies the following SDEs:

$$dp_i(t) = \sigma_i(t) dW_i(t), \quad i = 1, 2, \tag{3.2.5}$$

and

$$\begin{aligned} d\sigma_1^2(t) &= \theta_1[w_1 - \sigma_1^2]dt + \sqrt{2\lambda_1\theta_1}\sigma_1^2(t)dW_3(t), \\ d\sigma_2^2(t) &= \theta_2[w_2 - \sigma_2^2]dt + \sqrt{2\lambda_2\theta_2}\sigma_2^2(t)dW_4(t). \end{aligned} \tag{3.2.6}$$

where $\text{corr}(dW_1, dW_2) = \rho$ ranges from (-1, 1). We simulate a sample path of 10,000 seconds starting at R100 using the parameters from [12], [23] i.e. $\theta_1 = 0.035$, $\theta_2 = 0.054$, $w_1 = 0.636$, $w_2 = 0.476$, $\lambda_1 = 0.296$ and $\lambda_2 = 0.48$ [2]. This model will determine if stochastic volatility and volatility clustering will cause the estimators to differ.

The bivariate Ornstein Uhlenbeck process satisfies the following SDEs:

$$dp_i(t) = \theta_i(\mu_i - p_i(t))dt + \sigma_i dW_i(t), \quad i = 1, 2, \tag{3.2.7}$$

[2] We note the SDE specified by [12] is different to what [23] has, we followed [23].

where corr$(dW_1, dW_2) = \rho$ ranges from (-1, 1). A sample path of 10,000 seconds is simulated starting at R100 with parameters $\mu_1 = \mu_2 = 100$, $\sigma_1^2 = 0.1$, $\sigma_2^2 = 0.2$, $\theta_1 = 0.035$ and $\theta_2 = 0.054$. This model will see if the two estimators differ under mean-reversion.

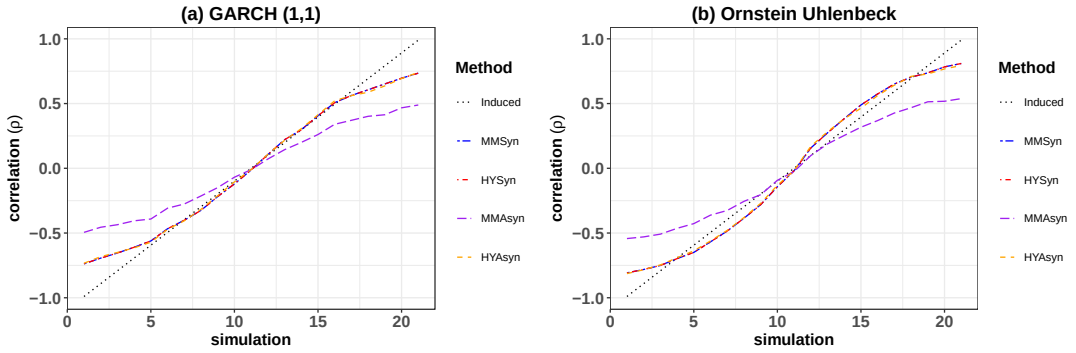


FIGURE 3.3: Comparing the two estimators based on volatility clustering (a) and mean-reversion (b). As per the figure legend, the blue and purple dotted lines are the MM estimator estimated using Algorithm 2 under synchronous and asynchronous observations respectively. The red and orange dotted line is the HY estimator estimated using Algorithm 3 under synchronous and asynchronous observations respectively. The black dotted line is the induced correlation from the various stochastic processes. (a) is 10,000 seconds simulated from a bivariate Geometric GARCH(1,1) satisfying (3.2.5) and numerically simulated using Algorithm 12. (b) is from a bivariate Geometric Ornstein Uhlenbeck satisfying (3.2.7) and numerically simulated using Algorithm 14. The figure shows that both estimators produce the same estimates in the synchronous case while for the asynchronous case, HY recovers the synchronous estimates but the correlation drops for MM.

Figure 3.2 [3] (a) through to (c), $\lambda_1 = \lambda_2 = 0$, 0.2 and 0.5 respectively. The asynchronicity is induced by down-sampling each price path by 20%. For all the plots in figure 3.2, the synchronous MM (blue dotted line) is the same as the synchronous HY (red dotted line). The asynchronous HY (orange dotted line) recovers the synchronous estimates while the asynchronous MM (purple dotted line) has a lower correlation estimate than the synchronous case. In figure 3.2 (a) through to (c), both the synchronous MM and HY estimators produce the same estimate which drops towards zero as λ increases. This drop in correlation is not due to any bias from the estimators but rather due to the fact that the jump process of the Merton model is independent of the underlying diffusion process. Therefore as the intensity of jumps increase, the impact from the independence, seeps through to change the correlation structure of the overall jump-diffusion process. This is the case because when the trades are synchronous, HY becomes the Realized Volatility (RV) [16] and the RV is consistent under jumps [24]. Therefore, the two estimators seem to

[3] Figure 3.2 can be reproduced using SDE1.R.

only differ under asynchrony and it is not dependent on the type of diffusion process.

Figure 3.3 [4], the asynchrony is induced by down-sampling each price path by 20%. Once again, the synchronous MM (blue dotted line) is the same as the synchronous HY (red dotted line) and the asynchronous HY (orange dotted line) recovers the synchronous estimates while the asynchronous MM (purple dotted line) has a lower correlation estimate than the synchronous case. Figure 3.3 provides the insight that neither volatility clustering nor mean-reversion causes the two estimators to differ under synchronicity. The two estimators only seem to produce different estimates under asynchronous conditions.

This experiment falsifies the idea that various stochastic processes will cause the two estimators to differ, rather it further validates that the two estimators differ only under asynchronous conditions, where the HY estimator is immune to the Epps effect brought through by asynchrony (under missing data conditions) while MM estimator picks up the Epps effect.

3.3 Effect of Asynchrony

In the previous experiments, asynchrony is induced by removing observations from the sample path. Although asynchrony is achieved; it is more of a missing data problem. The next experiment will follow a similar methodology used by [12] and [13]. The focus will be to achieve asynchronous sample paths that behave more like tick-by-tick TAQ data and investigate the effect the number of Fourier coefficients (N in (2.2.2)) has on the estimates.

Naturally, we first recover the results from [12]. The first thing to point out is that [12] has a different specification of the GARCH (1,1) compared to [23] - where the parameters were borrowed from. In figure 3.4 we present both specifications of the GARCH (1,1). The experiment in figure 3.4 is conducted by first simulating price paths of 86,400 seconds from a bivariate GARCH (1,1) with parameters from above. The asynchrony is induced by sampling the price path with an exponential inter-arrival time with a mean of 15 seconds and 45 seconds from asset 1 and asset 2 respectively. The synchronous case here is achieved by forcing the first time series to be observed at the same time as the second time series (i.e. the price paths were sampled with the same exponential inter-arrival time with a mean of 45 seconds). For each of the asynchronous and synchronous cases, we compute the correlation estimate using algorithm 2 for N ranging from 10 to 160. The effect achieved by studying the ranging Fourier coefficients (N) is that it allows various sampling frequencies to be studied through the relationship of the two from Fourier analysis.

[4] Figure 3.3 can be reproduced using SDE2.R.

From figure 3.4 [5], the Epps effect is clearly demonstrated. As N increases, so does the sampling frequency and from the asynchronous case (green dots) it is clear that the correlation drops as the sampling frequency increases. However, for the synchronous case (blue dots), the correlation does not drop as the sampling frequency increases. This is because the stochastic processes we have studied have dimensionless correlation (independent of the sampling intervals). The Epps effect recovered here is due to asynchrony which is different from the original Epps effect presented by Thomas Epps [4], where the correlation drops from synchronous observations as the sampling interval decreases. Researchers have investigated this - specifically [20] was able to derive an analytical expression for the Epps effect arising from smaller sampling intervals by decomposing the correlation of time scale Δt as a function of lagged autocorrelations and correlations of smaller time scales Δt_0. Additionally, [6] was able to further extend the results from [12] by analytically deriving the Epps effect arising from asynchrony as a function of Δt.

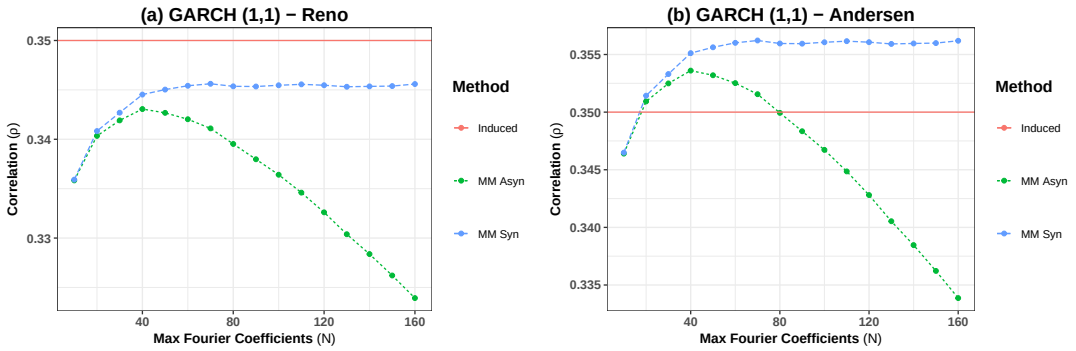


FIGURE 3.4: We recover the result from [12] using the complex exponential Fourier Transform 2. The average correlation as a function of the sampling frequency N in (2.2.2). Concretely, the asynchronous sample paths for (a) and (b) are exponential inter-arrival time samples from 86,400 seconds of simulated data. The exponential inter-arrival times have a mean of 15 seconds and 45 seconds respectively for asset 1 and asset 2. The synchronous sample paths for (a) and (b) are achieved by forcing the first time series to be observed at the same times as the second time series. (a) is simulated by adjusting (3.2.5) to how [12] defined the SDE and implemented by adjusting algorithm 12 accordingly. (b) is simulated from (3.2.5) using algorithm 12. As per the figure legend, the green dots and blue dots are the asynchronous and synchronous sample paths estimated using algorithm 2 respectively. The orange line is the induced correlation between (3.2.5). The results are obtained through 10,000 replications.

Drawing attention back to comparing the two estimators, we modify the experiment slightly. Specifically, for figure 3.5, the experiment is conducted

[5] Figure 3.4 can be reproduced using Reno Recovery.R.

by first simulating price paths of 10,000 seconds from the various stochastic processes from above, using their respective parameters as before [6]. The asynchrony is induced by sampling the first asset with an exponential inter-arrival time with mean 30 seconds and the second asset with a mean of 45 seconds. The synchronous case here is achieved by forcing the first time series to be observed at the same time as the second time series. The rationale behind adjusting the experiment is so that the Nyquist frequency can be indicated.

From figure 3.5 [7], it is clear that for the MM estimates, the correlations decrease for the asynchronous case as the number of Fourier coefficients (N) increase; whereas for the synchronous case the correlations become closer to the synchronous HY estimates as N increases. Additionally, the error bars calculated to be the standard deviation from the estimates decrease as N increases indicating that the estimates become more accurate with more Fourier coefficients. The HY estimates are not a function of N, but rather it a baseline to compare the MM estimate against. For figure 3.5 (a) through to (e), the asynchronous HY estimate recovers the synchronous HY estimate which is expected as Hayashi and Yoshida have claimed that their estimator is immune to the Epps effect. Oddly enough, the HY estimator demonstrates an Epps effect when using the Ornstein Uhlenbeck process. This was not picked up by the experiments before and a possible explanation for this is due to how this experiment is set up. In the previous experiments asynchrony was induced through a missing data manner, while in this experiment, the asynchrony is induced through exponential inter-arrival times to sample the price paths. Combined with the mean-reversion from the OU process, the sampling may have picked up different co-movements between the price paths. Another possible explanation is due to the combination of sampling method and mean-reversion, spurious lead-lag relations may have arisen due to the high levels of asynchrony which is another source for the Epps effect as investigated by [6], [12]. This further highlights the downfall of the HY estimator for high-frequency finance. Although the HY estimate may be immune to the Epps effect in a missing data manner, it is not immune to the Epps effect arising from lead-lag [10] nor from smaller synchronous sampling intervals [4] as seen in real financial data in later sections. Additionally, when levels of asynchrony is high - common for high-frequency data, the HY deletes observations [11] and therefore it is not well suited to study the co-movement between events.

[6] The Merton Model has $\lambda_1 = \lambda_2 = 0.2$

[7] Figure 3.5 can be reproduced using Reno Extended.R.

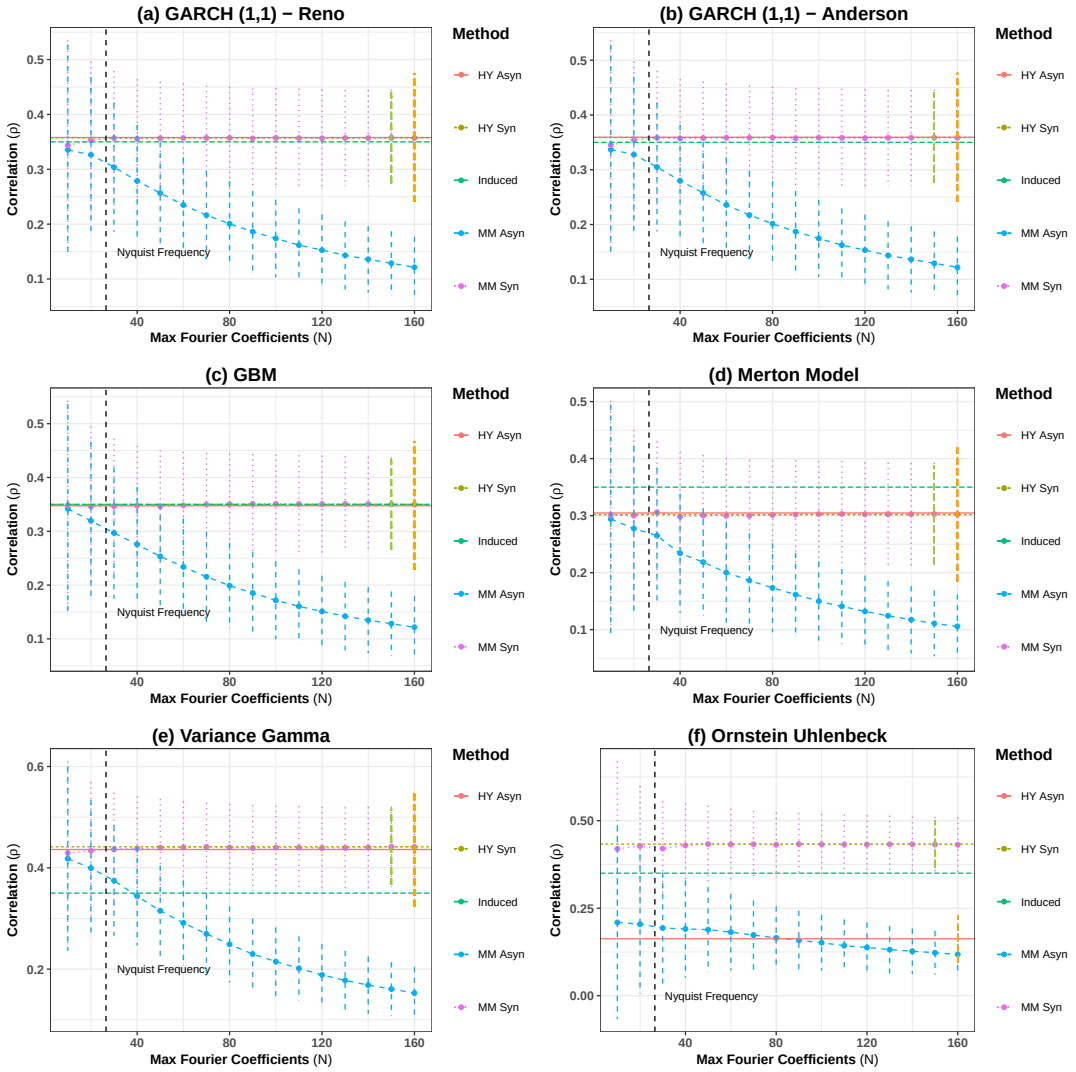


FIGURE 3.5: Comparing the two estimators by introducing asynchrony through sampling the time series with exponential inter-arrival times. The parameters used are the same as the previous experiments. We note that (a) uses the specification of GARCH (1,1) by [12] while (b) uses the specification of [23]. As per the legend, the synchronous (pink dots) and asynchronous (blue dots) average MM estimates using algorithm 2 differ more as the number of Fourier coefficients (N) increase. The synchronous (dark green line) and the asynchronous (orange line) average HY estimates using algorithm 3 in general recover the same estimates except for (f) - the OU process. The error bars are plotted as the standard deviation of the estimates across the replications. The green line is the induced correlation for each of the processes set to be $\rho = 0.35$. Additionally, the Nyquist frequency is indicated by the black dashed line. The results are obtained through 1,000 replications.

From figure 3.5, the Nyquist frequency is calculated using the *average* sampling frequency, this is not the true cutoff required to avoid any aliasing.

The true cutoff required is computed by first finding the highest sampling frequency present in the data, then computing the corresponding Nyquist frequency. Using the true cutoff is how we compute all the MM estimates in this paper except in figure 3.4 and 3.5. The rationale behind this is that we are trying to study the co-movement between high-frequency events, therefore picking a lower N to avoid market microstructure noise will lead to the aliasing of the event data. Picking a lower N, in essence, creates a smoothing effect due to aliasing of higher frequencies which is useful to identify the true signal under the market microstructure noise argument [9], but for the context of identifying the co-movement of events, picking a lower N will be a fatal choice to make.

This means that using the appropriate cutoff for the asynchronous cases in figure 3.5, all the correlations for the MM estimate diminish to zero. This result combined with the drop in correlation as the % of missing data increased in figure 3.1, indicates the importance the level of asynchrony has in contributing towards the Epps effect [13].

A point to be noted is that although this paper argues for the efficacy of the MM estimator over the HY estimator in the high-frequency paradigm through an event-based view of the world, we have presented results from the classical continuous-time stochastic processes which is a slight inconsistency to the event-based view we are taking. This is because of the limited techniques present in the literature to simulate a price process. Therefore we do note an extension on our results is to study how the correlations behave when the process is simulated from a Hawkes process [25] which can hopefully provide more insight into the co-movement between events and their relation to the Epps effect. However, our results are not futile because we have recovered and validated the work from previous researchers and have further performed a wholistic comparison of the two estimators with the various SDEs.

Although we have argued for the efficacy of the MM estimator over the HY estimator through an event-based view of the world, other researchers have also argued for the efficacy of the MM estimator in the market-microstructure noise view of the world. Specifically, [9] shows that the MM estimator is unbiased for the contaminated price process by an appropriate choice of n and N; while [11] points out that the HY estimator is infeasible in the setting of market microstructure noise.

Finally, a subtle point to notice is that all the SDEs used in this paper has dimensionless correlation which does not depend on time and therefore we could not study the Epps effect arising from smaller sampling intervals using Monte Carlo experiments, but from figure 3.4 and 3.5, we use the Fourier methods to study smaller sampling intervals. This subtle difference is due to what [6] showed. The correlation arising from asynchrony not only depends on the level of asynchrony, but also on the sampling intervals chosen. Therefore it seems that the Epps effect arising from smaller sampling intervals and asynchrony have some form of relation, and therefore more rigorous

research into this aspect is required - by attempting to decompose these two factors contributing towards the Epps effect. However, a possible complication that will arise is that [6] analytically showed the Epps effect arising from asynchrony as a function of Δt given by:

$$\tilde{\rho}^{12}_{\Delta t} = c\left(1 + \frac{1}{\lambda \Delta t}\left(e^{-\lambda \Delta t} - 1\right)\right), \tag{3.3.1}$$

where the correlation only decreases and does not change signs. However, in section 5, we show that there seems to be a structural change in the correlation that it is not only decreasing but also becoming positively correlated. Indicating that there is more to this problem than what meets the eye.

Chapter 4

Data Engineering

Managing the data is a difficult problem which is often downplayed, therefore we begin by describing the challenges faced with managing high-frequency data and how they were resolved. Furthermore, we outline all the algorithms employed to create the datasets used in the analysis. All the aggregation methods were built from the bottom up to ensure a white box process is adapted for reproducible research.

4.1 Data Collection

4.1.1 Collecting Data From Bloomberg

The data obtained for the analysis consists of Trade and Quote (TAQ) data from 10 equities listed on the Johannesburg Stock Exchange (JSE). The period considered is the week from 31/05/2019 to 07/06/2019. The equities considered are FirstRand Limited (FSR), Shoprite Holdings Ltd (SHP), Absa Group Ltd (ABG), Nedbank Group Ltd (NED), Standard Bank Group Ltd (SBK), Sasol Ltd (SOL), Mondi Plc (MNP), Anglo American Plc (AGL), Naspers Ltd (NPN) and British Am. Tobacco Plc (BTI). These equities are chosen due to their high liquidity, because we are interested in high-frequency event data.

There exists various APIs one can use to collect data from Bloomberg. These are outlined in table 4.1 provided by [26]. Each of the methods presented in table 4.1 have their strengths and weaknesses depending on the type of permission access one has on the shared Bloomberg terminal.

API	**Benefits**	**Issues**
Manual (GUI)	Good for content discovery	Documentation of user actions difficult.
Excel Add-In	Good for human-directed data refresh. Next-best for exploration	Scripting though VBA is slow. Only flat, tabular data or hard-to-machine-read data structures
C API	Fast, can retrieve multidimensional data objects, scriptable	Requires admin rights; low-level language. Poor for exploration.
Python API	Easy language, can retrieve data objects, scriptable	Depends on C under hood; requires admin rights. Poor for exploration.
R API	Fast(er than Excel), easy language, mature IDE, can retrieve data objects, scriptable	R does not require admin rights, but installing packages do. Poor for exploration.

TABLE 4.1: Bloomberg APIs compared

The manual (GUI) and excel add-in are unreliable when it comes to extracting large datasets from Bloomberg; furthermore, it has the added complication of not being easily reproducible. Thus we will consider more dynamic programmatic methods such as C, Python and R. The main complication that arises with C and Python is that one might not have the administrative rights to run the APIs, therefore we opt for the R API to extract TAQ data from Bloomberg.

For the purpose of collecting TAQ data, the choice of R comes with its caveats, mainly:

- Writing the TAQ data into flat files results in large flat files, which can take several hours to complete depending on the memory available on the terminal.
- Large flat files are complicated to read in R and often require Java or C++ interfaces to speed up the process.

This issue is not trivial due to the large nature of TAQ data. For example, one of the more liquid tickers - Naspers (NPN) has 15,544,244 data points for a period of 6 months which results in a flat-file of 450-500MB. Therefore obtaining data for all 10 tickers translates to roughly 5GB of data, illustrating the non-trivial nature of this problem.

After identifying these issues, we realised the need to find a more efficient and easily reproducible process to overcome the issues presented above.

4.1.2 Updated HF-Data Pipeline

The focus in our approach was to have an easily reproducible process that spends minimal time on the Bloomberg terminal and can easily be extended. In figure 4.1, we illustrate the data pipeline.

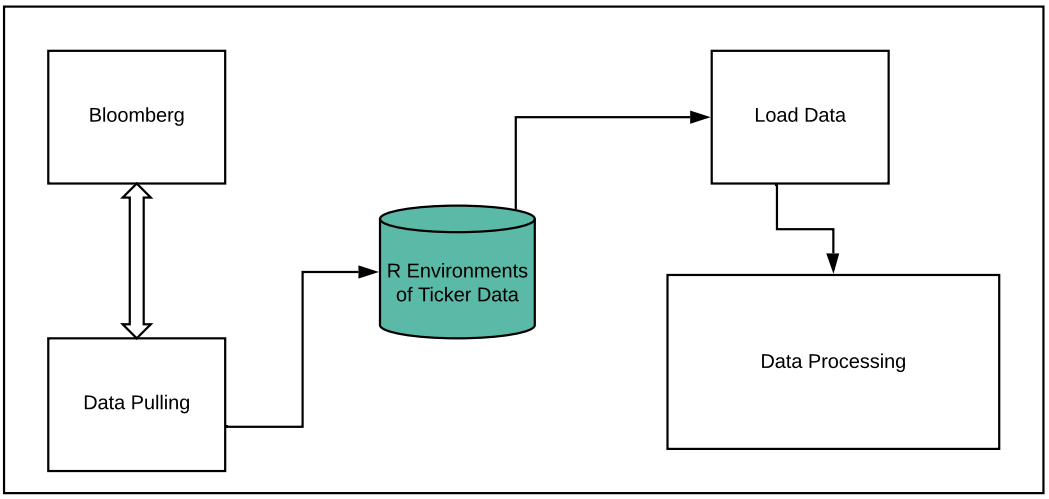


FIGURE 4.1: Data Collection Pipeline

The key factor that solves the issue of large files and long computation time is simply saving the data as R environments rather than flat files. The advantages achieved by doing so are exceptional, namely:

- R environments are significantly smaller when saving data compared to flat files. For example, the Naspers (NPN) ticker when saved to a flat-file results in a size of 500MB, while as an R environment, the size reduces to 47MB - a decrease of 90.6% in storage size.
- Reading in the R environment is significantly quicker - even more so than using third party interfaces such as Java or C++ and furthermore avoids any memory issues which arise from large file sizes.

Now the largest overhead left is simply the time it takes to extract data from Bloomberg since the other areas have been streamlined. To see the efficiency up the updated pipeline, we consider the file sizes and time spent in extracting and loading TAQ data. We initially pulled 24 tickers which took approximately 3 hours to extract from Bloomberg. Now saving the extracted data as an R environment resulted in a file of 557.6MB as opposed to the 4.9GB when saving the data as flat files. The point of significance is when we have to read the data. Loading the R environment takes on average 25-30 seconds while loading the flat files took several hours with no result [1]. Finally, the last advantage this approach presents is that it is ring-fenced within R and does not rely on third-party interfaces such as rJava or Rcpp [2].

The pipeline although advantageous, does present some potential pitfalls which we have not yet encountered. Specifically, if the R environment exceeds 2GB then reading in data might pose an issue. This can be pragmatically solved by writing various assets into their own R environment. However, this should never present itself as a real issue given the limitation of the Bloomberg terminal which only allows for 6 months of TAQ data to be extracted. Finally, the pipeline is designed to extract TAQ data, but this can easily be extended to pull other forms of data from Bloomberg.

4.2 Data Cleaning

4.2.1 Data Types

From the TAQ data, there are three types of observables: the bid, ask and actual trades. We are only interested in the actual trades as they form the price paths of interest. Furthermore, there are different trade types, examples include Automated Trade (AT), Late Trade (LT), Post Contra Trade (LC) and Indicative Auction Information (IP) [27]. For the analysis, we extract only the Automated Trades because only these form the continuous trading

[1] The reason it did not finish loading is because the file size was too large, coupled with the fact that R reads the entire dataset into RAM all at once, lack of RAM resulted in the loading to never finish. Solving this requires the package bigmemory to overcome the issue - the data gets read into the HDD, which comes with its own caveats.

[2] bigmemory is part of the Rcpp family.

process. The other trade types are after-hour trades (LT), correction of previous days published off book trade (LC) and an indicative auction price based on the volume maximising auction algorithm used to determine the auction uncrossing price (IP) [27] - which are irrelevant to the analysis.

4.2.2 Aggregation

An issue with Bloomberg data which is not present with Thomson Reuters data is that timestamps are only shown up to seconds, therefore there are multiple trades with the same timestamp - illustrated in figure 4.2. This poses an issue when using the MM and HY estimators - the two estimators require unique time stamps for each trade.

	times	type	value	size	condcode
3887	2018-12-28 12:26:39	TRADE	45337.15	162	AT
3888	2018-12-28 12:26:39	TRADE	45338.11	165	AT
3889	2018-12-28 12:26:39	TRADE	45338.11	200	AT
3890	2018-12-28 12:26:39	TRADE	45338.11	50	AT
3891	2018-12-28 12:26:39	TRADE	45338.11	53	AT
3892	2018-12-28 12:26:39	TRADE	45338.11	36	AT
3893	2018-12-28 12:26:39	TRADE	45338.11	169	AT
3894	2018-12-28 12:26:39	TRADE	45338.11	69	AT
3895	2018-12-28 12:26:39	TRADE	45338.11	35	AT
3896	2018-12-28 12:26:39	TRADE	45342.95	21	AT
3897	2018-12-28 12:26:39	TRADE	45342.95	19	AT
3898	2018-12-28 12:26:39	TRADE	45342.95	70	AT

FIGURE 4.2: Trades with the same time stamp.

To overcome this issue we first need to aggregate these "repeated" [3] trades. The aggregation algorithm is presented in algorithm 4 [4] and it uses a Volume Weighted Average Price (VWAP) method of aggregation. VWAP was employed because it gives a better representation of the data given by the fact that it weights each trade by the volume, which is directly linked to the price impact [28].

[3] The term repeated is used very loosely as these are not actually repeated trades.
[4] The implementation can be found in AsynchronousData.R.

Algorithm 4 Aggregation of Repeated Trades

Require:

1. T_i the trading times, $i = 1, ..., N$
2. S_i the observed prices, $i = 1, ..., N$
3. V_i the volume associated with the trade, $i = 1, ..., N$

Identify the unique trading times t_j^*, $j = 1, ...M$

Gather trades with the same trading times into a set J_j, $j = 1, ..., M$

Procedure for the j^{th} set:

1. Set
$$s_j^* = \frac{\sum_{i \in J_j} price_i * volume_i}{\sum_i volume_i}$$
2. Set $V_j^* = \sum_{i \in J_j} V_i$

return ($T^* = \{t_j^*\}_{j=1}^M, S^* = \{s_j^*\}_{j=1}^M, V^* = \{V_j^*\}_{j=1}^M$)

In figure 4.3 we demonstrate the output of algorithm 4 using one set of "repeated" trades J_j from figure 4.2.

	times	type	value	size
1220	2018-12-28 12:26:39	TRADE	45338.47	1049

FIGURE 4.3: Trades aggregated into a unique time stamp.

4.2.3 Overnight returns

The trading times for the JSE begin at 09:00 and end at 17:00 [5], while the closing auction begins at 16:50. Furthermore, the opening prices and closing prices are determined using a volume maximising auction algorithm to determine the uncrossing price, therefore these prices not determined by Market Orders (MO) hitting the Limit Order (LO) - which form the continuous trading process, but rather these prices follow a Walrasian equilibrium. Additionally, the opening prices and closing prices can differ vastly due to overnight information getting priced into the opening auction; therefore we opt to remove these overnight returns.

Removing the overnight returns is not a trivial task, and as such this process is performed as part of the data cleaning process. As a result, algorithm 1, 2 and 3 had to be adjusted to allow for logged-returns as an input, rather than the original prices [6]. In addition to the overnight return, we also opted to remove the first 10 minutes of the continuous trading session i.e. 09:00 to 09:10. This is because older trading algorithms are still calibrating during this period and thus the trades in this period will not be an accurate representation of the continuous trading session.

[5] We highlight the importance of pulling data with the right time zone UTC + 2 for South African data. We pulled UTC by mistake and thus our trading times are shifted back two hours.

[6] The pair-wise implementation can be found in ftcorr-RealData.R.

4.3 Creating Data Samples

To investigate the Epps effect, aggregation methods to create various sampling intervals are required. To this end, we will aggregate the TAQ data using calendar time aggregation methods; specifically the creation of our own bar data. Additionally, we will perform the novel task of aggregating TAQ data using intrinsic time aggregation methods; specifically using the framework provided by Derman [14] and our own pragmatic method of aggregation to highlight the differences between the two estimators. We highlight for all the aggregation methods, we first obtain the aggregated prices, then the conversion to returns is performed and finally the overnight returns are then removed.

4.3.1 Asynchronous Data

The first dataset to create is the cleaned version of the TAQ data, achieved by aggregating the trades using algorithm 4. Once the aggregation is complete for each ticker, they get merged into a data frame as shown in figure 4.4 where the non-trade times are represented with NaNs.

	Date	BTI	NPN
1467	2019-05-31 07:25:06	51133.41	NaN
1470	2019-05-31 07:25:09	51175.64	325700.0
1471	2019-05-31 07:25:10	NaN	325609.4
1472	2019-05-31 07:25:11	NaN	325697.2
1474	2019-05-31 07:25:13	NaN	325506.0
1476	2019-05-31 07:25:15	NaN	325556.0
1477	2019-05-31 07:25:16	51170.65	NaN
1486	2019-05-31 07:25:25	51172.69	NaN
1491	2019-05-31 07:25:30	51155.02	325501.0
1493	2019-05-31 07:25:32	NaN	325501.1
1496	2019-05-31 07:25:35	NaN	325446.1
1497	2019-05-31 07:25:36	NaN	325234.0

FIGURE 4.4: BTI and NPN Asynchronous Price Sample

The merging is achieved by first pre-populating a data frame with the highest available sampling frequency (1 second) over the period of consideration, then slotting the prices for each asset into their respective times and removing entire rows of NaNs afterwards.

To create the return matrix required, the returns are computed separately for each asset over each of the days considered, and the first return for each day takes on the time index of the second trade in that day, while the first trade for each day takes on NaN as a placeholder [7]. By computing the returns for each day separately, we have dealt with the over-night returns. The merging is then achieved in the same manner as the prices and the result is shown in figure 4.5.

	Date	BTI	NPN
1467	2019-05-31 07:25:06	-3.073838e-04	NaN
1470	2019-05-31 07:25:09	8.255380e-04	-1.350845e-04
1471	2019-05-31 07:25:10	NaN	-2.782109e-04
1472	2019-05-31 07:25:11	NaN	2.694873e-04
1474	2019-05-31 07:25:13	NaN	-5.870940e-04
1476	2019-05-31 07:25:15	NaN	1.536627e-04
1477	2019-05-31 07:25:16	-9.746177e-05	NaN
1486	2019-05-31 07:25:25	3.981549e-05	NaN
1491	2019-05-31 07:25:30	-3.453610e-04	-1.690235e-04
1493	2019-05-31 07:25:32	NaN	4.071573e-07
1496	2019-05-31 07:25:35	NaN	-1.690041e-04
1497	2019-05-31 07:25:36	NaN	-6.520136e-04

FIGURE 4.5: BTI and NPN Asynchronous Return Sample

The remaining aggregation methods will be computed using above asynchronous samples [8].

4.3.2 Calendar Time TAQ Data Aggregation

The calendar time aggregation is simply OHLCV data which is provided by Bloomberg; however, we opt to create our own bar for the main reason to append an additional VWAP column as another method to investigate the Epps effect. The creation of the bar data is done using algorithm 5 [9].

[7] after the computing the returns

[8] This is not an issue for the intrinsic time methods, because the aggregation done here is VWAP which is the same aggregation used in the intrinsic time samples.

[9] The implementation of all the aggregation methods are done using the function generate_data from TradeDataMain.R.

Algorithm 5 Bar Data

Require:

1. T_i the unique trading times, $i = 1, ..., N$
2. S_i the aggregated prices, $i = 1, ..., N$
3. V_i the volume associated with the trade, $i = 1, ..., N$
4. τ the bar length ▷ Units of time

Gather trades into their respective bars $\{\{J\}_j, j = 1, ..., M\}$ determined by τ.

for j = 1, ..., M **do**
  if j = 1 **then**
    Set $O_1 = S_1$
  else
    Set $O_j = C_{j-1}$
  end if
  Set $H_j = \max\{S_i \in J_j\}$
  Set $L_j = \min\{S_i \in J_j\}$
  Set $C_j =$ the last $S_i \in J_j$
  Set $V_j^* = \sum_{i \in J_j} V_i$
  Set

$$VWAP_j = \frac{\sum_{i \in J_j} S_i * V_i}{\sum_i V_i}$$

  Set $t_j^* = T_1 + j\tau$
end for
return $(t^* = \{t_j^*\}_{j=1}^M, O = \{O_j\}_{j=1}^M, H = \{H_j\}_{j=1}^M, L = \{L_j\}_{j=1}^M, C = \{C_j\}_{j=1}^M, V^* = \{V_j^*\}_{j=1}^M, VWAP = \{VWAP_j\}_{j=1}^M)$

The main difference between the bar data we created and the bar data extracted from Bloomberg is that Bloomberg's bar data clocks at exact minutes, whereas our bar data does not. This difference is due to the fact we wanted a function that can create bar data for any dataset given. Therefore to avoid data snooping the first opening price, we began the counter from the time of the first trade. This was a pragmatic choice because unlike Bloomberg which will always have a previous closing price to pull from, our finite dataset has its limitations.

Algorithm 5 [10] is presented for 1 asset, however the bar data for the analysis is for multiple assets. Therefore to make the bar data for more than one asset, the only point to note is that the T_1 used in $t_j^* = T_1 + j\tau$ is computed as the earliest trade of the day across all the assets across all the days considered [11]. Then algorithm 5 is applied for each asset.

Figure 4.6 and 4.7 shows the result of algorithm 5 applied to two assets and converting the closing price and VWAP price to returns respectively.

[10] The implementation of algorithm 5 can be found in SynchronousData.R.

[11] For assets which do not have an opening trade the same time as T_1, the opening price is set to NaN.

	Date	BTI	NPN
3	2019-05-31 07:20:40	2.877187e-04	1.210565e-03
4	2019-05-31 07:30:40	-7.691094e-05	-3.894474e-05
5	2019-05-31 07:40:40	0.000000e+00	1.528998e-06
6	2019-05-31 07:50:40	-1.100197e-03	-2.892516e-04
7	2019-05-31 08:00:40	2.124791e-04	4.293249e-04
8	2019-05-31 08:10:40	-5.753755e-04	5.539416e-05
9	2019-05-31 08:20:40	-4.949077e-04	-5.973955e-04
10	2019-05-31 08:30:40	5.225367e-04	-3.936299e-04
11	2019-05-31 08:40:40	-7.337887e-04	3.695216e-04
12	2019-05-31 08:50:40	1.741469e-04	0.000000e+00
13	2019-05-31 09:00:40	7.351942e-04	-1.061915e-04

FIGURE 4.6: BTI and NPN 10 Minute Closing Bar Return Sample

	Date	BTI	NPN
3	2019-05-31 07:20:40	8.453442e-05	-5.304664e-05
4	2019-05-31 07:30:40	-6.605707e-05	3.290135e-05
5	2019-05-31 07:40:40	-6.128537e-05	4.483050e-05
6	2019-05-31 07:50:40	-4.867560e-04	2.917930e-04
7	2019-05-31 08:00:40	1.696547e-06	8.431387e-06
8	2019-05-31 08:10:40	-1.006159e-04	-2.962207e-05
9	2019-05-31 08:20:40	2.584443e-05	-7.957232e-05
10	2019-05-31 08:30:40	1.160736e-04	2.168126e-04
11	2019-05-31 08:40:40	1.341311e-03	4.901715e-04
12	2019-05-31 08:50:40	-3.054909e-04	-1.568611e-05
13	2019-05-31 09:00:40	-8.169586e-05	2.266272e-05

FIGURE 4.7: BTI and NPN 10 Minute VWAP Bar Return Sample

A point of detail to note about the creation of figure 4.6 and 4.7 is that the OHLCV is computed for each asset, and for assets which do not have any trades within a bar, that row of OHLCV is not computed and therefore skipped. Returns are then computed for each assets closing and VWAP for each day, then merged into a data frame in the same manner as the creation of asynchronous returns [12].

[12] We initially made the error for figure 4.7 whereby we computed the OHLCV based on returns, rather than computing the returns after obtaining the OHLCV for the prices.

4.3.3 Intrinsic Time TAQ Data Aggregation

Intrinsic time, also known as event time is a method to measure time based on events rather than the traditional chronological time we humans perceive. This method of aggregation presents some statistical advantages - namely the logged-returns are made more gaussian. Additionally, this is the realm "silicon traders" [13] operate in [29], and they account for a large proportion of the volume traded [30]. Therefore as a novel application, we will examine the correlation structure produced from the estimators under this paradigm.

Derman Framework

The first method we will use to aggregate TAQ data in intrinsic time is the framework provided by Derman [14], where each stock has its own trading frequency v_j. The added benefit from this framework is that it is a natural way to deal with the asynchrony from high-frequency data through the fact that each stock has its own trading frequency, and more importantly it provides an elegant link between intrinsic time and calendar time. However, this framework has its drawbacks (discussed in section 5.2).

To implement this method of aggregation, we first need the average trades per day for each stock $\bar{V}_j$ over the given data period considered (31/05/2019 - 07/06/2019).

Algorithm 6 Derman Framework

Require:

1. T_i the unique trading times, $i = 1, ..., N$
2. S_i the aggregated prices, $i = 1, ..., N$
3. V_i the volume associated with the trade, $i = 1, ..., N$
4. $v_j = \bar{V}_j/(Number\ of\ Buckets)$ the bucket size

Expand the number of observations by repeating each observation S_i as many times as V_i, resulting in $I = \sum_i V_i$ observations of S_i ▷ Expand such that the initial ordering of S_i is not lost

Set $\tau = 0$

while $\tau v_j < I$ **do**

$\tau = \tau + 1$

$\forall i \in [(\tau - 1)v_j + 1, \tau v_j]$, compute

$$P_\tau = \frac{\sum_i S_i * V_i}{\sum_i V_i}$$

end while

return $P = \{P_1, ..., P_M\}$ ▷ $M = \tau$ at end of while loop

[13] High Frequency Traders

The first point to note about algorithm 6 [14] is that it is for one trading day, and thus the remaining trades at the end of each day which do not have enough volume to form a bucket are discarded. This choice although deviates from the framework, is justifiable. This is because even though intrinsic time operates on a separate measurement of time, trading is still performed on calendar time and at the end of each day the "silicon traders" stop trading. Furthermore, due to the overnight period, the opening auction can shift the prices to a completely different level and thus combining the remaining trades at the end of each day with the first few trades of the next day is not coherent. Due to this choice, the samples created are not completely synchronous as expected from the framework. This is because $\bar{V}_j$ is computed as the average trades per day over the given period while day to day volume traded can be different, therefore some assets will have more (less) prices than the *Number of Buckets* due to the volume traded in that day being more (less *resp.*) than the average. Thus the non-trading times (in intrinsic time) are filled in with NaNs. Algorithm 6 is computed for each trading day separately, then combined afterwards. Finally, the overnight returns are removed in the same manner as before, by computing the returns for each day separately and combining it afterwards. The overnight returns are removed for the reason that humans operate in calendar time and overnight information can get priced into the opening auction, therefore changing the price level, resulting in a return that is not consistent with the continuous trading process.

Figure 4.8 shows the resulting intrinsic time return samples for the first trading day of multiple assets where the average trades per day are computed over the period (31/05/2019 - 07/06/2019).

	volume_ticks	BTI	NPN	AGL	MNP	SOL
1	1	0.000000e+00	0.000000e+00	0.000000e+00	0.0000000000	0.0000000000
2	2	-8.992127e-04	-3.150050e-03	3.844936e-04	-0.0068279225	-0.0070950912
3	3	8.811849e-04	-4.848182e-03	5.611877e-03	-0.0029603412	-0.0004149761
4	4	1.742509e-03	-4.085905e-03	-6.511903e-04	0.0008151005	0.0013554047
5	5	7.175808e-04	-2.698087e-03	-6.218975e-04	-0.0001262050	NaN
6	6	1.198676e-03	1.234084e-03	1.676421e-03	0.0008667726	NaN
7	7	3.219217e-03	4.380357e-03	6.004078e-05	0.0011208474	NaN
8	8	2.846767e-03	NaN	2.060114e-03	NaN	NaN
9	9	NaN	NaN	NaN	NaN	NaN
10	10	NaN	NaN	NaN	NaN	NaN

FIGURE 4.8: Multiple Ticker Derman Volume Buckets

[14] The implementation of algorithm 6 can be found in DermanFrameworkVolumeBuckets.R.

Lining Up Events

Due to the synchronicity of the Derman framework, the resulting correlation estimates are very similar for both estimators. Thus it is not particularly meaningful in the context of comparing our estimators. Therefore we employ another method of aggregating TAQ data in intrinsic time which preserves the asynchrony of events while retaining the benefit of gaussian returns.

Algorithm 7 Lining Up Events

Require:

1. T_i^j the unique trading times of the j^{th} asset, $i = 1, ..., N_j$
2. S_i^j the aggregated prices of the j^{th} asset, $i = 1, ..., N_j$
3. V_i^j the volume associated with the trade of the j^{th} asset, $i = 1, ..., N_j$
4. $v = \bar{V}/(Number\ of\ Buckets)$ the bucket size ▷ $\bar{V}$ is computed from the most liquid asset

Obtain a sequence of unique trading times $\{T^*\}_{i^*=1}^{N^*}$ across all j assets ▷ $N^* \geq N_j \quad \forall j$

Set $\tau = 0$

for i^* in T^* **do**

$A_j = \{S_k^j\}_{k=1}^{K^j}$ ▷ Storage set for the j^{th} asset. Expand $S_{i^*}^j$ as many times as $V_{i^*}^j$ and append to the end of the set.

while $\vec{K} > \vec{v}$ **do** ▷ While **any** of the storage sets are larger than v

$\tau = \tau + 1$

for j^{th} asset **do**

if $K^j > v$ **then**

$\forall k \in [1, v]$, compute

$$P_\tau^j = \frac{\sum_k S_k}{v}$$

Remove the first v S_k^j from A_j

else

$P_\tau^j = NaN$

end if

end for

end while

end for

return $P^j = \{P_1^j, ..., P_M^j\}$ ▷ $M = \tau$ at end of the loops

The motivation behind algorithm 7 [15] is that we want to capture events. This is because when there is a shock or major event, assets will "synchronise" in response to the event and once again "de-synchronise" after the event. Therefore, algorithm 7 works by collecting trades in calendar time until the bucket is full and a price is printed. Another difference now is that we line up

[15] The implementation of algorithm 7 can be found in LiningUpEventsVolumeBuckets.R.

the events based on the most liquid asset which provides a baseline "clock" for which assets are lined up accordingly.

$\hat{V}$ in algorithm 7 is computed the same way as $\hat{V}_j$ - the average trades per day over the period considered. The main difference between algorithm 7 is that the prices need to be computed for all assets at the same time, whereas the previous algorithms permitted prices for individual assets to be created then combined. Due to the nature of this aggregation, computing the returns is not as simple as before, as in we cannot simply use the function diff() anymore. This is because the function computes $x[(1 + lag) : n] - x[1 : (n - lag)]$, and therefore if there are no successive prices, the returns will not be computed. Thus to overcome this issue, we need to first extract the actual prices for each asset, compute the returns then place them back into their respective positions.

Removing the overnight returns follows the same process as for algorithm 6, we apply algorithm 7 for each trading day, then combine them afterwards. Furthermore, the remaining trades at the end of each day which cannot fill up a bucket gets discarded - for the same reason we discarded them in the Derman framework, to focus only on the continuous trading process. However, this poses an issue for the less liquid stocks when the *Number of Buckets* are smaller. This is because the bucket sizes become very large and therefore some of the less liquid stocks do not have enough trades to form two prices and thus returns cannot be computed for that day. Due to this issue, we will ignore the Calendar time equivalent of 1 hour bar samples and focus only on a bucket frequency of 48 and 480.

Figure 4.9 below illustrates a data sample using algorithm 7 with a bucket frequency of 48 and the basis ticker is FSR.

	volume_ticks	FSR	BTI	NPN	AGL	MNP	SOL	SBK
1	1	0.000000e+00	0.000000000	0.0000000000	0.0000000000	0.000000000	0.000000e+00	0.000000e+00
3	2	-4.487499e-03	NaN	NaN	NaN	NaN	NaN	NaN
4	3	-5.458751e-03	NaN	NaN	NaN	NaN	NaN	NaN
6	4	-6.014010e-03	NaN	NaN	NaN	NaN	NaN	NaN
7	5	NaN	NaN	NaN	NaN	NaN	NaN	NaN
8	6	-3.617174e-03	NaN	NaN	NaN	NaN	NaN	NaN
9	7	NaN	NaN	NaN	NaN	NaN	NaN	-1.215534e-02
10	8	1.021923e-03	NaN	NaN	NaN	NaN	NaN	NaN

FIGURE 4.9: Multiple Ticker Lining Up Events Volume Buckets

From the data samples in figure 4.8 and figure 4.9, we note that there is a striking difference between these samples. Namely, in figure 4.8, the trades are near synchronous while in figure 4.9, we have high levels of asynchrony - allowing us to gain further insights into the two estimators and how they compare.

Chapter 5

Data Science

Turning our attention to real financial data, we perform the novel application by studying the Epps effect through various methods of aggregating Trade and Quote (TAQ) data. Specifically, we compare calendar time based sampling with volume time sampling methods.

5.1 Calendar Time

For the calendar time sampling methods we focus on the closing prices and Volume Weighted Average Price for different sized bar data.

5.1.1 Closing Prices

Using the TAQ data, we create 1 minute, 10 minute and 1 hour OHLCV bar data for the period. From the Monte Carlo experiment, we know that the two estimators produce the same results when trades are synchronous; however, some of the less liquid stocks do not have any trades within a given bar and therefore the bar data created is not truly synchronous. Thus the MM and HY estimates will differ slightly with HY having a higher estimate. We apply the MM and HY estimators to the closing prices of the various bar data along with the raw TAQ data to see the effect of increasing the sampling frequency to its highest available frequency.

Figure 5.1 [1] [2] (a) though to (d), the MM estimator is applied to 1 hour, 10 minute, 1 minute closing bar data and TAQ data respectively. Figure 5.1 (e) through to (h), the HY estimator is applied to 1 hour, 10 minute, 1 minute closing bar data and TAQ data respectively. On each of the plots, the average magnitude of the correlation is shown. It is clear that the Epps effect is present for both estimators when considering the closing price samples. This shows that the Epps effect is present even at much higher sampling frequencies that what [4] originally considered.

[1] Figure 5.1 can be reproduced using Closing.R and TAQ.R.
[2] Figure 5.2 can be reproduced using VWAP.R and TAQ.R.

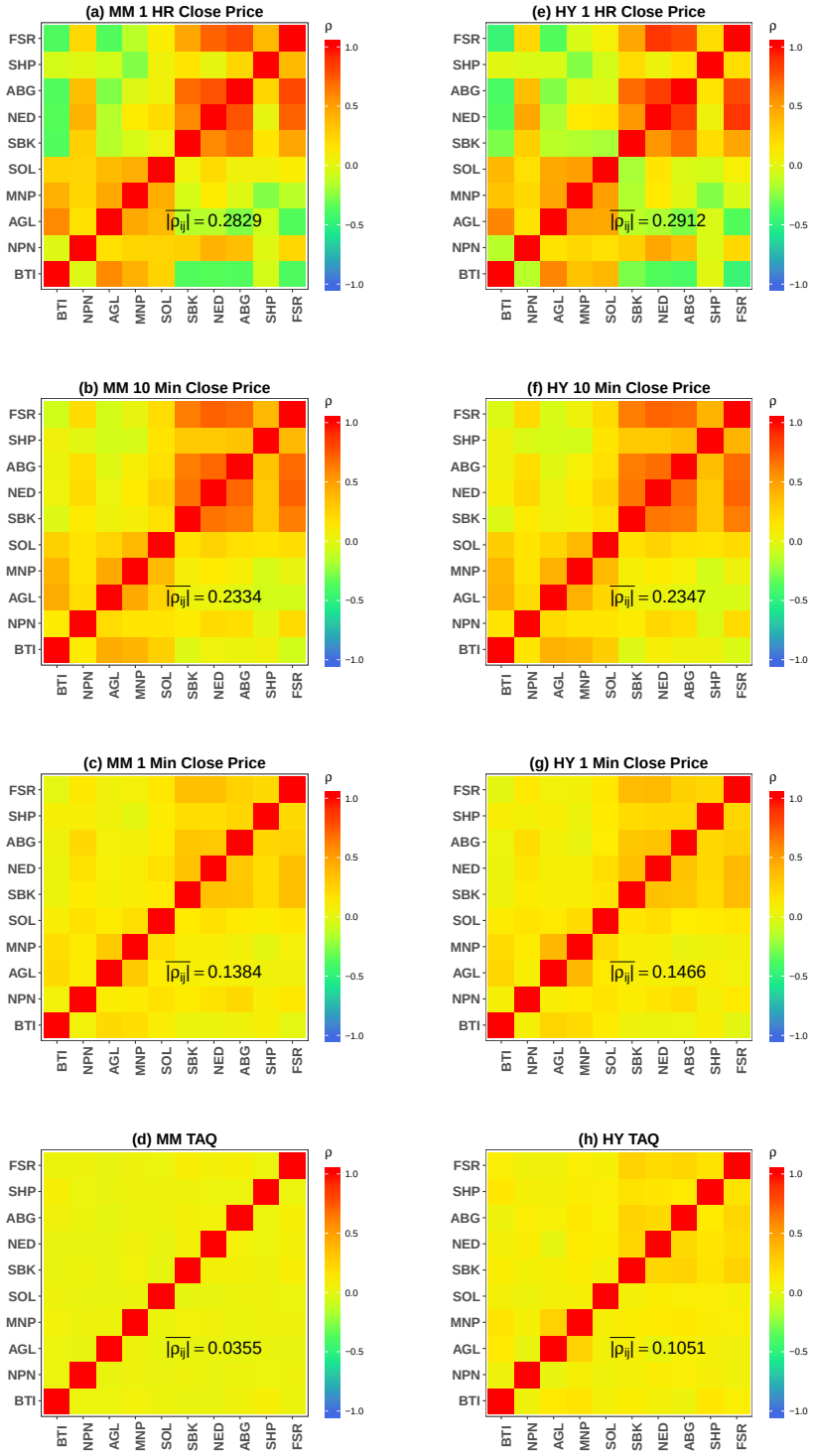


FIGURE 5.1: Investigating the Epps effect by aggregating TAQ data into closing bar prices. From (a) to (d), we have the MM estimator applied to 1 hour, 10 minute, 1 minute closing bar data and TAQ data respectively using algorithm 2. From (e) to (h), we have the HY estimator applied to 1 hour, 10 minute, 1 minute closing bar data and TAQ data respectively using algorithm 3. The Epps effect [4] is demonstrated with both estimators and it persists with the MM but only slightly for the HY estimator on the TAQ data.

What is more interesting is that when the sampling frequency increases to the highest available frequency, the correlation from the MM estimates drops towards zero as expected from the Epps effect, but for the HY estimates, the correlation from the TAQ data does not die out completely and stays near the correlation estimates from the 1 minute bars. Indicating that the HY estimator is manufacturing correlation and thereby not well suited for high-frequency data due to the upward bias. This is because the Epps effect is very real and present in both estimators.

Additionally, the drop in correlation from figure 5.1 (e) through to (g) goes against the claim of Hayashi and Yoshida that the Epps effect is a bias in the estimator for which their estimator is immune to, as we have clearly demonstrated otherwise. Even though their estimator does seem to be immune to the Epps effect arising from asynchrony, it does not seem to always be the case as seen in figure 3.5 (f), therefore given that there is no seamless way to decompose the various factors contributing towards the Epps effect it is hard to pinpoint what effect the multiple contributions are having with regards to the Epps effect.

The correlation estimates make quite a lot of sense, the top right corner for the various sub-figures in figure 5.1 are highly correlated, and these tickers are all from the banking sector, therefore validating the fact that we have performed the analysis correctly and algorithm 2 and 3 are implemented correctly.

Another interesting note is as the sampling intervals become smaller, the correlations drop, but become positively correlated. This is clearly demonstrated in figure 5.1 (a) to (b) and (e) to (f). Suggesting the argument of asynchrony and lead-lags may not be sufficient in fully explaining the entirety of the Epps effect, but rather there is an underlying change in correlation structure depending on the sampling interval.

Using closing prices for the various bar data discards a lot of the information given by the financial market as it is just another sample from the available sample of TAQ data. Thus we turn our attention to another method to create bar data, which encapsulates more of the information given within a bar.

5.1.2 Volume Weighted Prices

Looking at another method of aggregating TAQ data, we look towards the Volume Weighted Average Prices (VWAP). This representation of bar data is better than the closing prices because the VWAP includes the information from all trades within a given bar by means of averaging. Using this aggregation method, we apply the MM and HY estimators to the 1 hour, 10 minute, 1 minute and TAQ data to see how this compares against the closing prices.

From figure 5.2 (a) to (d), the MM estimator is applied to the 1 hour, 10 minute, 1 minute VWAP bar data and TAQ data respectively; and from figure 5.2 (e) to (h), the HY estimator is applied to the 1 hour, 10 minute, 1 minute VWAP bar data and TAQ respectively. Comparing the two estimators, the

Epps effect persists with the MM estimator into the TAQ data, while the correlation does not completely die out with the HY estimator as we go further into the high-frequency spectrum - indicating that the HY is manufacturing correlations through its multiple contributions. Further attesting to our argument that the HY estimate is biased for high-frequency event data.

The correlation structures in figure 5.2 is very similar to that of 5.1, the banking sectors are still strongly positively correlated. The main difference between the two correlation structures is that the VWAP aggregation seems to accentuate the existing correlation structure in the Closing aggregation. This is very interesting because the aggregation methods are quite different, the VWAP incorporates more information within the given bar by means of averaging while the closing prices are mere samples from the finite TAQ sample, yet their correlation structures remained very similar. Additionally, the Epps effect is once again present in this method of aggregation - once again suggesting there seems to be a structural change in correlation that is dependent on the sampling interval.

An interesting point to note in figures C.3 and C.4, is that we initially made the error of computing the OHLCV based on returns, rather than computing the returns after obtaining the OHLCV prices. Therefore, initially we had the closing and VWAP returns computed on the TAQ data for the various intervals, instead of computing the returns over the various intervals. The interesting thing about this is that we still saw an Epps effect under these circumstances, where the returns are computed from the highest available sampling frequency, rather than dependent on the various sampling intervals. Therefore the Epps effect was also inadvertently achieved by sampling the TAQ returns at various sampling frequencies. This begs the question as to what the Epps effect truly is. How and why did it still show up even when returns are computed at the highest available sampling frequency?

The VWAP bars share the same issue as the closing bars; some of the less liquid stocks will not have any trades within a given bar, thus the bar data is not truly synchronous and the two estimators differ slightly. It must be noted that although the VWAP incorporates more information from the stocks it has its own issues; namely that it hides any jumps the price paths may have into the average, therefore acting as a smoothing operator similar to that of a Moving Average. An additional issue is that the aggregation between the bars lacks consistency, this is because aggregating the data in calendar time means that different bars will be averaged with different volume sizes. Therefore a more suitable way to incorporate information from the price path is to look at intrinsic time aggregation.

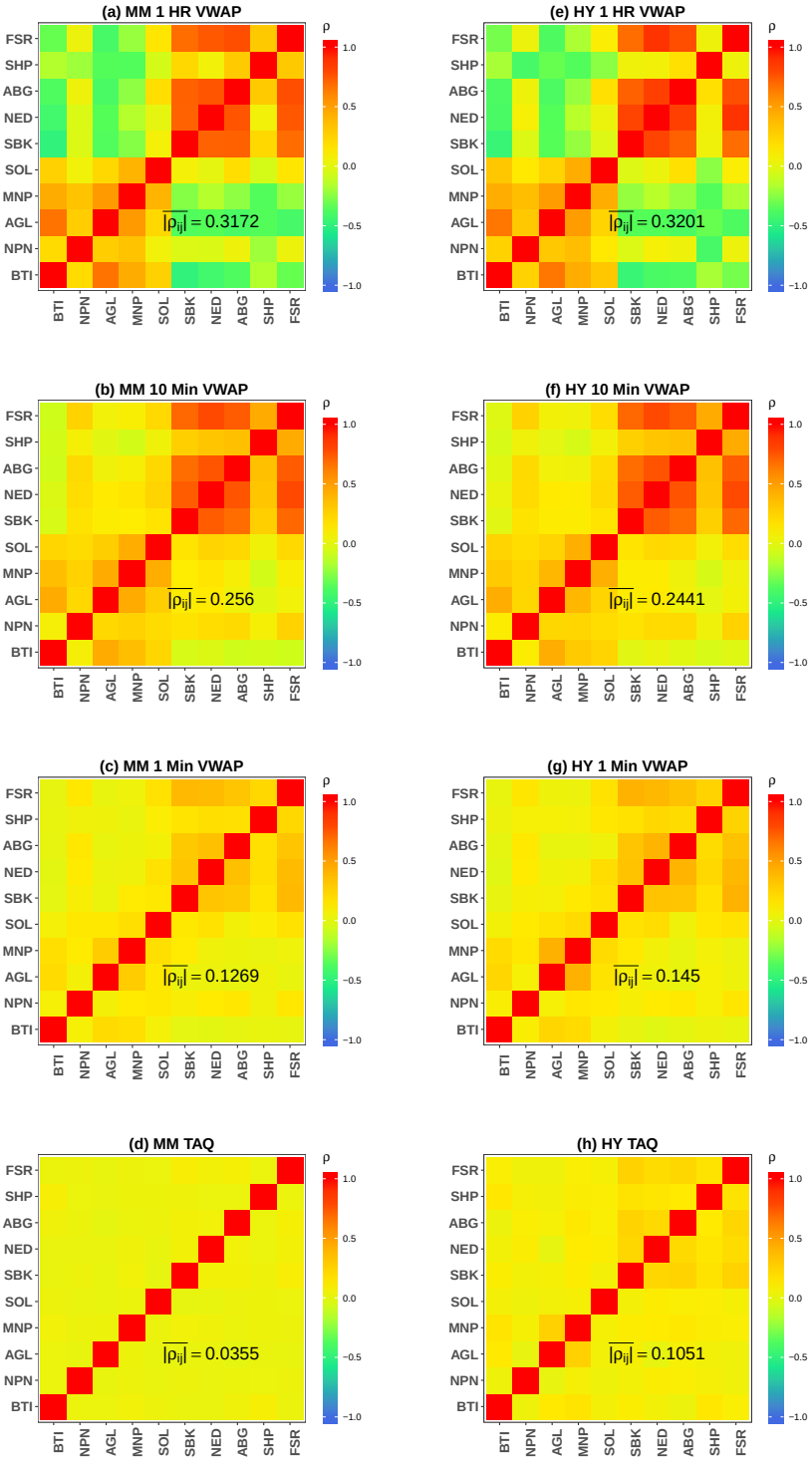


FIGURE 5.2: Investigating the Epps effect by aggregating TAQ data into VWAP bar prices. From (a) to (d), we have the MM estimator applied to 1 hour, 10 minute, 1 minute closing bar data and TAQ data respectively using algorithm 2. From (e) to (h), we have the HY estimator applied to 1 hour, 10 minute, 1 minute closing bar data and TAQ data respectively using algorithm 3. The Epps effect [4] is demonstrated with both estimators and it persists with the MM but only slightly for the HY estimator on the TAQ data.

5.2 Intrinsic Time

Intrinsic time is a different paradigm where time is now measured in terms of machine time - the clock ticks based on events rather than what humans perceive as time. Working in volume time presents significant statistical advantages. The benefit is that it allows the partial recovery of Normality and the IID assumption. Additionally, sampling in a volume clock metric deals with the issue of random and asynchronous trade data [29]. Furthermore, under this paradigm, there is a framework presented by Derman [14] which provides an intuitive method to synchronise the TAQ data across the various assets.

5.2.1 Derman Framework

Derman assumed that each stock has their its intrinsic time scale which is constant through time. He defines the stocks' trading frequency v_j as the number of intrinsic time ticks that occur for one calendar second [14].

He gives the relationship between the flow of calendar time t and the flow of intrinsic time τ_j as

$$d\tau_j = v_j dt. \tag{5.2.1}$$

More importantly, he was able to show that the correlation in intrinsic time π_{ij} is the same as the correlation in calendar time ρ_{ij} [14]. This framework provides a method to create synchronous price paths in intrinsic time which will allow the recovery of the correlation in calendar time. However, there is currently no framework that provides a method to create price paths in intrinsic time while allowing for the sampling intervals (in intrinsic time) to reduce down to each individual volumes. Thus we are unable to study the equivalent of TAQ data in intrinsic time.

To study the Epps effect with this framework, we have to alter Derman's methodology slightly. Instead of assuming v_j as the number of trades per calendar second, we assume v_j to be the number of trades per unit of sampling interval considered. Although we altered his method slightly, the maths showing that the correlations are dimensionless and independent of the various time measurements still holds.

Therefore to apply this sampling scheme, bucket sizes must be chosen for each stock. This is determined by the rough equivalent bar length in calendar time (i.e. to create the equivalent of 10 min calendar time bars in intrinsic time, we divide the average volume per day by 48). Since the bucket sizes are computed from the average volume per day, the price paths will not be fully synchronous due to different volume amounts traded per day. Some stocks will have more (less) volume buckets if the trades for the day are above (below) the average for the stock (respectively).

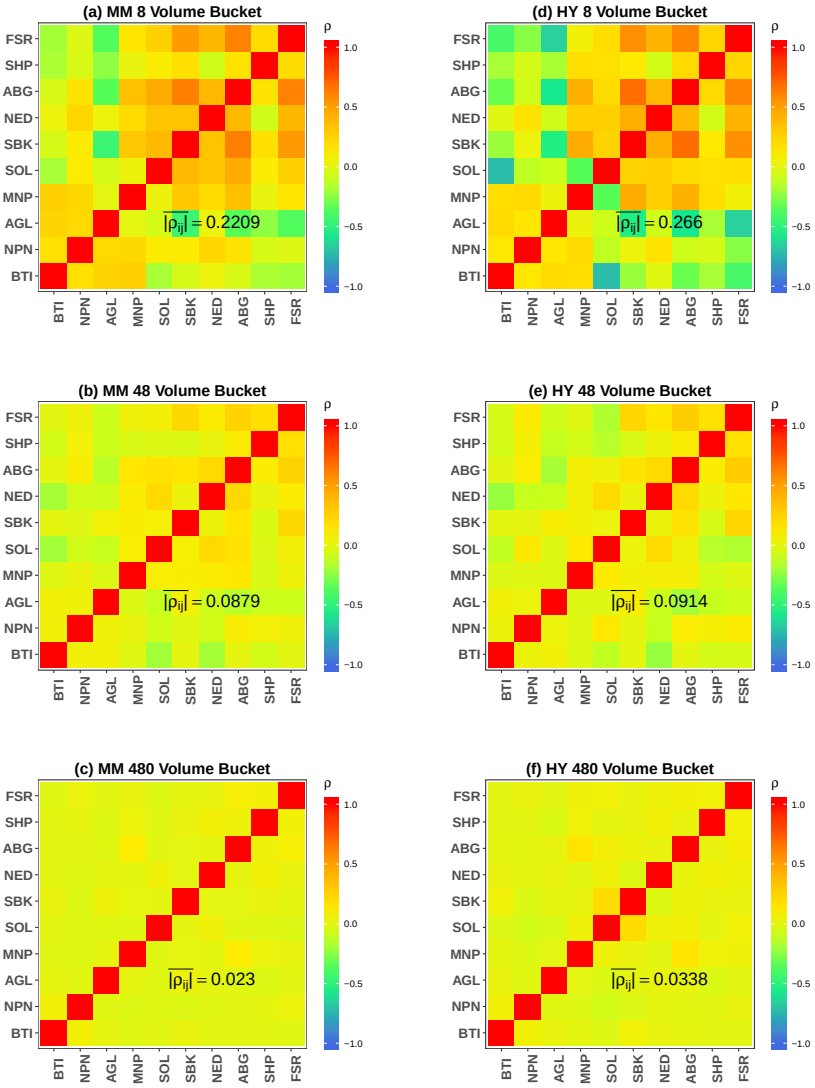


FIGURE 5.3: Using the Derman framework [14] to compare the two estimators. From (a) to (c), we have the MM estimator applied to the calendar time equivalent of 1 hour, 10 minute and 1 minute bar data respectively using algorithm 2. From (d) to (f), we have the HY estimator applied to the calendar time equivalent of 1 hour, 10 minute and 1 minute bar data respectively using algorithm 3. The Epps effect is clearly present for both estimators under the paradigm of intrinsic time.

This highlights the first issue with the framework. The assumption that each stocks' intrinsic time scale is constant through time is a strong assumption to make, where the validity is questionable. This is because the trading frequency changes over time depending on factors such as time of day or relevant news reports. For instance, if news comes out that a company is about to undergo liquidation, traders will try to square their positions therefore increasing the trading frequency. The second issue is that (5.2.1) is the key to linking the correlation in intrinsic time to the correlation in calendar time, but

this assumption induces a continuity assumption from the calendar time into the intrinsic time given the linear relationship which re-scales the time given by standard continuous-time stochastic processes. It must therefore be noted that it does not fully achieve the effect of converting stocks into intrinsic time - where time ticks purely on the events. Elaborating further on why this does not achieve the full conversion into intrinsic time is because a unit interval in intrinsic time can be thought of as a stochastic interval in calendar time, where the stopping rule is determined by the number of trades counted.

Using parts of algorithm A.1. from [30] in algorithm 6, we create the 1 hour, 10 minute and 1 minute calendar time equivalent data by using 8, 48 and 480 buckets per day respectively to create the intrinsic time samples.

From figure 5.3 [3] (a) through to (c), we have the MM estimator applied to the calendar time equivalent of 1 hour, 10 minute and 1 minute bar data respectively using algorithm 2. From figure 5.3 (d) through to (f), we have the HY estimator applied to the calendar time equivalent of 1 hour, 10 minute and 1 minute bar data respectively using algorithm 3. It is clear that the Epps effect still exists under this completely different method of aggregating TAQ data.

This is interesting because this shows that the Epps effect does not only exist in the paradigm of calendar time, it is also present under the event time paradigm. What is even more interesting is that the correlation structures change depending on the sampling interval used, indicating that correlations are not indeed dimensionless as suggested by Derman [14], and that the Epps effect seems to be intrinsically linked to the sampling intervals chosen, therefore further attesting to the idea that the Epps effect cannot be fully explained with asynchrony or lead-lags. In addition, the correlation structure in figure 5.3 is very different to that of figure 5.1 and 5.2, indicating that the correlations are not preserved across these various measurements of time as Derman proved. Finally, this method faces similar issues to that of the VWAP where the jumps are hidden into the averages. Although we have highlighted a few of the pitfalls regarding this framework, this is still the most seamless framework provided in the literature which ties together the ideas from intrinsic time to calendar time.

This however does not answer the main question as to which estimator is the more efficient of the two; simply because the aggregation method creates data which is very close to being synchronous, therefore the two estimators behave extremely similarly as seen in figure 5.3. To this end, by employing the ideas from the intrinsic time framework [14], we create our own method of aggregating TAQ data, specifically focused on determining how the two estimators differ.

[3] Figure 5.3 can be reproduced using Derman.R.

5.2.2 Lining Up Events

For this method of aggregation, the averaging method used in the Derman framework is also used here. However the difference is that we do not assume that each stock has its own intrinsic time. What we do is we count the events in terms of calendar time, and each stock "collects" trades in calendar time until the bucket is full and a price is printed [4]. The bucket sizes are determined by the average trades per day of the most liquid stock which provides the baseline "clock" for which all the assets are lined up accordingly.

The rationale behind this method of aggregation is that we want the prices to line up when there is an event triggering more trades. This is because when there is a large event, trades across various assets will "synchronise" in response to the event and once again "de-synchronise" after the event has passed. There is no framework linking this method of aggregation to the correlation in calendar time; however the advantages this method possesses is that by employing the volume time aggregation method we benefit from have log-returns that are more gaussian, in addition, this is a natural way to aggregate events in an asynchronous manner. Finally, since this method does not resolve the issue of asynchronous trading times, in the context for comparing the two estimators, this is exactly what we want - a clear example to demonstrate how the two estimators differ.

An issue was encountered when creating the 1 hour equivalent calendar time sample. Due to the large bucket sizes based on the most liquid stock, the stocks with low liquidity simply did not have enough trades in a given day to fill up more than two buckets. Since there is no clear method to fix the defective sample, we discard this sample and focus only on the 10 minute and 1 minute calendar time equivalent samples. This does not affect the analysis since correlations on short time scales is what is of interest.

Figure 5.4 [5] (a) and (b) is the MM estimator applied to the 10 minute and 1 minute calendar time equivalent with algorithm 2 and (c) and (d) is the HY estimator applied to the 10 minute and 1 minute calendar time equivalent with algorithm 3. The first thing to notice is that even under this asynchronous sampling scheme the Epps effect is still present with both estimators. The second thing to notice is how different the two estimators are: the MM has correlation near zero while HY has an extremely high correlation. The 1 minute equivalent calendar bar (d) has a significantly higher correlation when compared to the other aggregation methods - indicating that HY is manufacturing correlation.

Figure 5.4 is the key figure in this paper that demonstrates the HY is biased for computing the correlation of events. This is because HY does not treat missing observations as proper missing observations, it compensates the missing observations through the multiple contributions to bring up the estimate and therefore manufacturing correlation. The MM treats missing

[4] The time steps used here is still intrinsic time

[5] Figure 5.3 can be reproduced using LiningUpEvents.R.

observations as truly missing observations, it does not compensate the correlation but rather uses a lossless interpolation between events, therefore it does not overestimate the correlation. Therefore further showcasing our preference of the MM estimator over the HY estimator for high-frequency trading - where the events are truly discontinuous and asynchronous. The HY estimator is most certainly the better estimator of the two, given that the observed prices in the financial market are samples from an underlying continuous-time stochastic process; however the MM estimator is the better estimator of the two, given that the high-frequency finance world is truly discontinuous, discrete and asynchronous events.

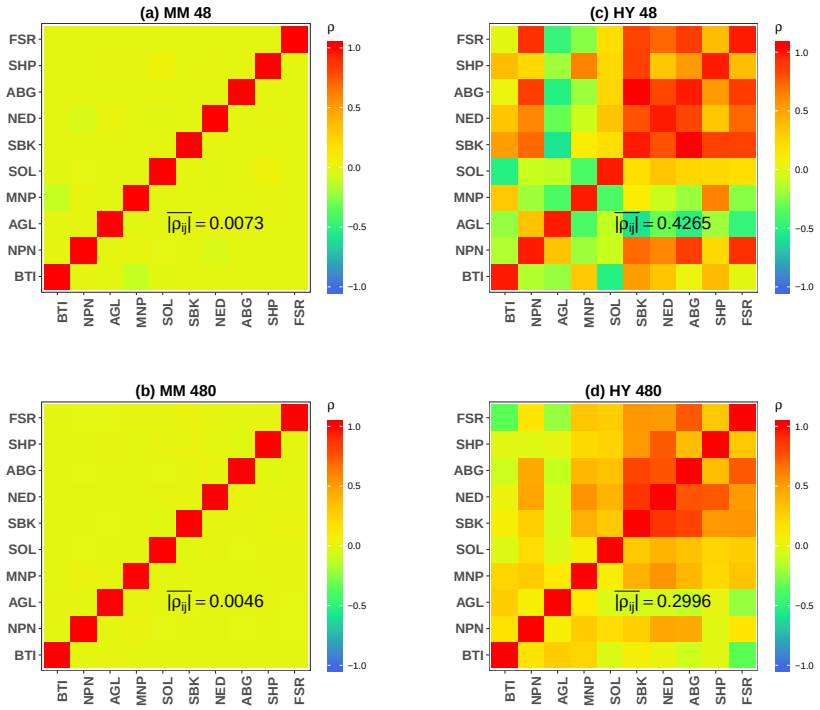


FIGURE 5.4: Comparing the two estimators using a sampling method to line up events. (a) and (b) is the MM estimator applied to the 10 minute and 1 minute calendar time equivalent with algorithm 2, (c) and (d) is the HY estimator applied to the 10 minute and 1 minute calendar time equivalent with algorithm 3. The Epps effect is once again present, but we see that HY is manufacturing correlation.

Chapter 6

Concluding Remarks

In this report, we have demonstrated that the MM and HY estimators differ under asynchronous conditions which is the exact conditions we are interested in addressing with regards to high-frequency finance. Therefore although we have argued for the efficacy of the MM estimator over the HY estimator from a data-informed view, this report does not concretely show which estimator is the better of the two. The whole argument as to why the MM estimator is more believable lies in the fact that it possesses lossless interpolation between event data and it correctly captures the existence of the Epps effect and the fact that it is currently unclear what the multiple contributions are truly doing with regards to the Epps effect in the HY estimator.

The reason why one cannot definitively show that the MM estimator performs better is because the two estimators only differ under tick-by-tick asynchronous trade data which is also the exact case where we want to determine the more effective estimator. Furthermore, given that there is currently no cohesive framework that ties together the data generating process and the continuous-time stochastic process; no seamless framework that ties up all the loose ends regarding the Epps effect; no satisfactory method to synchronise the TAQ data without interpolation and no method to falsify high-frequency finance being samples from a continuous-time stochastic process - one is left with very limited methods to definitively demonstrate which estimator performs better. What we have demonstrated in this report is that there is some inconsistency in the HY estimator when it comes to investigating the Epps effect and that the estimator seems to be manufacturing correlation which does not exist.

This report does not show anything new that was previously unknown about the Epps effect. What this report does show is the existence of the Epps effect on the JSE through various methods of aggregating TAQ data, which suggests that the Epps effect explained purely by asynchrony and lead-lag is insufficient, further validating Thomas Epps' initial thought that there seems to be an underlying change in the correlation structure which is dependent on the sampling intervals used [4].

A natural extension to what we have done here is to increase the computation speed of algorithm 2 using Fast Fourier techniques. Specifically, design a Fast Fourier Transform (FFT) version of algorithm 2 for the synchronous case [31]

and a Non-Uniform FFT (NUFFT) version for the more general asynchronous case.

Additional extensions regarding the Monte Carlo experiments in section 3 include the multivariate Hawkes process [25] method of generating asynchronous data and to study the effect of the correlation estimates by changing the various parameters pertaining to the Hawkes process calibration and simulation. Another extension on the Monte Carlo side will be to recover the results from [6] and firstly see if the MM estimator recovers the same estimates as their Fourier method, and secondly use these results to decompose the Epps effect into the component arising from asynchrony and the remaining factors.

Finally, all the code listing used in this report can be found on Github [32] and the steps outlining the recovery of the results can be found in the README.md document to ensure straightforward replication of the results.

Appendix A

Supporting Algorithms

Algorithm 8 Re-scale Trading Times

Require:

1. (n x m) matrix T of asynchronous sampled times

$t_{min} \leftarrow$ minimum value of T
$t_{max} \leftarrow$ maximum value of T
for j = 1 to m **do**
  for i = 1 to n **do**
    $\tau_{ij} \leftarrow \frac{2\pi(t_j - t_{min})}{t_{max} - t_{min}}$
  end for
end for
return (τ)

Algorithm 9 Kanatani Weight

Require:

1. (n x 2) matrix τ and P of re-scaled asynchronous sampled times and prices

I. Initialize W matrix
$W \leftarrow 0$ [N_i x N_j] matrix of 0's
for j = 1 to N_i **do**
  for i = 1 to N_j **do**
    if $(t^i_{k-1}, t^i_k] \cap (t^j_{l-1}, t^j_l] \neq \emptyset$ **then**
      $w_{kl} \leftarrow 1$
    end if
  end for
end for
return (W)

1 2

[1] Algorithm 8 can be found in ftcorr.R as an auxiliary function and was provided by [18].
[2] Algorithm 9 can be found in ftcorr.R as an auxiliary function.

Algorithm 10 Simulating Geometric Brownian motion

Require:

1. n number of price points to simulate
2. μ (d x 1) vector of drift parameters
3. Σ (d x d) covariance matrix
4. start price (d x 1) vector of $S(0)$

Procedure for the i^{th} asset:

1. generate $Z \sim N_d(0, I_{dxd})$
2. set $S_i(t_{k+1}) = S_i(t_k) \exp\left[(\mu_i - \frac{1}{2}\sigma_i^2)(t_{k+1} - t_k) + \sqrt{t_{k+1} - t_k} \sum_{k=1}^{d} A_{ik} Z_k\right]$

return (S)

Subject to the condition $S(0) =$ start price and A is the Cholesky decomposition of Σ. Algorithm 10 is provided by [22] [3].

Algorithm 11 Simulating Merton Model

Require:

1. n number of price points to simulate
2. μ (d x 1) vector of drift parameters
3. Σ (d x d) covariance matrix
4. λ (d x 1) vector of the Poisson process parameter
5. a (d x 1) vector of lognormal location parameter
6. b (d x 1) vector of lognormal standard deviation
7. start price (d x 1) vector of $S(0)$

Procedure for the i^{th} asset:

1. generate $Z \sim N_d(0, I_{dxd})$
2. generate $N_i \sim \text{Poisson}(\lambda_i(t_{k+1} - t_k))$
3. generate $Z_2 \sim N_1(0, 1)$
4. set $M = a_i N_i + b_i \sqrt{N_i} Z_2$
5. set $X_i(t_{k+1}) = X_i(t_k) + (\mu_i - \frac{1}{2}\sigma_i^2)(t_{k+1} - t_k) + \sqrt{t_{k+1} - t_k} \sum_{k=1}^{d} A_{ik} Z_k + M$
6. $S = \exp(X)$

return (S)

Subject to the condition $X(0) = \ln(\text{start price})$ and A is the Cholesky decomposition of Σ. Algorithm 11 is provided by [22] [4].

[3] Algorithm 10 can be found in GBM.R.

[4] Algorithm 11 can be found in Merton Model.R.

Algorithm 12 Simulating GARCH(1,1)

Require:

1. n number of price points to simulate
2. θ (2 x 1) mean reverting rate
3. λ (2 x 1)
4. w (2 x 1) vector long term variance
5. ρ correlation
6. starting variance (2 x 1) vector of starting variance
7. starting price (2 x 1) vector of starting price

Procedure for the i^{th} asset:

1. generate $Z \sim N(0,1)$
2. set $\sigma_i^2(t_{k+1}) = \sigma_i^2(t_k) + \theta_i(w_1 - \sigma_i^2(t_k))(t_{k+1} - t_k) + \sqrt{2\lambda_i\theta_i(t_{k+1} - t_k)}\sigma_i(t_k)Z$
3. create Σ^* correlation matrix based on $\sigma^2(t_{k+1})$
4. generate $Z^* \sim N_d(0, I_{dxd})$
5. set $X_i(t_{k+1}) = X_i(t_k) + \sqrt{t_{k+1} - t_k}\sum_{k=1}^{d} A_{ik}^* Z_k^*$
6. $S = \exp(X)$

return (S)

Subject to the condition $X(0) = \ln(\text{start price})$, $\sigma(0) =$ starting variance and A^* is the Cholesky decomposition of Σ^* [5].

Algorithm 13 Simulating Variance Gamma

Require:

1. n number of price points to simulate
2. μ (d x 1) vector of drift parameters
3. Σ (d x d) covariance matrix
4. β (d x 1) scale parameter of Gamma
5. start price (d x 1) vector of $S(0)$

Procedure for the i^{th} asset:

1. generate $Y_i \sim \text{Gamma}(t_{k+1} - t_k/\beta_i, \beta_i)$
2. generate $Z \sim N_d(0, I_{dxd})$
3. set $X_i(t_{k+1}) = X_i(t_k) + \mu Y_i + \sqrt{Y_i}\sum_{k=1}^{d} A_{ik}Z_k$

return (X)

Subject to the condition $X(0) =$ start price and A is the Cholesky decomposition of Σ. Algorithm 13 is provided by [22] [6].

[5] Algorithm 12 can be found in GarchAndersen.R, while the specification from [12] can be found in GarchReno.R.

[6] Algorithm 13 can be found in Variance Gamma.R.

Algorithm 14 Simulating Ornstein Uhlenbeck

Require:

1. n number of price points to simulate
2. μ (d x 1) vector of long term prices
3. Σ (d x d) covariance matrix
4. θ (d x 1) vector of mean reverting rate
5. start price (d x 1) vector of starting prices

Procedure for the i^{th} asset:

1. generate $Z \sim N_d(0, I_{dxd})$
2. set $X_i(t_{k+1}) = X_i(t_k) + \theta_i(\ln(\mu_i) - X_i(t_k))(t_{k+1} - t_k) + \sqrt{t_{k+1} - t_k} \sum_{k=1}^{d} A_{ik} Z_k$
3. $S = \exp(X)$

return (S)

Subject to the condition $X(0) = \ln(\text{start price})$ and A is the Cholesky decomposition of Σ [7].

[7] Algorithm 14 can be found in Ornstein Uhlenbeck.R.

Appendix B

Appendix Derivation

B.1 Proof for Theorem 2.1.1 and 2.1.2

B.1.1 Proof for $a_q(\Sigma)$

The proof is based on [1] and [15], and we complete the additional results that were omitted in their papers.

Assume the Bachelier paradigm and further suppose the price process follows a one-dimensional Itô process

$$dp(t) = \beta(t)dt + \sigma(t)dW. \tag{B.1.1}$$

Remark B.1.1 *The price process need not be a one-dimensional Itô process, we assume so for ease of derivation. The price process in general follows* (2.1.1)

Furthermore, we can assume the drift term has no contribution and can therefore ignore it. Thus the price process becomes

$$dp(t) = \sigma(t)dW. \tag{B.1.2}$$

This is acceptable because Malliavin and Mancino show that the contribution for the drift term is zero [1]. Additionally Malherbe argues that ignoring the drift term implies an efficient market [15].

We now introduce the Gaussian variables

$$G_k := a_k(dp), \quad G'_k := b_k(dp).$$

From Corollary (3.2.6) to the Martingale Representation theorem in [33], we know that G_k and G'_k is a martingale. Therefore, G_k and G'_k are Gaussian variables with $E(G_k) = E(G'_k) = 0$ [15].

We now calculate the covariance of the Gaussian variables

$$\begin{aligned} E(G_k G_l) &= E(a_k(dp)a_l(dp)) \\ &= E\Big[\frac{1}{\pi^2}\int_0^{2\pi}\cos(kt)\sigma(t)dW\int_0^{2\pi}\cos(lt)\sigma(t)dW\Big]. \end{aligned} \tag{B.1.3}$$

Now using the Itô isometry and (2.1.2), (B.1.3) becomes

$$E(G_k G_l) = \frac{1}{\pi^2}\int_0^{2\pi}\Sigma(t)\cos(kt)\cos(lt)dt. \tag{B.1.4}$$

Using the identity

$$\cos(kt)\cos(lt) = \frac{1}{2}(\cos(k-l)t + \cos(k+l)t),$$

(B.1.4) becomes

$$\begin{aligned} E(G_k G_l) &= \frac{1}{\pi^2}\int_0^{2\pi}\Sigma(t)\cos(kt)\cos(lt)dt \\ &= \frac{1}{2\pi^2}\int_0^{2\pi}\Sigma(t)[(\cos(k-l)t + \cos(k+l)t)]dt \\ &= \frac{1}{2\pi^2}\int_0^{2\pi}\Sigma(t)(\cos(k-l)tdt + \frac{1}{2\pi^2}\int_0^{2\pi}\Sigma(t)(\cos(k+l)tdt \\ &= \frac{1}{2\pi}(a_{|k-l|}(\Sigma) + a_{k+l}(\Sigma)). \end{aligned} \tag{B.1.5}$$

The energy identity [1] is

$$||\Sigma||_{L^2}^2 = \sum_k (a_k(\Sigma))^2 + (b_k(\Sigma))^2. \tag{B.1.6}$$

Furthermore,

$$\sum_k (a_k(\Sigma))^2 \leq ||\Sigma||_{L^2}^2. \tag{B.1.7}$$

Now for $q > 0$ consider the random variable U_N^q as the discrete convolution of the Gaussian variables, where

$$U_N^q := \frac{1}{N}\sum_{k=1}^{N} G_k G_{k+q}. \tag{B.1.8}$$

Using (B.1.5), we get that

[1] Better known as Parseval's theorem

$$\begin{aligned} E(U_N^q) &= \frac{1}{N}\sum_{k=1}^{N} E(G_k G_{k+q}) \\ &= \frac{1}{N}\sum_{k=1}^{N} \frac{1}{2\pi}(a_{|k-k-q|}(\Sigma) + a_{k+k+q}(\Sigma)) \\ &= \frac{1}{N}\sum_{k=1}^{N} \frac{1}{2\pi}(a_q(\Sigma) + a_{2k+q}(\Sigma)) \\ &= \frac{1}{2\pi}\left[a_q(\Sigma) + \frac{1}{N}\sum_{k=1}^{N} a_{2k+q}(\Sigma)\right] \\ 2\pi E(U_N^q) &= a_q(\Sigma) + \frac{1}{N}\sum_{k=1}^{N} a_{2k+q}(\Sigma) \\ &= a_q(\Sigma) + R_N. \end{aligned} \tag{B.1.9}$$

We note that Malliavin and Mancino omitted the 2π which is necessary in order to recover (2.1.1).

We now show that $2\pi E(U_N^q) \to a_q(\Sigma)$ as $N \to \infty$ by using the Cauchy-Schwarz inequality

$$\begin{aligned} R_n = \frac{1}{N}|\sum_{k=1}^{N} a_{2k+q}(\Sigma)| &\le \frac{1}{N}(\sum_{k=1}^{N} 1^2)^{\frac{1}{2}}(\sum_{k=1}^{N} a_{2k+q}^2(\Sigma))^{\frac{1}{2}} \\ &= \frac{1}{\sqrt{N}}(\sum_{k=1}^{N} a_{2k+q}^2(\Sigma))^{\frac{1}{2}} \\ &\le \frac{1}{\sqrt{N}}||\Sigma||_{L^2}. \end{aligned} \tag{B.1.10}$$

Therefore, $R_N \to 0$ as $N \to \infty$ [2].

We now want to show that $\lim_{N\to\infty} U_N^q = E(U_N^q)$ in L^2. We first compute

$$E[(U_N^q)^2] = \frac{1}{N^2}\sum_{0\le k,k'\le N} E(G_k^2 G_{k'+q}^2). \tag{B.1.11}$$

and consider an $\mathbb{R}^2$-valued normal variable (G_1, G_2) and denote

$$\lambda_i := E(G_i^2), \quad \mu := E(G_1 G_2),$$

and define

$$Z := G_2 - \frac{\mu}{\lambda_1} G_1.$$

[2] $\sum_{k=1}^{N} a_{2k+q}^2(\Sigma)) \le \sum_k a_k^2(\Sigma)$

We note that $E(ZG_1) = E(G_1G_2 - \frac{\mu}{\lambda_1}G_1^2) = E(G_1G_2) - \frac{\mu}{\lambda_1}E(G_1^2) = E(G_1G_2) - E(G_1G_2) = 0$ and therefore G_1 and Z are independent. Additionally, since $E(G_1) = E(G_2) = 0$, we get [3]

$$\begin{aligned}
E(G_1^2G_2^2) &= E[G_1^2(Z^2 + 2\frac{\mu}{\lambda_1}G_1Z + \frac{\mu^2}{\lambda_1^2}G_1^2)] \\
&= E(G_1^2Z^2) + 2\frac{\mu}{\lambda_1}E(G_1^3Z) + \frac{\mu^2}{\lambda_1^2}E(G_1^4) \\
&= E(G_1^2Z^2) + \frac{\mu^2}{\lambda_1^2}E(G_1^4) \\
&= E(G_1^2)E(Z^2) + \frac{\mu^2}{\lambda_1^2}E(G_1^4) \\
&= E(G_1^2)E(G_2^2 - 2\frac{\mu}{\lambda_1}G_1G_2 + \frac{\mu^2}{\lambda_1^2}E(G_1^2)) + \frac{\mu^2}{\lambda_1^2}E(G_1^4) \\
&= E(G_1^2)E(G_2^2) - 2\frac{\mu}{\lambda_1}E(G_1^2)E(G_1G_2) + \frac{\mu^2}{\lambda_1^2}E(G_1^2)E(G_1^2) + \frac{\mu^2}{\lambda_1^2}E(G_1^4).
\end{aligned} \tag{B.1.12}$$

Furthermore, since G_k is a Gaussian variable with mean 0. We know the fourth uncentered moment is $E(G_k^4) = 3E(G_k^2)^2$. Therefore, (B.1.12) becomes

$$\begin{aligned}
E(G_1^2G_2^2) &= E(G_1^2)E(G_2^2) - 2\frac{\mu}{\lambda_1}E(G_1^2)E(G_1G_2) + \frac{\mu^2}{\lambda_1^2}E(G_1^2)E(G_1^2) + \frac{\mu^2}{\lambda_1^2}E(G_1^4) \\
&= E(G_1^2)E(G_2^2) - 2\frac{\mu}{\lambda_1}E(G_1^2)E(G_1G_2) + 4\frac{\mu^2}{\lambda_1^2}E(G_1^2)E(G_1^2) \\
&= E(G_1^2)E(G_2^2) - 2\frac{\mu}{\lambda_1}\lambda_1\mu + 4\frac{\mu^2}{\lambda_1^2}\lambda_1\lambda_1 \\
&= E(G_1^2)E(G_2^2) + 2\mu^2 \\
&= E(G_1^2)E(G_2^2) + 2E(G_1G_2)^2.
\end{aligned} \tag{B.1.13}$$

Putting this together, we get

[3] $E(G_1^3Z) = 0$ due to independence and that the third uncentered moment $E(G_1^3) = 0$

$$
\begin{aligned}
&E[(U_N^q - E(U_N^q))^2] \\
&= E[(U_N^q)^2 - 2U_N^q E(U_N^q) + (E(U_N^q))^2] \\
&= E((U_N^q)^2) - [E(U_N^q)]^2 \\
&= \frac{1}{N^2} \sum_{0\le k,k'\le N} E(G_k^2 G_{k'+q}^2) - \frac{1}{(2\pi)^2} a_q(\Sigma)^2 \\
&= \frac{1}{N^2} \sum_{0\le k,k'\le N} [E(\tbinom{2}{k}) E(G_{k'+q}^2) + 2(E(G_k G_{k'+q}))^2] - \frac{1}{(2\pi)^2} a_q(\Sigma)^2 \\
&= \frac{1}{N^2} \sum_{0\le k,k'\le N} [E(\tbinom{2}{k}) E(G_{k'+q}^2) + \frac{1}{2\pi^2}(a_{|k-k'+q|}(\Sigma) + a_{k+k'+q}(\Sigma))^2] - \frac{1}{(2\pi)^2} a_q(\Sigma)^2 \\
&= (E(U_N^q))^2 + \frac{1}{N^2} \sum_{0\le k,k'\le N} [\frac{1}{2\pi^2}(a_{|k-k'+q|}(\Sigma) + a_{k+k'+q}(\Sigma))^2] - \frac{1}{(2\pi)^2} a_q(\Sigma)^2 \\
&= \big(\frac{1}{2\pi} a_q(\Sigma)\big)^2 + \frac{1}{N^2} \sum_{0\le k,k'\le N} [\frac{1}{2\pi^2}(a_{|k-k'+q|}(\Sigma) + a_{k+k'+q}(\Sigma))^2] - \frac{1}{(2\pi)^2} a_q(\Sigma)^2 \\
&= \frac{1}{2\pi^2 N^2} \sum_{0\le k,k'\le N} (a_{|k-k'+q|}(\Sigma) + a_{k+k'+q}(\Sigma))^2 \\
&\le \frac{1}{N} ||\Sigma||_{L^2}^2
\end{aligned}
\tag{B.1.14}
$$

As $N \to \infty$, $\frac{1}{N}||\Sigma||_{L^2}^2 \to 0$. Thus it follows that

$$\lim_{N\to\infty} U_N^q = E(U_N^q) \quad \text{in} \quad L^2.$$

Coupled with the fact that $2\pi E(U_N^q) \to a_q(\Sigma)$ as $N \to \infty$, we get

$$2\pi U_N^q \to 2\pi E(U_N^q) \to a_q(\Sigma),$$

in *probability* as a natural consequence from L^p convergence as $N \to \infty$.

Therefore,

$$a_q(\Sigma) = \lim_{N\to\infty} 2\pi U_N^q = \lim_{N\to\infty} \frac{2\pi}{N} \sum_{s=1}^{N} (a_s(dp) a_{s+q}(dp)), \quad \forall q > 0.$$

$a_q(\Sigma)$ has been proved. The remaining univariate and multivariate cases will be less rigorous and more focused on achieving the correct scaling factors.

B.1.2 Proof for $a_0(\Sigma)$

We consider $E(b_s^2(dp))$. Using the Itô isometry we get,

$$
\begin{aligned}
E(b_s^2(dp)) &= E\Big[\frac{1}{\pi^2}\int_0^{2\pi}\sin(st)\sigma(t)dW\int_0^{2\pi}\sin(st)\sigma(t)dW\Big] \\
&= \frac{1}{\pi^2}\int_0^{2\pi}\sin^2(st)\Sigma(t)dt.
\end{aligned}
\tag{B.1.15}
$$

Using the identity $\sin^2(st) = (\frac{1}{2} - \frac{1}{2}\cos(2st))$, (B.1.15) becomes

$$
\begin{aligned}
E(b_s^2(dp)) &= \frac{1}{\pi^2}\int_0^{2\pi}\sin^2(st)\Sigma(t)dt \\
&= \frac{1}{2\pi^2}\int_0^{2\pi}\Sigma(t)dt - \frac{1}{2\pi^2}\int_0^{2\pi}\cos(2st)\Sigma(t)dt.
\end{aligned}
\tag{B.1.16}
$$

Using the identity $\cos(2st) = 2\cos^2(st) - 1$, (B.1.16) becomes

$$
\begin{aligned}
E(b_s^2(dp)) &= \frac{1}{2\pi^2}\int_0^{2\pi}\Sigma(t)dt - \frac{1}{2\pi^2}\int_0^{2\pi}\cos(2st)\Sigma(t)dt \\
&= \frac{1}{2\pi^2}\int_0^{2\pi}\Sigma(t)dt - \frac{1}{\pi^2}\int_0^{2\pi}\cos^2(st)\Sigma(t)dt + \frac{1}{2\pi^2}\int_0^{2\pi}\Sigma(t)dt.
\end{aligned}
\tag{B.1.17}
$$

Using the fact that $E(a_s^2(dp)) = \frac{1}{\pi^2}\int_0^{2\pi}\cos^2(st)\Sigma(t)dt$, we get that

$$
\frac{2}{\pi}a_0(\Sigma) = E(a_s^2(dp)) + E(b_s^2(dp)). \tag{B.1.18}
$$

Thus we get a scaling factor of $\frac{1}{2}$ which is not present in [1].

B.1.3 Proof for $b_q(\Sigma)$

We consider $E(a_s(dp)b_{s+q}(dp))$. Using the Itô isometry we get,

$$
\begin{aligned}
E(a_s(dp)b_{s+q}(dp)) &= E\Big[\frac{1}{\pi^2}\int_0^{2\pi}\cos(st)\sigma(t)dW\int_0^{2\pi}\sin(s+q)t\sigma(t)dW\Big] \\
&= \frac{1}{\pi^2}\int_0^{2\pi}\cos(st)\sin(s+q)t\Sigma(t)dt.
\end{aligned}
\tag{B.1.19}
$$

Using the identity $\sin(lt)\cos(st) = \frac{1}{2}[\sin(l+s)t + \sin(l-s)t$, (B.1.19) becomes

$$E(a_s(dp)b_{s+q}(dp)) = \frac{1}{2\pi^2}\int_0^{2\pi}\sin(qt)\Sigma(t)dt + \frac{1}{2\pi^2}\int_0^{2\pi}\sin(2s+q)t\Sigma(t)dt$$
$$= \frac{1}{2\pi}(b_q(\Sigma) + b_{2s+q}(\Sigma)). \tag{B.1.20}$$

The remaining steps remains the same as the derivation for $a_q(\Sigma)$.

B.1.4 Proof for $a_q(\Sigma^{i,j})$

For the multivariate cases, we will index (B.1.2) with j,k and use the polarisation identity

$$\langle dp^j, dp^k\rangle_t = \frac{1}{2}\big(\langle dp^j + dp^k\rangle_t - \langle dp^j\rangle_t - \langle dp^k\rangle_t\big).$$

We first confirm that the polarization recovers the results we desire. We have

$$a_k(dp^j + dp^k) = \frac{1}{\pi}\int_0^{2\pi}\cos(kt)(\sigma^j(t) + \sigma^k(t))dW,$$

and

$$E[a_k(dp^j + dp^k)a_l(dp^j + dp^k)] = \frac{1}{\pi^2}\int_0^{2\pi}\big(\sigma^{jj}(t) + \sigma^{kk}(t) + 2\sigma^{jk}(t)\big)\cos(kt)\cos(lt)dt.$$

Therefore,

$$\frac{1}{2}\Big[E[a_k(dp^j + dp^k)a_l(dp^j + dp^k)] - E[a_k(dp^j)a_l(dp^j)] - E[a_k(dp^k)a_l(dp^k)]\Big]$$
$$= \frac{1}{\pi^2}\int_0^{2\pi}\sigma^{jk}(t)\cos(kt)\cos(lt)dt. \tag{B.1.21}$$

We note the linearity of Fourier transforms and thus $a_k(dp^j + dp^k) = a_k(dp^j) + a_k(dp^k)$, which means

$$a_k(dp^j + dp^k)a_l(dp^j + dp^k)$$
$$= \big[a_k(dp^j) + a_k(dp^k)\big]\big[a_l(dp^j) + a_l(dp^k)\big]$$
$$= a_k(dp^j)a_l(dp^j) + a_k(dp^j)a_l(dp^k) + a_k(dp^k)a_l(dp^j) + a_k(dp^k)a_l(dp^k). \tag{B.1.22}$$

Using (B.1.22), (B.1.21) can be simplified into

$$\begin{aligned}
&\frac{1}{2}\Big[E[a_k(dp^j+dp^k)a_l(dp^j+dp^k)]-E[a_k(dp^j)a_l(dp^j)]-E[a_k(dp^k)a_l(dp^k)]\Big]\\
&=\frac{1}{2}\Big[E[a_k(dp^j)a_l(dp^k)]+E[a_k(dp^k)a_l(dp^j)]\Big]\\
&=\frac{1}{\pi^2}\int_0^{2\pi}\sigma^{jk}(t)\cos(kt)\cos(lt)dt\\
&=\frac{1}{2\pi}(a_{|k-l|}(\Sigma^{jk})+a_{k+l}(\Sigma^{jk})).
\end{aligned}\tag{B.1.23}$$

The 2π in the last equation of (B.1.23) is the 2π in the numerator of $a_q(\Sigma^{i,j})$ which was left out in [1].

B.1.5 Proof for $a_0(\Sigma^{i,j})$

For $a_0(\Sigma^{i,j})$, we look only at $(a_s^2(dp)+b_s^2(dp))$ and using the polarisation identity

$$\frac{1}{2}\Big[E[a_s^2(dp^j+dp^k)+b_s^2(dp^j+dp^k)]-E[a_s^2(dp^j)+b_s^2(dp^j)]-E[a_s^2(dp^k)+b_s^2(dp^k)]\Big].\tag{B.1.24}$$

Using the linearity of Fourier transforms, the first expectation in (B.1.24) becomes

$$\begin{aligned}
&E\big[\big(a_s(dp^j)+a_s(dp^k)\big)\big(a_s(dp^j)+a_s(dp^k)\big)+\big(b_s(dp^j)+b_s(dp^k)\big)\big(b_s(dp^j)+b_s(dp^k)\big)\big]\\
&=E[a_s^2(dp^j)]+E[a_s^2(dp^k)]+2E[a_s(dp^j)a_s(dp^k)]\\
&\quad+E[b_s^2(dp^j)]+E[b_s^2(dp^k)]+2E[b_s(dp^j)b_s(dp^k)].
\end{aligned}\tag{B.1.25}$$

Using this in (B.1.24), we get the polarisation of $(a_s^2(dp)+b_s^2(dp))$ as

$$\frac{1}{2}\,2E[a_s(dp^j)a_s(dp^k)+b_s(dp^j)b_s(dp^k)].\tag{B.1.26}$$

Therefore the scaling factors for $a_0(\Sigma)$ and $a_0(\Sigma^{i,j})$ are the same.

B.1.6 Proof for $b_q(\Sigma^{i,j})$

We have

$$a_k(dp^j+dp^k)=\frac{1}{\pi}\int_0^{2\pi}\cos(kt)(\sigma^j(t)+\sigma^k(t))dW,$$

$$b_l(dp^j + dp^k) = \frac{1}{\pi}\int_0^{2\pi} \sin(lt)(\sigma^j(t) + \sigma^k(t))dW,$$

and

$$E[a_k(dp^j + dp^k)b_l(dp^j + dp^k)] = \frac{1}{\pi^2}\int_0^{2\pi} \left(\sigma^{jj}(t) + \sigma^{kk}(t) + 2\sigma^{jk}(t)\right)\cos(kt)\sin(lt)dt.$$

Therefore

$$\begin{aligned}&\frac{1}{2}\Big[E[a_k(dp^j + dp^k)b_l(dp^j + dp^k)] - E[a_k(dp^j)b_l(dp^j)] - E[a_k(dp^k)b_l(dp^k)]\Big] \\ &= \frac{1}{\pi^2}\int_0^{2\pi} \sigma^{jk}(t)\cos(kt)\sin(lt)dt.\end{aligned} \tag{B.1.27}$$

Using the linearity of Fourier transforms, we can simplify (B.1.27) to

$$\begin{aligned}\frac{1}{2}\Big[E[a_k(dp^j)b_l(dp^k)] + E[a_k(dp^k)b_l(dp^j)]\Big] &= \frac{1}{\pi^2}\int_0^{2\pi} \sigma^{jk}(t)\cos(kt)\sin(lt)dt \\ &= \frac{1}{2\pi}(b_{|k-l|}(\Sigma^{jk}) + b_{k+l}(\Sigma^{jk})).\end{aligned} \tag{B.1.28}$$

We note that the scaling factors we recovered are the same scaling factors recovered by [16].

B.2 Proof for Theorem 2.2.1

We present our own version for the proof that is different to [2], but the methods from [1] are used.

Assume first A-II and for ease of derivation that the price process follows a one-dimensional Itô process

$$dp(t) = \beta(t)dt + \sigma(t)dW. \tag{B.2.1}$$

Furthermore, we can assume that the drift term has no contribution which is proven in [2]. Therefore the price process is given by

$$dp(t) = \sigma(t)dW. \tag{B.2.2}$$

Further define

$$c_k(dp) = \frac{1}{2\pi}\int_0^{2\pi} e^{-ikt} dp(t).$$

Now using the Itô isometry

$$\begin{aligned} E(c_l c_k) &= \frac{1}{(2\pi)^2} E\Big[\int_0^{2\pi} e^{-ilt} dp(t) \int_0^{2\pi} e^{-ikt} dp(t)\Big] \\ &= \frac{1}{(2\pi)^2} E\Big[\int_0^{2\pi} e^{-ilt} \sigma(t) \int_0^{2\pi} e^{-ikt} \sigma(t)\Big] \\ &= \frac{1}{(2\pi)^2} E\Big[\int_0^{2\pi} e^{-i(l+k)t}\, \Sigma(t) dt\Big] \\ &= \frac{1}{2\pi} c_{l+k}(\Sigma). \end{aligned} \tag{B.2.3}$$

For $q \in \mathbb{Z}$, define the Random Variable U_N^q as the discrete convolution of the Fourier coefficients for the price process.

$$U_N^q := \frac{1}{2N+1} \sum_{s=-N}^{N} c_k c_{k-q}. \tag{B.2.4}$$

Therefore, it follows that

$$\begin{aligned} E\big[U_N^q\big] &= \frac{1}{2N+1} \sum_{s=-N}^{N} E\big[c_k c_{k-q}\big] \\ &= \frac{1}{2N+1} \sum_{|k|\le N} \frac{1}{2\pi} c_q(\Sigma), \end{aligned} \tag{B.2.5}$$

and thus

$$2\pi E\big[U_N^q\big] = c_q(\Sigma). \tag{B.2.6}$$

By setting the drift component $\beta(t) \equiv 0$, we have that $c_k(dp)$ is a complex martingale. Therefore by introducing $\mathbb{R}^2$ valued gaussian variables

$$G_k := c_k(dp),$$

where $E[G_k] = \mathbf{0}$. We can employ the same arguments used in the derivation of $a_q(\Sigma)$ and further noting that energy identity is

$$\|\Sigma\|_{L^2}^2 = \sum_k \big(c_k(\Sigma)\big)^2.$$

From which we get that

$$
\begin{aligned}
E\left[\left(U_N^q - E(U_N^q)\right)^2\right] &= \frac{1}{(2N+1)^2}\sum_{k,k'}\left[E(c_k^2)E(c_{q-k'}^2) + 2E(c_k c_{q-k'})^2\right] - \frac{1}{(2\pi)^2}c_q(\Sigma)^2 \\
&= E\left(U_N^q\right)^2 + \frac{1}{(2N+1)^2}\sum_{k,k'}\frac{1}{2\pi}c_{k+q+k'}(\Sigma) - \frac{1}{(2\pi)^2}c_q(\Sigma)^2 \\
&\leq \frac{1}{2N+1}\|\Sigma\|_{L^2}^2.
\end{aligned} \tag{B.2.7}
$$

By combining (B.2.6) and (B.2.7), we complete the theorem as

$$
\frac{1}{2\pi}c_k(\Sigma) = \lim_{N\to\infty}\frac{1}{2N+1}\sum_{s=-N}^{N} c_s(dp)c_{k-s}(dp), \tag{B.2.8}
$$

where the convergence of the *Bohr* convolution is attained in *probability*.

B.3 Proof for Theorem 2.3.1

The proof is based on [3], we add a few explanatory steps and details to their derivation

(i) We first assume that $\mu^l \equiv 0$ and $0 \leq t \leq T$. We now show that $U_n \to \theta$ in L^2 as $n \to \infty$. We introduce some auxiliary symbols: let $K_{ij} := I_{\{I^i \cap J^j \neq \emptyset\}}$. Furthermore, for each measurable set I on $[0,\infty)$, we define (signed) measures by

$$
\begin{aligned}
v(I) := v^0(I) &:= \int_I \sigma^1\sigma^2\rho dt, \\
v^k(I) &:= \int_I (\sigma^k)^2 dt, \qquad k = 1,2.
\end{aligned}
$$

We further introduce identities for $k = 0,1,2$

$$
\begin{aligned}
&\sum_i v^k\left(I^i\right)1_{\{I^i\neq\emptyset\}} = \sum_i v^k\left(I^i\right) = v^k((0,T]), \\
&\sum_{i,j} v^k\left(I^i \cap J^j\right)K_{ij} = v^k((0,T]), \\
&\sum_j v^k\left(I^i \cap J^j\right)K_{ij} = v^k\left(I^i\right),
\end{aligned}
$$

these identities hold due to the fact that I^i and J^j are the partitions over $(0,T]$. Finally, for each measurable set I on $[0,\infty)$, define

$$
\Delta P^k(I) := \int_0^T 1_I(t)\sigma^k dW^k, \quad k = 1,2.
$$

We can show that U_n is unbiased, for each n,

$$\mathrm{E}\left[U_n\right] = \mathrm{E}\left[\sum_{i,j} \mathrm{E}\left\{\Delta P^{\mathrm{l}}\left(I^i\right) \Delta P^2\left(J^j\right) |\Pi\right\} K_{ij}\right] = \mathrm{E}\left[\sum_{i,j} v\left(I^i \cap J^j\right) K_{ij}\right] = \theta.$$

Remark B.3.1 *We note that the inner expectation is for notation indicating that we know Π and that the outer expectation can be brought in due to linearity. This is to allow later parts of the derivation become more simple. The reason why we say the inner expectation is for notation is because we can recover θ with just the outer expectation:*

$$\begin{aligned} E[U_n] &= \mathrm{E}\left[\sum_{i,j} \Delta P^{\mathrm{l}}\left(I^i\right) \Delta P^2\left(J^j\right) K_{ij}\right] \\ &= \sum_{i,j} \mathrm{E}\left[\int_0^T 1_{I^i}(t)\sigma^1 dW^1 \int_0^T 1_{J^j}(t)\sigma^2 dW^2 \;\; K_{ij}\right] \\ &= \sum_{i,j} \mathrm{E}\left[\int_0^T 1_{I^i \cap J^j}(t)\sigma^{12} dt \;\; K_{ij}\right] \\ &= \sum_{i,j} \mathrm{E}\left[\int_{I^i \cap J^j} \sigma^1\sigma^2\rho dt \;\; K_{ij}\right] \\ &= \mathrm{E}\left[\sum_{i,j} v\left(I^i \cap J^j\right) K_{ij}\right] = \theta. \end{aligned} \tag{B.3.1}$$

The third equation follows from the second in (B.3.1) using Itô's Isometry. For the remainder of the proof we adopt inner expectation used by [3] to make things simpler.

We now want to show that $\mathrm{E}\left[U_n^2\right] = \theta^2 + o(1)$. This would mean that $\mathrm{var}\left[U_n\right] = o(1)$ so that $U_n \to \theta$ in L^2 as $n \to \infty$. To do so, we start with

$$\mathrm{E}\left[U_n^2\right] = \mathrm{E}\left[\sum_{i,j,i',j'} \mathrm{E}\left\{\Delta P^{\mathrm{l}}\left(I^i\right) \Delta P^2\left(J^j\right) \Delta P^{\mathrm{l}}\left(I^{i'}\right) \Delta P^2\left(J^{j'}\right) |\Pi\right\} K_{ij} K_{i'j'}\right],$$

for which the summations can be decomposed into

$$\sum_{i,j,i',j'} = \underset{\substack{i,j,i',j': \\ i'=i, j'=j}}{\sum} + \underset{\substack{i,j,i',j': \\ i'=i, j'\neq j}}{\sum} + \underset{\substack{i,j,i',j': \\ i'\neq i, j'=j}}{\sum} + \underset{\substack{i,j,i',j': \\ i'\neq i, j'\neq j}}{\sum} =: D_1 + D_2 + D_3 + D_4.$$

We will use this decomposition to calculate the expectation using four cases.

Case 1: $i = i'$, $j = j'$. Let $L_1 := I^i \cap J^j$, $L_2 := I^i \setminus L_1$ and $L_3 := J^j \setminus L_1$. We have

$$D_1 = \sum_{i,j} \mathrm{E}\left\{\Delta P^1 \left(I^i\right)^2 \Delta P^2 \left(J^j\right)^2 |\Pi\right\} K_{ij}.$$

Using the independence of the increments and identities such as $v^k(I^i) = v^k(L_2) + v^k(L_1)$ and $v^k(J^j) = v^k(L_3) + v^k(L_1)$, we get that

$$\begin{aligned}
\mathrm{E}\left\{\Delta P^1 \left(I^i\right)^2 \Delta P^2 \left(J^j\right)^2 |\Pi\right\} &= \mathrm{E}\left\{\left(\Delta P^1\left(L_2\right) + \Delta P^1\left(L_1\right)\right)^2 \left(\Delta P^2\left(L_3\right) + \Delta P^2\left(L_1\right)\right)^2 |\Pi\right\} \\
&= \mathrm{E}\left\{\Delta P^1\left(L_2\right)^2 \Delta P^2\left(L_1\right)^2 |\Pi\right\} + \mathrm{E}\left\{\Delta P^1\left(L_1\right)^2 \Delta P^2\left(L_1\right)^2 |\Pi\right\} \\
&\quad + \mathrm{E}\left\{\Delta P^1\left(L_1\right)^2 \Delta P^2\left(L_3\right)^2 |\Pi\right\} + \mathrm{E}\left\{\Delta P^1\left(L_2\right)^2 \Delta P^2\left(L_3\right)^2 |\Pi\right\} \\
&= v^1\left(L_2\right) v^2\left(L_1\right) + 2v\left(L_1\right)^2 + v^1\left(L_1\right) v^2\left(L_1\right) \\
&\quad + v^1\left(L_1\right) v^2\left(L_3\right) + v^1\left(L_2\right) v^2\left(L_3\right) \\
&= [v^1(L_2) + v^1(L_1)]v^2(L_1) + [v^1(L_1) + v^1(L_2)]v^2(L_3) + 2v(L_1)^2 \\
&= [v^1(L_1) + v^1(L_2)][v^2(L_1) + v^2(L_3)] + 2v(L_1)^2 \\
&= v^1(I^i)v^2(J^j) + 2v(I^i \cap J^j)^2.
\end{aligned}$$

The first equation comes from the definition that $I^i = L_2 + L_1$ and $J^j = L_3 + L_1$. The second equation comes from the fact that the cross products from each of the quadratic terms are 0. The third equation follows from the fact that for any (deterministic) interval I, $\Delta P^1(I)$ and $\Delta P^2(I)$ are jointly normal with respective mean and variance 0 and $v^k(I), k = 1, 2$, and with covariance $v(I)$. Thus $E[\Delta P^1(I)\Delta P^2(I)] = 2v(I)^2 + v^1(I)v^2(I)$ using some Multivariate reults. Therefore we have,

$$D_1 = \sum_{i,j} v^1\left(I^i\right) v^2\left(J^j\right) K_{ij} + 2\sum_{i,j} v\left(I^i \cap J^j\right)^2 K_{ij}. \tag{B.3.2}$$

Now looking at the first term on the right hand side of (B.3.2) and noting that the σ^k are bounded, we get

$$\begin{aligned}
\sum_{i,j} v^1\left(I^i\right) v^2\left(J^j\right) K_{ij} &= \sum_{i,j} \left(\int_{I^i} \left(\sigma^1\right)^2 \mathrm{d}t\right)\left(\int_{J^j} \left(\sigma^2\right)^2 \mathrm{d}t\right) K_{ij} \\
&\leq \sup_{0\leq t\leq T}\left(\sigma^1\right)^2 \sup_{0\leq t\leq T}\left(\sigma^2\right)^2 \sum_{i,j} |I^i||J^j| K_{ij}.
\end{aligned}$$

We now want to show that

$$\mathrm{E}\sum_{i,j}|I^i||J^j|K_{ij} = o(1).$$

To do so, we decompose

$$\sum_{i,j}|I^i||J^j|K_{ij} = \sum_{i,j}|I^i||J^j|K_{ij}1_{\{|I^i|\geq|J^j|\}} + \sum_{i,j}|I^i||J^j|K_{ij}1_{\{|I^i|<|J^j|\}}.$$

Noting that $\sum_j |J^j|K_{ij}1_{\{|I^i|\geq|J^j|\}} \leq 3|I^i|$ for some fixed i, we have

$$\sum_{i,j}|I^i||J^j|K_{ij}1_{\{|I^i|\geq|J^j|\}} = \sum_i |I^i|\sum_j |J^j|K_{ij}1_{\{|I^i|\geq|J^j|\}} \leq 3\sum_i |I^i|^2,$$

hence,

$$\mathrm{E}\sum_{i,j}|I^i||J^j|K_{ij}1_{\{|I|\geq|J|\}} \leq 3\mathrm{E}\sum_i |I^i|^2.$$

By symmetry, we have

$$\mathrm{E}\sum_{i,j}|I^i||J^j|K_{ij} \leq 3\mathrm{E}\sum_i |I^i|^2 + 3\mathrm{E}\sum_j |J^j|^2. \tag{B.3.3}$$

We see that (B.3.3) is $o(1)$ under Condition (A-IV) (ii), (C(ii) in remark 3.1 of [3]). Similarly, we can ascertain that for any random partition $(\tilde{I}^i)$ of $(0,T]$ satisfying (A-IV) (ii),

$$\mathrm{E}\sum_i v\left(\tilde{I}^i\right)^2 = o(1). \tag{B.3.4}$$

Thus the second term on the right hand side of (B.3.2) can be shown to be of $o_P(1)$ by choosing $(I^i \cap J^j)$ as the partition. Hence it follows that $E[D_1] = o(1)$.

Case 2: $i = i'$, $j \neq j'$. This yields

$$D_2 = \sum_{i,j':j\neq j'} \mathrm{E}\left\{\Delta P^1\left(I^i\right)^2 \Delta P^2\left(J^j\right)\Delta P^2\left(J^{j'}\right)|\Pi\right\}K_{ij}K_{ij'}.$$

Let $L_1 := I^i \cap J^j$, $L_2 := I^i \cap J^{j'}$, and $L_3 := I^i \setminus (L_1 \cup L_2)$. Then using the independence of increments,

$$
\begin{aligned}
&\mathrm{E}\left\{\Delta P^{1}\left(I^{i}\right)^{2}\Delta P^{2}\left(J^{j}\right)\Delta P^{2}\left(J^{j'}\right)|\Pi\right\}\\
&=\mathrm{E}\left\{\Delta P^{1}\left(I^{i}\right)^{2}\Delta P^{2}\left(L_{1}\right)\Delta P^{2}\left(L_{2}\right)|\Pi\right\}\\
&=\mathrm{E}\left\{\left(\Delta P^{1}\left(L_{1}\right)+\Delta P^{1}\left(L_{3}\right)+\Delta P^{1}\left(L_{2}\right)\right)^{2}\Delta P^{2}\left(L_{1}\right)\Delta P^{2}\left(L_{2}\right)|\Pi\right\}\\
&=2\mathrm{E}\left\{\Delta P^{1}\left(L_{1}\right)\Delta P^{2}\left(L_{1}\right)|\Pi\right\}\mathrm{E}\left\{\Delta P^{1}\left(L_{2}\right)\Delta P^{2}\left(L_{2}\right)|\Pi\right\}\\
&=2v\left(L_{1}\right)v\left(L_{2}\right)=2v\left(I^{i}\cap J^{j}\right)v\left(I^{i}\cap J^{j'}\right).
\end{aligned}
$$

The third equation follows because when we expand the second equation, all the non-overlapping increments reduces to 0 using the independence of increments and that for any (deterministic) interval I, $\Delta P^{1}(I)$ and $\Delta P^{2}(I)$ are jointly normal with respective mean and variance 0 and $v^{k}(I), k=1,2$, and with covariance $v(I)$.

Hence,

$$
\begin{aligned}
D_{2}&=2\sum_{i,j,j':j'\neq j}v\left(I^{i}\cap J^{j}\right)v\left(I^{i}\cap J^{j'}\right)K_{ij}K_{ij'}\\
&=2\sum_{i}\left\{\sum_{j}v\left(I^{i}\cap J^{j}\right)K_{ij}\left(\sum_{j'}v\left(I^{i}\cap J^{j'}\right)K_{ij'}-v\left(I^{i}\cap J^{j}\right)\right)\right\}\\
&=2\sum_{i}v\left(I^{i}\right)^{2}-2\sum_{i}\sum_{j}v\left(I^{i}\cap J^{j}\right)^{2}K_{ij},
\end{aligned}
$$

which follows becuase of the relation $\sum_{j}v^{k}\left(I^{i}\cap J^{j}\right)K_{ij}=v^{k}\left(I^{i}\right)$. Finally, we see that $E[D_{2}]=o(1)$ by using (B.3.4) and the fact that $(I^{i}\cap J^{j})$ partitions $]0,T]$.

Case 3: $i\neq i', j=j'$. The same argument in case 2 applies here by symmetry, thus we can obtain $E[D_{3}]=o(1)$.

Case 4: $i\neq i', j\neq j'$. Let $L_{1}:=I^{i}\cap J^{j}$, $L_{2}:=I^{i'}\cap J^{j'}$. Note that for i,i',j,j' such that $i\neq i', j\neq j'$ and $K_{ij}K_{i'j'}=1$ means that $K_{i'j}K_{ij'}=0$. Furthermore, due to the identity

$$
\left(1-K_{i'j}\right)\left(1-K_{ij'}\right)+K_{i'j}+K_{ij'}\equiv 1,
$$

we can decompose the event $\{K_{ij}K_{i'j'}=1\}$ further into three subcases, $\left\{I^{i'}\cap J^{j}=\emptyset, I^{i}\cap J^{j'}=\emptyset\right\}$, $\left\{I^{i'}\cap J^{j}\neq\emptyset, I^{i}\cap J^{j'}=\emptyset\right\}$ and $\left\{I^{i'}\cap J^{j}=\emptyset, I^{i}\cap J^{j'}\neq\emptyset\right\}$ each of which respectively corresponds to $\left\{\left(1-K_{i'j}\right)\left(1-K_{ij'}\right)=1\right\}$, $\left\{K_{i'j}=1\right\}$ and $\left\{K_{ij'}=1\right\}$.

Case 4(a): $\left\{I^{i'}\cap J^{j}=\emptyset, I^{i}\cap J^{j'}=\emptyset\right\}$. We have by analogy with case 2

$$\sum_{i,j,i',j':i\neq i',j\neq j'} \mathrm{E}\left\{\Delta P^1\left(I^i\right)\Delta P^2\left(J^j\right)\Delta P^1\left(I^{i'}\right)\Delta P^2\left(J^{j'}\right)|\Pi\right\} K_{ij}K_{i'j'}\left(1-K_{i'j}\right)\left(1-K_{ij'}\right)$$
$$= \sum_{i,j,i',j':i\neq i',j\neq j'} \mathrm{E}\left\{\Delta P^1\left(L_1\right)\Delta P^2\left(L_1\right)\Delta P^1\left(L_2\right)\Delta P^2\left(L_2\right)|\Pi\right\} K_{ij}K_{i'j'}\left(1-K_{i'j}\right)\left(1-K_{ij'}\right)$$
$$= \sum_{i,j,i',j':i\neq i',j\neq j'} v(L_1)v(L_2)K_{ij}K_{i'j'}\left(1-K_{i'j}\right)\left(1-K_{ij'}\right)$$

Case 4(b): $\left\{I^{i'}\cap J^j\neq\emptyset, I^i\cap J^{j'}=\emptyset\right\}$. Let $L_3 := I^{i'}\cap J^j$, $L_4 := J^j\setminus(L_1\cup L_3)$ and $L_5 := I^{i'}\setminus(L_2\cup L_3)$,

$$\begin{aligned}&\mathrm{E}\left\{\Delta P^1\left(L_1\right)\left(\Delta P^2\left(L_1\right)+\Delta P^2\left(L_4\right)+\Delta P^2\left(L_3\right)\right)\left(\Delta P^1\left(L_3\right)+\Delta P^1\left(L_5\right)+\Delta P^1\left(L_2\right)\right)\Delta P^2\left(L_2\right)|\Pi\right\}\\ &\quad=\mathrm{E}\left\{\Delta P^1\left(L_1\right)\Delta P^2\left(L_1\right)\Delta P^1\left(L_2\right)\Delta P^2\left(L_2\right)|\Pi\right\}=v\left(L_1\right)v\left(L_2\right)\end{aligned}$$

Therefore,

$$\begin{aligned}&\sum_{i,j,i',j':i\neq i',j\neq j'} \mathrm{E}\left\{\Delta P^1\left(I^i\right)\Delta P^2\left(J^j\right)\Delta P^1\left(I^{i'}\right)\Delta P^2\left(J^{j'}\right)|\Pi\right\} K_{ij}K_{i'j'}K_{i'j}\\ &=\sum_{i,j,i',j':i\neq i',j\neq j'} v\left(L_1\right)v\left(L_2\right)K_{ij}K_{i'j'}K_{i'j}.\end{aligned}$$

Case 4(c): $\left\{I^{i'}\cap J^j=\emptyset, I^i\cap J^{j'}\neq\emptyset\right\}$. By symmetry, we can obtain using the same technique as 4(b):

$$\sum_{i,j,i',j':i\neq i',j\neq j'} v\left(L_1\right)v\left(L_2\right)K_{ij}K_{i'j'}K_{ij'}.$$

Combining all three subcases together, we get

$$\begin{aligned}D_4 &= \sum_{i,j,i',j':i\neq i',j\neq j'}\left[v\left(L_1\right)v\left(L_2\right)K_{ij}K_{i'j'}\right]\left[\left(1-K_{i'j}\right)\left(1-K_{ij'}\right)+K_{ij'}+K_{ij'}\right]\\ &= \sum_{i,j,i',j':i\neq i',j\neq j'} v\left(I^i\cap J^j\right)v(I^{i'}\cap J^{j'})K_{ij}K_{i'j'}\\ &= \sum_{i,j} v\left(I^i\cap J^j\right)K_{ij}\Big(\sum_{i',j':i'\neq i,j'\neq j} v(I^{i'}\cap J^{j'})K_{i'j'}\Big).\end{aligned}$$

Now for fixed i and j,

$$\begin{aligned}
&\sum_{i',j':i'\neq i,j'\neq j} v(I^{i'}\cap J^{j'})K_{i'j'}\\
&=\sum_{i',j'} v\left(I^{i'}\cap J^{j'}\right)K_{i'j'}-v\left(I^{i}\cap J^{j}\right)-\sum_{j':j'\neq j} v\left(I^{i}\cap J^{j'}\right)K_{ij'}-\sum_{i':i'\neq i} v\left(I^{i'}\cap J^{j}\right)K_{i'j}\\
&=\sum_{i',j'} v\left(I^{i'}\cap J^{j'}\right)K_{i'j'}-v\left(I^{i}\cap J^{j}\right)-v(I^{i})-v(J^{j}),
\end{aligned}$$

we note the typo in [3], they have $v\left(I^i\cap J^j\right)$ instead of $-v\left(I^i\cap J^j\right)$. We now have

$$\begin{aligned}
D_4&=\sum_{i,j} v\left(I^{i}\cap J^{j}\right)K_{ij}\sum_{i',j'} v\left(I^{i'}\cap J^{j'}\right)K_{i'j'}-\sum_{i,j} v\left(I^{i}\cap J^{j}\right)^2K_{ij}\\
&\quad-\sum_{i,j} v\left(I^{i}\cap J^{j}\right)K_{ij}v(I^{i})-\sum_{i,j} v\left(I^{i}\cap J^{j}\right)K_{ij}v(J^{j})\\
&=v(]0,T])^2-\sum_{i,j} v\left(I^{i}\cap J^{j}\right)^2K_{ij}-\sum_{i} v(I^{i})^2-\sum_{j} v(J^{j})^2.
\end{aligned}$$

Thus $E[D_4]=\theta^2+o(1)$ by (B.3.4).

Therefore, $E[U_n^2]=E[D_1+D_2+D_3+D_4]=\theta^2+o(1)$.

(ii) Now we consider the case with non-zero drift such that $\sup_{0\leq t\leq T}|u^k|\in L^4$, $k=1,2$. Let $A^k_\cdot:=\int_0^\cdot u^k dt$, $M^k_\cdot:=\int_0^\cdot \sigma^k dW^k$, $k=1,2$ and

$$\begin{aligned}
B_0&:=\sum_{i,j}\Delta M^1\left(I^i\right)\Delta M^2\left(J^j\right)K_{ij}, \qquad B_1:=\sum_{i,j}\Delta A^1\left(I^i\right)\Delta M^2\left(J^j\right)K_{ij};\\
B_2&:=\sum_{i,j}\Delta M^1\left(I^i\right)\Delta A^2\left(J^j\right)K_{ij}, \qquad B_3:=\sum_{i,j}\Delta A^1\left(I^i\right)\Delta A^2\left(J^j\right)K_{ij}.
\end{aligned}$$

Note that

$$\begin{aligned}
|B_1|&=\left|\sum_i\int_{I^i}\mu^1\mathrm{d}t\left(\sum_j\int_{J^j}\sigma^2\mathrm{d}W^2K_{ij}\right)\right|\leq\sum_i\int_{I^i}\left|\mu^1\right|\mathrm{d}t\left|\sum_j\int_{J^j}\sigma^2\mathrm{d}W^2K_{ij}\right|\\
&\leq T\sup_{0\leq t\leq T}\left|\mu^1\right|\cdot\max_i\sup\left\{\left|\int_s^t\sigma^2\mathrm{d}W^2\right|\,|t-s|\leq\left|I^i\right|+2\max_j\left|J^j\right|,s,t\in[0,T]\right\}\\
&\leq T\sup_{0\leq t\leq T}\left|\mu^1\right|\cdot\sup\left\{\left|\int_s^t\sigma^2\mathrm{d}W^2\right|\,|t-s|\leq\max_i\left|I^i\right|+2\max_j\left|J^j\right|,s,t\in[0,T]\right\},
\end{aligned}\tag{B.3.5}$$

from which we see that B_1 is in L^2 because σ^2 is bounded and the supremum of μ^1 is in L^4. Moreover, under Condition (A-IV) (ii), $E[B_1^2] = o(1)$ as $n \to \infty$ by the dominated convergence theorem. $E[B_2^2] = o(1)$ and $E[B_3^2] = o(1)$ can be shown similarly.

Furthermore,

$$\mathrm{E}\left[(U_n - \theta)^2\right] \leq 2\mathrm{E}\left[(B_0 - \theta)^2\right] + 8\mathrm{E}\left[B_1^2 + B_2^2 + B_3^2\right],$$

and together with (i), we have shown that $U_n \to \theta$ in L^2 as $n \to \infty$ because B_0 is just the driftless case.

Showing $U_n \to \theta$ in probability as $n \to \infty$ follows from (B.3.5), as all that is needed is $B_k \to 0$ in probability as $n \to \infty$, k=1,2,3.

Appendix C

Miscellaneous

C.1 Run times and General issues

We compare first the various run times of algorithm 1, 2 and 3. We begin the experiment by first simulating 500 synchronous seconds from a bivariate GBM using algorithm 10, we then sample individual samples paths with an exponential inter-arrival time with a mean of 15 seconds to obtain asynchronous sample paths with an average of 33 data points. With these sample paths we measure the run-times for the various algorithms [1] 100 times and the distribution for each algorithm is shown in figure C.1.

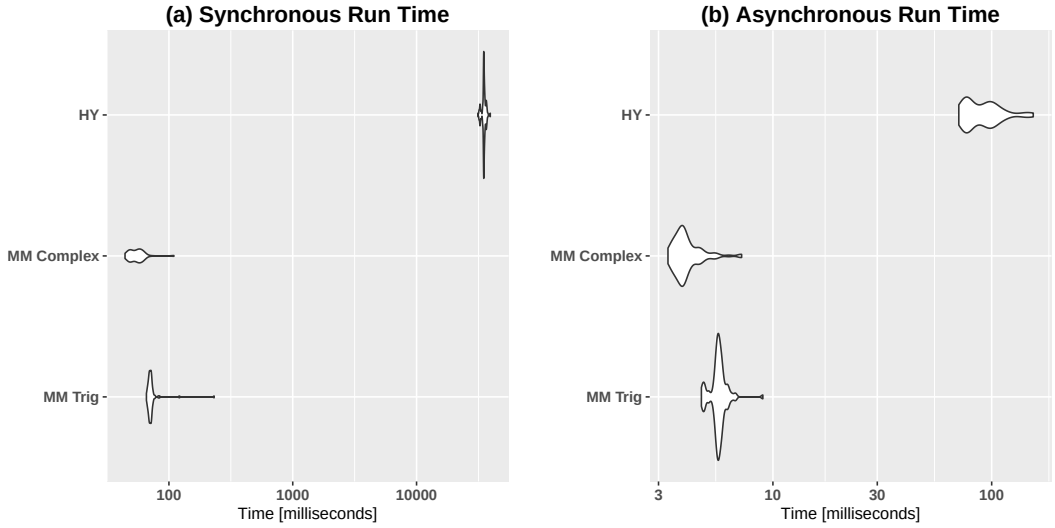


FIGURE C.1: We compare the run times of algorithm 1, 2 and 3. Specifically, (a) looks at the run time for 500 synchronous data points while (b) looks at the run time for an *average* of 33 data points where the asynchrony is induced by exponential inter-arrival time samples of the synchronous data points in (a). It is clear that the Complex MM has the fastest run time while HY has a significantly slower run time compared to the MM algorithms.

[1] For the computation of the MM estimators, the number of Fourier coefficients is chosen to be the Nyquist frequency based on the highest available sampling frequency present in the sample.

From figure C.1 [2], we see that for both the synchronous case and the asynchronous case, the HY algorithm is order of magnitudes longer than the MM algorithms. This is due to the Kanatani Weight matrix using a double for-loop to check for overlapping intervals. Furthermore, we notice that algorithm 2 is faster than algorithm 1, this is because algorithm 2 has been supplemented with Rcpp and RcppArmadillo code to improve the computation time.

We started experimenting with C++ code because memory issues started occurring with base R when computing the empirical data. This was because all the Fourier coefficients are computed with one matrix multiplication for computation efficiency; however, due to the large nature of empirical data, along with the fact that we computed Fourier coefficients based on the Nyquist frequency for the highest available sampling frequency present in the data - computing one pair of stock for one day of data required an matrix of dimensions [15,000 x 1,500,000] to be initialised.

Upon further investigation, we found that R uses 8 bytes of memory to store a double precision float, therefore a matrix with dimensions [15,000 x 1,500,000] uses 167.6Gb of memory to store the object - meaning that packages such as bigmemory which allows the data structure to be allocated to shared memory was not going to be particularly helpful due to physical constraints of our hardware. Thus the only solution left was to remove the vectorisation and compute each Fourier coefficients using a single for-loop.

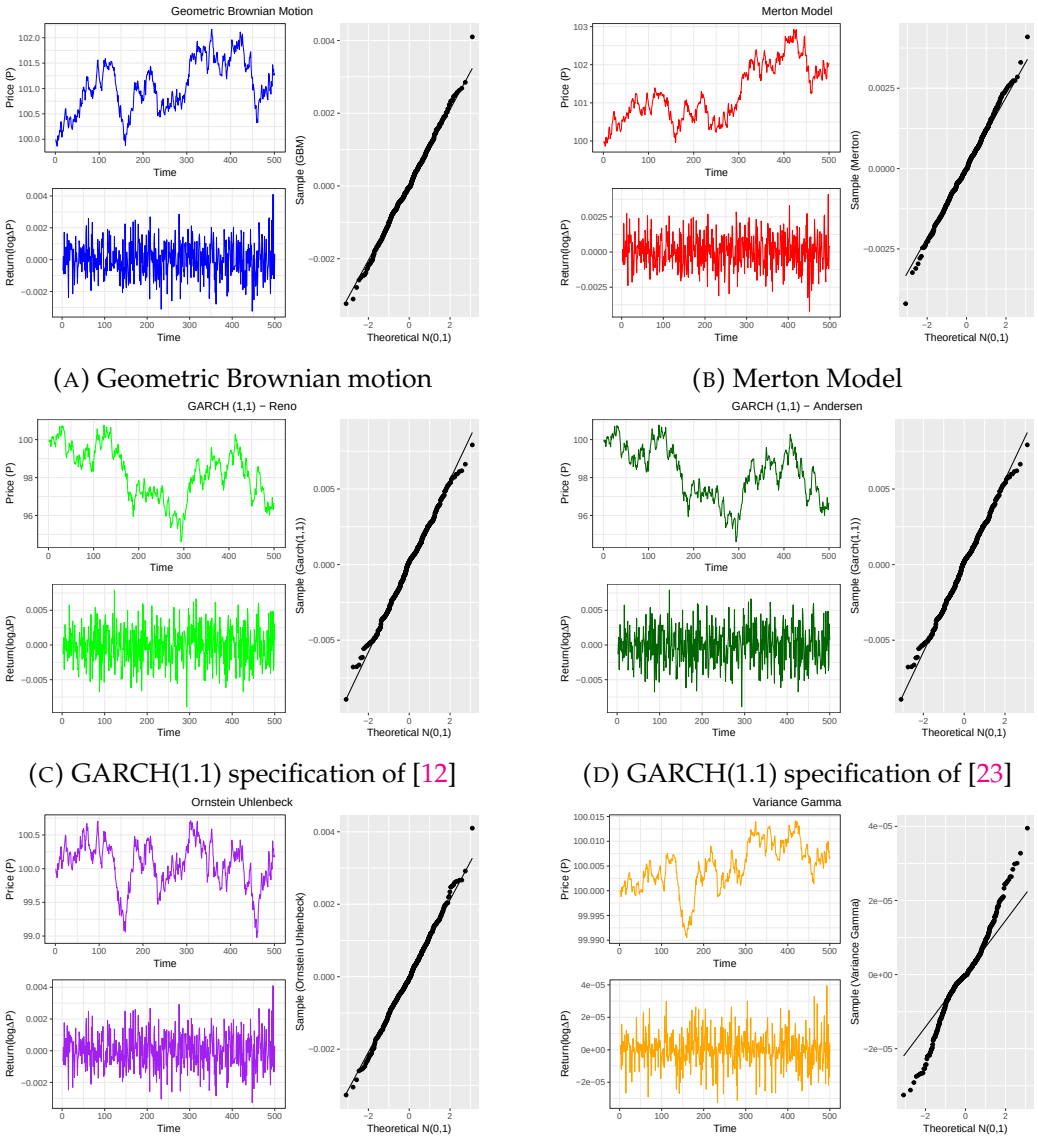
It must be noted, even though the vectorisation was removed from algorithm 2, it took 4 days to compute the correlation matrix for 1 week of data of the 10 assets considered, while algorithm 3 took 7 days. Furthermore, it must be noted that algorithm 3 will have memory issues that are not easily solvable as the Kanatani weight matrix becomes larger when considering longer periods such as the correlation for a month of data.

These problems are non-trivial computer science problems which severely impact the real-time implementation of these two estimators. Fixing these problems either requires very efficient parallelisation [5] or more efficient algorithms are needed. More effective algorithms for 2 include a Fast Fourier Transform (FFT) for the synchronous case and a Non-Uniform FFT (NUFFT) for the more practical asynchronous case.

Remark C.1.1 *One important point to notice is that we did not use algorithm 1 to compute any correlation results. This is because the algorithm is producing the wrong correlation estimates for the asynchronous cases. The exact reason for this error has not yet been figured out. We know that it is algorithm 1 which is wrong rather than algorithm 2 because for the asynchronous case, the integrated variance should be the same as the HY estimate (since the variance is synchronous) - however it is not.*

[2] Figure C.1 can be reproduced using Compute Time.R

C.2 Supporting plots



(A) Geometric Brownian motion

(B) Merton Model

(C) GARCH(1.1) specification of [12]

(D) GARCH(1.1) specification of [23]

(E) Ornstein Uhlenbeck

(F) Variance Gamma

FIGURE C.2: Price paths, returns and QQ-plot for the various SDEs.

Figure C.2 highlights the resulting price paths generated using the Algorithms from appendix A, and further shows the returns and QQ-plots associated with the various SDEs considered in this paper.

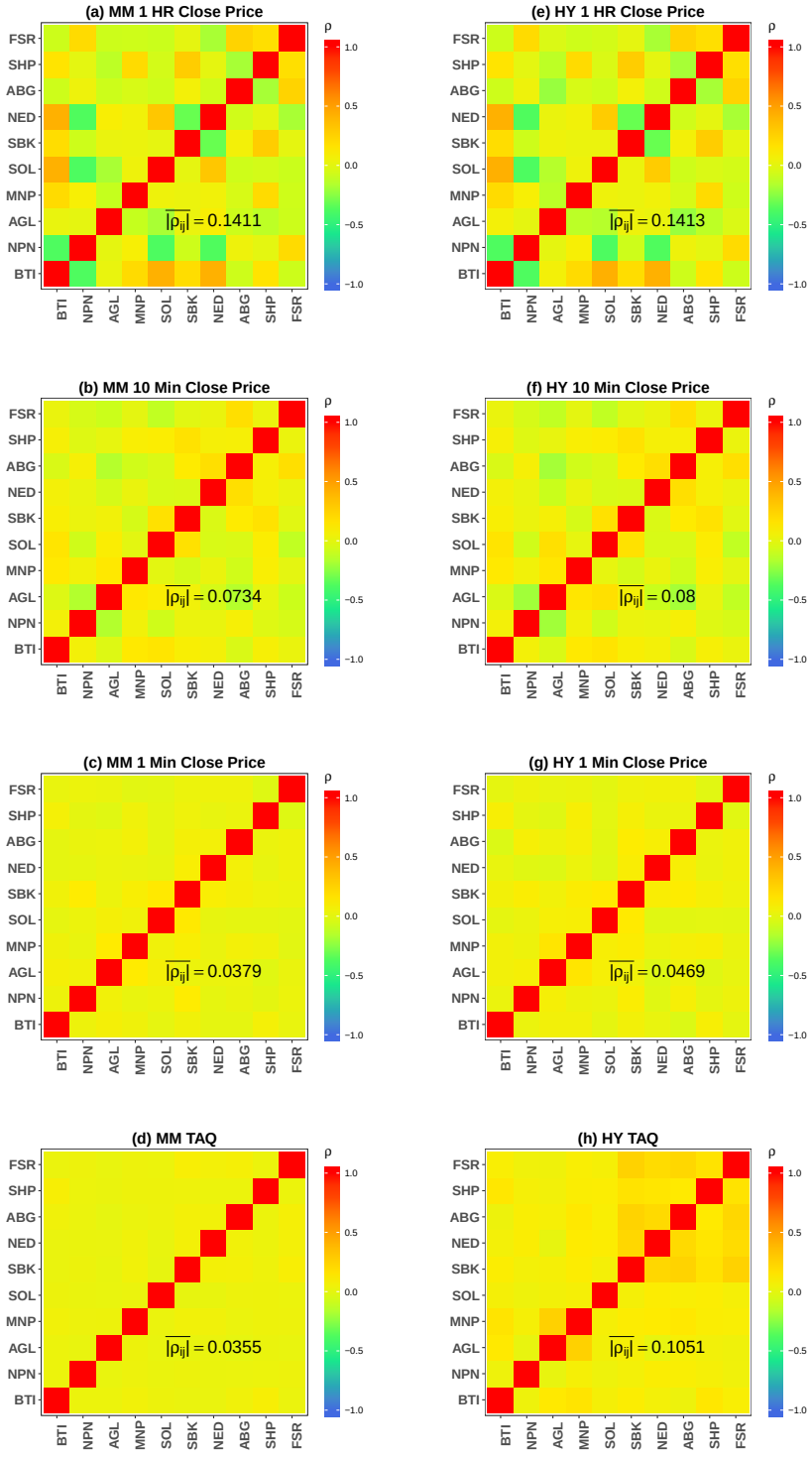


FIGURE C.3: Demonstrating the existence of the Epps effect when returns are all computed on the TAQ data rather than on the closing bar prices. From (a) to (d), we have the MM estimator applied to TAQ returns sampled every 1 hour, 10 minute, 1 minute and all TAQ return samples respectively using algorithm 2. From (e) to (h), we have the HY estimator applied to TAQ returns sampled every 1 hour, 10 minute, 1 minute and all the TAQ return samples respectively using algorithm 3. The Epps effect is still present even when all the returns are computed based on the highest available sampling frequency rather than on the respective sampling intervals.

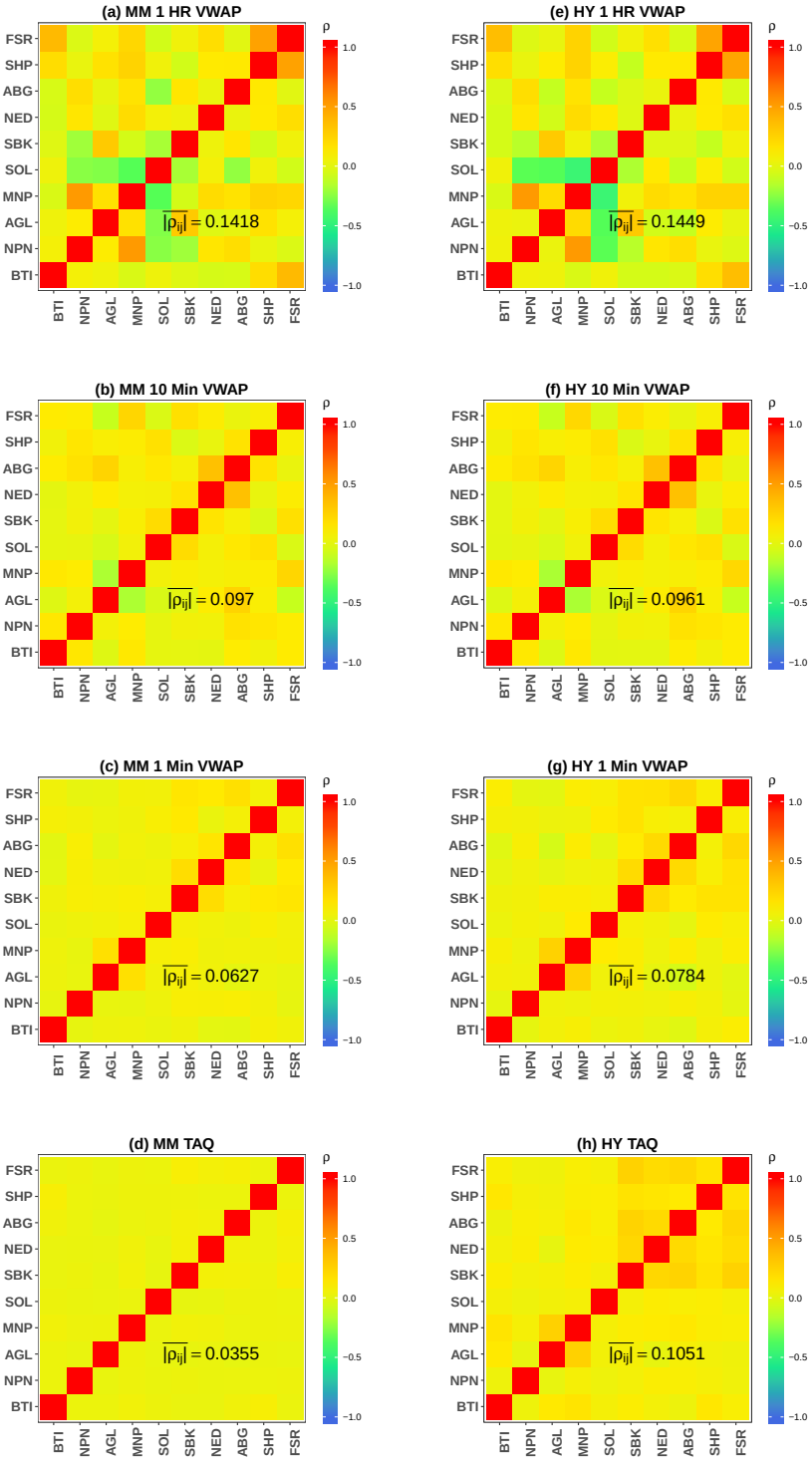


FIGURE C.4: Demonstrating the existence of the Epps effect when returns are all computed on the TAQ data and the VWAP aggregation is performed on the TAQ returns. From (a) to (d), we have the MM estimator applied to the VWAP aggregation of TAQ returns from 1 hour, 10 minute, 1 minute bars and pure TAQ samples respectively using algorithm 2. From (e) to (h), we have the HY estimator applied to the VWAP aggregation of TAQ returns from 1 hour, 10 minute, 1 minute bars and pure TAQ samples respectively using algorithm 3. The Epps effect is still present even under this bizarre aggregation of TAQ returns.

Figures C.3 and C.4 show the initial mistake we made with figures 5.1 and 5.2 respectively. The mistake was computing the returns before computing the OHCLV prices using algorithm 5, therefore we initially inputted the returns computed on the TAQ data into algorithm 5. Therefore the bar data in figure C.3 is rather the TAQ return samples sampled every 1 hour, 10 minute and 1 minute respectively while the bar data in figure C.4 is the VWAP algorithm applied to the TAQ return samples for 1 hour, 10 minute and 1 minute intervals respectively.

What is interesting about this mistake is that it still shows the existence of the Epps effect even though all the returns are computed based on the highest available sampling frequency rather than ranging sampling intervals. The only commonality with the samples in figures C.3, C.4 and 5.1, 5.2 is that the number of samples decrease as the sampling intervals decrease. This begs the question regarding the relationship between the Epps effect and the number of samples considered. Additionally, are there any factors other than asynchrony, lead-lag and smaller sampling intervals which contribute towards the Epps effect? Unfortunately, we have not been able to answer these questions.

Bibliography

[1] P. Malliavin and M. E. Mancino, "Fourier series method for measurement of multivariate volatilities", *Finance and Stochastics*, vol. 6, no. 1, pp. 49–61, 2002, ISSN: 0949-2984. DOI: 10.1007/s780-002-8400-6. [Online]. Available: https://doi.org/10.1007/s780-002-8400-6.

[2] ——, "A fourier transform method for nonparametric estimation of multivariate volatility", *Ann. Statist.*, vol. 37, no. 4, pp. 1983–2010, Aug. 2009. DOI: 10.1214/08-AOS633. [Online]. Available: https://doi.org/10.1214/08-AOS633.

[3] T. Hayashi and N. Yoshida, "On covariance estimation of non-synchronously observed diffusion processes", *Bernoulli*, vol. 11, no. 2, pp. 359–379, Apr. 2005. DOI: 10.3150/bj/1116340299. [Online]. Available: https://doi.org/10.3150/bj/1116340299.

[4] T. W. Epps, "Comovements in stock prices in the very short run", *Journal of the American Statistical Association*, vol. 74, no. 366, pp. 291–298, 1979, ISSN: 01621459. [Online]. Available: http://www.jstor.org/stable/2286325.

[5] D. Hendricks, "Using real-time cluster configurations of streaming asynchronous features as online state descriptors in financial markets", *Pattern Recognition Letters*, vol. 97, pp. 21 –28, 2017, ISSN: 0167-8655. DOI: https://doi.org/10.1016/j.patrec.2017.06.026. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865517302325.

[6] I. Mastromatteo, M. Marsili, and P. Zoi, "Financial correlations at ultra-high frequency: Theoretical models and empirical estimation", *The European Physical Journal B*, vol. 80, no. 2, pp. 243–253, 2011, ISSN: 1434-6036. DOI: 10.1140/epjb/e2011-10865-y. [Online]. Available: https://doi.org/10.1140/epjb/e2011-10865-y.

[7] E. Barucci and R. Renò, "On measuring volatility and the garch forecasting performance", *Journal of International Financial Markets, Institutions and Money*, vol. 12, pp. 183–200, Jul. 2002. DOI: 10.1016/S1042-4431(02)00002-1.

[8] M. Mancino, M. Recchioni, and S. Sanfelici, *Fourier-Malliavin Volatility Estimation Theory and Practice*. Springer International Publishing, Mar. 2017, ISBN: 978-3-319-50967-9. DOI: 10.1007/978-3-319-50969-3.

[9] M. Mancino and S. Sanfelici, "Robustness of fourier estimator of integrated volatility in the presence of microstructure noise", *Computational Statistics Data Analysis*, vol. 52, pp. 2966–2989, Feb. 2008. DOI: 10.1016/j.csda.2007.07.014.

[10] J. E. Griffin and R. C. Oomen, "Covariance measurement in the presence of non-synchronous trading and market microstructure noise", *Journal of Econometrics*, vol. 160, no. 1, pp. 58–68, 2011. [Online]. Available: https://ideas.repec.org/a/eee/econom/v160y2011i1p58-68.html.

[11] Y. AÃt-Sahalia, J. Fan, and D. Xiu, "High-frequency covariance estimates with noisy and asynchronous financial data", *Journal of the American Statistical Association*, vol. 105, pp. 1504–1517, Jun. 2010. DOI: 10.2139/ssrn.1631344.

[12] R. Renò, "A closer look at the epps effect", *International Journal of Theoretical and Applied Finance*, vol. 06, Nov. 2001. DOI: 10.2139/ssrn.314723.

[13] O. V. Precup and G. Iori, "Cross-correlation measures in the high-frequency domain", *The European Journal of Finance*, vol. 13, no. 4, pp. 319–331, 2007. DOI: 10.1080/13518470600813565. eprint: https://doi.org/10.1080/13518470600813565. [Online]. Available: https://doi.org/10.1080/13518470600813565.

[14] E. Derman, "The perception of time, risk and return during periods of speculation", *Quantitative Finance*, vol. 2, no. 4, pp. 282–296, 2002. DOI: 10.1088/1469-7688/2/4/304. eprint: https://doi.org/10.1088/1469-7688/2/4/304. [Online]. Available: https://doi.org/10.1088/1469-7688/2/4/304.

[15] C. Malherbe, "Fourier method for the measurement of univariate and multivariate volatility in the presence of high frequency data", Master's thesis, University of Cape Town, 2007.

[16] T. Hoshikawa, K. Nagai, T. Kanatani, and Y. Nishiyama, "Nonparametric estimation methods of integrated multivariate volatilities", *Econometric Reviews*, vol. 27, no. 1-3, pp. 112–138, 2008. DOI: 10.1080/07474930701853855. eprint: https://doi.org/10.1080/07474930701853855. [Online]. Available: https://doi.org/10.1080/07474930701853855.

[17] D. Hendricks, T. Gebbie, and D. Wilcox, "High-speed fourier method estimation of covariances from asynchronous data", working paper, 2017.

[18] T. Gebbie, D. Wilcox, C. Malherbe, and D. Hendricks, *Ftcorrgpu.m*, 2005.

[19] T. Kanatani, "Optimally weighted realized volatility", Jan. 2004.

[20] B. Tóth and J. Kertész, "The epps effect revisited", *Quantitative Finance*, vol. 9, no. 7, pp. 793–802, 2009. DOI: 10.1080/14697680802595668. eprint: https://doi.org/10.1080/14697680802595668. [Online]. Available: https://doi.org/10.1080/14697680802595668.

[21] W Moon and J. S. Wettlaufer, "On the interpretation of stratonovich calculus", *New Journal of Physics*, vol. 16, no. 5, p. 055 017, 2014. DOI: 10.1088/1367-2630/16/5/055017. [Online]. Available: https://doi.org/10.1088%2F1367-2630%2F16%2F5%2F055017.

[22] P. Glasserman, *Monte Carlo methods in financial engineering*. New York: Springer, 2004, ISBN: 0387004513 9780387004518 1441918221 9781441918222. [Online]. Available: http://www.amazon.com/Financial-Engineering-

Stochastic-Modelling-Probability/dp/0387004513/ref=pd_sim_b_68?ie=UTF8&refRID=1AN8JXSDGMEV2RPHFC2A.

[23] T. G. Andersen and T. Bollerslev, "Answering the skeptics: Yes, standard volatility models do provide accurate forecasts", *International Economic Review*, vol. 39, no. 4, pp. 885–905, 1998, ISSN: 00206598, 14682354. [Online]. Available: http://www.jstor.org/stable/2527343.

[24] T. G. Andersen and T. Teräsvirta, "Realized volatility", in *Handbook of Financial Time Series*, T. Mikosch, J.-P. Kreiß, R. A. Davis, and T. G. Andersen, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 555–575, ISBN: 978-3-540-71297-8. DOI: 10.1007/978-3-540-71297-8_24. [Online]. Available: https://doi.org/10.1007/978-3-540-71297-8_24.

[25] A. G. Hawkes, "Spectra of some self-exciting and mutually exciting point processes", *Biometrika*, vol. 58, no. 1, pp. 83–90, 1971, ISSN: 00063444. [Online]. Available: http://www.jstor.org/stable/2334319.

[26] R. Arbi, "A reproducible approach to equity backtesting", Master's thesis, University of Cape Town, 2019.

[27] *Volume 00e-trading and information overview*, Johannesburg Stock Exchange, Sandown, Johannesburg, South Africa, 2019.

[28] D. Easley, R. F. Engle, M. O'Hara, and L. Wu, "Time-Varying Arrival Rates of Informed and Uninformed Trades", *Journal of Financial Econometrics*, vol. 6, no. 2, pp. 171–207, Feb. 2008, ISSN: 1479-8409. DOI: 10.1093/jjfinec/nbn003. eprint: http://oup.prod.sis.lan/jfec/article-pdf/6/2/171/2594496/nbn003.pdf. [Online]. Available: https://doi.org/10.1093/jjfinec/nbn003.

[29] D. Easley, M. M. López de Prado, and M. O'Hara, "The volume clock: Insights into the high-frequency paradigm", *The Journal of Portfolio Management*, vol. 39, no. 1, pp. 19–29, 2012, ISSN: 0095-4918. DOI: 10.3905/jpm.2012.39.1.019. eprint: https://jpm.pm-research.com/content/39/1/19.full.pdf. [Online]. Available: https://jpm.pm-research.com/content/39/1/19.

[30] D. Easley, M. Lopez de Prado, and M. O'Hara, "Flow toxicity and liquidity in a high frequency world", *Review of Financial Studies*, vol. 25, Feb. 2012. DOI: 10.2139/ssrn.1695596.

[31] T. Gebbie, D. Wilcox, C. Malherbe, and D. Hendricks, *Fftcorrgpu.m*, 2005.

[32] P. Chang, R. Bukuru, and T. Gebbie, 2019. [Online]. Available: https://github.com/rogerbukuru/Honours-Project.

[33] B. Oksendal, *Stochastic Differential Equations (3rd Ed.): An Introduction with Applications*. Berlin, Heidelberg: Springer-Verlag, 1992, ISBN: 3-387-53335-4.