Instituto Nacional de Matemática Pura e Aplicada

Optimal Execution of Co-integrated Brazilian Stocks with Multivariate Ornstein-Uhlenbeck Dynamics

Author: **Adolfo Correia**

Supervisor: **Yuri Saporito**

Rio de Janeiro
June 20, 2023

To God and to my family.

ACKNOWLEDGMENTS

I would like to acknowledge everyone who somehow helped or supported me during the personal and academic journey that culminated in the present work.

At first, I thank all professors, teaching assistants and both IMPA's and FGV's staff for the inciting mathematical knowledge bestowed throughout the course, as well as the auspicious learning environment that was provided. In this regard, I must render thanks to Jorge Zubelli, Margaret Armstrong, Rodrigo Targino, Vinicius Ramos, Hugo Cancino, Moacyr Alvim, Paulo Cezar Carvalho, Yuri Saporito, Fernando Aiube, Edison Tito, Daniel Tordecilla and Dyego Soares.

Next, I extend my gratitude to all classmates with whom I had the opportunity to share this undertaking, particularly, Ciro Paolucci, Eduardo Brasil, Felipe Fernandes, Francis Araújo, Lucas Carvalho, Luiz Martins, Maurício Darós and Yuri Fonseca. Additionally, I extend my thanks to each teammate in FGV's Financial Mathematical Team Challenge, namely, Ali Al-Aradi, Danilo Naiff, Gabriel Jardim and to our supervisor Yuri Saporito. Much of the inspiration for the present work originally came from those intense ten days.

Naturally, this work would not have been possible without the priceless supervision of Yuri Saporito, whom I must acknowledge for the research guidance, valuable insights and incentive, both as professor and as supervisor. I also specially acknowledge David Evangelista and Moacyr Alvim for reviewing this work as members of my committee and providing opportune suggestions.

Not least of all, I thank my family for their unconditional support through this time, in particular my parents, Paulo and Elizabeth, my sister Raquel and my lovely and inspiring wife Silvana.

ABSTRACT

In this work, we apply the Bergault-Drissi-Guéant (BDG) optimal liquidation model to stocks in the Brazilian market. The model at hand assumes the price processes follow a mean-reverting, multidimensional Ornstein-Uhlenbeck dynamics, which allows for both correlation and co-integration dependency structures between the assets. As a baseline, we compare the results, measured in terms of the resulting inventory and wealth processes, to both single and multidimensional versions of the seminal Almgren-Chriss (AC) optimal liquidation model. In the studied case, the BDG model is able to outperform the AC models in terms of both the mean wealth level and its variance, notwithstanding its unconventional inventory process profile. As an additional line of work, we apply the Deep Galerkin Method for partial differential equation solving to the BDG model's Hamilton-Jacobi-Bellman equation. Even though such novel method is able to satisfactorily tackle simpler forms of the problem, it does not successfully generalize to the case at hand, arguably because of the numerically problematic profile of the solution.

Keywords: Limit Order Book, Optimal Portfolio Liquidation, Almgren-Chriss Model, Bergault-Drissi-Guéant Model, Ornstein-Uhlenbeck Process, Co-integrated Series, Hamilton-Jacobi-Bellman Equation, Deep Galerkin Method.

RESUMO

No presente trabalho, o modelo de liquidação ótima de Bergault-Drissi-Guéant (BDG) é aplicado a ações do mercado brasileiro. Tal modelo assume que os preços dos ativos seguem uma dinâmica multidimensional de reversão à média do tipo Ornstein-Uhlenbeck, que suporta tanto estruturas de correlação quanto de co-integração no que diz respeito à dependência entre as ações. Os resultados, mensurados com base nos processos de inventário e de riqueza obtidos, são comparados com o modelo de Almgren-Chriss (AC) em suas versões uni e multidimensional. No caso estudado, o modelo BDG é capaz de superar os modelos AC, tanto em relação ao nível médio de riqueza quanto em relação à sua variância, apesar do perfil pouco convencional do processo de inventário resultante. De modo adicional, também são aplicadas às equações de Hamilton-Jacobi-Bellman do modelo BDG técnicas de solução de equações diferenciais parciais baseadas no Método Profundo de Galerkin. Apesar de tal método ser capaz de solucionar satisfatoriamente formas mais simples do problema, ele não é capaz de atender o caso mais geral em tela, presumivelmente em razão do perfil numérico problemático da solução.

Palavras-chave: Livro de Ordens, Liquidação Ótima de Portfólios, Modelo de Almgren-Chriss, Modelo de Bergault-Drissi-Guéant, Processo de Ornstein-Uhlenbeck, Séries Co-integradas, Equação de Hamilton-Jacobi-Bellman, Método Profundo de Galerkin.

CONTENTS

1 INTRODUCTION

Capital markets are essential to the modern world economies and, as comprehensively and deeply as possible, it is critical to examine and understand their characteristics and behavior. Particularly with respect to secondary markets, a notable phenomenon is that of the impact on price levels due to successive or sizable buy and sell orders. Besides being intrinsically linked to the basilar supply and demand dynamics of any market, this phenomenon can also be concretely explained as a result of the organization of exchanges around Limit Order Books (LOBs). As later expounded, it is possible both to empirically quantify such impacts and to develop financial models that take them into account. A critical application of such knowledge is that of asset liquidation, particularly when the assets' disposal is required to take place in a short time frame or, equivalently, in large volumes with respect to the available liquidity pool. In such scenarios, the liquidating agent may incur in significant losses if a naive strategy is adopted without better discernment of potential price impacts. On the other hand, with proper risk management, a well balanced strategy can better handle the inescapable trade-off between liquidation speed and the expected losses due to price impacts.

After the present introduction, this work begins by describing in Chapter 2 some fundamental concepts related to the structure of exchanges, particularly (i) the orders agents use to materialize their buying or selling intentions to other market participants and (ii) the LOB, which is the logical data structure in which the orders are consolidated and matched against each other to result in trades. Further, it is also illustrated how LOBs can be used to quantify potential price impacts on incoming orders. The chapter ends with a description of the raw order data that are used in the present work.

In Chapter 3, we present two optimal liquidation models, namely the Almgren-Chriss (AC) and the Bergault-Drissi-Guéant (BDG) ones. The first is a classic and well known model in the finance literature, while the second is a more recent development which represents a generalization for the case with co-integrated assets. All the models' main concepts and equations are sub-sequentially presented, and their respective solutions are explained in detail. The chapter ends with a discussion on the procedures used in order to estimate all models' parameters.

In Chapter 4, the general concept of time series co-integration is defined and explained. We also present some statistical tests that can be used in order to identify co-integrated series. The chapter is concluded with a brief discussion of co-integration in the context of continuous time processes.

All theory presented in the previous chapters is finally applied in Chapter 5. We begin it by describing the real price series used and the respective co-integration tests. We then estimate the models' parameters from such price and order data and solve all models, obtaining their respective control and inventory curves. The models are then compared to each other in relation to their wealth process, both in terms of their mean levels and variances. The chapter is concluded with a discussion about the sensitivity of the models to some of their parameters, particularly the risk aversion and terminal penalty ones, which are neither observed nor estimated, and must be discretionarily chosen.

In Chapter 6, we present an alternative numerical solution to the BDG model equations, which is based on neural networks techniques. We begin by describing the general method and then we apply it to the model's partial differential equation using a simplified parameter set. We gradually increase the difficulty of the optimization problem by changing the parameter values and adjusting the algorithm accordingly. Unfortunately, such efforts fail to generalize when applied to the parameter set with the previously estimated values. We conclude the chapter with a brief discussion on the possible reasons for such result.

We finally conclude this work in Chapter 7 by wrapping up all addressed themes, including the obtained results and suggestions for future work.

2 MARKET STRUCTURE AND PRICE IMPACT

In general terms, exchanges can be characterized as organized markets where securities, such as stocks, options, or futures, are exchanged between matching buyers and sellers upon financial settlement. Besides their role as trading facilitators, exchanges also typically provide many other core services such as market-making, securities clearing, settlement, registration and deposit, collateral management, among others. They are also responsible for much of the enabling infrastructure for the smooth operation of financial markets worldwide. As such, exchanges are paramount for liquidity provision, price discovery, and transparency in the markets in which they operate, performing an essential economic role.

Even though bilateral, over-the-counter (OTC) transactions are common in some less liquid markets (*e.g.*, corporate bond markets), exchange-based trading platforms became dominant in recent times in more developed ones (*e.g.*, stock markets). Such trading platforms, mostly electronic nowadays, are predominantly organized around *Limit Order Books* (LOB) to which prospective buyers and sellers post buy and sell orders during trading periods. In addition to the more regular continuous trading phase where individual buyers are matched against individual sellers, an order pair at a time, there can also be periods of auctions where several agents (both buyers and sellers) interact simultaneously to form a resulting settled price for a group of orders. Auctions are common during opening and closing hours, but they can also take place throughout the trading day, in response to unusual events (*e.g.*, volatility shocks) accordingly to the exchange's rules. In any case, both continuous trading and auctions rely on the LOB as a central repository for all available orders.

In the following sections, we describe, in general terms, how LOBs work, and present the main and most relevant types of orders with respect to the present work. We also discuss the concept of price impact which arises due to this market structure. This chapter's exposition has as its main references the introductory chapters in Cartea et al. (2015) and Guéant (2016).

2.1 Order types and the Limit Order Book

For any trade to come about in a marketplace, it is essential that (i) market agents are able to signal their willingness to buy or sell the assets at hand and that (ii) corresponding buyers and sellers are matched against each other. These two conditions are satisfied, respectively, by orders and the matching algorithms that rely on the LOB.

In a basic setup, an agent's buy (or sell) deal intentions can be signaled to the market via orders that express his transaction conditions. There are two main types of orders: *limit orders* and *market orders*. A trader issuing a limit order signals the market his intention of buying (or selling), at a predetermined price, a maximum quantity of shares of a given stock[1]. This type of order can be considered "limited" in the sense that the price paid (or received) per share in the case of buy (or sell) orders is limited in the agent's favor[2]. On the other hand, a trader issuing a market order signals to the market his intention of buying (or selling), at the best available price, a predetermined quantity of shares of a given stock. Since a market order usually results in an immediate trade, market orders are considered *aggressive*. Equivalently, limit orders are considered *passive*, since they usually do not result in immediate trades and require a new upcoming order to be matched against and eventually executed[3]. Therefore, a key distinction between both types of orders is that limit orders guarantee the price level of the transaction but do not assure its occurrence, while market orders guarantee execution and quantity but do not guarantee price levels.

Still taking into account this basic setup, it is necessary for orders to be matched against each other and eventually executed. Matching algorithms typically depend on the LOB, which can be characterized as a data structure that holds and keeps track of all current and valid orders. For each price level (*e.g.*, one cent ticks), the LOB maintains a list of all current (and available for execution) limit orders. Figure 2.1 illustrates a portion[4] of PETR4's LOB on November 27$^{\text{th}}$, 2019, exactly at 15:00. Each vertical bar represents the total available volume, indicated by the y-axis, at several price levels, indicated by the x-axis. The total volume at each price level is obtained by the sum of the (remaining) volumes of all valid limit orders with the corresponding price[5]. LOBs are composed of two sides that represent the available buy and sell limit orders. On Figure 2.1, each side is represented with a different color, with the buy side to the left and the sell side to the right of the graph. The highest price on the buy side is usually called *bid-price* and represents the maximum price a buyer is willing to pay for a share of stock at a given moment. Analogously, the lowest price on the sell side is usually called *ask-price* and represents the minimum price a seller is willing to take for a share of the same stock. The difference between the bid and ask prices is denoted as *bid-ask spread* and the average between these prices is called *mid-price*, which is a common indicator for the spot price for a given stock[6]. It should be noted that, by construction,

1 The present discussion is broadly applicable to financial securities in general, but we restrict our scope to stock shares for convenience's sake.

2 It is possible, nevertheless, that the buyer pays less per share (or, respectively, that the seller receives more) in special circumstances (*e.g.*, auctions).

3 In the real world, nothing prevents agents from issuing limit buy (sell) orders at price levels above the current ask price (below the bid price). These limit orders should also be considered aggressive since they trigger trades. Nonetheless, for simplification's sake, we assume aggressive limit orders are not allowed.

4 Only a few price levels are displayed at the graph.

5 Individual orders are not represented on Figure 2.1.

6 Another usual indicator is the volume weighted average of the bid and ask prices.

the bid price is always strictly smaller than the ask price, since only unmatched limit orders are kept in the LOB.

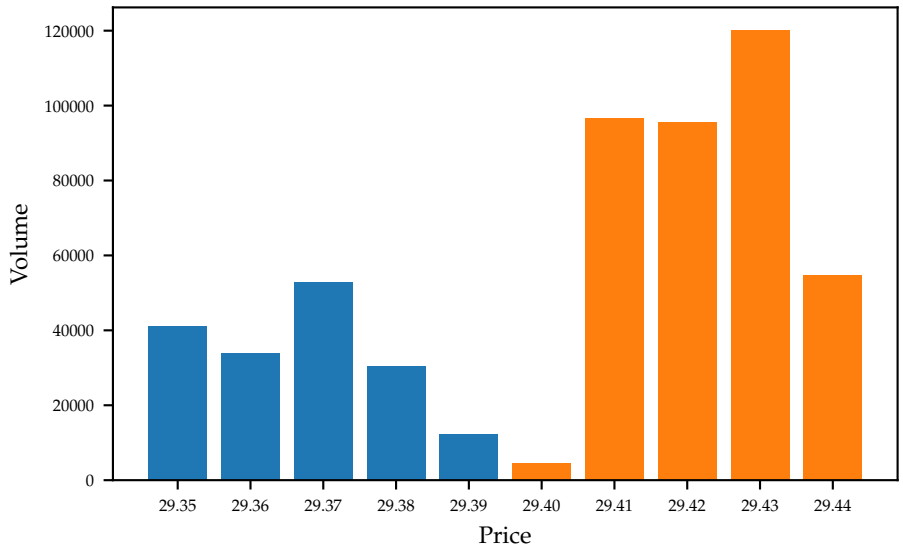


Figure 2.1: PETR4 Limit Order Book (2019–11–27 15:00)

Each time a new order arrives at the market during trading hours, an accompanying matching engine checks if one or more trades can take place and if so, processes the resulting transactions. Otherwise, if a new (limit) order cannot be matched against any other order, it is added to the corresponding price level in the LOB. In general, market orders are always matched against previous limit orders according to the following (simplified) rules:

- At first, the new buy (sell) market order is matched against the available sell (buy) limit orders that offer the best, lowest (highest), price in the LOB;
- If there are more than one sell (buy) limit order available at the best price level, the market order is matched against the older (first to arrive) limit order;
- Matched orders are processed (executed) against each other taking into account the minimum available volume among both, which may give rise to partially executed orders;
- Completed orders (with all requested volume matched and processed) are removed from the LOB;
- If the market other at hand is bigger (in terms of quantity demanded) than the previously matched limit order, then the next older limit order at the best price level is matched (and executed);
- When the quantity demanded by the market order is larger than the volume available at the best price, it is sequentially matched against limit orders at the next best price level, following the rules above, until the whole order volume is processed.

Although the description presented above is somewhat simplistic when compared to what happens on exchanges in the real world, it is sufficient for the purposes at hand. It also highlights one last key distinction between market and limit orders concerning liquidity: while limit orders usually provide liquidity to the market by adding volume to the LOB, market orders consume it by removing volume.

2.2 Price impact

Given the trading framework presented in the previous section, one can infer that market orders always take place at worse price levels than the current mid-price (spot). At best, small buy (sell) market orders can occur at the ask (bid) prices which, by definition, are different from the mid-price. On the other hand, big market orders may be executed at significantly worse prices insofar as they are matched against continuously worse priced orders. This process is usually referred as "walking" the LOB. The resulting price of an executed market order, which corresponds to the volume weighted average prices of all matched limit orders, is usually denoted as its *effective price*.

As an illustration of walking the LOB, consider the order data displayed in Table 2.1, which correspond to the same LOB indicated by Figure 2.1. Suppose that an agent sends a small buy market order for just 3,000 shares. The available volume present at the first price level on the sell side of the LOB (4,400 shares at \$29.40) is sufficient to cover this order. The agent will pay[7] a total of \$88,200 monetary units[8] for 3,000 stocks, which correspond to an effective price of \$29.40 per share. Alternatively, suppose the agent sends a bigger buy market order of 300,000 shares. Now, it is clear that the volume available at the first price level on the sell side of the LOB is not enough to cover this order. In fact, we need to resort to the first four price levels on the sell side of the LOB to account for the demanded volume. The agent will now pay a total of \$8,825,979 monetary units[9] for 300,000 stocks, which correspond to an effective price of approximately \$29.42 per share. Since the mid-price is \$29.395, the price impacts for both scenarios are approximately \$0.005 and \$0.025, respectively. The effective price for buy market orders increases monotonically with volume. Conversely, the effective price for sell market orders decreases monotonically with volume. This phenomenon is illustrated in Figure 2.2 which presents the effective price curves as functions of order volume for both buy and sell orders extracted from the same LOB. Note, in particular, that outlier limit orders with price levels significantly different from the mid-price can severely impact the effective price curves for larger volumes. The Figure 2.2 also exhibits the linear approximations of such effective price curves, considering only 75% of total volume in each side of the LOB (in order to avoid outlier effects).

7 For simplification's sake, we are ignoring transaction costs in this illustration.

8 Amount given by 29.40×3000.

9 Amount given by $29.40 \times 4400 + 29.41 \times 96700 + 29.42 \times 95500 + 29.43 \times 103400$.

Table 2.1: PETR4 Limit Order Book (2019–11–27 15:00)

Buy Side		Sell Side	
Volume	Price	Price	Volume
12 300	$29.39	$29.40	4400
30 500	$29.38	$29.41	96 700
52 800	$29.37	$29.42	95 500
33 900	$29.36	$29.43	120 200
41 200	$29.35	$29.44	54 800
31 100	$29.34	$29.45	59 100
32 600	$29.33	$29.46	124 700
34 000	$29.32	$29.47	36 600

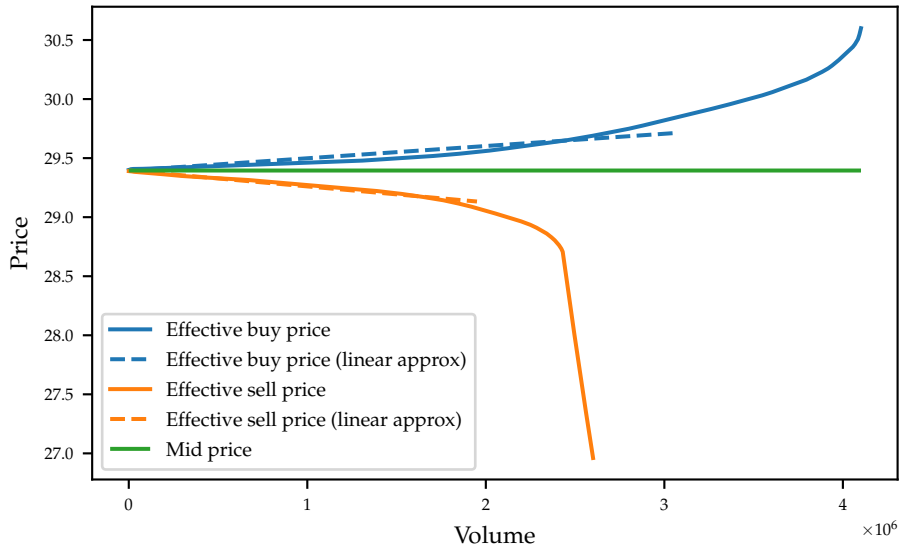


Figure 2.2: PETR4 effective price curves (2019–11–27 15:00)

It is clear from the discussion above that market agents can affect prices when trading, in particular when their market orders walk the LOB consuming liquidity. Some authors distinguish between two kinds of price impact: *temporary* and *permanent*. According to Almgren and Chriss (2001), "temporary impact refers to temporary imbalances in supply and demand caused by trading leading to temporary price movements away from equilibrium" and "permanent impact means changes in the equilibrium price due to trading". Thus, the key distinction between both types of impact is influence on the "equilibrium" price. In their seminal work on optimal execution, Almgren and Chriss model both impacts as linear functions that depend on the rate of trade. For an alternative and somewhat different approach, also considering a more empirical point of view, Cartea et al. (2015) estimate the permanent price impact linearly relating it to the order flow[10] using time intervals of five minutes and the temporary price impact linearly

10 Order flow is the difference between the volumes of buy and sell market orders during a time interval.

relating it to order volume using one second apart snapshots of the LOB.

2.3 Brazilian stock market order data

Recall that, so as to build LOBs, one needs to resort to the complete order dataset and not merely rely on intraday price series or carried out trades information. The LOB is the fundamental data structure that allows one to estimate temporary price impact parameters. Taking into account the fact the optimal liquidation models presented in the present work rely on price impact information and that we want to apply such models in real world based asset liquidation simulation scenarios, we have chosen to use real LOBs built from intraday order data obtained from the Brazilian stock market (B3).

Daily data files with all orders submitted and processed during each trading day used to be freely available on B3's FTP servers up until mid 2020 via the "Market Data" service[11]. Unfortunately, access to B3's file servers has been revoked since then[12], and, to the best of our knowledge, B3 does not provide order data any longer (even though intraday trade data are still available). Anyhow, we had access to complete order data from 2019, which we use in the present work.

For any given trading day, there used to be two files with buy and sell orders, respectively. A third, somewhat redundant, file was also available which contained trades data. Each line in the orders files corresponds to an order record with information such as:

- Trading session date;
- Stock ticker;
- Order direction (buy/sell);
- Order sequence (identification) codes;
- Order event, which refers to the order's stage in relation to its life-cycle (*e.g.*, order entry, order update, order cancellation, order expiration, order trade);
- Order priority indicator;
- Order price;
- Order total and traded quantities;
- Order inclusion date and time;
- Order status (*e.g.*, new, partially executed, totally executed, canceled, updated, expired);

11 Market Data service accessible at https://b3.com.br/en_us/market-data-and-indices/data-services/market-data.

12 For the file servers deactivation notice, see B3's Circular Letter 018/2020-PRE, issued on February 27th 2020, retrievable at https://b3.com.br/en_us/regulation/circular-letters-and-external-communications.

- Order trade condition (neutral, passive or aggressive order).

For more information on B3's order and trade files, refer to the (now deprecated) R language package `GetHFData` (Perlin & Ramos, 2016).

As part of the present work, we have developed a computer program to read the raw order data files, parse all order records and add each incoming order into an evolving and dynamic LOB. The program also takes periodic snapshots (every one minute) of the LOB and extracts or computes several statistics, such as:

- Bid, ask and mid-prices;
- Bid-ask spreads;
- Temporary price impact coefficients.

3 THE OPTIMAL LIQUIDATION PROBLEM

Portfolio managers and other financial agents frequently need to liquidate positions in short notice in order to comply with risk mandates or to fulfill withdrawal requests from clients, for instance. Often enough, a trade-off emerges between execution speed and cost minimization, because of the price impacts typically entailed by fast trading. Almgren and Chriss (2001) provide some of the earliest rigorous treatments of optimal execution, and their work remain influential to this day. They frame the optimal execution problem in terms of maximizing the expected revenue of trading (or equivalently minimizing trading costs) with a penalty for the uncertainty of revenue (or cost). Ensuing this seminal work, many other models and extensions have been proposed by the academia ever since, and many of those have made their way into the real world practice of financial markets.

The original AC model is discrete in time and uni-dimensional with respect to the number of assets, whose price processes are described with a Bachelier dynamics (arithmetic Brownian motion). Both temporary and permanent price impacts are linear functions in relation to the trade rate. The efficient trading strategies are obtained as solutions to mean-variance optimization problems. Furthermore, given its relative simplicity and straightforwardness, its solution (*i.e.*, the optimal execution rate) can be written in analytical form.

Extensions to this benchmark model have been proposed in several directions (*e.g.*, continuous time, multi-asset, mean-reverting price dynamics, non-linear price impacts, alternative optimization techniques). With respect to the work at hand, we are particularly interested in exploring multi-asset trades with possibly mean-reverting or co-integrated dynamics. In practice, market agents often hold multi-asset portfolios and need to liquidate them simultaneously and, even in single asset liquidation scenarios, it might be beneficial to factor other assets into the strategy. Evidently, single-asset models are not able to capture eventual joint price dynamics such as innovation correlation or even series co-integration. Besides, many existing multi-asset models simply consider correlated Brownian motions when modeling the joint dynamics of the price processes, but such models may be overly unrealistic and unable to capture portfolio-level impacts and costs. Multivariate Ornstein-Uhlenbeck (OU) dynamics are particularly attractive in this context, since such models are still relatively lean and sufficiently general to cover many multidimensional dynamics, including correlated Brownian motions and mean-reverting or co-integrated ones. Besides, such models are widely used in the financial world, both by academia and practitioners.

Almgren and Chriss propose a multi-asset extension to their original model in an appendix of their seminal paper. As before, this model is discrete in time and assumes a Bachelier dynamics for the price processes. The multidimensional facet of the model is handled by the employment of correlated Brownian increments. This pioneer model can be considered as a benchmark for comparison with subsequent extensions. One of the first multi-assets model to incorporate an OU dynamics is developed in Cartea et al. (2018), in which the execution problem is treated as an optimal stochastic control problem. The paper applies the model to the execution of a basket of co-integrated stocks and shows how the additional dynamics information can be used to outperform Almgren-Chriss' model. The authors assume the agent wants to maximize an objective function given by the expected profit and loss (PnL) abated by a running penalty linked to the variance of the portfolio. Another similar model is the one proposed in Bergault et al. (2022), hereby referred as Bergault-Drissi-Guéant (BDG), which uses as its objective function an expected exponential utility of the PnL. In the following sections, the multivariate AC and the BDG models are described in detail.

3.1 Almgren-Chriss multi-asset model

In this section, we describe the modeling framework, notation and main results presented in Almgren and Chriss (2001). The present exposition focus on the multi-asset version of the model. For a more thorough discussion of the Almgren-Chriss (AC) model, including the original mathematical results, we refer the reader to the paper itself.

The AC model considers a market with $m \in \mathbb{N}$ assets and an agent who wishes to completely liquidate a portfolio over a period of time $[0, T]$, with $T > 0$. The authors divide the trading period into $N \in \mathbb{N}$ time intervals of length $\tau = T/N$ and define the discrete times $t_k = k\tau$, for $k = 0, \dots, N$. They also define the *trading trajectory* to be a list $x_0, \dots, x_N$, where $x_k = (x_{1,k}, \dots, x_{m,k})^\intercal$ are column vectors representing the inventory, *i.e.*, the quantity of each stock, held at time t_k. The initial inventory is $x_0 = X = (X_1, \dots, X_m)^\intercal$ and complete liquidation requires $x_N = 0$. Equivalently, it is possible to specify a trading strategy by the *trade list* $n_1, \dots, n_N$, where $n_k = x_{k-1} - x_k$ is the column vector indicating the number of stocks sold between times t_{k-1} and t_k. The trading trajectory, the trade list and the initial inventory are related by the equations

$$x_k = X - \sum_{j=1}^{k} n_j = \sum_{j=k+1}^{N} n_j,$$

for $k = 0, \dots, N$. Note that, if $x_{jk} < 0$, then the stock j is held short at time t_k; and if $n_{jk} < 0$, then the stock j is being bought between t_{k-1} and t_k.

As usual, the AC model assumes that the stock prices S_k evolve according to the two exogenous factors of volatility and drift. Volatility and drift are exogenous in the sense that they are assumed to be

the result of market forces that occur randomly and independently of the agent's trading. Nonetheless, as a simplification, the authors assume the drift factor is null since the agent has no information about the direction of future price movements. As expected, the model also incorporates the endogenous factor of permanent market impact, which model deviations from the "equilibrium" price due to the agent's trading. Therefore, S_k follow a multidimensional arithmetic Brownian random walk with zero drift and a permanent market impact term. Its dynamics is given by

$$S_k = S_{k-1} + \sigma\sqrt{\tau}\,\xi_k - \tau\, g\Big(\frac{n_k}{\tau}\Big)\,, \tag{3.1}$$

for $k = 1, \ldots, N$, where $\xi_k = (\xi_{1k}, \ldots, \xi_{rk})^\intercal$ are vectors of r independent and identically distributed standard Gaussian random variables, with $r \leq m$, and $\sigma \in \mathbb{R}^{m\times r}$ represents the volatility matrix of the assets. Furthermore, σ is such that $C = \sigma\sigma^\intercal$, with $C \in \mathbb{R}^{m\times m}$, is a symmetric positive-definite covariance matrix. Additionally, $g, h : \mathbb{R}^m \to \mathbb{R}^m$ are, respectively, the permanent and temporary impact functions of the average rate of trading $v_k = n_k/\tau$ during the interval $(t_{k-1}, t_k]$. It is assumed that both impact functions are linear, with expressions given by

$$g(v) = \Gamma v \tag{3.2}$$

and

$$h(v) = \epsilon \odot \operatorname{sgn}(v) + Hv, \tag{3.3}$$

where $\Gamma, H \in \mathbb{R}^{m\times m}$ are matrices whose ij-element represents the price depression on stock i caused by selling stock j at a unit rate, $\epsilon \in \mathbb{R}^m$ is a column vector representing fixed costs of selling (*e.g.*, half the bid-ask spread plus fees), and $\odot$ represents the Hadamard (element-wise) product operator. It is required that H be positive-definite (*i.e.*, $v^\intercal Hv > 0$ for any non-zero column vector v), but it is not necessary for Γ and H to be symmetric. If H were not positive-definite, then there could be a non-zero selling rate v with $v^\intercal Hv \leq 0$ that would result in a net benefit (or at least no loss) obtained from instantaneous market impact. Given the temporary price impact, the actual (or effective) prices per share received on sale k can be obtained by

$$\tilde{S}_k = S_{k-1} - h\Big(\frac{n_k}{\tau}\Big)\,. \tag{3.4}$$

As expected, the temporary effect due to h does not influence the price process S_k.

The authors define the *capture* of a trajectory to be the full trading revenue upon completion of all trades. At each instant t_k, the transaction revenue corresponds to the dot product $n_k^\intercal \tilde{S}_k$. Therefore, the full trading revenue is $\sum_{k=1}^{N} n_k^\intercal \tilde{S}_k$. From (3.1) and (3.4), we can find the

expression for the capture of a trajectory as

$$\begin{aligned}
\sum_{k=1}^{N} n_k^\intercal \tilde{S}_k &= \sum_{k=1}^{N} n_k^\intercal \left(S_{k-1} - h\Big(\frac{n_k}{\tau}\Big)\right) \\
&= \sum_{k=1}^{N} n_k^\intercal S_{k-1} - \sum_{k=1}^{N} n_k^\intercal \, h\Big(\frac{n_k}{\tau}\Big) \\
&= \sum_{k=1}^{N} \left(x_{k-1}^\intercal - x_k^\intercal\right) S_{k-1} - \sum_{k=1}^{N} n_k^\intercal \, h\Big(\frac{n_k}{\tau}\Big) \\
&= \sum_{k=1}^{N} x_{k-1}^\intercal S_{k-1} - \sum_{k=1}^{N} x_k^\intercal \left(S_k - \sigma\sqrt{\tau}\xi_k + \tau \, g\Big(\frac{n_k}{\tau}\Big)\right) - \sum_{k=1}^{N} n_k^\intercal \, h\Big(\frac{n_k}{\tau}\Big) \\
&= X^\intercal S_0 + \sum_{k=1}^{N} x_k^\intercal \left(\sigma\sqrt{\tau}\xi_k - \tau \, g\Big(\frac{n_k}{\tau}\Big)\right) - \sum_{k=1}^{N} n_k^\intercal \, h\Big(\frac{n_k}{\tau}\Big),
\end{aligned}$$

where we rely on the fact that AC is a complete liquidation model (*i.e.*, $x_N = 0$)[1]. The market value of the initial position is $X^\intercal S_0$ and the loss in value (*i.e.*, the total cost of trading) incurred by a liquidation profile $x = (x_1, \ldots, x_N)$ is the difference $X^\intercal S_0 - \sum_{k=1}^{N} n_k^\intercal \tilde{S}_k$. The term $\sum x_k^\intercal \sigma\sqrt{\tau}\xi_k$ represents the gain (or loss) arising from price volatility, while the terms $\sum x_k^\intercal \, \tau \, g(n_k/\tau)$ and $\sum n_k^\intercal \, h(n_k/\tau)$ represent losses due to permanent and temporary price impacts, respectively. Given that the total cost of trading, given by $C^x = X^\intercal S_0 - \sum_{k=1}^{N} n_k^\intercal \tilde{S}_k$, is a random variable[2], where x indicates the trading trajectory which represents the control parameters of the optimization problem at hand, we can write the expressions for its expected value $\mathbb{E}[C^x]$ and variance $\mathbb{V}[C^x]$ as

$$\mathbb{E}[C^x] = \sum_{k=1}^{N} x_k^\intercal \, \tau \, g\Big(\frac{n_k}{\tau}\Big) + \sum_{k=1}^{N} n_k^\intercal \, h\Big(\frac{n_k}{\tau}\Big) \tag{3.5}$$

and

$$\mathbb{V}[C^x] = \sum_{k=1}^{N} \tau x_k^\intercal \sigma\sigma^\intercal x_k = \sum_{k=1}^{N} \tau x_k^\intercal C x_k. \tag{3.6}$$

Given this modeling framework, the authors frame the optimal liquidation problem in terms of finding trading profiles that minimize $U(x) = \mathbb{E}[C^x] + \lambda\mathbb{V}[C^x]$ for some risk aversion coefficient $\lambda \geq 0$.

Replacing into equation (3.5) the linear price impact functions defined by (3.2) and (3.3), the expected value $\mathbb{E}[C^x]$ is

$$\begin{aligned}
\mathbb{E}[C^x] &= \sum_{k=1}^{N} n_k^\intercal \, \epsilon \odot \operatorname{sgn}\Big(\frac{n_k}{\tau}\Big) + \sum_{k=1}^{N} \tau x_k^\intercal \Gamma v_k + \sum_{k=1}^{N} n_k^\intercal H v_k \\
&= \sum_{k=1}^{N} \left|n_k^\intercal\right| \epsilon + \sum_{k=1}^{N} \tau x_k^\intercal \left(\Gamma^S + \Gamma^A\right) v_k + \sum_{k=1}^{N} \tau v_k^\intercal \left(H^S + H^A\right) v_k \\
&= \epsilon^\intercal \left|X\right| + \sum_{k=1}^{N} \tau x_k^\intercal \Gamma^S v_k + \sum_{k=1}^{N} \tau x_k^\intercal \Gamma^A v_k + \sum_{k=1}^{N} \tau v_k^\intercal H^S v_k + \sum_{k=1}^{N} \tau v_k^\intercal H^A v_k,
\end{aligned}$$

1 This is generally not the case for other liquidation models, which admit remaining inventories at $t = T$.

2 Also referred as *implementation shortfall* by some authors.

where the superscripts S and A denote, respectively, the symmetric and antisymmetric parts of the corresponding matrix[3]. It is also assumed that all $v_k = n_k/\tau$ have the same sign (*i.e.*, we have a pure sell or buy trading profile. Note that the authors recognize that it is possible for some components of the trading velocity to be non-monotonic in time, in particular when there are highly correlated assets in the portfolio being liquidated. In this case, the previous equations are not exactly correct because of the changing sign of the fixed cost (ϵ) terms. In such scenarios, one could simply pragmatically assume this sign changing effect is negligible or, in order to preserve mathematical rigor, assume $\epsilon = 0$.). In any case, the expression above can be simplified by noting that, for any column vector v, we have $v^\intercal H^A v = v^\intercal (H^A)^\intercal v = -v^\intercal H^A v = 0$, which implies the last summation term above is null. Further, we can rewrite the first summation term above as

$$\begin{aligned}
\sum_{k=1}^{N} \tau x_k^\intercal \Gamma^S v_k &= \sum_{k=1}^{N} x_k^\intercal \Gamma^S n_k = \sum_{k=1}^{N} x_k^\intercal \Gamma^S \left(x_{k-1} - x_k\right) \\
&= \sum_{k=1}^{N} \left[x_k^\intercal \Gamma^S x_{k-1} - x_k^\intercal \Gamma^S x_k \right] \\
&= \frac{1}{2} \sum_{k=1}^{N} \left[x_{k-1}^\intercal \Gamma^S x_{k-1} - x_k^\intercal \Gamma^S x_k - \left(x_k^\intercal \Gamma^S x_k - 2 x_k^\intercal \Gamma^S x_{k-1} + x_{k-1}^\intercal \Gamma^S x_{k-1} \right) \right] \\
&= \frac{1}{2} \sum_{k=1}^{N} \left[x_{k-1}^\intercal \Gamma^S x_{k-1} - x_k^\intercal \Gamma^S x_k - \left(x_k^\intercal - x_{k-1}^\intercal \right) \Gamma^S \left(x_k - x_{k-1} \right) \right] \\
&= \frac{1}{2} \sum_{k=1}^{N} \left[x_{k-1}^\intercal \Gamma^S x_{k-1} - x_k^\intercal \Gamma^S x_k \right] - \frac{1}{2} \sum_{k=1}^{N} n_k^\intercal \Gamma^S n_k \\
&= \frac{1}{2} X^\intercal \Gamma^S X - \frac{1}{2} \sum_{k=1}^{N} n_k^\intercal \Gamma^S n_k,
\end{aligned}$$

in which we rely on the identity $x_k^\intercal \Gamma^S x_{k-1} = x_{k-1}^\intercal \Gamma^S x_k$, since Γ^S is symmetric. Thus, we arrive at the following final expression for $\mathbb{E}[C^x]$

$$\mathbb{E}[C^x] = \epsilon^\intercal |X| + \frac{1}{2} X^\intercal \Gamma^S X + \sum_{k=1}^{N} \tau x_k^\intercal \Gamma^A v_k + \sum_{k=1}^{N} \tau v_k^\intercal \tilde{H} v_k, \tag{3.7}$$

where $\tilde{H} = H^S - \frac{1}{2}\tau\Gamma^S$ is symmetric. We also assume τ is small enough so as to allow $\tilde{H}$ to be positive-definite and hence invertible.

With $\mathbb{E}[C^x]$ given by (3.7) and $\mathbb{V}[C^x]$ given by (3.6), it is clear that $U(x)$ is a quadratic function of the control parameters[4] $x_1, \ldots, x_{N-1}$ and it is strictly convex for all $\lambda \geq 0$. Therefore, it is possible to determine its unique global minimum by setting all its partial derivatives to zero (*i.e.*, $\partial U / \partial x_{jk} = 0$ for $j = 1, \ldots, m$ and $k = 1, \ldots, N-1$). The (vectorial)

3 Note that any square matrix M can be expressed as a sum of a symmetric and an antisymmetric matrix such that $M = M^S + M^A$, with $M^S = \frac{1}{2}(M + M^\intercal)$ and $M^A = \frac{1}{2}(M - M^\intercal)$. Additionally, recall that, if S and A are, respectively, symmetric and antisymmetric, then $S^\intercal = S$ and $A^\intercal = -A$.

4 Note that we can always replace v_k in (3.7) with $(x_{k-1} - x_k)/\tau$.

expression for such partial derivatives can be written as

$$\frac{\partial U}{\partial x_k} = \frac{\partial}{\partial x_k}\sum_{i=1}^{N}\tau x_i^\intercal \Gamma^A v_i + \frac{\partial}{\partial x_k}\sum_{i=1}^{N}\tau v_i^\intercal \tilde{H} v_i + \lambda\frac{\partial}{\partial x_k}\sum_{i=1}^{N}\tau x_i^\intercal C x_i. \tag{3.8}$$

The first term of this expression can be simplified as

$$\begin{aligned}
&\frac{\partial}{\partial x_k}\sum_{i=1}^{N}\tau x_i^\intercal \Gamma^A v_i = \frac{\partial}{\partial x_k}\sum_{i=1}^{N} x_i^\intercal \Gamma^A \left(x_{i-1}-x_i\right) = \frac{\partial}{\partial x_k}\sum_{i=k}^{k+1} x_i^\intercal \Gamma^A \left(x_{i-1}-x_i\right)\\
&= \frac{\partial}{\partial x_k}\left[x_k^\intercal \Gamma^A \left(x_{k-1}-x_k\right)\right] + \frac{\partial}{\partial x_k}\left[x_{k+1}^\intercal \Gamma^A \left(x_k - x_{k+1}\right)\right]\\
&= x_k^\intercal \Gamma^A \frac{\partial}{\partial x_k}[x_{k-1}-x_k] + (x_{k-1}-x_k)^\intercal (\Gamma^A)^\intercal \frac{\partial x_k}{\partial x_k} + x_{k+1}^\intercal \Gamma^A \frac{\partial}{\partial x_k}[x_k - x_{k+1}]\\
&= -x_k^\intercal \Gamma^A - (x_{k-1}-x_k)^\intercal \Gamma^A + x_{k+1}^\intercal \Gamma^A\\
&= \left(-x_{k-1}^\intercal + x_{k+1}^\intercal\right)\Gamma^A,
\end{aligned}$$

the second one can be written as

$$\begin{aligned}
&\frac{\partial}{\partial x_k}\sum_{i=1}^{N}\tau v_i^\intercal \tilde{H} v_i = \frac{\partial}{\partial x_k}\sum_{i=1}^{N}\frac{1}{\tau}(x_{i-1}-x_i)^\intercal \tilde{H}\left(x_{i-1}-x_i\right)\\
&= \frac{\partial}{\partial x_k}\sum_{i=k}^{k+1}\frac{1}{\tau}(x_{i-1}-x_i)^\intercal \tilde{H}\left(x_{i-1}-x_i\right)\\
&= \frac{\partial}{\partial x_k}\left[\frac{1}{\tau}(x_{k-1}-x_k)^\intercal \tilde{H}\left(x_{k-1}-x_k\right)\right]\\
&\quad + \frac{\partial}{\partial x_k}\left[\frac{1}{\tau}(x_k-x_{k+1})^\intercal \tilde{H}\left(x_k-x_{k+1}\right)\right]\\
&= \frac{1}{\tau}(x_{k-1}-x_k)^\intercal \tilde{H}\frac{\partial}{\partial x_k}\left[x_{k-1}-x_k\right] + \frac{1}{\tau}(x_{k-1}-x_k)^\intercal \tilde{H}^\intercal \frac{\partial}{\partial x_k}\left[x_{k-1}-x_k\right]\\
&\quad + \frac{1}{\tau}(x_k-x_{k+1})^\intercal \tilde{H}\frac{\partial}{\partial x_k}\left[x_k-x_{k+1}\right] + \frac{1}{\tau}(x_k-x_{k+1})^\intercal \tilde{H}^\intercal \frac{\partial}{\partial x_k}\left[x_k-x_{k+1}\right]\\
&= \frac{1}{\tau}\left[-(x_{k-1}-x_k)^\intercal - (x_{k-1}-x_k)^\intercal + (x_k-x_{k+1})^\intercal + (x_k-x_{k+1})^\intercal\right]\tilde{H}\\
&= \frac{1}{\tau}\left[-2(x_{k-1}-x_k)^\intercal + 2(x_k-x_{k+1})^\intercal\right]\tilde{H}\\
&= -\frac{2}{\tau}(x_{k-1}-2x_k+x_{k+1})^\intercal \tilde{H}
\end{aligned}$$

and the last one is equal to

$$\begin{aligned}
\lambda\frac{\partial}{\partial x_k}\sum_{i=1}^{N}\tau x_i^\intercal C x_i &= \lambda\frac{\partial}{\partial x_k}\sum_{i=k}^{k}\tau x_i^\intercal C x_i = \lambda\frac{\partial}{\partial x_k}\left[\tau x_k^\intercal C x_k\right]\\
&= \lambda\tau x_k^\intercal\left(C + C^\intercal\right) = 2\lambda\tau x_k^\intercal C,
\end{aligned}$$

where we apply the matrix calculus properties (in the numerator layout convention)

$$\frac{\partial \boldsymbol{x}^\intercal \boldsymbol{A}\boldsymbol{x}}{\partial \boldsymbol{x}} = \boldsymbol{x}^\intercal\left(\boldsymbol{A} + \boldsymbol{A}^\intercal\right) \quad \text{and} \quad \frac{\partial \boldsymbol{u}^\intercal \boldsymbol{A}\boldsymbol{v}}{\partial \boldsymbol{x}} = \boldsymbol{u}^\intercal \boldsymbol{A}\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}} + \boldsymbol{v}^\intercal \boldsymbol{A}^\intercal \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{x}}.$$

Equating (3.8) to zero, we get the resulting relations

$$\begin{aligned}
&\left(-x_{k-1}^{\intercal} + x_{k+1}^{\intercal}\right)\Gamma^A - \frac{2}{\tau}(x_{k-1} - 2x_k + x_{k+1})^{\intercal}\tilde{H} + 2\lambda\tau x_k^{\intercal} C = 0 \\
&\quad\Rightarrow \Gamma^A\left(x_{k-1} - x_{k+1}\right) - \frac{2}{\tau}\tilde{H}\left(x_{k-1} - 2x_k + x_{k+1}\right) + 2\lambda\tau C x_k = 0 \\
&\quad\Rightarrow \frac{x_{k-1} - 2x_k + x_{k+1}}{\tau^2} = \lambda\tilde{H}^{-1}C x_k + \tilde{H}^{-1}\Gamma^A\frac{x_{k-1} - x_{k+1}}{2\tau}.
\end{aligned}$$

As suggested by the authors, it is convenient to define a transformation of the solution variable

$$y_k = \tilde{H}^{\frac{1}{2}} x_k,$$

considering $\tilde{H}^{-1}C$ is not necessarily symmetric and $\tilde{H}^{-1}\Gamma^A$ is not necessarily antisymmetric. In this case, the resulting equations are given by

$$\frac{y_{k-1} - 2y_k + y_{k+1}}{\tau^2} = \lambda A\, y_k + B\,\frac{y_{k-1} - y_{k+1}}{2\tau},$$

in which $A = \tilde{H}^{-\frac{1}{2}} C \tilde{H}^{-\frac{1}{2}}$ and $B = \tilde{H}^{-\frac{1}{2}}\Gamma^A\tilde{H}^{-\frac{1}{2}}$ are symmetric positive-definite and antisymmetric, respectively. This is a linear system of equations in $(N-1)m$ variables which can be numerically solved.

Unlike its uni-dimensional version, the multidimensional formulation of the AC model does not admit an explicit solution in the general case. Nevertheless, if it is assumed that trading in each stock does not affect prices of other stocks, then it is possible to obtain an analytical solution for the problem. Mathematically, this assumption corresponds to the matrices Γ and H being diagonal, with $\Gamma_{jj} = \gamma_j$ and $H_{jj} = \eta_j$, where it is required that each $\gamma_j > 0$ and $\eta_j > 0$. Even though this diagonal model is not able to capture cross-asset price impact effects, the covariances still couple the whole system. Additionally, $\tilde{H}$ is also diagonal with

$$\tilde{H}_{jj} = \eta_j\left(1 - \frac{\gamma_j\tau}{2\eta_j}\right),$$

where it is required that $\tau < \min_j(2\eta_j/\gamma_j)$ so all these diagonal elements are positive.

Since Γ is now symmetric, we have $\Gamma^A = 0$ and hence $B = 0$. Thus, y_k must satisfy the second-order linear difference equations

$$y_{k+1} = \left(2I + \tau^2\lambda A\right) y_k - y_{k-1}. \tag{3.9}$$

As λA is symmetric and positive definite, it admits the spectral decomposition $\lambda A = U\Lambda U^{\intercal}$, where U is an orthogonal matrix (*i.e.*, such that $U^{\intercal} = U^{-1}$) whose columns are orthonormal eigenvectors of λA and Λ is a diagonal matrix whose main diagonal is composed of the corresponding eigenvalues of λA, denoted as $\tilde{\kappa}_1^2, \ldots, \tilde{\kappa}_m^2$, with $\tilde{\kappa}_j^2 > 0$. Defining $z_k = U^{\intercal} y_k$, with $y_k = U z_k$, we can rewrite (3.9) as

$$\begin{aligned}
y_{k+1} &= \left(2\,UU^{\intercal} + \tau^2\,U\Lambda U^{\intercal}\right) y_k - y_{k-1} \\
&\Rightarrow y_{k+1} = U\left(2I + \tau^2\Lambda\right) U^{\intercal} y_k - y_{k-1} \\
&\Rightarrow U z_{k+1} = U\left(2I + \tau^2\Lambda\right) z_k - U z_{k-1} \\
&\Rightarrow z_{k+1} = \left(2I + \tau^2\Lambda\right) z_k - z_{k-1},
\end{aligned}$$

which gives rise to the m second-order homogeneous difference equations

$$z_{j,k+1} = \left(2 + \tau^2 \tilde{\kappa}_j^2\right) z_{j,k} - z_{j,k-1}. \tag{3.10}$$

Taking $\omega_j = 2 + \tau^2 \tilde{\kappa}_j^2$, the characteristic polynomials corresponding to (3.10) are given by

$$r_j^2 - \omega_j r_j + 1 = 0,$$

whose solutions are

$$r_{j\pm} = \frac{\omega_j \pm \sqrt{\omega_j^2 - 4}}{2}.$$

One can show that $r_{j+} = 1/r_{j-}$ with a straightforward algebraic expedient. Notice as well that $\omega_j > 2$, which implies $r_{j+} > 1$. Therefore, we can write $r_{j\pm} = e^{\pm \kappa_j \tau}$ for some $\kappa_j > 0$. In particular, note that $z_{jk} = \alpha_+ r_{j+}^k = \alpha_+ e^{\kappa_j \tau k}$ is a solution to the difference equation, and we can substitute this expression back into (3.10) to find how to write κ_j in terms of $\tilde{\kappa}^2$. Hence, we get

$$\begin{aligned}
e^{\kappa_j \tau} z_{jk} &= \left(2 + \tau^2 \tilde{\kappa}_j^2\right) z_{jk} - e^{-\kappa_j \tau} z_{jk} \\
&\Rightarrow e^{\kappa_j \tau} + e^{-\kappa_j \tau} = 2 + \tau^2 \tilde{\kappa}_j^2 \\
&\Rightarrow 2\cosh(\kappa_j \tau) = 2 + \tau^2 \tilde{\kappa}_j^2 \\
&\Rightarrow \kappa_j = \frac{1}{\tau} \cosh^{-1}\left(\frac{\tau^2 \tilde{\kappa}_j^2}{2} + 1\right).
\end{aligned}$$

Since the roots of the characteristic polynomials are real and distinct, the general solution for the difference equations (3.10) is given by

$$z_{jk} = \alpha_- r_{j-}^k + \alpha_+ r_{j+}^k = \alpha_- e^{-\kappa_j \tau k} + \alpha_+ e^{\kappa_j \tau k} = \alpha_- e^{-\kappa_j t_k} + \alpha_+ e^{\kappa_j t_k},$$

for some yet to be determined constants $\alpha_\pm$. Given that $z_N = 0$, we can write

$$z_{j0} = \alpha_- e^{-\kappa_j 0} + \alpha_+ e^{\kappa_j 0} = \alpha_- + \alpha_+$$

and

$$z_{jN} = \alpha_- e^{-\kappa_j T} + \alpha_+ e^{\kappa_j T} = 0.$$

Solving these equations for α_- and α_+, we obtain

$$\alpha_- = \frac{e^{\kappa_j T}}{e^{\kappa_j T} - e^{-\kappa_j T}} z_{j0} \quad \text{and} \quad \alpha_+ = -\frac{e^{-\kappa_j T}}{e^{\kappa_j T} - e^{-\kappa_j T}} z_{j0}.$$

Therefore, we can write

$$\begin{aligned}
z_{jk} &= \left[\frac{e^{\kappa_j T}}{e^{\kappa_j T} - e^{-\kappa_j T}} e^{-\kappa_j t_k} - \frac{e^{-\kappa_j T}}{e^{\kappa_j T} - e^{-\kappa_j T}} e^{\kappa_j t_k}\right] z_{j0} \\
&= \left[\frac{e^{\kappa_j (T - t_k)}}{e^{\kappa_j T} - e^{-\kappa_j T}} - \frac{e^{-\kappa_j (T - t_k)}}{e^{\kappa_j T} - e^{-\kappa_j T}}\right] z_{j0} \\
&= \frac{\sinh\left(\kappa_j \left(T - t_k\right)\right)}{\sinh\left(\kappa_j T\right)} z_{j0}
\end{aligned}$$

with $z_0 = U^\intercal y_0 = U^\intercal \tilde{H}^{\frac{1}{2}} X$. Finally, we have the solution

$$x_k = \tilde{H}^{-\frac{1}{2}} U z_k.$$

3.2 Bergault-Drissi-Guéant model

In this section, we describe the modeling framework, notation and main results presented in Bergault et al. (2022). For a more thorough discussion of the BDG model, including the original mathematical results, we refer the reader to the paper itself. If deemed necessary, we also refer the reader to Appendix A, where we describe some foundational mathematical results on stochastic optimal control theory and the Hamilton-Jacobi-Bellman (HJB) Partial Differential Equation (PDE) that will be alluded to in the model description.

The BDG model considers a market with $d \in \mathbb{N}$ assets and an agent who wishes to liquidate a portfolio over a period of time $[0, T]$, with $T > 0$. The inventory process is denoted by $(q_t)_{t\in[0,T]} = (q_t^1, \dots, q_t^d)_{t\in[0,T]}^\intercal$ with dynamics

$$dq_t = v_t\, dt, \tag{3.11}$$

for a given $q_0 \in \mathbb{R}^d$, where $(v_t)_{t\in[0,T]} = (v_t^1, \dots, v_t^d)_{t\in[0,T]}^\intercal$ represents the column vector of instant trading rates. The fundamental prices of the d assets are modeled as a d-dimensional OU process $(S_t)_{t\in[0,T]} = (S_t^1, \dots, S_t^d)_{t\in[0,T]}^\intercal$ with dynamics

$$dS_t = R\,(\bar{S} - S_t)\; dt + V\, dW_t, \tag{3.12}$$

for a given $S_0 \in \mathbb{R}^d$, where $R \in \mathbb{R}^{d\times d}$ represents the mean-reversion intensity (speed)[5], $\bar{S} \in \mathbb{R}^d$ represents the mean-reversion level, $V \in \mathbb{R}^{d\times k}$ represents the process diffusion and $(W_t)_{t\in[0,T]} = (W_t^1, \dots, W_t^k)_{t\in[0,T]}^\intercal$ is a k-dimensional standard Brownian motion process for some $k \in \mathbb{N}$. The covariation matrix of the process is given by $\Sigma = VV^\intercal$. To account for the market impact, the authors introduce the market price process $(\tilde{S}_t)_{t\in[0,T]} = (\tilde{S}_t^1, \dots, \tilde{S}_t^d)_{t\in[0,T]}^\intercal$ with dynamics

$$d\tilde{S}_t = dS_t + Kv_t\, dt = dS_t + K\, dq_t \tag{3.13}$$

and $\tilde{S}_0 = S_0$, where $K \in \mathbb{R}^{d\times d}$ is a symmetric matrix that represents the linear permanent impact the agent has on prices. At last, the authors also introduce the (uni-dimensional) trader's cash account process $(\tilde{X}_t)_{t\in[0,T]}$ with dynamics

$$d\tilde{X}_t = -v_t^\intercal \tilde{S}_t\, dt - L(v_t)\, dt, \tag{3.14}$$

for a given $\tilde{X}_0 \in \mathbb{R}$, where $L : \mathbb{R}^d \to \mathbb{R}^+$ is a function representing the temporary market impact of trades. It is assumed that L is a positive-definite quadratic form given by

$$L(v) = v^\intercal \eta v, \tag{3.15}$$

where $\eta \in \mathbb{R}^{d\times d}$ is a positive-definite matrix.

The trader wants to maximize the (exponential) expected utility of his wealth at the end of the trading window. This wealth is given by the

5 The matrix R must also be positive definite for the price processes to be co-integrated, as discussed in Section 4.3.

sum of the amount on the cash account ($\tilde{X}_T$) and the market value of any remaining inventory at time T ($q_T^\intercal \tilde{S}_T$), deducted by a discount term $\tilde{\ell}(q_T)$ that penalizes any non-zero terminal position. The expression for the final wealth is therefore $\tilde{X}_T + q_T^\intercal \tilde{S}_T - \tilde{\ell}(q_T)$ and it also assumed that $\tilde{\ell}$ is a positive-definite quadratic form given by

$$\tilde{\ell}(q) = q^\intercal \tilde{\Gamma} q,$$

where $\tilde{\Gamma} \in \mathbb{R}^{d\times d}$ is a positive-definite matrix. This problem can be mathematically expressed as the dynamic optimization problem

$$\sup_{v\in\mathcal{A}_0} \mathbb{E}\left[-e^{-\gamma(\tilde{X}_T + q_T^\intercal \tilde{S}_T - \tilde{\ell}(q_T))}\right], \tag{3.16}$$

where $\gamma > 0$ is the absolute risk aversion parameter of the trader and $\mathcal{A}_0$ is the set of admissible controls[6] beginning at $t = 0$.

The authors propose an equivalent optimization problem defined with respect to the fundamental price process instead of the market price[7]. At first, note that we can rewrite (3.14) as

$$\tilde{X}_T = \tilde{X}_0 - \int_0^T v_t^\intercal \tilde{S}_t \, dt - \int_0^T L(v_t)\, dt,$$

and, through (3.11), (3.13) and the Itô product rule, we have

$$\begin{aligned} d[q_t^\intercal \tilde{S}_t] &= q_t^\intercal \, d\tilde{S}_t + \left(\tilde{S}_t^\intercal \, dq_t\right)^\intercal + dq_t^\intercal \, d\tilde{S}_t \\ &= q_t^\intercal \left(dS_t + K\, dq_t\right) + v_t^\intercal \tilde{S}_t \, dt + 0, \end{aligned}$$

with

$$q_T^\intercal \tilde{S}_T = q_0^\intercal \tilde{S}_0 + \int_0^T q_t^\intercal \, dS_t + \int_0^T q_t^\intercal K \, dq_t + \int_0^T v_t^\intercal \tilde{S}_t \, dt$$

and

$$\begin{aligned} d\left[q_t^\intercal S_t\right] &= q_t^\intercal \, dS_t + \left(S_t^\intercal \, dq_t\right)^\intercal + dq_t^\intercal \, dS_t \\ &= q_t^\intercal \, dS_t + \left(S_t^\intercal v_t \, dt\right)^\intercal + 0, \end{aligned}$$

with

$$q_T^\intercal S_T = q_0^\intercal S_0 + \int_0^T q_t^\intercal \, dS_t + \int_0^T v_t^\intercal S_t \, dt.$$

Now let $(X_t)_{t\in[0,T]}$ be given as

$$dX_t = -v_t^\intercal S_t \, dt - L(v_t)\, dt = -\left(v_t^\intercal S_t + L(v_t)\right) \, dt, \tag{3.17}$$

6 For all $t \in [0,T]$, the admissible control set $\mathcal{A}_t$ is composed of all $\mathbb{R}^d$-valued and adapted processes $(\xi_s)_{s\in[t,T]}$ that satisfy a linear growth condition with respect to the process $(S_s)_{s\in[t,T]}$. Refer to the original paper for a precise definition of the admissible controls sets.

7 The rationale for the selection of this optimization problem is not explicitly stated by the authors in Bergault et al. (2022). A possible motivation for such choice is the fact that, unlike $\tilde{S}_t$, S_t follows a more manageable OU process.

with $X_0 = \tilde{X}_0$. Considering the identities above, we can rewrite the wealth expression as

$$\begin{aligned}
q_T^\intercal \tilde{S}_T &= q_T^\intercal S_T + \int_0^T q_t^\intercal K\, dq_t + \int_0^T v_t^\intercal \tilde{S}_t\, dt - \int_0^T v_t^\intercal S_t\, dt \\
&\Rightarrow \tilde{X}_T + q_T^\intercal \tilde{S}_T = X_0 + q_T^\intercal S_T + \int_0^T q_t^\intercal K\, dq_t - \int_0^T v_t^\intercal S_t\, dt - \int_0^T L(v_t)\, dt \\
&\Rightarrow \tilde{X}_T + q_T^\intercal \tilde{S}_T = X_T + q_T^\intercal S_T + \int_0^T q_t^\intercal K\, dq_t \\
&\Rightarrow \tilde{X}_T + q_T^\intercal \tilde{S}_T - \tilde{\ell}(q_T) = X_T + q_T^\intercal S_T - \tilde{\ell}(q_T) + \frac{1}{2} q_T^\intercal K q_T - \frac{1}{2} q_0^\intercal K q_0,
\end{aligned}$$

We can also define the penalty function $\ell : \mathbb{R}^d \to \mathbb{R}$ as

$$\ell(q) = \tilde{\ell}(q) - \frac{1}{2} q^\intercal K q = q^\intercal \tilde{\Gamma} q - \frac{1}{2} q^\intercal K q = q^\intercal \Gamma q, \tag{3.18}$$

with $\Gamma = \tilde{\Gamma} - \frac{1}{2}K$, where it is assumed that ℓ is a positive semi-definite form, that is, $\Gamma \in \mathbb{R}^{d\times d}$ is a positive semi-definite matrix.

Then, it can be shown that the problem given by (3.16) is equivalent to the problem

$$\sup_{v\in\mathcal{A}_0} \mathbb{E}\left[-e^{-\gamma\left(X_T + q_T^\intercal S_T - \ell(q_T)\right)}\right], \tag{3.19}$$

which can be tackled with the tools of stochastic optimal control. Let $w : [0,T] \times \mathbb{R} \times \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be the optimal value function associated with the problem (3.19):

$$w(t,x,q,S) = \sup_{v\in\mathcal{A}_0} \mathbb{E}\left[-e^{-\gamma\left(X_T + q_T^\intercal S_T - \ell(q_T)\right)} \,\middle|\, X_t = x, q_t = q, S_t = S\right].$$

Note that w depends on three non-temporal variables, namely x, q and S, associated with three different random processes, which imply that the corresponding infinitesimal generator[8] will be given by

$$\mathcal{L} = \mu_x^\intercal \cdot \partial_x + \mu_q^\intercal \cdot \nabla_q + \mu_S^\intercal \cdot \nabla_S + \frac{1}{2}\operatorname{Tr}(\sigma_S \sigma_S^\intercal \cdot D_{SS}^2),$$

where μ_x, μ_q, μ_S and σ_S correspond to the drift and diffusion terms in the dynamics equations (3.17), (3.11) and (3.12). In particular, note that there are no second-order derivatives of x and q in the expression above because these variable's dynamics do not include any Brownian motion term. Note, as well, that only μ_x and μ_q depend on v.

Therefore, the HJB equation associated with (3.19) is given by

$$\begin{aligned}
0 = {} & \partial_t w(t,x,q,S) \\
& + \sup_{v\in\mathbb{R}^d} \left[-\left(v^\intercal S + L(v)\right) \partial_x w(t,x,q,S) + v^\intercal \nabla_q w(t,x,q,S)\right] \\
& + (\bar{S} - S)^\intercal R^\intercal \nabla_S w(t,x,q,S) \\
& + \tfrac{1}{2}\operatorname{Tr}\left[\Sigma D_{SS}^2 w(t,x,q,S)\right],
\end{aligned} \tag{3.20}$$

8 See a brief discussion about infinitesimal generators in Appendix A.

for all $(t, x, q, S) \in [0, T) \times \mathbb{R} \times \mathbb{R}^d \times \mathbb{R}^d$, with terminal condition

$$w(T, x, q, S) = -e^{-\gamma(x+q^\intercal S-\ell(q))}, \tag{3.21}$$

for all $(x, q, S) \in \mathbb{R} \times \mathbb{R}^d \times \mathbb{R}^d$.

The authors propose using the *ansatz*

$$w(t, x, q, S) = -e^{-\gamma(x+q^\intercal S+\theta(t,q,S))}, \tag{3.22}$$

for all $(t, x, q, S) \in [0, T] \times \mathbb{R} \times \mathbb{R}^d \times \mathbb{R}^d$. Differentiating both sides of (3.22) for each variable, we get the following differential relations between w and θ:

$$\begin{aligned}
\partial_t w &= -\gamma w\, \partial_t \theta \\
\partial_x w &= -\gamma w \\
\nabla_q w &= -\gamma w \left(S + \nabla_q \theta\right) \\
\nabla_S w &= -\gamma w \left(q + \nabla_S \theta\right) \\
D^2_{SS} w &= -\gamma w\, D^2_{SS}\theta + \gamma^2 w (q + \nabla_S \theta)^\intercal \left(q + \nabla_S \theta\right).
\end{aligned}$$

Replacing each differential term in w in (3.20) for the corresponding expression in θ shown above and dividing each term of the equation by $-\gamma w$ (which is always positive), we get the following HJB equation for θ:

$$\begin{aligned}
0 = \partial_t \theta(t, q, S) &+ \sup_{v \in \mathbb{R}^d} \left[v^\intercal \nabla_q \theta(t, q, S) - L(v)\right] \\
&- \tfrac{1}{2}\gamma (q + \nabla_S \theta(t, q, S))^\intercal \Sigma \left(q + \nabla_S \theta(t, q, S)\right) \\
&+ (\bar{S} - S)^\intercal R^\intercal \left(q + \nabla_S \theta(t, q, S)\right) \\
&+ \tfrac{1}{2} \operatorname{Tr}\left[\Sigma\, D^2_{SS}\theta(t, q, S)\right],
\end{aligned} \tag{3.23}$$

for all $(t, q, S) \in [0, T) \times \mathbb{R}^d \times \mathbb{R}^d$, with terminal condition

$$\theta(T, q, S) = -\ell(q), \tag{3.24}$$

for all $(q, S) \in \mathbb{R}^d \times \mathbb{R}^d$. If θ is a solution to (3.23), then w, defined as (3.22), is a solution to (3.20).

By replacing $L(v)$ for $v^\intercal \eta v$ in (3.23), it becomes clear that we have the quadratic expression (in v)

$$\phi(v) = v^\intercal \nabla_q \theta(t, q, S) - v^\intercal \eta v$$

in the optimization (supremum) term, which can be solved analytically. Differentiating it with respect to v and equaling the resulting expression to zero, we get

$$\arg\sup_v \phi(v) = \frac{1}{2}\eta^{-1} \nabla_q \theta(t, q, S),$$

with

$$\sup_v \phi(v) = \frac{1}{4} \nabla_q \theta(t, q, S)^\intercal\, \eta^{-1}\, \nabla_q \theta(t, q, S).$$

At last, we get the resulting HJB equation for θ

$$\begin{aligned}0 = \partial_t\theta(t,q,S)&\\ &+ \tfrac{1}{4}\nabla_q\theta(t,q,S)^\intercal\,\eta^{-1}\,\nabla_q\theta(t,q,S)\\ &+ \tfrac{1}{2}\,\mathrm{Tr}\big[\Sigma\,D^2_{SS}\theta(t,q,S)\big]\\ &- \tfrac{1}{2}\gamma(q+\nabla_S\theta(t,q,S))^\intercal\Sigma\,(q+\nabla_S\theta(t,q,S))\\ &+ (\bar{S}-S)^\intercal R^\intercal\,(q+\nabla_S\theta(t,q,S))\,,\end{aligned} \tag{3.25}$$

with terminal condition

$$\theta(T,q,S) = -q^\intercal\Gamma q. \tag{3.26}$$

At this point, the authors propose using a second *ansatz* and looking for a solution θ of the following (quadratic in q and S) form:

$$\begin{aligned}&\theta(t,q,S) =\\ &\quad q^\intercal A(t)\,q + q^T B(t)\,S + S^T C(t)\,S + D(t)^\intercal\,q + E(t)^\intercal\,S + F(t),\end{aligned} \tag{3.27}$$

for all $(t,q,S) \in [0,T]\times\mathbb{R}^d\times\mathbb{R}^d$, where A, B, C, D, E and F are functions which take values from $[0,T]$ and whose co-domains are $\mathbb{R}^{d\times d}$, $\mathbb{R}^{d\times d}$, $\mathbb{R}^{d\times d}$, $\mathbb{R}^d$, $\mathbb{R}^d$ and $\mathbb{R}$, respectively. Additionally, note that A and C only return symmetric matrices. Assuming these functions satisfy the following system of Ordinary Differential Equations (ODEs)

$$\begin{aligned}A'(t) &= \tfrac{1}{2}\gamma\,(B(t)+I_d)\,\Sigma\,(B(t)^\intercal+I_d) - A(t)\,\eta^{-1}A(t)\\ B'(t) &= (B(t)+I_d)\,R + 2\gamma\,(B(t)+I_d)\,\Sigma\,C(t) - A(t)\,\eta^{-1}B(t)\\ C'(t) &= R^\intercal\,C(t) + C(t)\,R + 2\gamma\,C(t)\,\Sigma\,C(t) - \tfrac{1}{4}B(t)^\intercal\,\eta^{-1}B(t)\\ D'(t) &= -\,(B(t)+I_d)\,R\,\bar{S} + \gamma\,(B(t)+I_d)\,\Sigma\,E(t) - A(t)\,\eta^{-1}D(t)\\ E'(t) &= -2\,C(t)\,R\,\bar{S} + R^\intercal E(t) + 2\gamma\,C(t)\,\Sigma\,E(t) - \tfrac{1}{2}B(t)^\intercal\,\eta^{-1}D(t)\\ F'(t) &= -\bar{S}^\intercal R^\intercal E(t) - \mathrm{Tr}[\Sigma\,C(t)] + \tfrac{1}{2}\gamma E(t)^\intercal\,\Sigma\,E(t) - \tfrac{1}{4}D(t)^\intercal\,\eta^{-1}D(t),\end{aligned} \tag{3.28}$$

where I_d corresponds to the identity matrix in $\mathbb{R}^{d\times d}$, with terminal conditions

$$A(T) = -\Gamma, \quad B(T) = C(T) = D(T) = E(T) = F(T) = 0, \tag{3.29}$$

it can be shown that (3.27) is a solution of (3.25) and (3.26).

As a demonstration, suppose A, B, C, D, E and F satisfy (3.28) and (3.29) and θ is given by (3.27). Then, the following equalities hold

$$\begin{aligned}\partial_t\theta &= q^\intercal A'(t)\,q + q^\intercal B'(t)\,S + S^\intercal C'(t)\,S + D'(t)^\intercal\,q + E'(t)^\intercal\,S + F'(t)\\ \nabla_q\theta &= 2\,A(t)\,q + B(t)\,S + D(t)\\ \nabla_S\theta &= B(t)^\intercal\,q + 2\,C(t)\,S + E(t)\\ D^2_{SS}\theta &= 2\,C(t).\end{aligned}$$

If we replace each of these differential equalities in the right-hand side of (3.25), we obtain

$$
\begin{aligned}
&\partial_t\theta(t,q,S)+\tfrac{1}{4}\nabla_q\theta(t,q,S)^\intercal\,\eta^{-1}\,\nabla_q\theta(t,q,S)+\tfrac{1}{2}\,\mathrm{Tr}\big[\Sigma\,D^2_{SS}\theta(t,q,S)\big]\\
&\quad-\tfrac{\gamma}{2}(q+\nabla_S\theta(t,q,S))^\intercal\Sigma\,(q+\nabla_S\theta(t,q,S))+(\bar{S}-S)^\intercal R^\intercal\,(q+\nabla_S\theta(t,q,S))\\
&=q^\intercal A'(t)\,q+q^\intercal B'(t)\,S+S^\intercal C'(t)\,S+D'(t)^\intercal\,q+E'(t)^\intercal\,S+F'(t)\\
&\quad+\tfrac{1}{4}(2A(t)\,q+B(t)\,S+D(t))^\intercal\eta^{-1}\,(2A(t)\,q+B(t)\,S+D(t))\\
&\quad-\tfrac{\gamma}{2}(q+B(t)^\intercal\,q+2C(t)\,S+E(t))^\intercal\Sigma\,(q+B(t)^\intercal\,q+2C(t)\,S+E(t))\\
&\quad+(\bar{S}-S)^\intercal R^\intercal\,(q+B(t)^\intercal\,q+2C(t)\,S+E(t))+\mathrm{Tr}[\Sigma\,C(t)]\\
&=q^\intercal A'(t)\,q+q^\intercal B'(t)\,S+S^\intercal C'(t)\,S+D'(t)^\intercal\,q+E'(t)^\intercal\,S+F'(t)\\
&\quad+q^\intercal A(t)\eta^{-1}A(t)\,q+q^\intercal A(t)\eta^{-1}B(t)\,S+\tfrac{1}{4}S^\intercal B(t)^\intercal\eta^{-1}B(t)\,S\\
&\quad+D(t)^\intercal\eta^{-1}A(t)\;q+\tfrac{1}{2}D(t)^\intercal\eta^{-1}B(t)\,S+\tfrac{1}{4}D(t)^\intercal\eta^{-1}D(t)\\
&\quad-\tfrac{\gamma}{2}(q+B(t)^\intercal\,q+2C(t)\,S+E(t))^\intercal\Sigma\,(q+B(t)^\intercal\,q+2C(t)\,S+E(t))\\
&\quad+\mathrm{Tr}[\Sigma\,C(t)]+\bar{S}^\intercal R^\intercal\,(q+B(t)^\intercal\,q+2C(t)\,S+E(t))\\
&\quad-S^\intercal R^\intercal\,(q+B(t)^\intercal\,q+2C(t)\,S+E(t))\\
&=q^\intercal\left[A'(t)+A(t)\eta^{-1}A(t)-\tfrac{\gamma}{2}\,(I+B(t))\,\Sigma(I+B(t))^\intercal\right]q\\
&\quad+q^\intercal\left[B'(t)+A(t)\eta^{-1}B(t)-2\gamma\,(I+B(t))\,\Sigma C(t)-(I+B(t))\,R\right]S\\
&\quad+S^\intercal\left[C'(t)+\tfrac{1}{4}B(t)^\intercal\eta^{-1}B(t)-2\gamma C(t)\Sigma C(t)-R^\intercal C(t)-C(t)R\right]S\\
&\quad+\left[D'(t)+A(t)\eta^{-1}D(t)^\intercal-\gamma\,(I+B(t))\,\Sigma E(t)+(I+B(t))\,R\bar{S}\right]^\intercal q\\
&\quad+\left[E'(t)+\tfrac{1}{2}B(t)^\intercal\eta^{-1}D(t)-2\gamma C(t)\Sigma E(t)+2C(t)R\bar{S}-R^\intercal E(t)\right]^\intercal S\\
&\quad+\left[F'(t)+\tfrac{1}{4}D(t)^\intercal\eta^{-1}D(t)-\tfrac{\gamma}{2}E(t)^\intercal\Sigma E(t)+\bar{S}^\intercal R^\intercal E(t)+\mathrm{Tr}[\Sigma\,C(t)]\right]\\
&=0.
\end{aligned}
$$

where we resorted to the equalities

$$
\begin{aligned}
q^\intercal A(t)\,\eta^{-1}B(t)\,S &= S^\intercal B(t)^\intercal\,\eta^{-1}A(t)\,q\\
q^T A(t)\,\eta^{-1}D(t) &= D(t)^\intercal\,\eta^{-1}A(t)\;q\\
S^\intercal\,B(t)^\intercal\,\eta^{-1}D(t) &= D(t)^\intercal\,\eta^{-1}B(t)\,S\\
q^\intercal B(t)\,\Sigma\,C(t)\,S &= S^\intercal\,C(t)\,\Sigma\,B(t)^\intercal\,q\\
S^\intercal R^\intercal\,C(t)\,S &= S^\intercal\,C(t)\,R\,S\\
q^\intercal B(t)\,\Sigma\,E(t) &= E(t)^\intercal\,\Sigma\,B(t)^\intercal\,q\\
S^\intercal\,C(t)\,\Sigma\,E(t) &= E(t)^\intercal\,\Sigma\,C(t)\,S\\
S^\intercal\,R^\intercal\,E(t) &= E(t)^\intercal\,R\,S,
\end{aligned}
$$

which hold since $A(t)$, $C(t)$, Σ and η are symmetric. Additionally, note that (3.27) also satisfy the terminal condition (3.26), which concludes the proof.

Notice that the first three equations on the ODE system above are independent of the others and can be solved as a first step. Similarly, the equations for D and E are independent of the last one and can be solved for after A, B and C are obtained. As a last step, F can

be found through an integration procedure. We can apply numerical ODE solvers to incrementally obtain the solutions to these three sub-problems and find θ numerically. As a final result, the authors also show that the optimal control process $(v_t)_{t\in[0,T]} \in \mathcal{A}_t$ can be written as

$$v_t = \frac{1}{2}\eta^{-1}\,\nabla_q\theta = \frac{1}{2}\eta^{-1}\left(2\,A(t)\,q_t + B(t)\,S_t + D(t)\right). \tag{3.30}$$

3.3 Model parameters estimation

3.3.1 Temporary price impact coefficient

While prices and spreads can be directly extracted from the LOB snapshots, price impact coefficients must be estimated. In order to compute these coefficients, we assemble the effective price curve (as a function of trade volume) at each side of the LOB for each time snapshot. Then we perform a null-intercept linear regression on the difference between the effective price and the current mid-price and retrieve the regression's slope as the temporary price impact coefficient estimator. Check Figure 2.2 for an illustration of the temporary price impact curves and the respective regressed lines. Note that, in order to mitigate the effect of outlier orders typically found at the extremes of real world LOBs (clearly seen in Figure 2.2, for instance), the regressions only take into account the slices of the effective prices curves corresponding to 75% of the available volume at each side of the LOB. Figure 3.1 illustrates the obtained price impact coefficients for all LOB snapshots in a trading session.

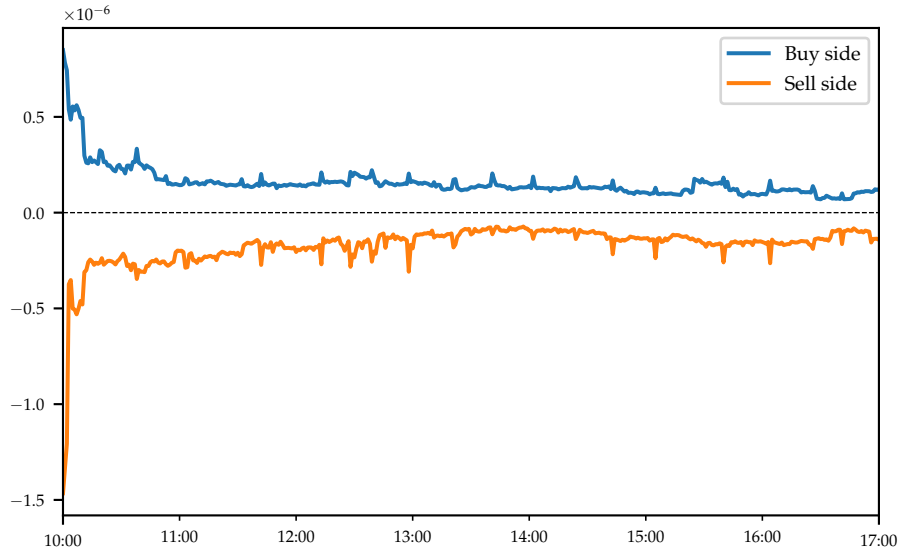


Figure 3.1: PETR4 price impact coefficients (2019–11–27)

Subsequently, the obtained buy and sell side price impact coefficients are averaged together to produce a single output value for each one-minute snapshot. Similarly, in order to get a single price impact coefficient for a trading period (*e.g.*, a day), the one-minute snapshots coefficients are averaged together. In order to avoid low liquidity effects

of opening and closing hours on the LOBs, the first and last 30 minutes of the daily trading window are discarded when computing the final price impact coefficient mean.

3.3.2 Ornstein-Uhlenbeck parameters

In order to estimate the OU model parameters, the only required data are the mid-price series, which can be directly obtained from the LOB snapshots. Also notice that the time discretization of an OU process is equivalent to a vector auto-regressive (VAR) process of order 1. Thus, it is possible to fit an OU model by means of a VAR(1) model parameter estimation.

To see why this is the case, recall that the multidimensional OU process dynamics is given by

$$dS_t = R\left(\bar{S} - S_t\right)\, dt + V\, dW_t, \tag{3.31}$$

where we use the same notation presented in Section 3.2. Through the Euler-Maryuama discretization, we can write a discrete version of the equation above as

$$\begin{aligned} S_{k+1} - S_k &= R\bar{S}\Delta t - RS_k\Delta t + V\left(W_{k+1} - W_k\right) \\ S_{k+1} &= R\bar{S}\Delta t + \left(I - R\Delta t\right)S_k + V\sqrt{\Delta t}\,\epsilon_{k+1}, \end{aligned}$$

where $\epsilon_k \sim N(0, I_d)$ are d-dimensional Gaussian noise vectors. Note, as well, that the process $(S_k)_{k\in\mathbb{N}}$ admits the following VAR(1) representation

$$S_{k+1} = a + BS_k + C\varepsilon_{k+1}, \tag{3.32}$$

where $a \in \mathbb{R}^d$, $B \in \mathbb{R}^{d\times d}$ and $C \in \mathbb{R}^{d\times d}$. Matching each term in both equations, one can find the following expressions for the OU process parameters:

$$\begin{aligned} R &= \frac{1}{\Delta t}\left(I - B\right) \\ \bar{S} &= \frac{1}{\Delta t}R^{-1}a \\ V &= \frac{1}{\sqrt{\Delta t}}C. \end{aligned}$$

Therefore, one can estimate a VAR(1) model to the mid-price series and obtain the OU parameters using the relations above.

Alternatively, instead of obtaining a discretization for the OU process from its dynamics equation (3.31), we can obtain a discretization directly from its analytical solution. Recall that the analytical expression for the OU process defined by (3.31) is given by

$$S_{t+\tau} = \left(I - e^{-R\tau}\right)\bar{S} + e^{-R\tau}S_t + \int_t^{t+\tau} e^{-R(t+\tau-s)}V\, dW_s, \tag{3.33}$$

where the last term (the Itô integral) is a multi-variate normally distributed random variable with zero mean and covariance matrix $\Sigma_\tau \in \mathbb{R}^{d\times d}$ given by

$$\Sigma_\tau = \int_t^{t+\tau} e^{-R(t+\tau-s)}VV^\intercal e^{-R^\intercal(t+\tau-s)}\, ds \approx \tau e^{-R\tau}VV^\intercal e^{-R^\intercal\tau},$$

where the approximation holds for $\tau \to 0$ (which makes $t - s \approx 0$). According to Meucci (2009) and Vatiwutipong and Phewchean (2019), we can apply the identities $\text{vec}(ABC) = (C^\intercal \otimes A)\,\text{vec}(B)$ and $e^{A \oplus B} = e^A \otimes e^B$, expressed in terms of the vector operator[9] as well as the Kronecker product ($\otimes$) and sum ($\oplus$), to obtain

$$\begin{aligned}
\text{vec}(\Sigma_\tau) &= \int_t^{t+\tau} e^{-R(t+\tau-s)} \otimes e^{-R(t+\tau-s)}\, ds\ \text{vec}(VV^\intercal) \\
&= \int_t^{t+\tau} e^{-(R\oplus R)(t+\tau-s)}\, ds\ \text{vec}(VV^\intercal) \\
&= (R \oplus R)^{-1}(I - e^{-(R\oplus R)\tau})\,\text{vec}(VV^\intercal).
\end{aligned}$$

Taking $\tau = \Delta t$, we can write a discrete version of (3.33) as

$$S_{k+1} = \left(I - e^{-R\,\Delta t}\right)\bar{S} + e^{-R\,\Delta t} S_k + \sqrt{\Sigma_{\Delta t}}\,\epsilon_{k+1},$$

where $\epsilon_k \sim N(0, I_d)$. Matching each term of the equation above with (3.32), one can find the following expressions for the OU parameters:

$$\begin{aligned}
R &= -\frac{\log B}{\Delta t} \\
\bar{S} &= \left(I - e^{-R\,\Delta t}\right)^{-1} a \\
V &= \left\{ \text{vec}^{-1}\left[\left(I - e^{-(R\oplus R)\Delta t}\right)^{-1} (R \oplus R)\ \text{vec}(\Sigma_{\Delta t}) \right] \right\}^{\frac{1}{2}}.
\end{aligned}$$

It must be noted as a final remark that the mean-reverting speed parameter R is notoriously difficult to estimate accurately, even with numerous observations (see Pei, 2021).

9 The vector operator (vec) stacks the columns of a matrix into a vector. The inverse operator (vec^{-1}) reshapes a vector back into a square matrix.

4 FINANCIAL SERIES CO-INTEGRATION

In financial markets, it is often empirically observed that some groups of assets evolve in a highly interrelated manner. Such coupling may take one of different forms, requiring diverse and appropriate mathematical tools in the modeling endeavor. In the case of random variables, one of the simpler and most commonly used form of statistical relationship modeling is that of correlation. Generally speaking, correlation can be understood as a measure of the linear dependency[1] between two random variables (*e.g.*, stock returns) and is ubiquitously applied in the realm of quantitative finance. In the case of time series analysis, particularly when resorting to parametric approaches, another fruitful form of relationship modeling is that of co-integration, which is able to capture the dynamics of a stronger type of coupling between financial series (*e.g.*, stock prices). The concept of co-integration has been put to use as the foundation for mean reverting (pair) trading strategies and, as exemplified in the present work, portfolio liquidation programs, among other possible applications.

Both models presented in Chapter 3 use correlation structures to represent the coupling between the price processes' innovations, which is a widespread practice in quantitative finance. On the other hand, the application of co-integration dependency structures, as in the case of the BDG model, is much less common, particularly in the realm of optimal liquidation. In the present chapter, we present the concept of co-integration, along with the statistical tests that can be used to identify its presence on a time series. We also briefly discuss co-integration in the context of continuous processes and its connection to the OU process, as applied in the BDG model.

4.1 Co-integration concepts and definition

Before formalizing the definition of co-integration in the context of time series analysis, it is necessary to introduce the two basilar concepts of stationarity and integration. A time series $(X_n)_{n=1,\dots,N}$ is said to be *strictly stationary* if its unconditional joint probability distribution does not change with respect to time. More precisely, if, for every $m, n \in \mathbb{N}$, with $n + m \leq N$, the distributions of $(X_1, \dots, X_n)$ and $(X_{1+m}, \dots, X_{n+m})$ are the same, then $(X_n)_{n=1,\dots,N}$ is said to be a stationary time series. This strong sense of stationarity implies, among other outcomes, that

1 This common linear notion of dependency is typically quantified through Pearson's correlation coefficient. Naturally, there are other non-linear measures of dependency between random variables such as rank-based correlation (*e.g.*, Spearman's and Kendall's correlation coefficients) and even more general and complex copula-based structures.

all the series' statistics (*e.g.*, its distribution moments) are immutable with respect to time shifts. There is also a weaker sense of stationarity which only requires the process' mean, variance, and covariance to be limited and static with respect to time.

Financial series such as price processes are usually not stationary, since price levels, and hence the process mean, clearly change from a moment to the other. On the other hand, closely related processes such as returns can be stationary, specially in the weaker sense mentioned above. Notice, in particular, that the return series can be obtained from the price series through the application of a differencing operator Δ defined by $\Delta X_t = X_t - X_{t-1}$. To account for this phenomenon, it is useful to introduce the concepts of *integration*, which is the inverse operation of differencing, and *order of integration*, which represents the number of times the differencing operator needs to be applied to a non-stationary series in order to obtain a stationary one. If a time series has order of integration equal to d, we say it is integrated of order d or, as a shorter notation, that it is $I(d)$.

Having established the concepts of stationarity and integration, it is now possible to properly define co-integration. A collection $(X_{1,n})_n, \ldots, (X_{m,n})_n$, for $n = 1, \ldots, N$, of m $I(d)$ time series is said to be *co-integrated* if there is a linear combination determined by scalar constants $k_1, \ldots, k_m$, with $k_i \in \mathbb{R}$, not all null, such that the series $(Y_n)_{n=1,\ldots,N}$ given by $Y_n = k_1 X_{1,n} + \cdots + k_m X_{m,n}$, has an order of integration smaller than d. When disposed as a column vector, the collection of linear combination coefficients is referred as a *co-integrating vector*. An equivalent and more intuitive way to describe the co-integration property is to realize that a collection of time series is co-integrated if and only if the series share a common non-stationary stochastic trend component, the remainder terms being stationary.

A customary application of co-integration in finance is the identification of assets, typically pairs of stocks, that exhibit some sort of joint mean reversion dynamics. Once the co-integrated property is identified, it is possible for an arbitrageur to exploit eventual price unbalances for financial gain, based on the assumption that the co-integrated process will, in due course, revert to its mean. Usually, in this type of application, the price (or log-price) processes are $I(1)$ and the return (or log-return) processes are stationary ($I(0)$).

4.2 Co-integration estimation and tests

Two fundamental requirements when applying co-integration techniques in practice are (i) to estimate the linear combination coefficients k_i and (ii) to assert the confidence levels of both the co-integration property per se and such coefficient estimations.

A simple method to estimate k_i can be obtained by performing linear regressions. To see why this is the case, we can consider a bi-variate $I(d)$ example. Assume $(X_{1,n})_{n=1,\ldots,N}$ and $(X_{2,n})_{n=1,\ldots,N}$ are two distinct co-integrated $I(1)$ time series and $k_1, k_2 \in \mathbb{R}$ are such that $Y_n = k_1 X_{1,n} + k_2 X_{2,n}$ is a stationary series. Assume as well that both

series share a common $I(1)$ trend component $(T_n)_{n=1,\dots,N}$. Then we can write

$$X_{1,n} = c_1 T_n + \epsilon_{1,n}$$
$$X_{2,n} = c_2 T_n + \epsilon_{2,n},$$

where $c_i \in \mathbb{R}$, $c_i \neq 0$, and $(\epsilon_{i,n})_{n=1,\dots,N}$ are noise processes, for $i = 1, 2$. We can assume, without loss of generality, that $k_1 = 1$ and $k_2 = -c_1/c_2$, which gives us

$$Y_n = X_{1,n} - \frac{c_1}{c_2} X_{2,n} = c_1 T_n + \epsilon_{1,n} - \frac{c_1}{c_2} c_2 T_n - \frac{c_1}{c_2} \epsilon_{2,n} = \epsilon_{1,n} - \frac{c_1}{c_2} \epsilon_{2,n},$$

which satisfy the stationarity requirement. Therefore, we can write

$$X_{1,n} = \alpha + \beta X_{2,n} + \epsilon_n,$$

for $\alpha, \beta \in \mathbb{R}$ and $(\epsilon_n)_{n=1,\dots,N}$ a mean-zero stationary noise process. The equation above corresponds to the typical formulation of linear regressions, with $\alpha = \mathbb{E}[\epsilon_{1,n} - {}^{c_1}/_{c_2}\, \epsilon_{2,n}]$ and $\beta = {}^{c_1}/_{c_2}$. Therefore, $[1, -\beta]^\intercal$ should be a co-integration vector.

In relation to testing the co-integration property, it must be noted that this property is strongly linked with auto-regressive (AR) models and unit root tests. Recall that, when analyzing AR processes for stationarity or non-stationarity, one can check the roots of the respective characteristic polynomial. The necessary and sufficient condition for non-stationarity is that all roots of this polynomial have absolute values greater than one.

Therefore, a possible way to test for co-integration between two or more series, at least as a first approximation, is to perform the aforementioned linear regression and check, using unit root tests, for the stationarity of the series obtained through the linear combination of the original series designated by the prospect co-integration vector. This method is essentially what the Engle and Granger (1987) co-integration test performs, using a variation of the Dickey-Fuller unit root test. The test's null hypothesis is that there is no co-integration and the alternative hypothesis is that there is a co-integration relationship. Therefore, if the p-value is small, we can reject the hypothesis of no co-integration. See also MacKinnon (2010) for updated and improved versions of the Engle-Granger (EG) test, including critical values tables.

Even though the EG co-integration test is easy to apply, it suffers from some problems and weaknesses, which led to the development of more general and robust tests, such as the Johansen (1991) test. The Johansen test is not limited to the bi-variate case and can check for numerous co-integration relationships, unlike the EG test. It models the time series as a VAR process[2] and uses vector error correction model (VECM)[3] techniques to test for the number of co-integrating factors. Given a d-variate problem, the method sequentially tests[4] whether the

2 The VAR model can be understood as a multidimensional extension of the AR model.

3 VECMs are used to study short-term deviations in a series from one or more permanent stochastic trends.

4 Johansen proposes two distinct variants for his co-integration test: the trace test and the maximum eigenvalue test, which have different statistics and critical values.

number of co-integrating vectors is equal to k, for $k = 0, \ldots, d$. At each step, the null hypothesis is that the number of co-integrating vectors is equal to k and the alternative hypothesis is that it is greater than k. The first non-rejection of the null hypothesis corresponds to the resulting estimate for the number of co-integration factors.

4.3 Continuous co-integration and the OU process

Even thought co-integration has traditionally been studied in the context of discrete time processes, there have been some efforts to generalize the concept for continuous time processes. One of the earliest papers that explore this subject is Comte (1999), where the author proposes a definition for continuous time co-integration which is akin to the discrete case. In broad terms, according to such definition, non stationary continuous time processes with stationary increments are said to be co-integrated if there is a linear combination of such processes that results in a stationary time process.

At any rate, a more general definition for co-integration in the context of continuous time processes is the one offered in Benth and Süss (2018), which is construed in terms of convergence of probability distributions. Such definition is more useful for our current discussion and is hereby succinctly presented. Let $(S_t)_{t\geq 0} \in \mathbb{R}^d$, for $d \geq 2$, be a stochastic process representing the asset prices in a given market, and let also $\mathbb{P}_X(t, \cdot)$ denote the probability distribution of some stochastic process $(X_t)_{t\geq 0}$. Then, $(S_t)_{t\geq 0}$ is said to be continuously co-integrated if there exist a non-null vector $c \in \mathbb{R}^d$ and a probability distribution μ_c such that $\mathbb{P}_{c^\intercal S}(t, \cdot)$ converges in distribution to μ_c when $t \to \infty$. Such convergence can also be denoted as

$$c^\intercal S \xrightarrow{d} \mu_c.$$

In more practical terms, such definition means that there is a linear combination, indicated by c, of the components of the price process vector $(S_t)_{t\geq 0}$ which admits a limiting probability distribution.

The multidimensional OU process is a natural candidate to use when working with continuous time co-integration. Take, for instance, the OU process whose dynamics is given by (3.12) with a positive definite mean reversion matrix R and suppose $\alpha \in \mathbb{R}^d$ and $\kappa > 0$ are, respectively, a left-eigenvector of R and its corresponding eigenvalue, such that $\alpha^\intercal R = \kappa \alpha^\intercal$. Then, we can write the dynamics for the unidimensional process $\alpha^\intercal S_t$ as

$$\begin{aligned} d\left(\alpha^\intercal S_t\right) &= \alpha^\intercal \, dS_t \\ &= \alpha^\intercal R \left(\bar{S} - S_t\right) \, dt + \alpha^\intercal V \, dW_t \\ &= \kappa \alpha^\intercal \left(\bar{S} - S_t\right) \, dt + \alpha^\intercal V \, dW_t \\ &= \kappa \left(\alpha^\intercal \bar{S} - \alpha^\intercal S_t\right) \, dt + \alpha^\intercal V \, dW_t \\ &= \kappa \left(\tilde{\mu} - \alpha^\intercal S_t\right) \, dt + \tilde{\sigma} \, d\tilde{W}_t, \end{aligned}$$

where $\tilde{\mu} = \alpha^\intercal \bar{S}$, $\tilde{\sigma} \, d\tilde{W}_t = \alpha^\intercal V \, dW_t$ and the process $(\tilde{W}_t)_{t>0}$ is the unidimensional Brownian motion resulting from the linear combination of

$(W_t)_{t>0}$ with factors defined by $\alpha^\intercal V$. Thus, $\tilde{\mu}$ and $\tilde{\sigma}$ are the mean and volatility coefficients for the $\alpha^\intercal S_t$ uni-dimensional OU process, which is known to have a limiting stationary Gaussian distribution as $t \to \infty$. Therefore, since

$$c^\intercal S \xrightarrow{d} N\left(\tilde{\mu}, \frac{\tilde{\sigma}^2}{2\kappa}\right),$$

the multidimensional OU process $(S_t)_{t>0}$ is continuously co-integrated.

5 NUMERICAL RESULTS

In the present chapter, we apply the previously presented optimal execution models to Brazilian stocks. As a test case, we use both common and preferred shares of the oil company Petrobras[1], which, besides being some of the most liquid stocks in the Brazilian market, should also be highly correlated and co-integrated, since they are linked to the same company. We also provide a second, though briefer, discussion of the application of the models to another pair of stocks, namely the common and preferred shares of the Brazilian bank Bradesco[2]. The overall goal in our analysis is to evaluate the impact and potential benefits for a market agent (*e.g.*, a fund manager) of taking into account the correlation and co-integration properties of assets in a portfolio liquidation scenario.

In order to achieve such goal, we simulate the inventory processes obtained via the liquidation strategies for the following models: uni-dimensional AC, multidimensional AC and BDG. The initial position being liquidated should be entirely composed of a single stock, not least to allow for the application of the uni-dimensional AC model, but, more fundamentally, because we wish to evaluate correlation and co-integration effects in a single asset liquidation scenario. Particularly, given a single asset position that needs to be liquidated, we wish to find out if and how an agent could take advantage of other correlated or co-integrated assets to mitigate his loss or even increase his wealth along and after the liquidation process[3]. Recall that the uni-dimensional AC model does not consider any of these two properties, while its multidimensional version only considers correlation. BDG, being the most comprehensive model of this set, takes into account both correlation and co-integration properties, even though it also assumes mean-reversion (OU). The uni-dimensional AC model serves, therefore, as a baseline reference for the multidimensional AC and the BDG models.

The models are fit using with real data, obtained from four consecutive trading sessions, from which we build the corresponding LOBs. We also verify that the mid-prices are co-integrated through statistical tests. We then run some simulations in order to evaluate and compare the final wealth resulting from each model's liquidation strategy. Finally, we study the models' sensitivity to some of their parameters.

1 With tickers PETR3 and PETR4, respectively.
2 With tickers BBDC3 and BBDC4, respectively.
3 We could, alternatively, study a multiple asset liquidation scenario, but we assume the single asset one to be much more common for practitioners and, therefore, more interesting.

5.1 Stock data, co-integration and model fitting

In order to fit the models' parameters, we use order data from November 25th to 28th, 2019. The mid-prices series (with a 60-seconds resolution) are presented in Figure 5.1. Visual inspection of both series suggests co-integration, which is confirmed by Engle-Granger and Johansen tests, as presented in Table 5.1. Note that we show two EG tests, because the results may differ depending on which of the two stocks we choose as the dependent variable. The obtained co-integrating vector is $[1, -1.257]^\intercal$ and the resulting co-integrated process is shown in Figure 5.2.

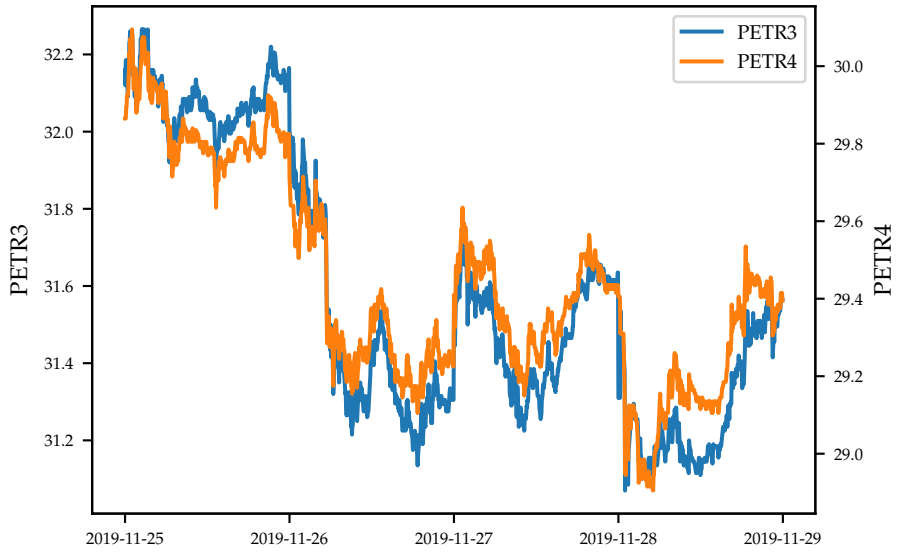


Figure 5.1: PETR3/PETR4 mid-prices (2019–11–25 to 2019–11–28)

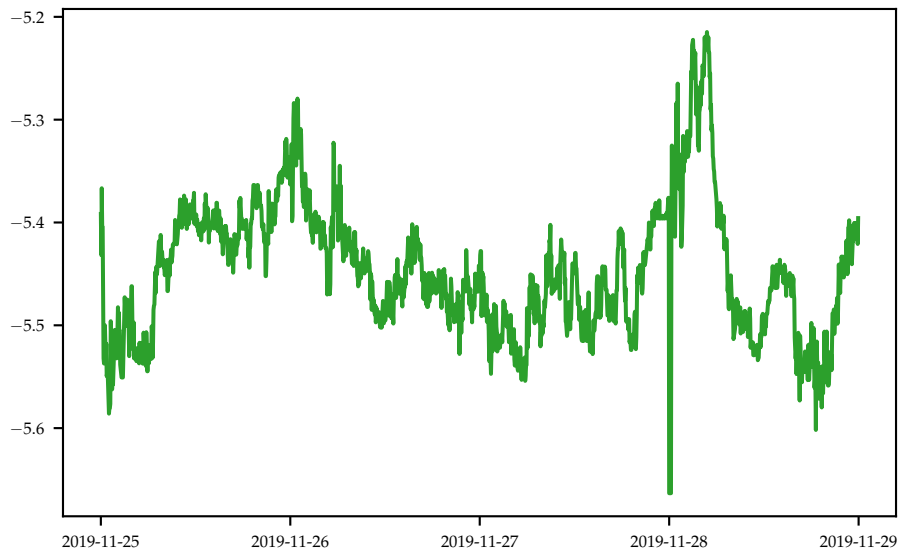


Figure 5.2: Co-integrated process (PETR3 $-\ \beta \times$ PETR4)

We fit the single and multidimensional AC models, as well as the BDG model, to these price series. The fitted parameters for the three models are shown in Tables 5.2, 5.3 and 5.4, respectively. In particular, note the similarity between the volatility parameters of the multidimensional models (σ and V, in Tables 5.3 and 5.4, respectively). The values for V are a little bit smaller than the ones for σ since some of the price process dispersion is captured by the mean reversion term in the BDG model. Additionally, note that the values for ϵ in both AC models were fixed at zero in order to preserve the mathematical rigor in the derivation of (3.5), since we expect non-monotonic trading profiles in the multidimensional case.

Table 5.1: Co-integration tests results

Test	Test Variant	Statistic	Result[a]
EG	PETR3 as dep. variable	-3.44	Reject H_0
EG	PETR4 as dep. variable	-3.56	Reject H_0
Johansen	Trace, $k = 0$	32.41	Reject H_0
Johansen	Trace, $k = 1$	3.72	Fail to reject H_0
Johansen	Max eigenvalue, $k = 0$	28.69	Reject H_0
Johansen	Max eigenvalue, $k = 1$	3.72	Fail to reject H_0

[a] The null hypothesis for the Engle-Granger test is that of no co-integration. The null hypothesis for the Johansen test is that the number of co-integration vectors is equal to k. The results are taken at a 0.05 significance level.

Table 5.2: Uni-dimensional Almgren-Chriss (AC) model parameters

Parameter	Meaning	Value
η	Temporary impact	4.82×10^{-7}
ϵ	Fixed costs	0.0
γ	Permanent impact	0.0
σ	Volatility	0.178
λ	Risk aversion	1.0×10^{-5}

5.2 Liquidation and wealth results

Next, for each model, we compute the corresponding liquidation strategy and the resulting inventory process, considering real prices from the following day (November 29$^{\text{th}}$, 2019) and an initial inventory q_0 of 100 thousand units of PETR3[4], which are to be liquidated during the next trading session (T)[5]. The resulting inventory processes are shown in Figures 5.3 and 5.4, respectively. Note that, for the BDG case, the

4 Which correspond to approximately 2% of the total volume traded on November 28$^{\text{th}}$, 2019.

5 The trading hours on B3 go from 10 to 17, totaling 7 hours.

Table 5.3: Multidimensional Almgren-Chriss (AC) model parameters

Parameter	Meaning	Value
H	Temporary impact	$\begin{bmatrix} 4.82 \times 10^{-7} & 0.0 \\ 0.0 & 1.62 \times 10^{-7} \end{bmatrix}$
ϵ	Fixed costs	$\begin{bmatrix} 0.0 \\ 0.0 \end{bmatrix}$
Γ	Permanent impact	$\begin{bmatrix} 0.0 & 0.0 \\ 0.0 & 0.0 \end{bmatrix}$
σ	Volatility	$\begin{bmatrix} 0.162 & 0.074 \\ 0.074 & 0.147 \end{bmatrix}$
λ	Risk aversion	1.0×10^{-5}

Table 5.4: Bergault-Drissi-Guéant (BDG) model parameters

Parameter	Meaning	Value
R	Mean reversion rate	$\begin{bmatrix} 2.20 & -1.28 \\ -1.17 & 2.69 \end{bmatrix}$
$\bar{S}$	Mean reversion level	$\begin{bmatrix} 31.55 \\ 29.42 \end{bmatrix}$
V	Volatility	$\begin{bmatrix} 0.161 & 0.074 \\ 0.074 & 0.146 \end{bmatrix}$
η	Temporary impact	$\begin{bmatrix} 4.82 \times 10^{-7} & 0.0 \\ 0.0 & 1.62 \times 10^{-7} \end{bmatrix}$
Γ	Inventory penalty	$\begin{bmatrix} 1.0 \times 10^{-4} & 0.0 \\ 0.0 & 1.0 \times 10^{-4} \end{bmatrix}$
γ	Risk aversion	5.0×10^{-3}

mid-price processes for both stocks are also shown with their respective estimated $\bar{S}$ values. In Figure 5.5, it is possible to visually compare the inventory processes for the multidimensional AC and BDG models.

When comparing the inventory processes for both versions of the AC model (Figure 5.3), we can observe that the uni-dimensional one exhibits a higher liquidation speed, particularly at the beginning of the trading window. For instance, by two hours into the trading session, the inventory for PETR3 is approximately 20 thousand units (around one fifth of the initial amount) for the uni-dimensional version, versus approximately 32 thousand (around one third of the initial amount) for the multidimensional one. Even more significant is the fact that, in the multidimensional version, the inventory process assumes a short position for the other stock (PETR4) which reaches a maximum of approximately 30 thousand units early on. Because of the correlation between these two assets, and taking into account the fact that

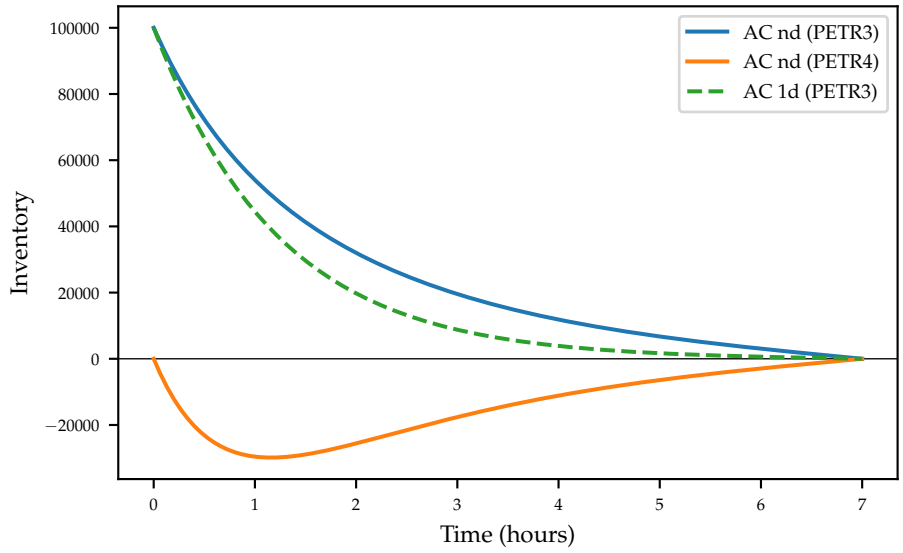


Figure 5.3: Inventory processes — Uni-dimensional and multidimensional AC models

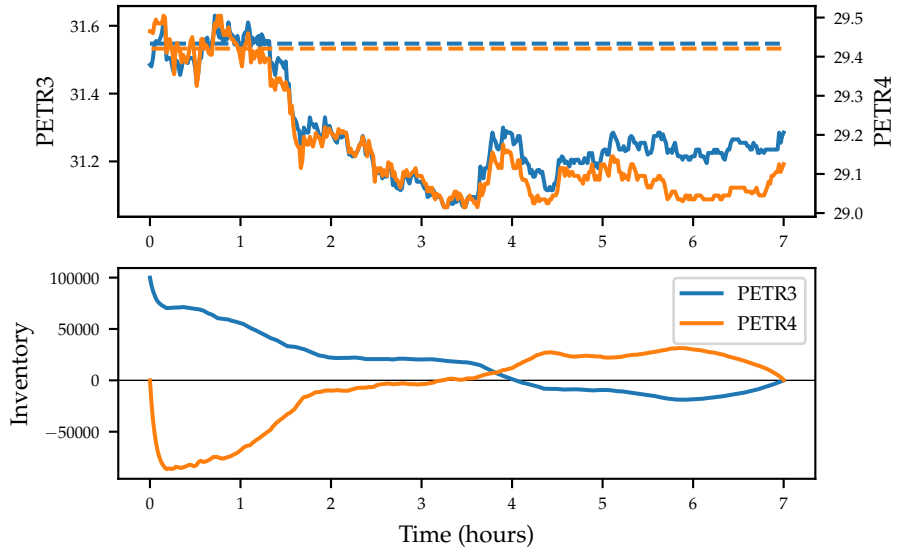


Figure 5.4: Inventory processes — BDG model

PETR4 is more liquid[6], the model is able to "hedge" some volatility, improving on the optimization criteria with respect to the more restrict uni-dimensional version. This is also evidenced in the wealth discussion to follow.

When comparing the inventory processes for the multidimensional AC and BDG models (Figure 5.5), one can notice their remarkably different liquidation profiles. The AC model inventory curves are smooth throughout the whole trading session, with faster trading speeds early on (the first hour), followed by a slower convergence towards zero until the end of the trading session. On the other hand,

6 Note that the value corresponding to PETR4 in the H matrix (see Table 5.3) is smaller than the one corresponding to PETR3.

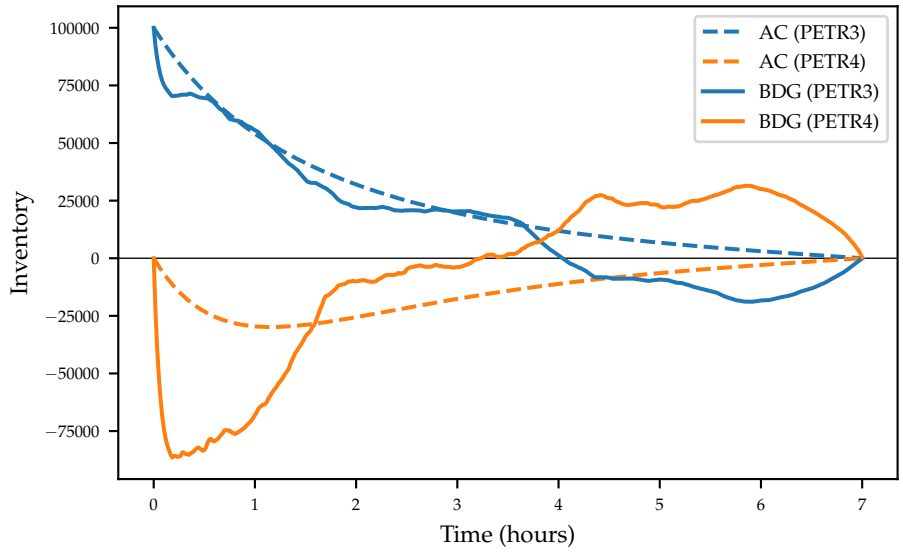


Figure 5.5: Inventory processes — AC and BDG models

the BDG model inventory curves are rougher and fairly irregular as the model responds to the price dynamics of the underlying stocks. Similarly to the AC model, the inventory process for the BDG model also assumes significant short positions for PETR4 early on during the trading day. These short position are considerably larger, though, when compared to the AC model, reaching an astonishing maximum level of approximately 90 thousand units in just some minutes into the trading session. After this initial sell movement in the first minutes, the control process for the PETR4 stock is reversed and a buying signal is maintained for the next four hours, finally reaching a long position of around 30 thousand units. Regarding the main PETR3 stock, the inventory curve begins with a brief fast selling movement, shifts into a trading profile comparable to the AC model, and then it stays relatively flat at around 25 thousand units until the middle of the trading session. The control process then assumes a selling signal for approximately two hours, reaching a short position of about 20 thousand units. In the final hour, both PETR3 and PETR4 positions are definitely unwound. As a last remark, one can argue that, except for the opening and closing moments of the trading session, the trade (buy or sell) signals, along with their relative intensity, are directly related to the price difference between the underlying stocks, as visibly suggested in Figure 5.4. Thus, besides the correlation properties that the multidimensional AC model could exploit, we can verify that the BDG model is able to take advantage of the co-integration properties of the underlying price series.

A metric that can be used to compare the models under evaluation is the final wealth, which is the sum of the cash account and the value of any eventual remaining inventory[7]. Mathematically, the wealth process is given by $\Omega_t = X_t + q_t S_t$ and the final wealth is simply Ω_T. Recall that the initial wealth value is $\Omega_0 = q_0 S_0$, which is the same

7 Here we do not penalize any remaining inventory.

for all test cases ($\$3.1485 \times 10^6$). The resulting final wealth value for each model, considering the real price processes (for both PETR3 and PETR4), is approximately $\$3.1442 \times 10^6$, $\$3.1501 \times 10^6$ and $\$3.1589 \times 10^6$, respectively. Figure 5.6 shows the wealth process corresponding to each model. Note that the processes corresponding to the AC models are more stable than the one corresponding to the BDG model, since the main source of volatility for the wealth processes is price variation and the inventory levels are generally lower throughout most of the trading session with the AC liquidation profile, as can be seen in Figures 5.3 and 5.5. Besides the wealth values obtained from the real price processes, we can also obtain wealth values for simulated price processes. The relative wealth gain[8] of the multidimensional AC and the BDG models (with respect to the uni-dimensional AC model) are shown as histograms in Figures 5.7a and 5.7b, respectively. The models achieve progressively better results, considering their average wealth values of (approximately) $\$3.1533 \times 10^6$, $\$3.1540 \times 10^6$ and $\$3.1663 \times 10^6$, respectively for the uni-dimensional, multidimensional AC and the BDG models. In relation to the baseline model, the multidimensional AC and BDG models achieve improvements of 2.35 and 41.19 basis points, respectively. In order to further illustrate the variance of the wealth processes corresponding to each model, we plot in Figure 5.8 the 0.05, 0.50 and 0.95 quantile curves corresponding to wealth processes obtained through simulations. Unlike before, here we use $S_0 = \bar{S}$ so as to remove variance effects related to the difference between the initial and the mean price levels. Note that the median wealth increases throughout all trading session in the case of the BDG model, while it stays flat for the AC models. Additionally, the wealth variance is significantly smaller in the case of the BDG model. This result is particularly remarkable given that the liquidation profile for the BDG model assumes more aggressive positions than its counterpart model, as seen in Figure 5.5.

As a second application of the models, we also run them for another pair of stocks, namely the common and preferred shares of the Brazilian bank Bradesco. As before, we fit the models' parameters using orders data from November 25$^{\text{th}}$ to 28$^{\text{th}}$, 2019, and test their performance with real data corresponding to the following trading day. The obtained liquidation profiles are illustrated in Figures 5.9 and 5.10. As before, the inventory curves corresponding to the AC model are smoother and less irregular than those corresponding to the BDG, albeit now the liquidation speed for AC is slower when compared to the Petrobras case. The BDG curves are also less dramatic than before in the sense that there is no reversion between long and short positions and that the short position for the second stock is not as large. Interestingly, however, the multidimensional models are not able to significantly outperform the base model in relation to the wealth criterion. Unlike the previous case, the multidimensional AC model performs worse

8 Computed as $(\Omega_T^{\text{M}}/\Omega_T^{\text{UAC}} - 1) \times 10^4$, where Ω_T^{M} and Ω_T^{UAC} represent the final wealth corresponding to the model in question and the uni-dimensional AC model, respectively.

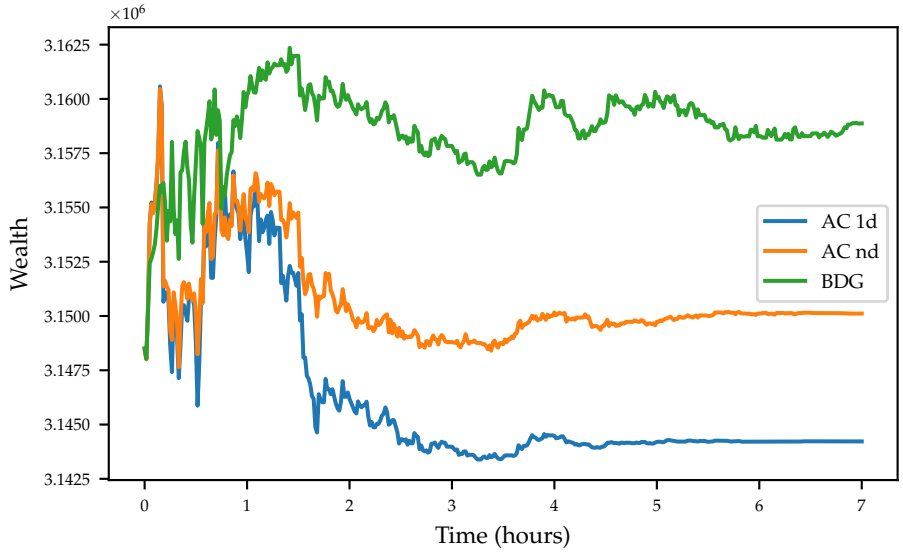
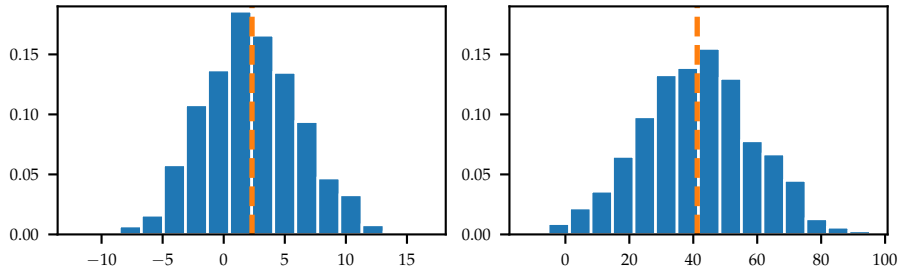


Figure 5.6: Wealth processes — AC and BDG models



(a) Multidimensional AC *(b)* BDG

Figure 5.7: Relative wealth gain (basis points) histograms

(minus 2.33 basis points) and BDG performs only slightly better (5.85 basis points) than the baseline. The respective wealth distributions are shown in Figures 5.11a and 5.11b.

5.3 Sensitivity to model parameters

The values for most of the AC and BDG models' parameters are exogenously determined from real world circumstances. Some, such as S_0 and q_0 are directly observed or given, while others, such as H, ϵ, Γ and σ (AC) and R, $\bar{S}$, V and η (BDG), may be estimated from the data through statistical means. The value for T is also constrained because of externally defined time to liquidation limits (usually dependent on trading windows). The value for Γ in the BDG model is somewhat less restricted in the sense that it is not directly derived from real data. Nevertheless, since we are considering the problem of optimal liquidation, we want the remaining inventory q_T to be significantly close to zero, which, in turn, requires Γ to be sufficiently big.

The remaining risk aversion parameters λ (AC) and γ (BDG) are the ones whose value assignment is more flexible, in the sense that they

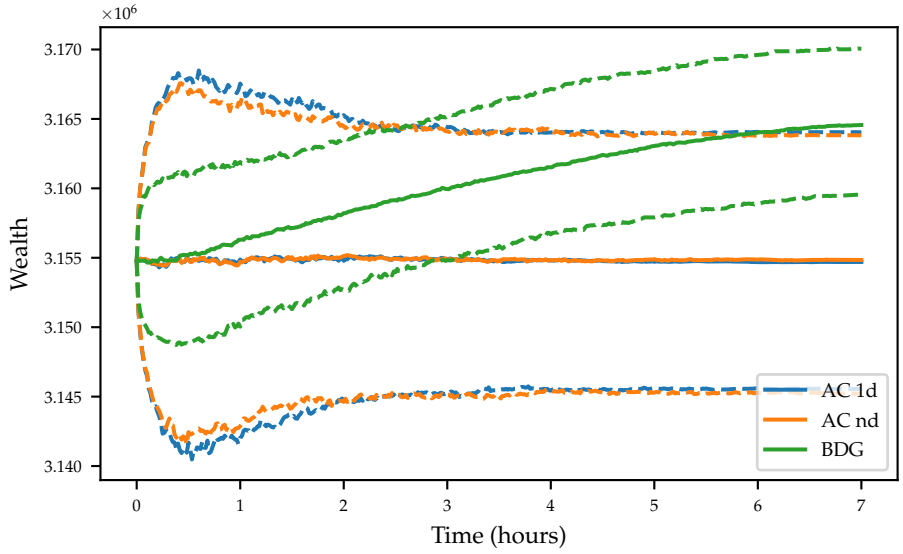


Figure 5.8: Wealth processes variance (0.05, 0.50 and 0.95 quantiles)

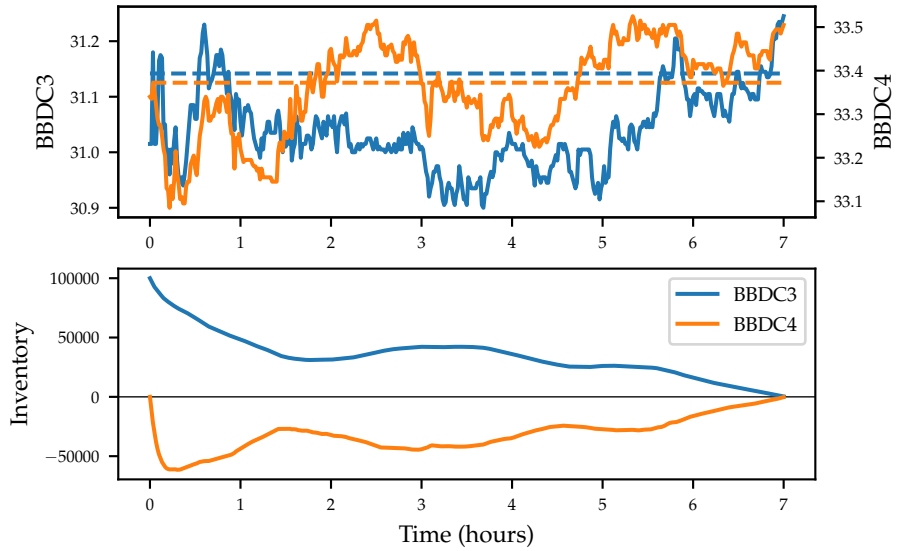


Figure 5.9: Inventory processes — BDG model

admit a greater range of reasonable choices. In general, bigger values for λ and γ result in faster liquidation profiles in which the correlation and the mean-reverting (co-integrated) properties of the series are not adequately taken advantage of. On the other hand, smaller values of λ lead to essentially constant trading speeds, resulting in basically linear inventory profiles. As for γ, smaller values induce the inventory to assume unrealistic large positions, even beyond the initial inventories, which is contrary in spirit to the original problem of liquidation. This behavior is illustrated in Figure 5.12, where we show the inventory profiles resulting from the application of different values of the risk aversion parameters for both models.

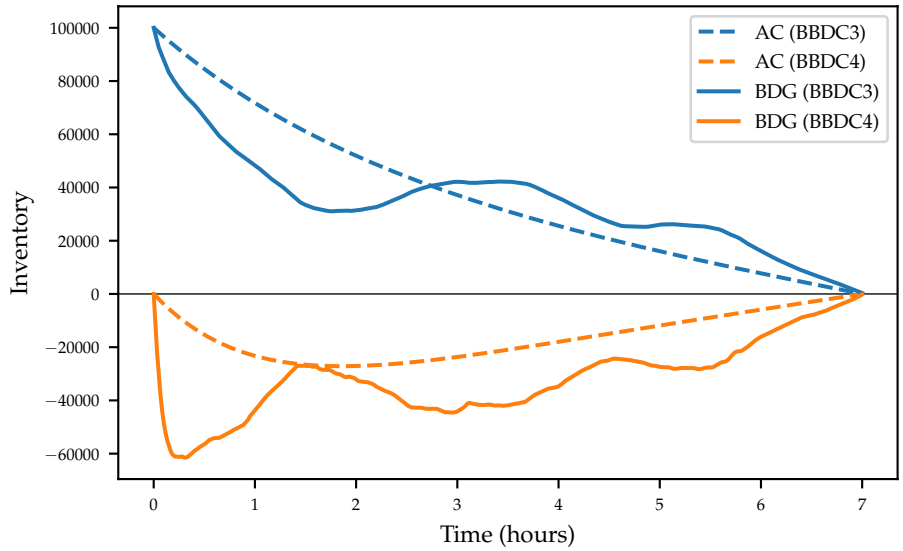
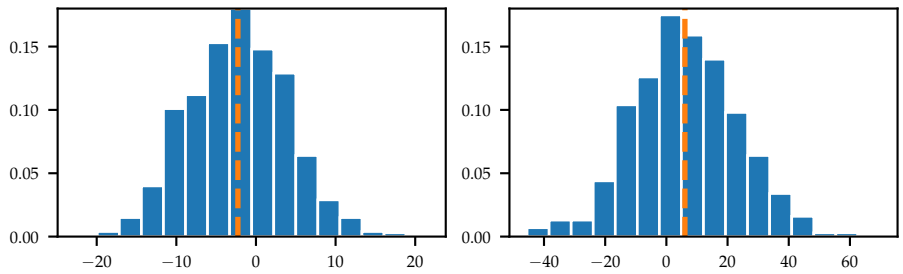


Figure 5.10: Inventory processes — AC and BDG models



(a) Multidimensional AC *(b)* BDG

Figure 5.11: Relative wealth gain (basis points) histograms

Specifically in the case of the BDG model, it is also useful to understand how big Γ should be in order to force liquidation by the end of the trading session. Recall that this model does not guarantee $q_T = 0$, but we should be able to get sufficiently close to a complete liquidation with proper parameterizing. Figure 5.13 and Table 5.5 illustrate the model's sensitivity with respect to the terminal inventory penalty parameter. In order to get close to total liquidation, we need Γ to be at least around 10^{-4}. Note that even with $\Gamma = 10^{-5}$, we still get somewhat close to complete liquidation. On the other extreme, when we set $\Gamma = 10^{-8}$, the terminal liquidation penalty is so low that it is virtually indistinguishable from zero. The Table 5.5 shows the remaining inventories corresponding to some values of Γ.

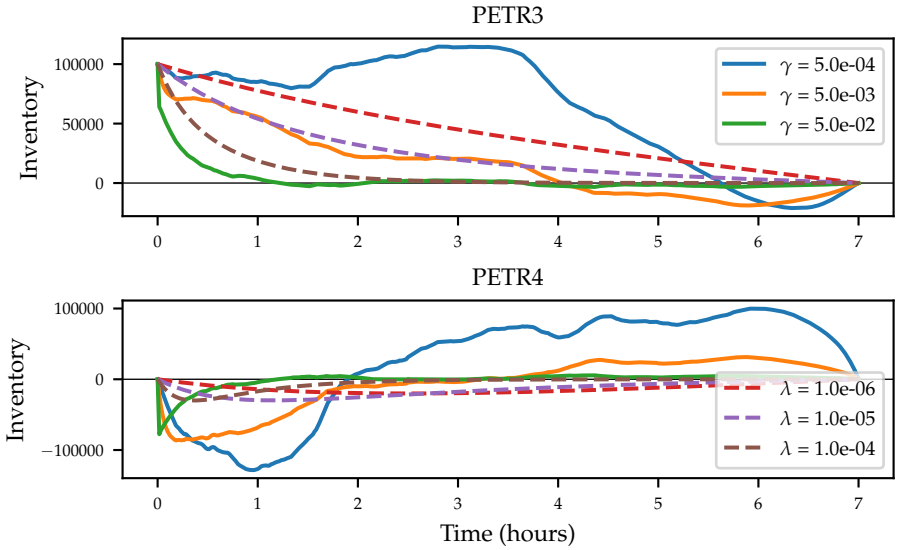


Figure 5.12: Risk aversion sensitivity

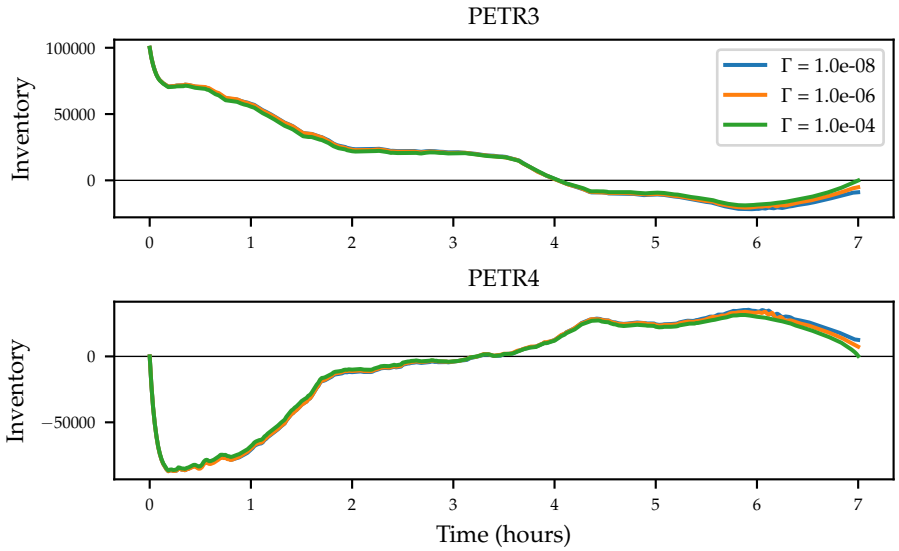


Figure 5.13: Terminal inventory penalty sensitivity

Table 5.5: Remaining inventories per Γ

Γ	PETR3	PETR4
0	−9040.7	12 599.3
1×10^{-8}	−8974.2	12 507.9
1×10^{-7}	−8412.7	11 748.0
1×10^{-6}	−5184.0	7367.0
1×10^{-5}	−1077.7	1720.5
1×10^{-4}	−109.7	343.4
1	17.5	160.2

6 NUMERICAL SOLUTION VIA NEURAL NETWORKS

As discussed in detail in Section 3.2, the solution to the BDG model's PDE given by (3.20) and its corresponding terminal condition (3.21) was obtained by a sequence of steps:

1. At first, the problem was simplified by dropping the dimension corresponding to the variable x, which corresponds to the cash account process. This was done through the *ansatz* (3.22) and resulted in the PDE on θ given by (3.23) and its corresponding terminal condition (3.24).

2. Then, the optimization (supremum) term in (3.23) was analytically handled through a simple differential technique, which resulted in a more manageable PDE given by (3.25).

3. Lastly, the authors cleverly propose applying the *ansatz* (3.27), which essentially transforms the previous PDE problem into a more amenable ODE problem. The resulting system of equations can then be numerically handled by readily available ODE solvers[1].

These simplification steps were welcome because, in its original primal form, the BDG model's PDE is rather intractable. Some of the most relevant challenges it presents are (i) its relatively high number of dimensions[2], (ii) its general non-linearity and, in particular, (iii) the optimization term typical of HJB equations.

Unfortunately, however, it is not always the case that these or alternative simplifying mathematical artifices are available and applicable when one is confronted with the task of solving a general PDE. Except for the simpler cases, one should not expect to find analytical solutions, and even common numerical approaches (such as finite difference or finite element methods) may not be practical. In particular, many conventional grid-based numerical methods are known to be susceptible to the curse of dimensionality, since their discretization strategies require the object function to be evaluated on an exponentially increasing number of points with respect to the number of dimensions of the problem.

In recent years, there have been some novel developments which brought together the fields of scientific computing and machine learning through the application of neural networks-based techniques to

1 Such as the `DifferentialEquations.jl` Julia package (Rackauckas & Nie, 2017) or the SciPy (scipy.org) Python package and its `solve_ivp` method.

2 Excluding the time dimension, the total number of "spacial" dimensions in this problem is equal to $2d + 1$, where d represents the number of financial assets under consideration.

numerical problems such as PDE solving (which we refer hereby as neural PDE solvers). Two of the earliest and most influential papers are Raissi et al. (2019), which introduces the concept of Physics-Informed Neural Networks (PINNs), and Sirignano and Spiliopoulos (2018), which introduces the Deep Galerkin Method (DGM). Both rely on neural networks with their high expressivity and universal approximation properties[3]. In both cases, neural networks are trained via loss functionals that take into account (i) the physical (or, in our application, the "financial") laws that govern the system, expressed in terms of PDEs, (ii) the required boundary conditions, including initial or terminal ones, and, optionally, (iii) known data points.

Such prodigious numerical methods have recently been computationally enabled in great measure by Automatic Differentiation (AD), which is a set of algorithmic techniques that allow the evaluation of the derivatives of functions along with their respective computation. Unlike symbolic and numerical differentiation, AD admits a more straightforward implementation and is not as afflicted by round-off and discretization errors. AD is also particularly well-suited to the computation of higher order derivatives. Such AD techniques recently came to prominence in machine learning suites (*e.g.*, TensorFlow and PyTorch), where they are heavily used for the backpropagation step during the network training step. In the present application, besides being used for backpropagation, AD is also paramount to evaluate the differential loss terms.

Despite their common kernels, both of the aforementioned papers have slightly different approaches and emphases. The first one focuses on physics-based applications and takes as its starting point the typical data-driven optimization scenario of machine learning applications, whose loss functions only consider boundary conditions and experimentally measured data points. At this point, the authors propose including an additional physical law term (*i.e.*, the PDE term) to the loss expression (hence the "physics-informed" designation), which is arguably able to overcome low data availability scenarios. On the contrary, the second paper focuses on finance-based applications, where hidden state variables are typically not available for empirical observation (hence the absence of the data loss term in this work). Moreover, the authors focus on the question of high dimensionality, typical of financial applications, where each asset under consideration usually corresponds to one (or more) dimensions in the mathematical models. Therefore, they propose a mesh-free algorithm that randomly samples points from the internal and boundary domains to compute loss values corresponding to the differential and the boundary terms, respectively. Another key contribution is a novel neural network architecture, similar to Long Short-Term Memory (LSTM) networks, which is arguably more performant and better suited to capture rapidly changing function profiles than regular feed forward fully connected networks.

The inherent flexibility of neural networks and the devised PDE solving algorithm allows for the specialization of the general method

3 Assuming, of course, the object functions under analysis are sufficiently regular.

to some particular classes of PDEs. In this regard, we refer to Al-Aradi et al. (2022), where we were able to improve on the original DGM with respect to two classes of PDEs, namely, Fokker-Planck and HJB equations. In the first case, given that the solution of the equation is a probability density function, which is everywhere positive and which integrates to unity, we were able to re-parameterize the solution as the exponential of a properly normalized neural network. This re-parameterizing ensures that both positivity and unity requirements are obtained, which was not generally the case before. Regarding the HJB class of equations, we were able to directly tackle the unsimplified primal form of the equation, which includes the optimization (supremum) term. In this case, we adapted the algorithm to solve for the value and the control function simultaneously by representing both functions as distinct neural networks. The networks are trained in an alternating manner, similarly to policy improvement algorithms.

6.1 Description of the neural PDE solver

In general terms, given an unknown function u of time (t) and space ($\boldsymbol{x}$), whose domain is $[0,T]\times\Omega$ for $\Omega\subset\mathbb{R}^d$, we are interested in solving PDE problems of the form

$$\begin{cases}(\partial_t+\mathcal{L})\,u(t,\boldsymbol{x})=0 & \forall(t,\boldsymbol{x})\in[0,T]\times\Omega\\ u(0,\boldsymbol{x})=u_0(\boldsymbol{x}) & \forall\boldsymbol{x}\in\Omega\\ u(t,\boldsymbol{x})=g(t,\boldsymbol{x}) & \forall(t,\boldsymbol{x})\in[0,T]\times\partial\Omega,\end{cases}$$

where $\mathcal{L}$ is some differential operator and $\partial\Omega$ is the boundary of Ω. The equations above represent the PDE per se, the initial condition (expressed it terms of u_0) and the boundary condition (expressed in terms of g), respectively. Note that the previous structure can be easily adapted for terminal condition problems by replacing the initial condition expression above with a terminal one. Similarly, we could suppress boundary conditions if they are not relevant to the problem at hand.

The essence of the neural PDE solver method is to approximate u by a neural network $f(t,\boldsymbol{x};\boldsymbol{\theta})$, whose parameter set is denoted by $\boldsymbol{\theta}$. In order to train the neural network, we define a loss functional $L(\boldsymbol{\theta})$ composed of three loss terms related to each equation, namely:

- The differential loss term:

$$L_1(\boldsymbol{\theta})=\|(\partial_t+\mathcal{L})\,f(\cdot,\cdot;\boldsymbol{\theta})\|^2_{[0,T]\times\Omega,\,\nu_1};$$

- The initial condition loss term:

$$L_2(\boldsymbol{\theta})=\|f(0,\cdot;\boldsymbol{\theta})-u_0(\cdot)\|^2_{\Omega,\,\nu_2};$$

- The boundary condition loss term:

$$L_3(\boldsymbol{\theta})=\|f(\cdot,\cdot;\boldsymbol{\theta})-g(\cdot,\cdot)\|^2_{[0,T]\times\partial\Omega,\,\nu_3}.$$

In all three loss terms above, the error is measured in terms of the L^2-norm, such that

$$\|\phi(\cdot)\|^2_{\Theta,\nu} = \int_\Theta |\phi(z)|^2 \, d\nu(z),$$

where ν is a probability measure defined on the region Θ. The final loss functional is simply given by the sum of those three terms

$$L(\boldsymbol{\theta}) = L_1(\boldsymbol{\theta}) + L_2(\boldsymbol{\theta}) + L_3(\boldsymbol{\theta}).$$

In practice, nonetheless, the integrals present in the loss terms above are approximated by sampling points (z_n) belonging to each region, according to the respective distribution (ν), and then averaging $|\phi(z_n)|^2$.

We then use some optimization technique (*e.g.*, stochastic gradient descent) to search for a parameter set that minimizes the loss functional L. The general neural PDE method is presented in Algorithm 6.1.

Algorithm 6.1: Neural PDE method

Initialize the parameter set $\boldsymbol{\theta}_1$ and the learning rate λ;
repeat for $n = 1, 2, \ldots$
  begin Generate random samples
    Sample $(t_n, \boldsymbol{x}_n)$ from $[0, T] \times \Omega$ according to ν_1;
    Sample $\boldsymbol{y}_n$ from Ω according to ν_2;
    Sample $(\tau_n, \boldsymbol{z}_n)$ from $[0, T] \times \partial\Omega$ according to ν_3;
    $s_n \leftarrow \{(t_n, \boldsymbol{x}_n), \boldsymbol{y}_n, (\tau_n, \boldsymbol{z}_n)\}$;
  end
  begin Compute the loss value
    Compute $L_1(\boldsymbol{\theta}_n; t_n, \boldsymbol{x}_n)$ as the mean of
     $[(\partial_t + \mathcal{L}) f(t_n, \boldsymbol{x}_n; \boldsymbol{\theta}_n)]^2$;
    Compute $L_2(\boldsymbol{\theta}_n; \boldsymbol{y}_n)$ as the mean of
     $[f(0, \boldsymbol{y}_n; \boldsymbol{\theta}_n) - u_0(\boldsymbol{y}_n)]^2$;
    Compute $L_3(\boldsymbol{\theta}_n; \tau_n, \boldsymbol{z}_n)$ as the mean of
     $[f(\tau_n, \boldsymbol{z}_n; \boldsymbol{\theta}_n) - g(\tau_n, \boldsymbol{z}_n)]^2$;
    $L(\boldsymbol{\theta}_n; s_n) \leftarrow L_1(\boldsymbol{\theta}_n; t_n, \boldsymbol{x}_n) + L_2(\boldsymbol{\theta}_n; \boldsymbol{y}_n) + L_3(\boldsymbol{\theta}_n; \tau_n, \boldsymbol{z}_n)$;
  end
  begin Perform the SGD optimization step
    Compute the gradient of the loss with respect to the
     parameters: $\nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}_n; s_n)$;
    $\boldsymbol{\theta}_{n+1} \leftarrow \boldsymbol{\theta}_n - \lambda \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}_n; s_n)$;
  end
until $L(\boldsymbol{\theta}_n)$ or $\|\boldsymbol{\theta}_{n+1} - \boldsymbol{\theta}_n\|$ is small enough;

6.2 Numerical results

In this section, we apply the neural PDE solver method to the BDG model's equations (3.25 and 3.26) using the ODE solver as a comparison base. As a first step, we begin with a simple uni-dimensional case with

all parameters (*i.e.*, R, $\bar{S}$, V, η, Γ, γ, S_0, q_0 and T) set to unity. The uni-dimensional version of (3.25) is given by

$$0 = \partial_t\theta\left(t,q,S\right) + \frac{1}{4}\eta^{-1}\,\partial_q\theta(t,q,S)^2 + \frac{1}{2}\Sigma\,\partial_{SS}\theta\left(t,q,S\right) - \frac{\gamma}{2}\Sigma(q+\partial_S\theta\left(t,q,S\right))^2 + \left(\bar{S}-S\right)R\,\left(\partial_S\theta\left(t,q,S\right)+q\right) \quad (6.1)$$

and the uni-dimensional terminal condition corresponding to (3.26) is

$$\theta(T,q,S) = -\Gamma q^2. \quad (6.2)$$

Figures 6.1, 6.2, 6.3, 6.4 and 6.5 show, the inventory (q), value (θ) and control (v) processes obtained[4] by both numerical methods. From such results, it is clear that the neural PDE solver is able to satisfactorily handle the problem in a simple uni-dimensional scenario. The training loss after all iterations is around 10^{-7}.

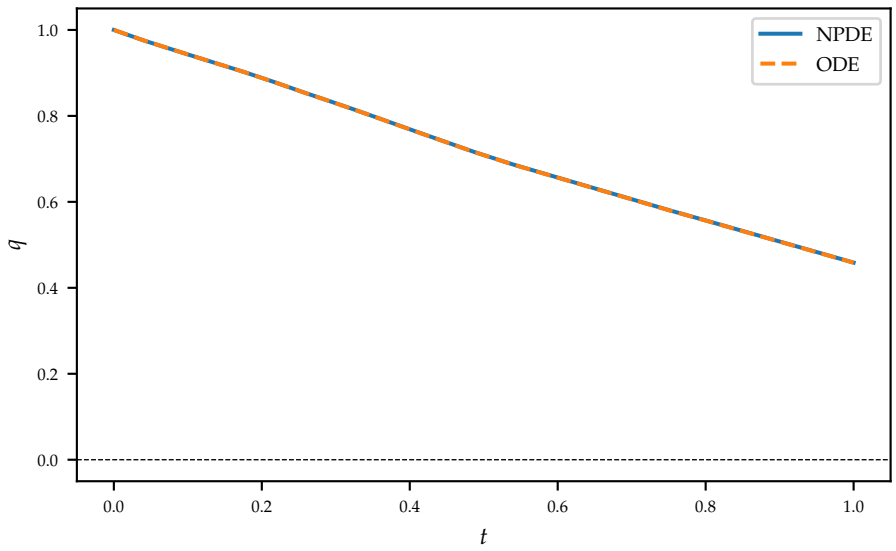


Figure 6.1: Inventory (q) process (parameters equal to 1)

Unfortunately, however, the neural PDE solver is not robust with respect to the model parameters, particularly the domain scale. For instance, setting S_0, $\bar{S}$ and q_0 to 100 while keeping all other parameters equal to one, we get the disappointing results illustrated by Figures 6.6, 6.7, 6.8, 6.9 and 6.10. The training loss after all iterations is still at an extremely high level of around 10^7.

We can improve on such predicament by noting that a best practice when using neural networks is to normalize the input values they are fed with. Thus, in the current scenario, it is desirable to normalize both q and S variables[5] which, in real applications, can easily assume values of several orders of magnitude. This implies that we need to

4 In all cases presented in this section, we run the neural network training process for five thousand iterations.

5 We could normalize the time variable as well, but it typically does not vary as significantly as the other two.

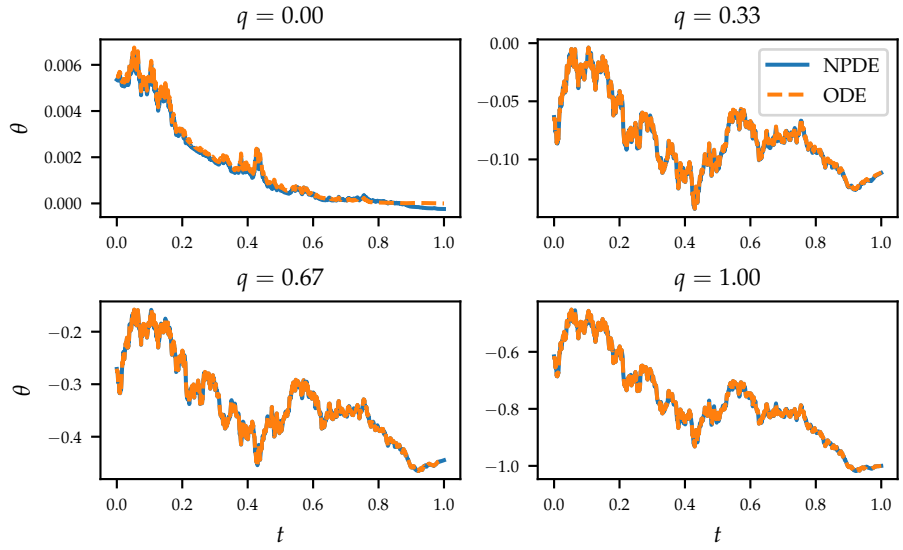


Figure 6.2: Value (θ) process (parameters equal to 1, inventory slices)

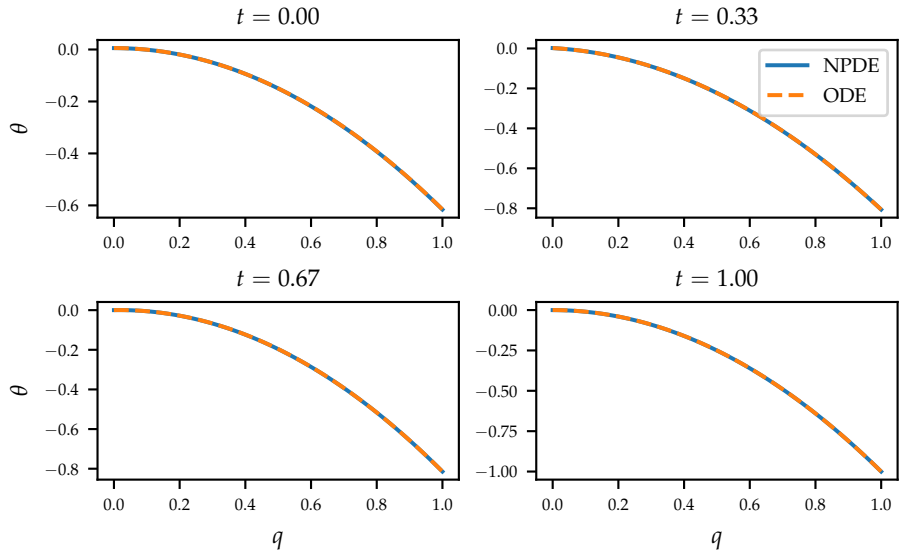


Figure 6.3: Value (θ) process (parameters equal to 1, time slices)

rewrite the PDE problem with respect to the variables $\tilde{q}$ and $\tilde{S}$, given by $\tilde{q} = \frac{1}{A}q$ and $\tilde{S} = \frac{A}{B}S$, for some normalizing constants $A, B > 0$.

The unusual choice of the normalizing factor for the S variable is due to financial coherence considerations. Even tough the variables x, q and S are mathematically independent in the original PDE problem, one could argue that they are interdependent in the real world, at least with respect to their units of measurement. These variables refer, respectively, to the cash process, measured in terms of monetary units ($), to the inventory process, measured in terms of number of assets (#), and to the price process, measured in terms of monetary units per number of assets ($/#). Thus, any scale change in x or q should also

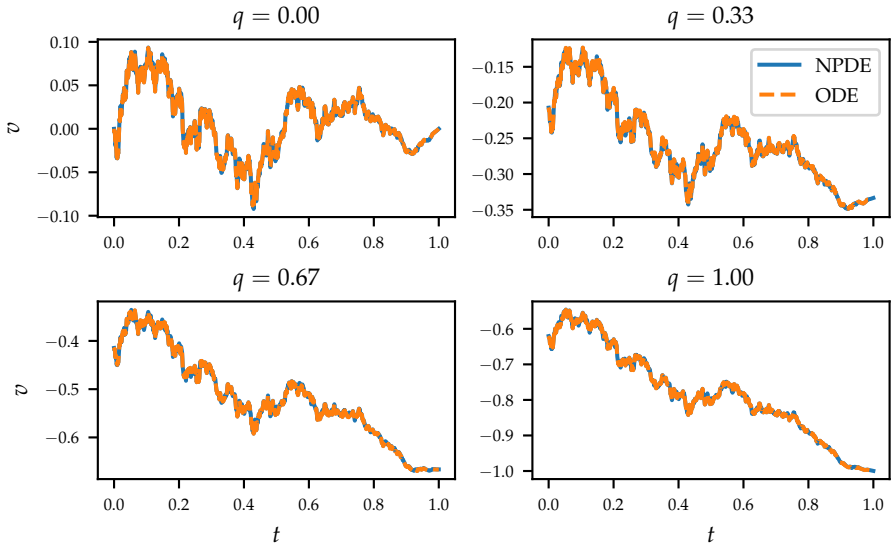


Figure 6.4: Control (v) process (parameters equal to 1, inventory slices)

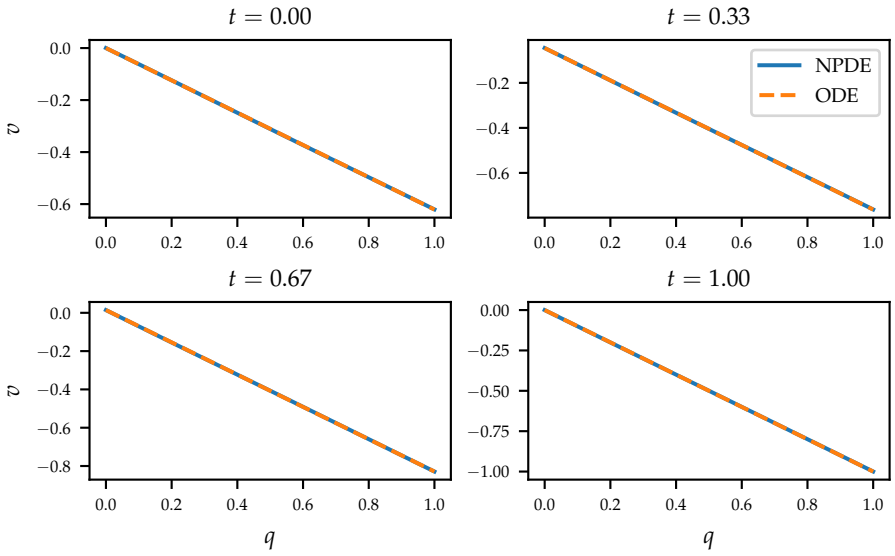


Figure 6.5: Control (v) process (parameters equal to 1, time slices)

impact S. For instance, if we decide to measure q in terms of lots of assets instead of asset units, the price per lot should be different from the price per unit. Analogously, if we change the monetary unit (*e.g.*, by using another currency), the price per asset should be different as well. Thus, in order to maintain financial coherence among all variables, we use A as the normalizing factor in the inventory (#) dimension and B as the normalizing factor in the monetary dimension ($).

Similarly, it is also useful to take into account the dimensions associated with the other mathematical entities of the model, such as its parameters, its functions and their respective derivatives. With some abuse of notation, using τ, # and $ to denote, respectively, the time,

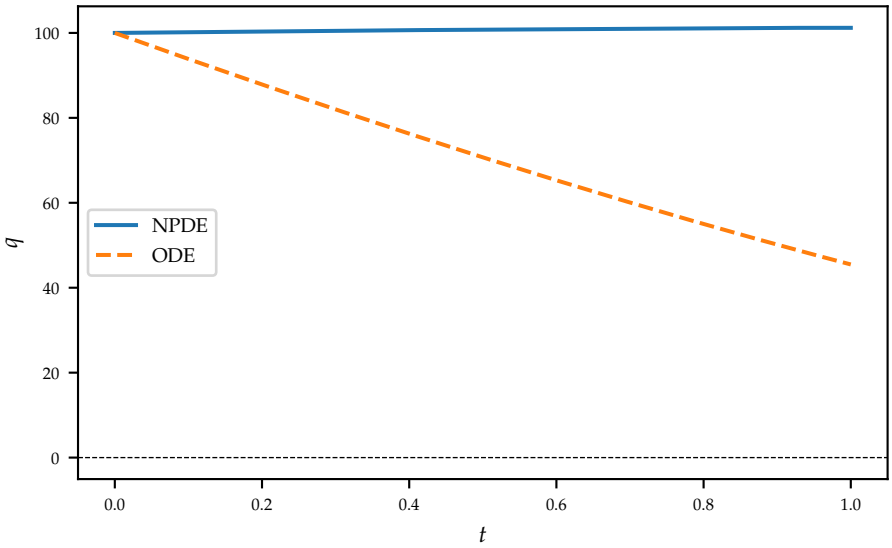


Figure 6.6: Inventory (q) process ($S_0 = \bar{S} = q_0 = 100$)

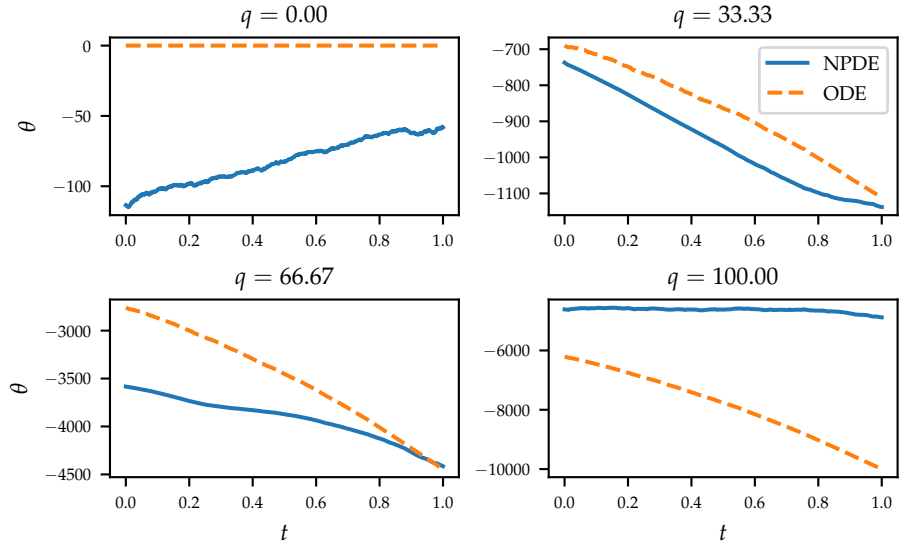


Figure 6.7: Value (θ) process ($S_0 = \bar{S} = q_0 = 100$, inventory slices)

inventory and price unity measures, we can perform the dimensionality analysis of each mathematical object in the model, taking into consideration its equations. From the inventory dynamics (3.11), we get that v is measured in terms of $^{\#}/_{\tau}$, and from the price dynamics (3.12), we obtain that R, $\bar{S}$ and V are measured in terms of τ^{-1}, $^{\$}/_{\#}$ and $^{\$}/_{\#\sqrt{\tau}}$, respectively (we assume the Brownian motion term dW_t has $\sqrt{\tau}$ as its dimension). From the cash dynamics (3.17) and the temporary market impact expression (3.15), we can establish that L is measured in terms of $^{\$}/_{\tau}$ and η as $^{\tau\$}/_{\#^2}$. From the final wealth expression present in (3.19) and the permanent market impact equation (3.18), we can assert that ℓ is measured in terms of $\$$ and Γ as $^{\$}/_{\#^2}$. Assuming that the value expression in (3.19) is dimensionless, we can take γ as being measured

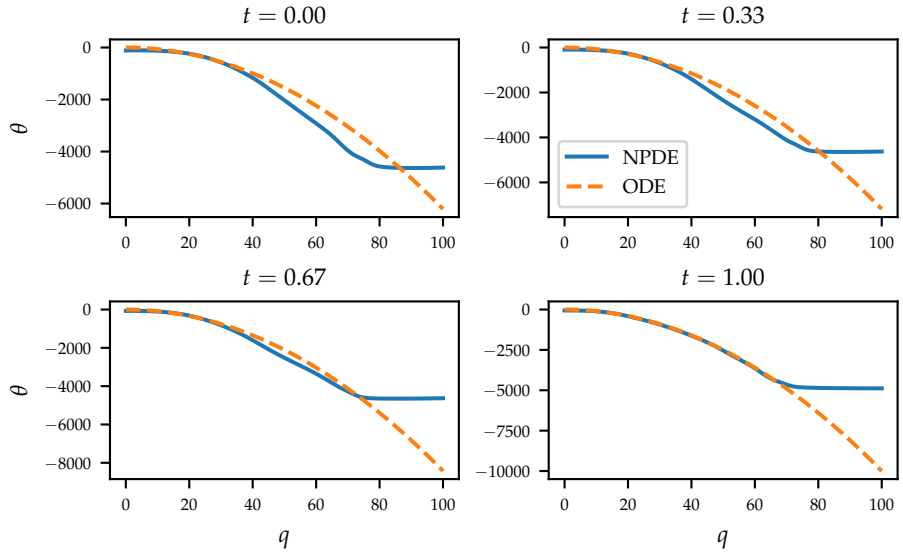


Figure 6.8: Value (θ) process ($S_0 = \bar{S} = q_0 = 100$, time slices)

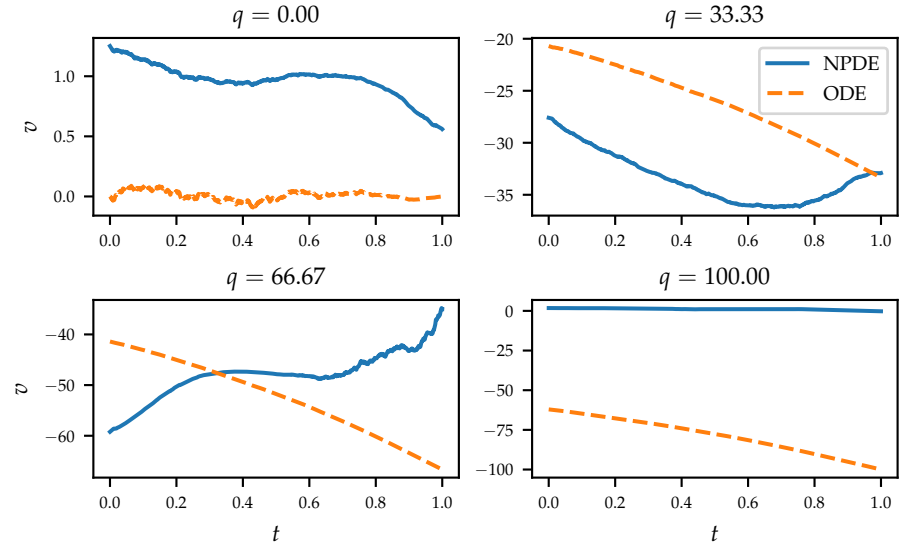


Figure 6.9: Control (v) process ($S_0 = \bar{S} = q_0 = 100$, inventory slices)

in terms of $\$^{-1}$. Lastly, from the terminal condition (3.24) we can affirm that the dimension for θ is \$ and, accordingly, the ones for its partial derivatives $\partial_t\theta$, $\partial_q\theta$, $\partial_S\theta$ and $\partial_{SS}\theta$ are $\$/\tau$, $\$/\#$, $\#$ and $\#^2/\$$, respectively. Building upon this dimensionality analysis, we can define the normalized version of all variables, parameters, and functions of the BDG model, written in terms of the original ones and the normalizing factors A and B, as presented in Tables 6.1a and 6.1b.

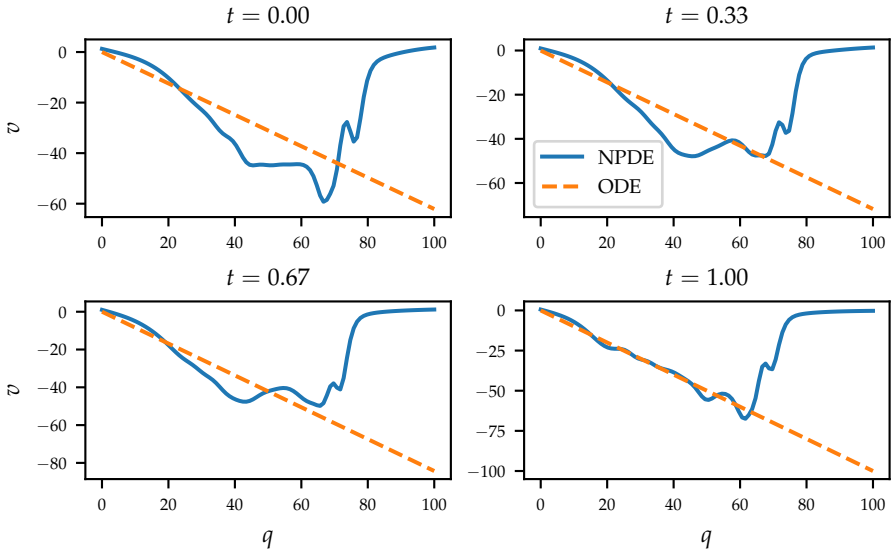


Figure 6.10: Control (v) process ($S_0 = \bar{S} = q_0 = 100$, time slices)

Table 6.1: Normalized functions and parameters

(a) Parameters

Parameter	Value
$\tilde{R}$	R
$\tilde{\bar{S}}$	$\frac{A}{B}\bar{S}$
$\tilde{V}$	$\frac{A}{B}V$
$\tilde{\Sigma}$	$\frac{A^2}{B^2}\Sigma$
$\tilde{\eta}$	$\frac{A^2}{B}\eta$
$\tilde{\Gamma}$	$\frac{A^2}{B}\Gamma$
$\tilde{\gamma}$	$B\gamma$

(b) Functions

Function	Value
$\tilde{\theta}$	$\frac{1}{B}\theta$
$\partial_t\tilde{\theta}$	$\frac{1}{B}\partial_t\theta$
$\partial_{\tilde{q}}\tilde{\theta}$	$\frac{A}{B}\partial_q\theta$
$\partial_{\tilde{S}}\tilde{\theta}$	$\frac{1}{A}\partial_S\theta$
$\partial_{\tilde{S}\tilde{S}}\tilde{\theta}$	$\frac{B}{A^2}\partial_{SS}\theta$
$\tilde{v}$	$\frac{1}{A}v$

Thus, we can write the normalized version of (6.1) as

$$
\begin{aligned}
0 &= B\,\partial_{\tilde{t}}\tilde{\theta}\left(\tilde{t},\tilde{q},\tilde{S}\right) + \frac{1}{4}\eta^{-1}\frac{B^2}{A^2}\,\partial_{\tilde{q}}\tilde{\theta}\left(\tilde{t},\tilde{q},\tilde{S}\right)^2 + \frac{1}{2}\Sigma\frac{A^2}{B}\,\partial_{\tilde{S}\tilde{S}}\tilde{\theta}\left(\tilde{t},\tilde{q},\tilde{S}\right) \\
&\quad - \frac{\gamma}{2}\Sigma\left(A\tilde{q} + A\,\partial_{\tilde{S}}\tilde{\theta}\left(\tilde{t},\tilde{q},\tilde{S}\right)\right)^2 + \left(\bar{S} - \frac{B}{A}\tilde{S}\right)R\left(A\,\partial_{\tilde{S}}\tilde{\theta}\left(\tilde{t},\tilde{q},\tilde{S}\right) + A\tilde{q}\right) \\
&= \partial_{\tilde{t}}\tilde{\theta}\left(\tilde{t},\tilde{q},\tilde{S}\right) + \frac{1}{4}\eta^{-1}\frac{B}{A^2}\,\partial_{\tilde{q}}\tilde{\theta}\left(\tilde{t},\tilde{q},\tilde{S}\right)^2 + \frac{1}{2}\Sigma\frac{A^2}{B^2}\,\partial_{\tilde{S}\tilde{S}}\tilde{\theta}\left(\tilde{t},\tilde{q},\tilde{S}\right) \\
&\quad - \frac{\gamma}{2}\Sigma\frac{A^2}{B}\left(\tilde{q} + \partial_{\tilde{S}}\tilde{\theta}\left(\tilde{t},\tilde{q},\tilde{S}\right)\right)^2 + \left(\frac{A}{B}\bar{S} - \tilde{S}\right)R\left(\partial_{\tilde{S}}\tilde{\theta}\left(\tilde{t},\tilde{q},\tilde{S}\right) + \tilde{q}\right) \\
&= \partial_{\tilde{t}}\tilde{\theta}\left(\tilde{t},\tilde{q},\tilde{S}\right) + \frac{1}{4}\tilde{\eta}^{-1}\,\partial_{\tilde{q}}\tilde{\theta}\left(\tilde{t},\tilde{q},\tilde{S}\right)^2 + \frac{1}{2}\tilde{\Sigma}\,\partial_{\tilde{S}\tilde{S}}\tilde{\theta}\left(\tilde{t},\tilde{q},\tilde{S}\right) \\
&\quad - \frac{\tilde{\gamma}}{2}\tilde{\Sigma}\left(\tilde{q} + \partial_{\tilde{S}}\tilde{\theta}\left(\tilde{t},\tilde{q},\tilde{S}\right)\right)^2 + \left(\tilde{\bar{S}} - \tilde{S}\right)R\left(\partial_{\tilde{S}}\tilde{\theta}\left(\tilde{t},\tilde{q},\tilde{S}\right) + \tilde{q}\right).
\end{aligned}
$$

Additionally, the normalized expression corresponding to the terminal condition (6.2) is

$$\tilde{\theta}(T, \tilde{q}, \tilde{S}) = \frac{1}{B}\theta(T, q, S) = -\frac{1}{B}\Gamma q^2 = -\frac{1}{B}\frac{B}{A^2}\tilde{\Gamma}\left(A\tilde{q}\right)^2 = -\tilde{\Gamma}\tilde{q}^2.$$

Even tough it presents some deviations with respect to the solution provided by the ODE solver (particularly when q is close to zero), the normalized version of the neural PDE solver is now able to perform much better than the previous one, as attested by Figures 6.11, 6.12, 6.13, 6.14 and 6.15, where we used the same set of parameters as before and the normalizing factors $A = q_0$ and $B = q_0 S_0$, which should keep the magnitudes of $\tilde{q}$ and $\tilde{S}$ close to one. The training loss after all iterations is at a much more reasonable level of around 10^{-7}. Thus, at least to some extent, the applied normalization technique results in an improvement of the numerical method at hand.

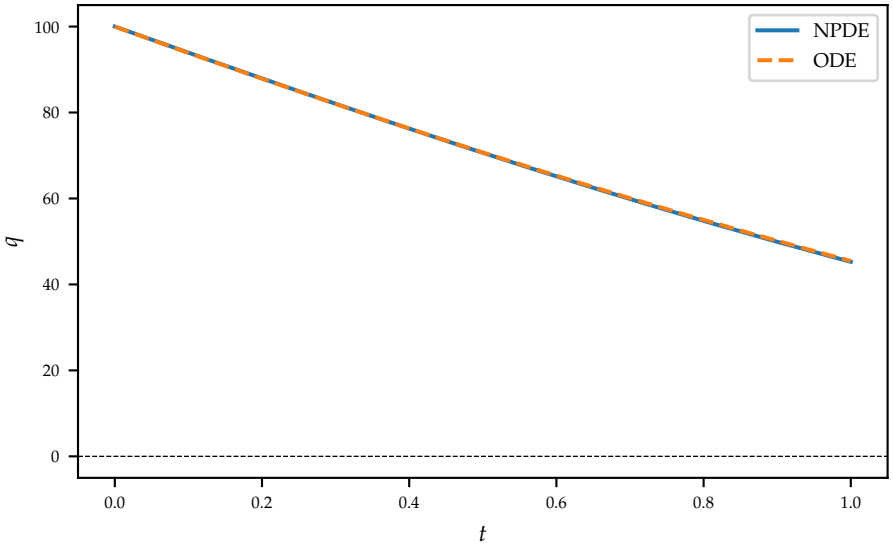


Figure 6.11: Inventory (q) process (normalized variables)

At any rate, even after the normalization adjustments, the neural PDE solver is still not generally robust with respect to the model's parameters. Figures 6.16, 6.17, 6.18, 6.19 and 6.20 illustrate the results when using the parameter values corresponding to PETR3 in Table 5.3. The training loss value is relatively high (around 10^{-2}) and the model is not able to converge to a proper solution.

Notice, however, that the value and control surfaces corresponding to the solution of the PDE with this set of parameters are somewhat problematic. They assume an extreme, singularity-like, behavior near the terminal boundary $t \approx T$. This effect can get even more pronounced when using different sets of parameters, reaching differences of several orders of magnitude. Remark, however, that this behavior does not occur when $q \approx 0$. In this case, the value curves obtained by the ODE solver remain relatively flat. We understand such explosive behavior as a simple consequence of the fact that the model forces any remaining inventory near the end of the trading interval to be quickly liquidated,

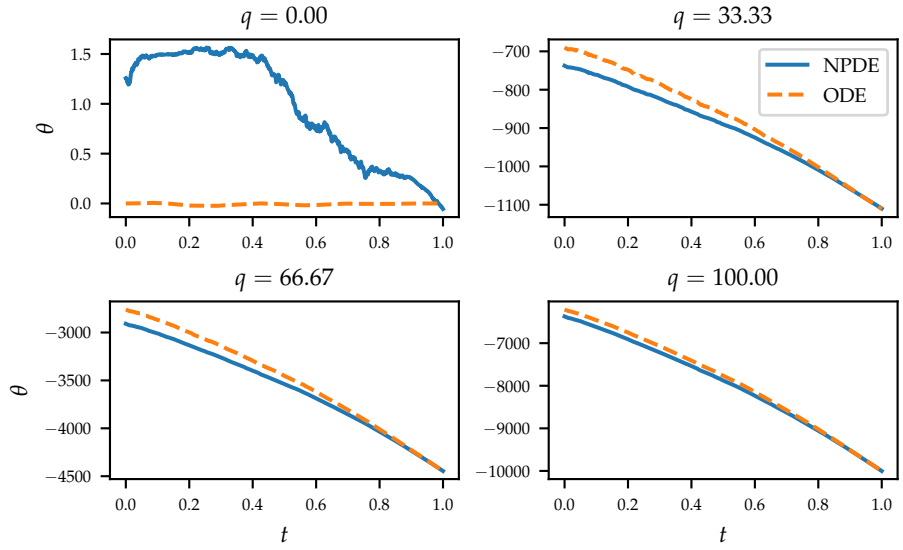


Figure 6.12: Value (θ) process (normalized variables, inventory slices)

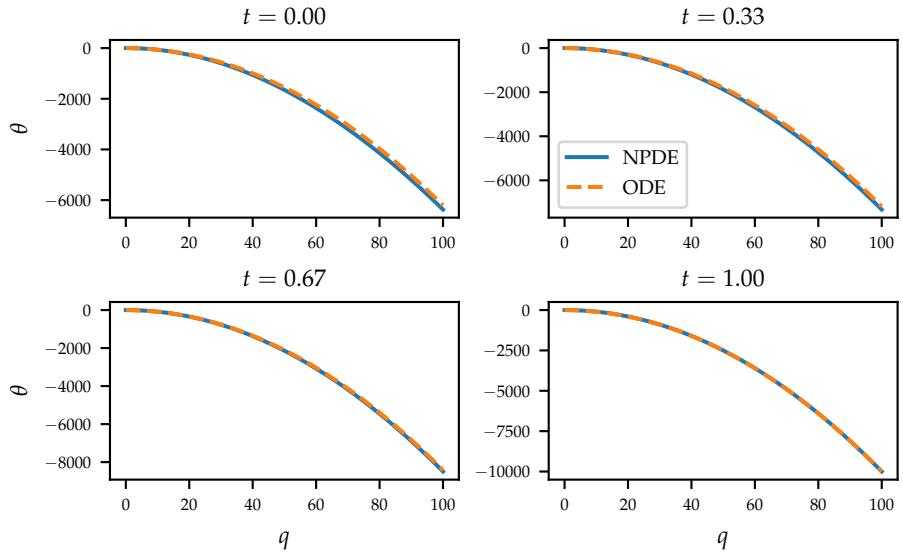


Figure 6.13: Value (θ) process (normalized variables, time slices)

and the corresponding liquidation speed must be roughly inversely proportional to the time still remaining. Therefore, this intense behavior is, in fact, expected. Mind, however, that these regions of high inventory levels near the terminal boundary should rarely be reached in practice, if the optimal control liquidation policy is being applied consistently throughout the trading session.

Notwithstanding the previous discussion, we argue that it is not tenable to expect a neural network-based model to easily converge to such ill-disposed surfaces. This is specially the case when taking into account that, on the neural PDE method, the boundaries and their surrounding regions act as an anchor or as an initial point of

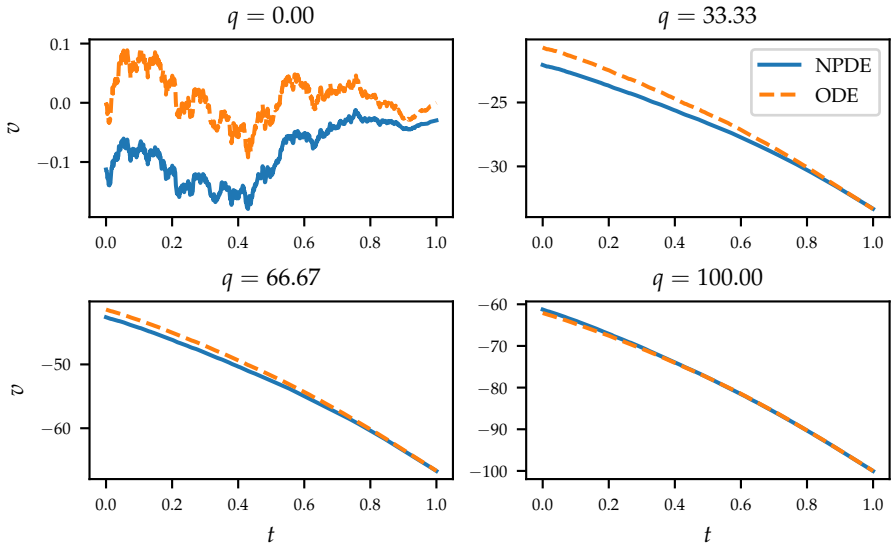


Figure 6.14: Control (v) process (normalized variables, inventory slices)

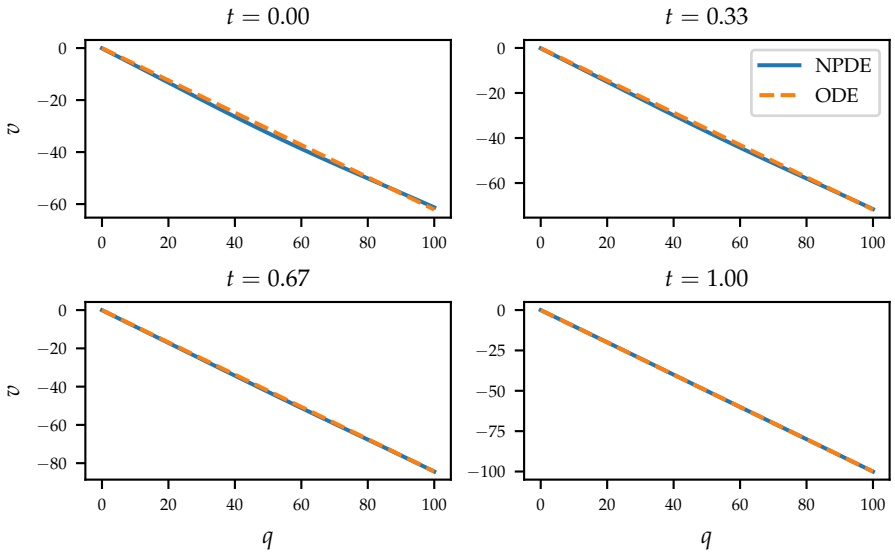


Figure 6.15: Control (v) process (normalized variables, time slices)

convergence of the model, from where all other regions' convergence should gradually emanate. However, since the object function values in such problematic boundary region are so much greater than elsewhere, the respective training loss term is expected to dominate over the PDE loss term elsewhere, hindering the prospects of convergence in the domain as a whole.

Motivated by the fact that such troublesome regions (with high inventory levels near the terminal time) should not be common in real world liquidation scenarios, one might try, as a possible workaround, avoiding such problematic subdomain by simply not sampling points with high values for both q and t. This could be achieved by using

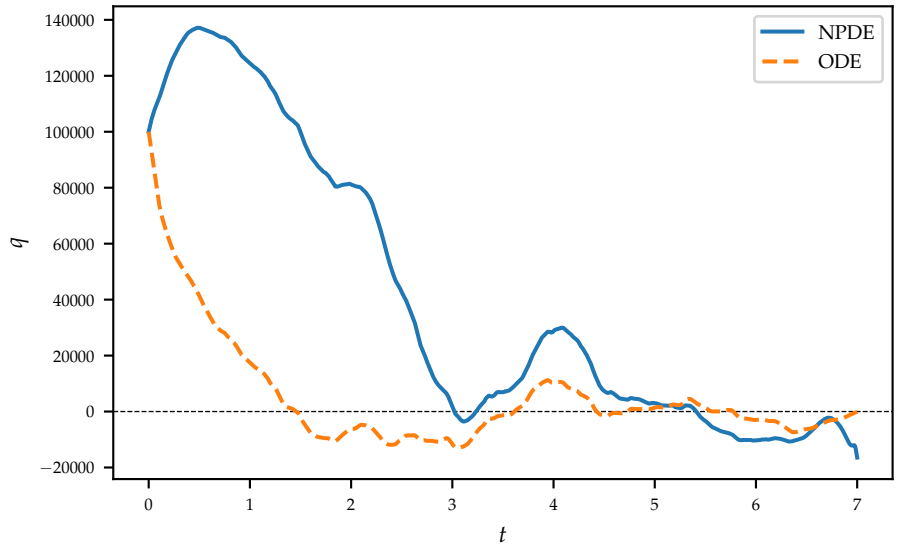


Figure 6.16: Inventory (q) process (PETR3)

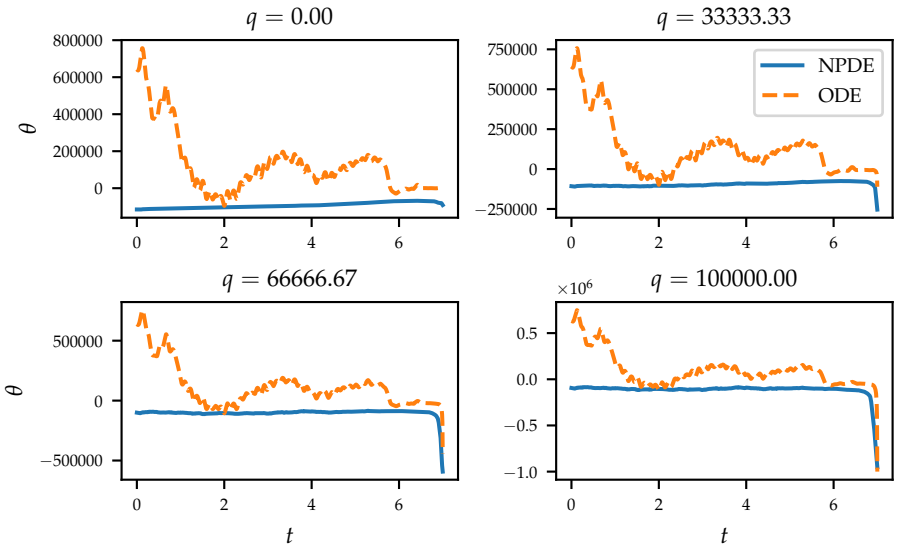


Figure 6.17: Value (θ) process (PETR3, inventory slices)

some sort of triangular domain or, analogously, a triangular sampler. Similarly, one could use the ODE based solution to come up with a distribution of commonly visited regions and use it instead of a more uniform sampler. Unfortunately, such strategies proved unfruitful even for the simpler unity parameter set.

We conjecture that, in other general cases where the object function being approximated by neural networks is ill-behaved in specific regions, one could circumvent such issue simply by avoiding sampling from such problematic regions. Sadly this strategy is not applicable here since the problematic region is precisely the boundary and its neighboring subdomain.

Alternatively, one might try to use self-adapting weights for the loss terms in order to avoid the boundary loss term completely dominating

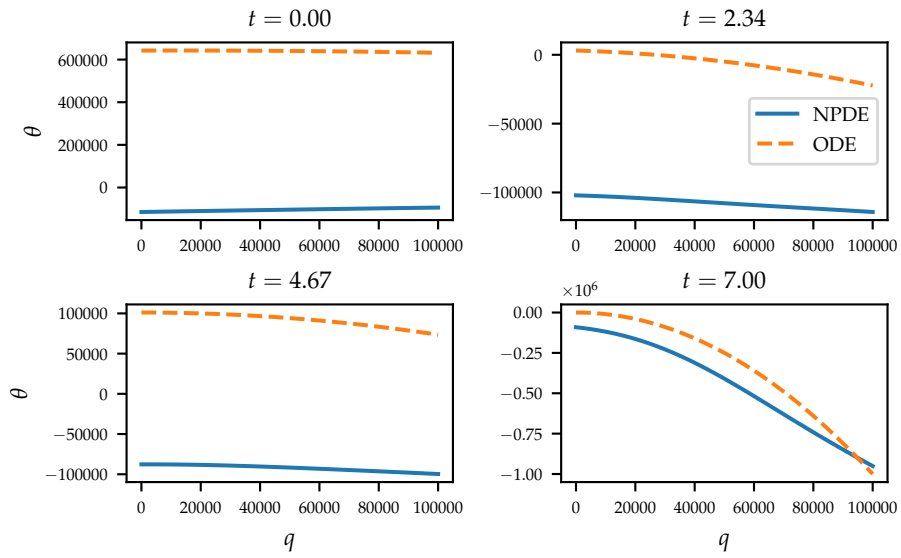


Figure 6.18: Value (θ) process (PETR3, time slices)

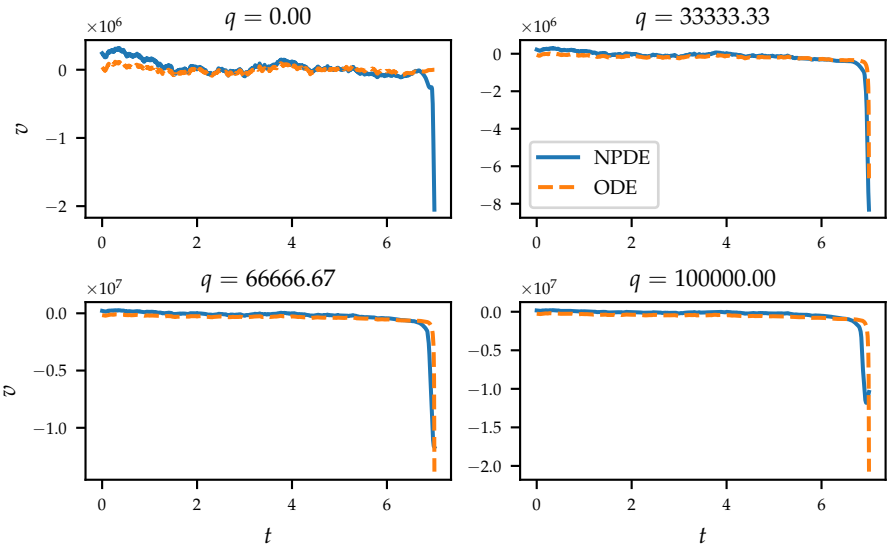


Figure 6.19: Control (v) process (PETR3, inventory slices)

the other ones, or, as another class of workaround, one could try to come up with some variable or function substitution scheme which could level the resulting function values across all of its domain, including the terminal boundary[6]. We leave those lines of work as a possibility for further research, however. As a final remark, since the neural PDE numerical results are not satisfactory even in the unidimensional case, we do not extend its application to the original multidimensional version of the problem.

6 Such scheme would be similar in spirit to what is done in the case of Fokker-Planck equations in Al-Aradi et al. (2022).

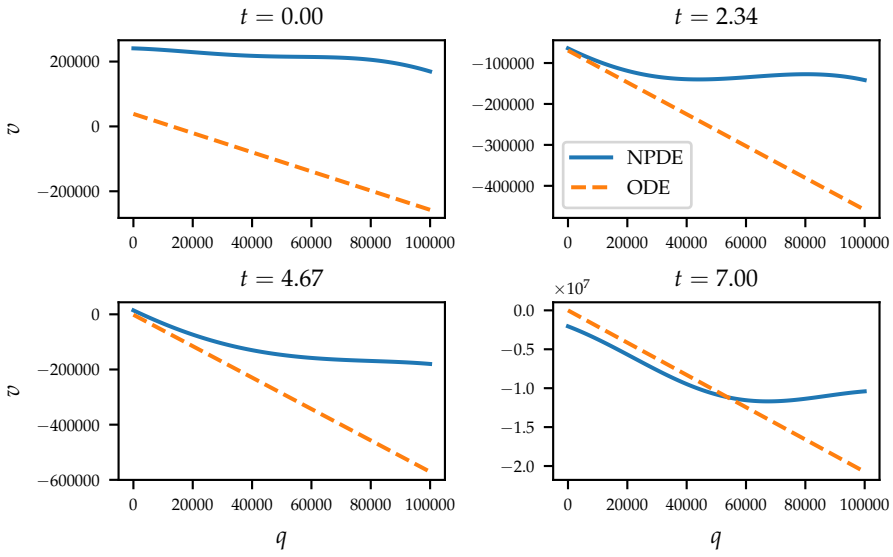


Figure 6.20: Control (v) process (PETR3, time slices)

7 CONCLUSION

In this work, we consider models for the optimal execution of financial assets given price impacts resulting from trading. We also apply such models to Brazilian stocks.

We launch this project by providing an account of the general operation of exchanges, particularly with respect to the two main types of orders, namely limit and market orders, and how such orders are collected and arranged via LOBs. From these general market structure principles and concepts, we are able to grasp the mechanism that gives rise to price impacts and how we can resort to LOBs to estimate and quantify them.

Next, we begin our study of the optimal execution models per se, firstly by presenting a classic result, namely the AC model, that introduces the main mathematical concepts related to portfolio liquidation (*e.g.*, the trade-off between expected wealth and risk), and secondly, by expounding a more recent one, namely the BDG model, which expands from the previous one in several general directions. Compared to AC, the BDG model is continuous in time, has a different optimization goal and requires distinct mathematical tools to be solved. More fundamentally, however, the BDG model assumes a different asset price dynamics[1], which allows for one of the key distinctive features of the BDG model: its ability to capture not only the correlation but also any eventual co-integration dependency properties between assets. This lets the model take current price information into account and adjust the liquidation profile accordingly in order to potentially achieve better results, both in terms of higher returns (expected wealth) and less incurred risks (wealth variance). Another distinction between both models is that, while the AC model's equations generally admit closed-form solutions[2], the BDG model's solution is obtained via a system of ordinary differential equations, which only admit numerical solutions. The derivation of both models' solutions are thoroughly demonstrated in the aforementioned discussion, which is concluded with some remarks about some available techniques which can be used to estimate the models' parameters from price series and LOB snapshots.

Following, we present a brief overview of some concepts and the definition of co-integration, particularly in relation to financial series. We also discuss some techniques to estimate co-integration vectors and introduce some statistical co-integration tests. We conclude such discussion with a remark about co-integration for continuous series.

1 Random walk in AC versus mean-reverting in BDG.

2 Only the multidimensional version with cross-asset price impact requires a numerical solution via a linear system of equations.

Finally, we apply the models to real data obtained from the Brazilian stock market. In order to fit the models, *i.e.*, find the models' parameters matching the data at hand, we first need to process the raw order data flow to gradually reconstruct the corresponding LOB at periodic time intervals. From such snapshots, we are then able to recover the price impact coefficients and the intraday price series, and from such price series we are able to estimate the price processes' parameters.

In the main studied case, the BDG model is able to outperform the AC model in terms of both the mean resulting wealth level and its variance, notwithstanding its unconventional inventory process profile, which maintained significant positions (both long and short) throughout most of the trading session. Such behavior contrasts with what one should usually expect from the AC model, where a major portion of the initial inventory is typically liquidated early on, even though short positions buildup is also common for this otherwise more conservative model. Naturally, such results are grounded on the premise of a mean-reverting price dynamics and any deviation from such hypothesis may result in undesired losses in a real world application. Such risk analysis is not in the scope of the present work, though.

As a second line of work, we also investigate the application of neural networks to the numerical solution of the PDE problem that arises in the context of the BDG model solution. Such application is done through the DGM framework, where neural networks are used to approximate the desired functions, and the differential and boundary equations are used as terms in a loss functional that is fed into the optimization algorithm for the network's parameters. Even though it is possible to successfully solve the problem for simpler sets of parameters, the method does not generalize well to the more realistic tried settings. We argue that the failure of convergence to the solution is due to the singularity-like profile of the curves near the terminal boundary where significant amounts of inventory need to be liquidated in increasingly small intervals of time, resulting in extreme trade speeds which bring convergence difficulties to the numerical optimization algorithms. Further work is required to devise and evaluate alternative mathematical or numerical techniques that could eventually circumvent such convergence challenge. Such work is, however, not in the scope of the present work.

STOCHASTIC CONTROL AND THE HAMILTON-JACOBI-BELLMAN EQUATION

In the context of stochastic control problems, consider the control process $(u_t)_{t\in[0,T]}$ associated with a d-dimensional controlled state process $(X_t^u)_{t\in[0,T]}$. Suppose we want to maximize some expected profit (or utility) expression, which depends on the state process, subject to a required terminal condition. Alternatively, one might be interested in minimizing some cost (or loss) expression, but it is usually trivial to convert between maximization and minimization problems by just toggling the signal of the expression at hand.

The controlled state process dynamics can be expressed as

$$dX_t^u = \mu(t, X_t^u, u_t)\, dt + \sigma(t, X_t^u, u_t)\, dW_t, \tag{A.1}$$

with $X_0^u = x_0$, where μ and σ represent, respectively, its drift and diffusion properties and W_t is a Brownian motion. The utility expression for a given control u_t can be represented as a value function $V^u : [0,T] \times \mathbb{R}^d \to \mathbb{R}$ given by

$$V^u(t,x) = \mathbb{E}\left[\int_t^T F(s, X_s^u, u_s)\, ds + G(X_T^u) \,\middle|\, X_t^u = x\right],$$

where $F(t,x,u)$ and $G(x)$ indicate running and terminal utility (or loss) functions, respectively. The terminal condition on the value function is given by

$$V^u(T,x) = G(x).$$

We can define the optimal control $u^* = \arg\sup_u V^u$, where the supremum is taken among all admissible control processes, as well as the optimal value function

$$V(t,x) = V^{u^*}(t,x) = \sup_u V^u(t,x).$$

Given this setup, and assuming V to be smooth, the dynamic programming principle implies that the optimal value function must satisfy the following HJB equation

$$\partial_t V + \sup_u \{\mathcal{L}^u V + F(t,x,u)\} = 0,$$

where $\mathcal{L}^u$ is the infinitesimal generator of the controlled state process. Finally, recall that the expression for the infinitesimal generator of the multivariate Itô process with dynamics given by (A.1) is

$$\begin{aligned}\mathcal{L}^u &= \sum_{i=1}^d \mu_i \frac{\partial}{\partial x_i} + \frac{1}{2}\sum_{i=1}^d (\sigma\sigma^\intercal)_{i,i} \frac{\partial^2}{\partial x_i \partial x_i} \\ &= \mu^\intercal \cdot \nabla_x + \frac{1}{2}\operatorname{Tr}(\sigma\sigma^\intercal \cdot D_{xx}^2),\end{aligned}$$

where ∇_x and D_{xx}^2 represent the gradient and Hessian differential operators, respectively.

BIBLIOGRAPHY

Al-Aradi, A., Correia, A., Jardim, G., de Freitas Naiff, D., & Saporito, Y. (2022). Extensions of the Deep Galerkin Method. *Applied Mathematics and Computation*, *430*, 127287. https://doi.org/10.1016/j.amc.2022.127287

Almgren, R., & Chriss, N. (2001). Optimal Execution of Portfolio Transactions. *The Journal of Risk*, *3*(2), 5–39. https://doi.org/10.21314/jor.2001.041

Benth, F. E., & Süss, A. (2018). Cointegration in Continuous Time for Factor Models. *Mathematics and Financial Economics*, *13*(1), 87–114. https://doi.org/10.1007/s11579-018-0221-8

Bergault, P., Drissi, F., & Guéant, O. (2022). Multi-asset Optimal Execution and Statistical Arbitrage Strategies under Ornstein-Uhlenbeck Dynamics. *SIAM Journal on Financial Mathematics*, *13*(1), 353–390. https://doi.org/10.1137/21m1407756

Cartea, Á., Gan, L., & Jaimungal, S. (2018). Trading Co-integrated Assets with Price Impact. *Mathematical Finance*, *29*(2), 542–567. https://doi.org/10.1111/mafi.12181

Cartea, Á., Jaimungal, S., & Penalva, J. (2015). *Algorithmic and High-Frequency Trading*. Cambridge University Press.

Comte, F. (1999). Discrete and Continuous Time Cointegration. *Journal of Econometrics*, *88*(2), 207–226. https://doi.org/10.1016/s0304-4076(98)00025-6

Engle, R. F., & Granger, C. W. J. (1987). Co-Integration and Error Correction: Representation, Estimation, and Testing. *Econometrica*, *55*(2), 251. https://doi.org/10.2307/1913236

Guéant, O. (2016). *The Financial Mathematics of Market Liquidity: From Optimal Execution to Market Making*. Taylor & Francis Group.

Johansen, S. (1991). Estimation and Hypothesis Testing of Cointegration Vectors in Gaussian Vector Autoregressive Models. *Econometrica*, *59*(6), 1551. https://doi.org/10.2307/2938278

MacKinnon, J. (2010). *Critical Values For Cointegration Tests* (Working Paper No. 1227). Economics Department, Queen's University. https://EconPapers.repec.org/RePEc:qed:wpaper:1227

Meucci, A. (2009). Review of Statistical Arbitrage, Cointegration, and Multivariate Ornstein-Uhlenbeck. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.1404905

Pei, H. (2021, August). *Mean-Reverting Spread Modeling: Caveats in Calibrating the OU Process*. Retrieved December 3, 2022, from https://hudsonthames.org/caveats-in-calibrating-the-ou-process

Perlin, M., & Ramos, H. (2016). GetHFData: A R Package for Downloading and Aggregating High Frequency Trading Data from Bovespa. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.2824058

Rackauckas, C., & Nie, Q. (2017). DifferentialEquations.jl A Performant and Feature-Rich Ecosystem for Solving Differential Equations

in Julia. *Journal of Open Research Software*, *5*(1), 15. https://doi.org/10.5334/jors.151

Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations. *Journal of Computational Physics*, *378*, 686–707. https://doi.org/10.1016/j.jcp.2018.10.045

Sirignano, J., & Spiliopoulos, K. (2018). DGM: A Deep Learning Algorithm for Solving Partial Differential Equations. *Journal of Computational Physics*, *375*, 1339–1364. https://doi.org/10.1016/j.jcp.2018.08.029

Vatiwutipong, P., & Phewchean, N. (2019). Alternative Way to Derive the Distribution of the Multivariate Ornstein-Uhlenbeck Process. *Advances in Difference Equations*, *2019*(1). https://doi.org/10.1186/s13662-019-2214-1