Imperial College
London

Imperial College London

Department of Mathematics

Detecting and repairing arbitrage: from European to American options

Author: Jeroen NELIS (CID: 02304425)

A thesis submitted for the degree of

MSc in Mathematics and Finance, 2022-2023

Declaration

The work contained in this thesis is my own work unless otherwise stated.

Acknowledgements

I would like to thank Qube Research and Technologies for providing access to market data and the opportunity to work on this project. In particular, I want to thank Kaitai Chan for his guidance and all the helpful feedback he gave me throughout. Finally, I would like to express my gratitude to prof. Jack Jacquier for his insightful comments and support whenever needed.

Abstract

The use of option price data is widespread in finance, ranging from risk management applications to the development of trading strategies. The existence of arbitrage within this data can significantly impede the effectiveness or even result in the failure of these tasks. This results in a need for preprocessing to remove arbitrages. Historically, most research has been devoted to arbitrage-free smoothing and filtering (i.e. removal) of data. Recently a framework to repair quotes using linear programming has been developed. These type of methods result in sparse perturbations that keep most prices within its bid-ask bounds. Furthermore, they are shown to be fast when applied to real world large-scale problems and to improve model calibration.

A shortcoming of this repair method is that it only makes use of European call prices. This severely limits the range of strikes for which the volatility surface can be repaired, since calls at low strikes are not liquid enough to be a good representation of the implied volatility. To overcome this, we extend the method to include European put prices in a natural manner. It considers both call and put price data and weighs them on the basis of their bid-ask spread to arrive to an improved implied volatility estimate. The method is shown to be accurate on a significantly wider range of strikes, while retaining all the benefits of the original one.

The method is then further extended to include American options, which is the type most commonly found on single-name stocks. This is more challenging, since put-call parity does not hold and their arbitrage conditions cannot be written as a system of linear inequalities. These challenges are resolved by adding the *early exercise premium* as a variable in the linear program. Using the same principles as for the European case, a linear programming problem is constructed that this method leads to reasonable implied volatility surfaces as well.

Finally, applications of these repair methods are outlined. In this context, it is shown that the developed methods lead to a more robust calibration of the Merton jump-diffusion model. Additionally, it is shown that these methods can detect executable arbitrage, and some important details that were previously overlooked are clarified.

Contents

List of Figures

Introduction

The implied volatility surface is an essential tool used by traders, risk managers and other market participants. For a variety of applications it is imperative that the implied volatility surface be arbitrage-free. Until recently, there were two main methods used to construct arbitrage-free implied volatility surfaces: smoothing and filtering [12]. *Smoothing* methods use parametric interpolation to remove noise from options data and to extrapolate the surface to prices that are not (or not liquid enough) in the market. Examples of smoothing methods can be found in [1], [16], [19] and [27]. However, the main goal of smoothing is to produce a $C^{1,2}$ implied volatility function $(T, K) \mapsto \sigma(T, K)$, while it can be argued that obtaining arbitrage-free data is merely a by-product [12]. Additionally, for smoothing usually an l^2-norm penalization is used, resulting in a change of nearly all data.

The other class of methods are *filtering* methods, summarized in [23] and [29]. This filtering of data refers to simply removing data that is perceived to be of low-quality. This is typically done according to criteria in terms of moneyness, expiration, trading volume, etc. However, filtering is relatively subjective, causes information loss and is not always feasible (e.g. in OTC markets).

Recently, a new framework was developed in which the raw call price quotes are made arbitrage-free by solving a constrained linear program [12]. This *repair* method has an L^1-norm type penalty function which punishes deviations from the market prices. In addition, the marginal cost of deviating outside of the bid-ask spread is made higher than inside the spread. Moreover, the authors constructed the static no-arbitrage constraints as a matrix inequality, thereby ensuring the solution to be arbitrage-free. This solution is both sparse (not too many quotes are changed) and has most quotes within the bid-ask spread. The authors then demonstrate the usefulness of this method, primarily as a pre-processing tool for option price calibration or as a way to detect executable arbitrage.

This dissertation is structured as follows. In Chapter 1, we start with a summary of European and American option pricing theory. This treatment is descriptional and not fully rigorous, since that would take us too far. It includes an overview of the Black-Scholes model, risk-neutral pricing, Black's model and binomial tree pricing. Furthermore, we show how to derive a robust estimate of the forward price and discount factor from real market data. Next, we state the no-arbitrage conditions for European options and observe that it can be written as a matrix inequality, following [12]. We conclude with a deeper investigation of the implied volatility surface, its typical shape and the reasons that it is almost guaranteed to contain arbitrage when not pre-processed.

In Chapter 2, we investigate in detail the repair method L1BA, developed in [12]. We analyze its behavior on real market data, thus verifying that the method works. However, we spot one big problem with the method: since it only uses call price data it cannot be applied for a wide range of strikes. This is because calls with small strikes are rarely traded, resulting in low liquidity, a big bid-ask spread and thus inaccurate price data. Usually, implied volatility at low strikes is calculated with puts, while calls are used only for high strikes. It is therefore an obvious idea to add put prices to the framework, but we must be careful to ensure that put-call parity holds. We develop a method called L1BA-PC that is a straightforward extension of the L1BA method, includes put prices and preserves put-call parity. The properties of this method are then discussed and an intuitive explanation is provided. Finally, we compare the two methods and find that the extended method L1BA-PC produces meaningful results over a wider range of strikes. This is due to the much higher accuracy for low strikes while maintaining the same accuracy for high strikes.

In Chapter 3 we make an attempt to extend this framework to American options. This is much more challenging because put-call parity doesn't necessarily hold and there is no system of linear inequalities describing the no-arbitrage constraints. The first repair method we develop is straightforward: use the pseudo-European option quotes as input for the previously developed L1BA-PC method. This method is quite problematic and does not give satisfactory results mainly because of the noisiness of the early exercise premium estimates. They key insight to solve this problem is to add the early exercise premium inside the optimization framework. This allows us to implicitly convert American option quotes to corresponding (pseudo-)European option quotes, for which put-call parity and the no-arbitrage constraints are valid. In addition, we can take advantage of the fact that the early exercise premiums should be monotonic as a function of strike. By adding the put-call parity constraint and penalties that discourage deviations from early exercise premium estimates, we arrive at the L1BAA-OPT-PC method. We compare our methods on real market data, including a complex case where a significant dividend is about to be paid. In the end, we can conclude that the L1BAA-OPT-PC method also gives reasonable results.

Finally, in Chapter 4, we discuss some applications in which repair methods are useful: enabling more robust model calibration and the detection of executable arbitrage. In [12] it is shown that pre-processing option price data to make it arbitrage-tree results in less variation in the parameters of a calibrated Heston model. We test this with the Merton jump-diffusion model and our own L1BA-PC method. We reach the same conclusions, and show that adding put prices (as L1BA-PC does implicitly) further improves model calibration. Hereafter, we investigate how repair methods can be used to detect executable arbitrage (where we buy at ask and sell at bid). But, we observe that in order to guarantee this, the parameter δ_0 needs to be chosen infinitesimally small, a fact that was overlooked in [12]. We show with real market data that otherwise the repair method might result in false positives and propose a value of δ_0 which significantly reduces this.

Chapter 1

Option pricing theory

In this chapter we will summarize the basics of option pricing theory, more specifically for European and American vanilla options. This summary is by no means exhaustive and contains only the theory necessary to understand the rest of this thesis. For a more comprehensive treatment, we refer to [34] and [35]. For an introduction to measure-theoretic probability theory, see [39].

We first start with the absolute basics of options, mostly to establish notation. Hereafter, we look at the pricing of European options in the Black-Scholes model. In the next section, the pricing of American options is outlined and its framework is compared with the European equivalent. In Section 1.4, no-arbitrage conditions for calls are derived. Finally, we investigate the implied volatility surface, consider why there is often arbitrage present and summarize methods to overcome this.

1.1 Option basics

We start with considering a stock S, with price S_t at time t. A (European) *call* option on S, with *strike* K and *maturity* T, is a contract that gives the owner the right (but not the obligation) to buy S at time T for a price K. A (European) *put* option on S, with *strike* K and *maturity* T, is a contract that gives the owner the right to sell S at time T for a price K. Since the value of both contracts clearly depends on the underlying S, these European options are considered *derivative contracts*.

Since the owner of a call can buy the underlying at time T for a price of K, if the stock price at that time is greater than K, the owner can easily draw in a profit of $S_T - K$ by exercising the call (i.e. buying the stock for a price of K) and immediately selling it in the market for S_T. However, since the owner of the call has the right but not the obligation to exercise, if the price at expiration is less than K, they will simply not exercise it. This means that the payout of a call option is $(S_T - K)^+ := \max(S_T - K, 0)$. Similarly, the payout of a put option is $(K - S_T)^+ := \max(K - S_T, 0)$. This implies that a holder of a call option profits when the stock prices moves up, while the holder of a put option wants the stock price to decrease.

The value of options can be separated in two parts: the *intrinsic value* is the value of the option if it where exercised right now (e.g. $(S_t - K)^+$ for a call at time t). The intrinsic value conveys how *deep in the money* the option is. Options for which intrinsic value is (significantly) positive are called *in the money* (ITM), options with strike close to spot are called *at the money* (ATM), while options that are not close to being in the money are called *out of the money* (OTM). It follows that for low strikes calls are ITM and puts are OTM, for strikes around spot both calls and puts are ATM, and for high strikes calls are OTM and puts ITM.

The *extrinsic value* (or *time value*) is the remaining portion of an options value that is not accounted for by its intrinsic value. It is affected by factors such as time to expiration, market volatility, interest rates, and other market dynamics. Extrinsic value is generally positive, to account for the fact that the option might gain in value in the future, but is negative in some special cases.

1.2 European option pricing

One of the earliest accounts about options is that of the ancient Greek mathematician and philosopher Thales, who bought something similar to a call option to profit from a larger than usual olive harvest [33]. In London, they have been traded from at least the 1690s, mainly as insurance products [36]. In 1973 the famous the Black-Scholes model was developed [7], [30]. Using a no-arbitrage argument, they derived a closed form solution to option prices as a function of the underlying asset's price, the option's strike price, time to expiration, volatility of the underlying asset, and risk-free interest rate. Before the Black-Scholes model, there was no widely accepted method for valuing options, which led to inconsistencies and pricing inefficiencies in the options market. However, others argue that very similar methods were already used before and that the model resembles an economic argument rather than an option pricing formula [20]. However, the model's elegant solution helped standardize and rationalize options pricing, which was a major advancement in financial theory. Nowadays, it is not really used for option pricing, but rather for mapping market option prices to a single real number, implied volatility. Instead, extensions and variations of the original model have been developed to better capture real market dynamics such as dividends, transaction costs, non-constant volatility and jumps [24].

1.2.1 Black-Scholes model

From looking at historical data it is abundantly clear that stock prices don't move in a deterministic way, but rather in a stochastic manner. This is a consequence of the complex interplay of various factors and participants in the financial markets that move stock prices: information flow, heterogeneity of market participants, macro- and micro-economic factors and psychological factors such as market sentiment and herd behaviour. It therefore makes sense to model stock prices as a stochastic process in some probability space. The following introduction to the Black-Scholes model is mainly based on [8].

Let's define the underlying stock price process $(S_t)_{t\geq 0}$ on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$, on which $(W_t)_{t\geq 0}$ is a standard Wiener process. In the *Black-Scholes* model, we model the stock price as a geometric Brownian motion [7]:

$$dS_t = \mu S_t dt + \sigma S_t dW_t, \tag{1.2.1}$$

where μ is the expected rate of return for the stock and σ is its volatility, which is assumed to be constant. The idea of this model is that in some small time step Δt, the stock price return $\frac{S_{t+\Delta t}-S_t}{S_t}$ is normally distributed with mean $\mu\Delta t$ and standard deviation $\sigma\sqrt{\Delta t}$.

Next to this asset containing risk, we consider a riskless asset, the bond B_t whose price process moves deterministic according to:

$$dB_t = B_t r dt, \quad B_0 = 1. \tag{1.2.2}$$

where r is the risk-free rate. Finally, in this model one considers the *Black-Scholes ideal conditions*:

- There are no transaction costs in trading the stock.
- The stock pays no dividends.
- Shares are infinitely divisible.
- Short selling is allowed without any restriction or penalty.

Next, we consider a pair of stochastic processes $\phi = (\phi^B, \phi^S)$ on $(\Omega, \mathcal{F}, \mathbb{P})$, called the *trading strategy*. The idea is that these processes describe the amount of bonds and stocks to be held at time t. This is mathematically represented through the *value process* V, which describes the value of the portfolio constructed by following the strategy ϕ:

$$V_t(\phi) = \phi_t^B B_t + \phi_t^S S_t. \tag{1.2.3}$$

It is clear that in practice a trading strategy can only be dependent on information that is known at time t. To translate this mathematically, we consider a filtration $(\mathcal{F}_t)_{t\geq 0} \subset \mathcal{F}$ which represents all known information on (S_t) at time t, and require $(\phi_t)_{t\geq 0}$ to be $\mathcal{F}_t$-adapted.

In this context, we are most interested in *self-financing* strategies. The main idea of such a strategy ϕ is that the changes in value of the portfolio described by ϕ is only due to gains/losses coming from price movements (in B and S), without any cash inflow or outflow. This can be expressed in differential terms as:

$$d(\phi_t^B B_t + \phi_t^S S_t) = \phi_t^B dB_t + \phi_t^S dS_t \tag{1.2.4}$$

Finally, to construct the option pricing problem in the context of no-arbitrage, we need a few extra concepts.

Definition 1.2.1. A *contingent claim* Y for the maturity T is any-square integrable and positive random variable in $(\Omega, \mathcal{F}_T, \mathbb{P})$, which is in particular $\mathcal{F}_T$-measurable.

The idea behind a claim is that it represents an amount that will be paid at maturity to the holder of the contract. The European options described earlier are clearly examples of contingent claims.

Definition 1.2.2. A contingent claim Y is *attainable* if there exists a self-financing strategy ϕ such that $V_T(\phi) = Y$.

Definition 1.2.3. An *arbitrage strategy* is a self-financing strategy ϕ such that:

$$\begin{gathered} \phi_0^B B_0 + \phi_0^S S_0 = 0, \\ \text{but } \mathbb{P}(\phi_T B_T + \phi_S S_T > 0) > 0, \\ \text{and } \mathbb{P}(\phi_T B_T + \phi_S S_T \geq 0) = 1. \end{gathered}$$

In other words, an arbitrage strategy is a strategy which creates a positive cash inflow, starting from nothing, with positive probability and never creates a loss. One of the core ideas of financial markets on which basically all pricing methods are built is that there is no arbitrage, since when an arbitrage opportunity would exist, it is immediately exploited by fast and specialized market participants, after which the arbitrage opportunity is no longer there.

Moreover, if we for now assume that European options are attainable contingent claims, to avoid arbitrage opportunities, the corresponding initial portfolio value $V_0(\phi_r)$ of a self-financing strategy ϕ_r (also known as a *replicating strategy*) must be equal to the initial price of the claim to avoid arbitrage opportunities (this is usually called the *Law of one price*). This is the main idea of the Black-Scholes model: we create a (dynamically) replicating self-financing strategy using only the underlying and a risk-free asset. This portfolio is guaranteed to match the options' payoff at maturity, and using no-arbitrage arguments it follows that the initial price of the option should be equal to the initial price of the replicating portfolio.

To construct this self-financing strategy, let $C(t, S_t)$ be the option value at time t. We now assume for the function $C(t, S_t)$ to have regularity: $C \in C^{1,2}([0,T] \times \mathbb{R}^+)$, so we can apply Ito's Lemma [39]:

$$\begin{aligned} dC(t, S_t) &= \frac{\partial C(t, S_t)}{\partial t} dt + \frac{\partial C(t, S_t)}{\partial S} dS_t + \frac{1}{2} \frac{\partial^2 C(t, S_t)}{\partial S^2} dS_t dS_t && (1.2.5) \\ &= \left(\frac{\partial C}{\partial t}(t, S_t) + \frac{\partial C}{\partial S}(t, S_t) \mu S_t + \frac{1}{2} \frac{\partial^2 C}{\partial S^2}(t, S_t) \sigma^2 S_t^2 \right) dt + \frac{\partial C}{\partial S}(t, S_t) \sigma S_t dW_t && (1.2.6) \end{aligned}$$

Now construct the trading strategy $(\phi)_t = (\phi_t^B, \phi_t^S)$ defined as:

$$\phi_t^S = \frac{\partial C}{\partial S}(t, S_t), \quad \phi_t^B = (V_t - \phi_t^S S_t)/B_t. \tag{1.2.7}$$

It is immediately clear that this trading strategy is replicating, since $V_t(\phi_t) = C_t$. If we now require ϕ to be self-financing, we obtain:

$$\begin{aligned} dV_t &= \phi_t^B dB_t + \phi_t^S dS_t && (1.2.8) \\ &= \left[V(t, S_t) - \frac{\partial C}{\partial S}(t, S_t) S_t \right] r dt + \frac{\partial C}{\partial S}(t, S_t) S_t (\mu dt + \sigma dW_t) && (1.2.9) \end{aligned}$$

If we now equate (1.2.6) and (1.2.9), while taking into account that $V_t = C_t$, we obtain the famous Black-Scholes PDE:

$$\frac{\partial V}{\partial t}(t, S_t) + \frac{\partial V}{\partial S}(t, S_t) r S_t + \frac{1}{2}\frac{\partial^2 V}{\partial S^2}(t, S_t)\sigma^2 S_t^2 = rV(t, S_t), \tag{1.2.10}$$

with terminal condition $V_T = (S_T - K)^+$. In the case of a European put, this terminal condition becomes $V_T = (K - S_T)^+$.

We also note that the trading strategy is made a replicating one, by choosing ϕ_t^S in that way that the stochastic component of V_t corresponds exactly to the stochastic component of C, namely $\frac{\partial C}{\partial S}(t, S_t)\sigma S_t dW_t$. However, in doing so, it also replicates a part of the deterministic part of C, namely $\frac{\partial C}{\partial S}(t, S_t)\mu S_t$. In the end, this results in the drift term μ not appearing in the Black-Scholes PDE (1.2.10), which means that the option price is not dependent on the drift of the stock price process. This is the basic principle of *risk-neutral pricing*, which we will now investigate.

1.2.2 Risk neutral pricing

A way to obtain the solution of the Black-Scholes PDE (1.2.10) is by using the Feynman-Kac Theorem. This theorem allows to interpret the solutions of a parabolic PDE as the expected value of a diffusion process.

Theorem 1.2.4. *Feynman-Kac: Given suitable regularity and integrability conditions, the solution of the PDE*

$$\frac{\partial V}{\partial t}(t, x) + \frac{\partial V}{\partial x}(t, x) b(x) + \frac{1}{2}\frac{\partial^2 V}{\partial x^2}(t, x)\sigma^2(x) = rV(t, x), \quad V(T, x) = f(x),$$

can be expressed as

$$V(t, x) = e^{-r(T-t)}\mathbb{E}^{\mathbb{Q}}[f(X_T) \mid \mathcal{F}_t]$$

where the probability measure $\mathbb{Q}$ *is taken such that* X *is an Ito process driven by*

$$dX_s = b(X_s)ds + \sigma(X_s)dW_s^{\mathbb{Q}}, \quad s \geq t, \quad X_t = x,$$

and $(W_t^{\mathbb{Q}})_{t\geq 0}$ *is a standard Wiener process under* $\mathbb{Q}$.

By substituting $b(x) = rx, \sigma(x) = \sigma x$, we let the Black-Scholes PDE coincide with the one in the Feynman-Kac theorem. This theorem then implies that the value of a payout $g(\cdot)$ at time t can be expressed as

$$V(t) = \mathbb{E}^{\mathbb{Q}}[e^{-r(T-t)}g(S_T) \mid \mathcal{F}_t], \tag{1.2.11}$$

where the expectation is taken with respect to the so-called martingale measure $\mathbb{Q}$, a probability measure $\mathbb{Q} \sim \mathbb{P}$ under which the discounted stock price $S_t/B_t = e^{-rt}S_t$ is a martingale. This is equivalent to S having drift rate r under $\mathbb{Q}$:

$$dS_t = rS_t dt + \sigma S_t dW_t^{\mathbb{Q}}. \tag{1.2.12}$$

Using *Girsanov's theorem* [39] an explicit conversion from the *physical measure* $\mathbb{P}$ to the *pricing measure* $\mathbb{Q}$ can be obtained:

$$\frac{d\mathbb{Q}}{d\mathbb{P}} = \exp\left(-\frac{1}{2}\left(\frac{\mu - r}{\sigma}\right)^2 T - \frac{\mu - r}{\sigma}W_T\right). \tag{1.2.13}$$

The main conclusion follows from Equations (1.2.11) and (1.2.12): we can view the option price as the expected value of the discounted payoff of the option. However, this expectation is taken with respect to the pricing measure $\mathbb{Q}$, under which the stock price process has risk-neutral drift, and not with respect to the physical measure $\mathbb{P}$. Again, we observe that the drift term μ doesn't influence the option price. This indicates that investors, though having different risk preferences or predictions about the future stock price behaviour, must agree on the option price. Equivalently, one can state that the measure $\mathbb{Q}$ defines the *risk-neutral world*, in which the expected rate of return on all securities is the risk-free interest rate r, implying that investors do not require any risk premium for trading stocks.

1.2.3 Black-Scholes formula

Since the solution of (1.2.12) is:

$$S_t = S_0 \exp\left((r - \frac{1}{2}\sigma^2)t + \sigma W_t^{\mathbb{Q}}\right), \tag{1.2.14}$$

we obtain that S_T is log-normally distributed under the risk-neutral measure. Plugging this in into (1.2.11), one obtains the famous Black-Scholes formula:

$$C_0 = S_0\mathcal{N}(d_1) - Ke^{-rT}\mathcal{N}(d_2), \tag{1.2.15}$$

$$P_0 = Ke^{-rT}\mathcal{N}(-d_2) - S_0\mathcal{N}(-d_1), \tag{1.2.16}$$

where

$$\begin{aligned} d_1 &= \frac{\log(S_0/K) + (r + \frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}} \\ d_2 &= d_1 - \sigma\sqrt{T} \end{aligned}$$

where $\mathcal{N}$ is the normal cumulative distribution function. This formulation is widely regarded as one of the main results in option pricing and financial mathematics in general. It was the first widely accepted mathematical model to determine the fair market value of options, which were relatively new and poorly understood financial instruments at the time.

1.2.4 Extensions of the Black-Scholes model

The Black-Scholes model is arguably the most basic model in option prices, and relies on some strong assumptions that are clearly not realistic. We here present an extension to the original model called Black's model. It allows us to handle non-constant risk-free interest rates and discrete dividend payments.

Stochastic interest rates

In practice, there is not such a thing as a constant risk-free rate. The risk-free asset changes with an interest rate that changes over time (i.e. is stochastic):

$$dB_t = B_t r_t dt, \quad B_0 = 1. \tag{1.2.17}$$

This gives a solution $B_t = \exp(\int_0^t r_s ds)$, which is stochastic since it is not $\mathcal{F}_0$ measurable. The bank account B_t is the value of risk-free money at time t. When considering payouts at time t, it makes sense to discount them by comparing them to the value of the bank account. In other words, we multiply by the *stochastic discount factor* $1/B_t$ to discount future payouts. From Feynman-Kac, one can derive that the value of a future derivative payout $g(\cdot)$ becomes:

$$V(t) = \mathbb{E}^{\mathbb{Q}}[\frac{B_t}{B_T} g(S_T) \mid \mathcal{F}_t], \tag{1.2.18}$$

or in other words, $(B_t)_{t\geq 0}$ is the *numeraire* corresponding to the risk-neutral measure $\mathbb{Q}$ which makes the stock price a martingale. Additionally, we consider the *(risk-free) T–maturity zero–coupon bond* $P_{0,T}$, a contract which guarantees the payment of one unit of currency at time T. From (1.2.18) it follows that:

$$\begin{aligned} P_{0,T} &= \mathbb{E}^{\mathbb{Q}}[\frac{1}{B_T}.1], \\ &= \mathbb{E}^{\mathbb{Q}}[\exp\left(-\int_0^t r_s ds\right)]. \end{aligned}$$

This price $P_{0,T}$ is known from the market's yield curve, it contains the market's expectation of the risk-neutral rate in the future, see Section 1.2.5. To avoid confusion with the notation for a put, we will write it as the *discount factor* $D_T := P_{0,T}$.

Forwards and put-call parity

A forward contract is a derivative contract in which two parties agree to buy or sell an asset at a specified price F on a future date T. At time T it has a payoff $S_T - F$, where S_T is the asset price at time T. In general, when one refers to the *forward* or *forward price*, we mean the pre-agreed price F which makes this contract fair (i.e. without money needing to be exchanged now). Since at the conception of the forward no money is exchanged, for the contract to be fair, the expected value of the discounted payout under the risk-neutral measure should be zero. This leads to:

$$\begin{aligned}
&E^{\mathbb{Q}}[\frac{1}{B_T}(S_T - F)] = 0, \\
\iff &E^{\mathbb{Q}}[\frac{S_T}{B_T}] = FE^{\mathbb{Q}}[\frac{1}{B_T}], \\
\iff &F = \frac{E^{\mathbb{Q}}[S_T/B_T]}{E^{\mathbb{Q}}[1/B_T]}, \\
\iff &F = \frac{S_0}{D_T},
\end{aligned}$$

where we used the definition of the discount factor and the fact that S_T/B_T is a martingale under the risk-neutral measure $\mathbb{Q}$.

A very important property of European options is that a call can be converted into a put and vice versa.

Theorem 1.2.5. *Put-call parity: For European options, the following equality holds:*

$$C_0 - P_0 = D_T(F_T - K), \tag{1.2.19}$$

where C_0 and P_0 are the price of a call and put with strike K and expiration T, DF_T is the corresponding discount factor and F_T is the corresponding forward price.

Proof. We start by looking at the payout of the left hand side in (1.2.19) at expiration T:

$$\begin{aligned}
C_T - P_T &= (S_T - K)^+ - (K - S_T)^+, \\
&= S_T - K.
\end{aligned}$$

In other words, by buying a call and selling a put with the same expiration and strike, we create a *synthetic long position.* Multiplying both sides by $1/B_T$ and taking the risk-neutral expectation:

$$\begin{aligned}
E^{\mathbb{Q}}[C_T/B_T] - E^{\mathbb{Q}}[P_T/B_T] &= E^{\mathbb{Q}}[S_T/B_T] - E^{\mathbb{Q}}[K/B_T], \\
\iff C_0 - P_0 &= E^{\mathbb{Q}}[S_T/B_T] - KE^{\mathbb{Q}}[1/B_T], \\
\iff C_0 - P_0 &= D_T(F_T - K),
\end{aligned}$$

where we used the definitions of the forward and the discount factor. □

Handling income streams

The majority of assets provide an income stream for investors that hold the asset, typically in the form of dividend payments. These payments influence the price of the underlying in the future and therefore also the price of a forward contract. This is because an investor who buys the forward and hedges by selling the stock, needs to be compensated for missing the income that comes with owning the stock. For a forward at time T, we define the income streams as $(I_k)_{k=1}^N$, paying at time $(t_k)_{k=1}^N$, where $t_k \in [0, T]$ for $k = 1, ..., N$. From no-arbitrage theory it then follows that:

$$D_T F_T + \sum_{k=1}^{N} D_{t_k} I_k = S_0 \tag{1.2.20}$$

$$\iff F_T = \frac{S_0 - \sum_{k=1}^N D_{t_k} I_k}{D_T}. \tag{1.2.21}$$

We can also consider the case where the asset provides a continuous income stream with a known yield q rather than a known cash payment I. Mathematically this means that a when one buys S_0 amount of a stock at time 0, it will have accrued to $S_t e^{qt}$ at time t. This corresponds to a forward price:

$$F_T = \frac{S_0 e^{-qT}}{D_T}, \tag{1.2.22}$$

which is equal to $S_0 e^{(r-q)T}$ in the case of a constant risk-free interest rate r.

Black's model

In the case of non-constant interest rates and discrete dividend payments, the relationship between the spot price and the forward price is not as clear. The main insight of Black was to generalize the Black-Scholes formula (1.2.16) to *Black's formula* [6], which describes options on futures. It can be seen as an extension of Black-Scholes, where we use the discounted forward price instead of the spot price. The forward prices implicitly takes into account interest rates, dividend payments, borrowing costs,... This leads to:

$$C_0 = D_T(F_T\mathcal{N}(d_1) - K\mathcal{N}(d_2)), \tag{1.2.23}$$
$$P_0 = D_T(K\mathcal{N}(-d_2) - F_T\mathcal{N}(-d_1)), \tag{1.2.24}$$

where

$$d_1 = \frac{\log(F_T/K) + \frac{1}{2}\sigma^2 T}{\sigma\sqrt{T}}$$
$$d_2 = d_1 - \sigma\sqrt{T}$$

and with $\mathcal{N}$ the normal cumulative distribution function. It is this formula that we will use to calculate implied volatilities of European options. However, in order to do so, we first need an estimate of the forward price and discount rate.

1.2.5 Estimating forward and discount factor

As an example, we fetched quotes of the SPX index and its full option chain (all available expiries and all strikes for each expiration) on 2023/07/13. We choose SPX because it is very liquid and there are therefore a lot of option quotes available. Moreover, the options are (like all index options) of the European type. We then proceeded by filtering out quotes that contain missing values or don't have any volume in either bid or ask.

The main goal of this thesis is to construct an implied volatility surface from this raw option price data without being dependent on other data sources. The method we propose achieves this. However, one of the main problems one encounters in practice, is that most market parameters are not unambiguously defined (there isn't *one* price, *one* discount factor, *one* forward price, etc.). In the market we often find multiple possible values of these parameters, each of which gives slightly different results. We will shortly explain how we unambiguously fetch the discount factor and the forward price from the market.

Discount factor

As explained earlier, the discount factor D_T is the expectation of the stochastic discount factor $1/B_T$ under the risk-neutral measure. It encompasses the markets view of the future movement of risk-free interest rates and is used to discount future cashflows to the present. However, it is important to understand that the risk-free interest rate is a purely theoretical concept (it represents the return on an investment with zero risk of default), while in practice no investment is totally risk-free. The risk-free rate is therefore often approximated using financial instruments that have very low levels of risk such as interbank lending rates. Since for SPX we have USD as the underlying currency, we choose the Secured Overnight Financing Rate (SOFR) curve. For an index with EUR as the underlying currency we would take the Euro Short-Term Rate (ESTR).

Forwards

Earlier we defined the forward price F_T as the price for which delivery of the asset at time T is fair, given the information that is available now. This includes incoming cashflows such as dividend payments and takes into account the time value of money using the discount factor D_T, see (1.2.21). Sometimes, forward contracts are traded directly in the market, from which an accurate forward price can be derived. This is unfortunately often not the case, which means that an estimate of the forward price needs to be made. The accuracy of this estimate is important to calculate implied volatilities using Black's formula (1.2.24).

The main idea to estimate forward prices is that they can be inferred from European option prices using put-call parity (1.2.19). However, for a chosen expiration T, the implied forward price will typically be (slightly) different for each strike K:

$$F_{T,K} := \frac{C_K - P_K}{D_T} + K. \tag{1.2.25}$$

This can be seen in Figure 1.1a. There are now multiple ways to derive a single forward price estimate $\hat{F}_T$ given all implied forward prices $\{F_{T,K}\}_{K \in \mathcal{K}}$.

The simplest and most straightforward method is to use the forward estimate corresponding to the strike which is closest to ATM (the *ATM strike* K_{ATM}). We then have:

$$\hat{F}_{T,K} = \frac{C_{K_{ATM}} - P_{K_{ATM}}}{D_T} + K_{ATM}. \tag{1.2.26}$$

This is shown in Figure 1.1a. The main idea is that options that are ATM are usually the most liquid, which means that their prices should be quite accurate and therefore the implied forward at that strike should be accurate as well. This method is very simple, but uses only information at one strike, while ignoring information available from the other strikes.

An arguably better method that includes more strikes is to fit a linear regression to the put-call parity equation:

$$\underbrace{C_{K^i} - P_{K^i}}_{:=Y^i} = D_T F_T - D_T \underbrace{K^i}_{:=X^i}. \tag{1.2.27}$$

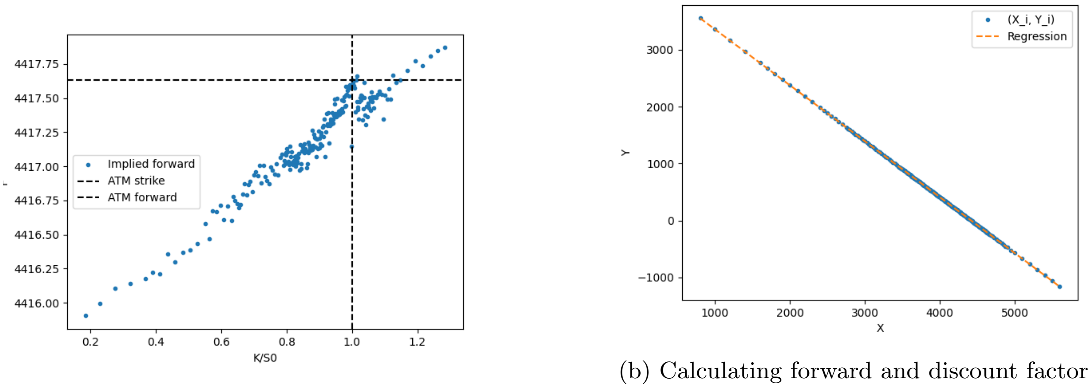
In other words, we fit a linear regression $Y^i = a + bX^i$ through the points $(X^i, Y^i) := (K^i, C_{K^i} - P_{K^i})$ and derive

$$\begin{cases} \hat{D}_T &= -b, \\ \hat{F}_T &= -a/b. \end{cases} \tag{1.2.28}$$

An example of this regression is shown in Figure 1.1b. In practice, we can opt to only use points with strikes that are relatively close to the money, since they are the most liquid and therefore the most accurate. Compared to the ATM forward, this forward should be more accurate, since it includes information from multiple strikes. Another big advantage is that it also immediately provides an estimate of the discount factor $\hat{D}_T$. This value can be used when there is no market price available, or to assess the accuracy of the regression by comparing $\hat{D}_T$ to the price in the market.

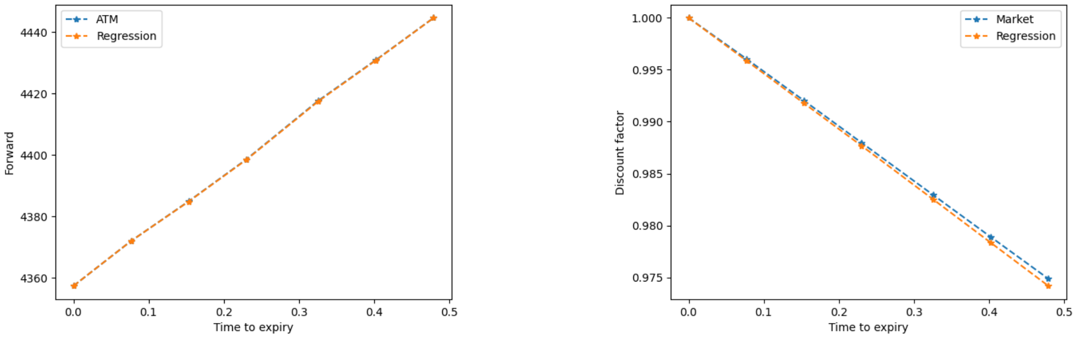
Comparison

We can now compare the different forward and discount factor calculation methods. When plotting the obtained forward as a function of time to expiration, we see on Figure 1.2a that both methods produce very similar results. From Figure 1.2b we notice that the discount factor using regression is typically also close to the one observed in the market. From these results one could argue to either use the ATM forward and the discount factor from the market (least amount of calculations) or using the forward and discount factor obtained from the linear regression (less market input needed). However all methods will generate very similar results, so we argue that the exact choice isn't too important.



(a) The ATM forward

(b) Calculating forward and discount factor using linear regression



(a) Forward as function of expiration

(b) Discount factor as function of expiration

1.3 American option pricing

American options are options that can be exercised at any time before expiration. This simple feature makes them considerably more complicated to price or hedge, since one must take into account all different possible exercise policies [10]. Again, we will consider the pricing of American options in the Black-Scholes setting: the underlying stock price process $(S_t)_{t\geq 0}$ is defined on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$, on which $(W_t)_{t\geq 0}$ is a standard Wiener process. The stock price is hereby modelled as a geometric Brownian motion as in (1.2.1), but to price options, we use the unique pricing measure $\mathbb{Q}$ under which the discounted stock price process $(D_t S_t)_{t\geq 0}$ is a martingale [39].

It is easy to show that for a non-dividend paying stock and with positive interest rates, it is never optimal to exercise an American call option before its expiration date.

Theorem 1.3.1. *With positive interest rates, and with an underlying that doesn't pay it is never optimal to exercise an American call option before expiration.*

Proof. This can be observed from the following inequality:

$$\begin{aligned} C^{AM} &\geq C^{EU} \\ &\geq D_T(F-K) \\ &= S_0 - D_T K \\ &> S_0 - K. \end{aligned}$$

We used the fact that American options are worth more than European ones due to optionality (see Section 3.1), put-call parity for European options (and non-negativity of puts), the fact that the stock doesn't pay dividends so $S_0 = DF$ and $D_T < 1$ since interest rates are positive. This inequality shows that the value of an American call is always greater than its intrinsic value (the amount of money one gets when exercising now). □

This is intuitive in the context of *opportunity cost*: when you hold an American call, you have the right to buy the underlying stock at the strike price at any time until expiration. By early exercising the option, you are essentially giving up the remaining *time value.* This time value represents the potential profit that could be earned by holding the option and waiting for a favorable price movement in the underlying stock. By exercising early, you forfeit this opportunity to capture potential future gains. Additionally, you have the possibility to invest the cash you would use to buy the stock if you exercised early. By waiting until expiration to exercise, you can earn interest on this cash, which can contribute to your overall returns.

However, the same doesn't apply for American puts: since a put can never be worth more than its strike price K, it might make sense to exercise when the stock price is low. For this reason we will now focus on the American put pricing problem, but the techniques shown extend to any convex payoff structure that is equipped with an early exercise feature [17].

1.3.1 Optimal stopping problem

A first way to characterize the American put pricing problem is as an optimal stopping problem in terms of the time at which it is exercised (called the *exercise time*). The exercise time τ should be a *stopping time* with respect to $(\mathcal{F}_t)$, that is a random time such that the event $\{\tau < t\}$ belongs to $\mathcal{F}_t$ for any $t \leq T$. Using *optimal stopping* theory, it can then be shown that the no-arbitrage price of an American put is obtained by maximizing, over all stopping times, the expected value of the discounted payoff under the risk-neutral measure [17]:

$$P(t,s) = \sup_{t \leq \tau \leq T} \mathbb{E}^{\mathbb{Q}}[e^{-r(\tau-t)}(K - S_\tau)^+ \mid \mathcal{F}_t], \quad S_t = s \tag{1.3.1}$$

where $P(t,s)$ is the price of the put at time t and current stock price s. Another way to state this is that the value of an American option can be represented by the *Snell envelope*, see [4]. When comparing this to the formula for European options (1.2.11), we see that for American options the supremum is taken over all possible exercising strategies, while for European options the option is always exercised at maturity.

The *optimal stopping time* $\tau^*(t)$ is now defined as the stopping time where the supremum is reached:

$$\tau^*(t) = \inf\{t \leq u \leq T, P(u, S_u) = h(S_u)\}, \tag{1.3.2}$$

or in other words the first time when the price is equal to its payoff. To determine $\tau^*(t)$ directly, one needs to obtain the price process (S_t) first. This is obviously much harder to solve because of the uncertainty of the stock price process.

1.3.2 PDI formulation

Similar to the way that European derivatives must satisfy the Black-Scholes partial differential equation (1.2.10), one can construct a system of partial differential *inequalities* that the pricing function of American derivatives need to satisfy. In [17] it is shown that the price of an American put $P(t,s)$ is the solution of the system:

$$\begin{aligned}
\frac{\partial P}{\partial t} + \frac{1}{2}\sigma^2 s^2 \frac{\partial^2 P}{\partial s^2} + rs\frac{\partial P}{\partial s} - rP \leq 0,& \\
P \geq (K-s)^+,& \\
\left(\frac{\partial P}{\partial t} + \frac{1}{2}\sigma^2 s^2 \frac{\partial^2 P}{\partial s^2} + rs\frac{\partial P}{\partial s} - rP\right)\left((K-s)^+ - P\right) = 0,&
\end{aligned} \tag{1.3.3}$$

to be solved in $\{(t,s) : 0 \leq t \leq T, s > 0\}$ with the final condition $P(t,s) = (K-s)^+$.

The main difference with European options is that there is no analytical solution for the equations (1.3.3). There are however multiple approximate numerical methods to price American options such as tree pricing [13], finite difference methods [40], least-squares Monte Carlo [28], and analytical approximations [3], [5]. We will go deeper into the tree pricing method, as it's the main method we will later use. This is because the method is quite intuitive and can handle early exercise and discrete dividend payments.

1.3.3 Tree pricing

The main idea of tree pricing methods is to evaluate the expectation in (1.3.1) by approximating the continuous process (1.2.12) as a discrete one [26] [32]. In order to do so, we fix an integer $N \geq 1$ and let $(\xi_n)_{n=1}^N$ be a sequence of i.i.d. random variables on $(\Omega, \mathcal{F}, \mathbb{Q})$. The discrete stock price process $(S_n)_{n=1}^N$ is then defined by:

$$S_n = S_{n-1}\xi_n \qquad \Longleftrightarrow \qquad S_n = S_0 \prod_{k=1}^{n} \xi_k, \quad n = 1, \ldots, N. \tag{1.3.4}$$

Furthermore, we let $\mathcal{F}_n = \sigma(S_0, S_1, \ldots, S_n)$, which means that $\mathcal{F}_n$ contains all information up until time n.

The simplest example is a recombining binomial tree, where each ξ_i is a binary random variable:

$$\xi_k := \begin{cases} u, & \text{with probability q} \\ d, & \text{with probability 1-q} \end{cases}$$

This way, at each fixed time step n, the random variable S_n takes value on the set $\{S_0u^n, S_0u^{n-1}d, \ldots, S_0d^n\}$. If we define $s_k^n := S_0u^{n-k}d^k$, then:

$$\mathbb{Q}(S_n = s_k^n) = \binom{n}{k} q^{n-k}(1-q)^k, \tag{1.3.5}$$

since each ξ_k is a Bernoulli random variable, and the sum of Bernoulli random variables is a binomial random variable.

In practice, we set $\delta_t := T/N$, so that the binomial tree covers the lifetime of the option. From (1.2.12) if follows that:

$$\frac{S_{t+\Delta t}}{S_t} \sim \text{log-normal}\left((r - \frac{\sigma^2}{2})\Delta t, \sigma^2 \Delta t\right). \tag{1.3.6}$$

This implies that the first two moments of the stochastic return are:

$$\mathbb{E}^{\mathbb{Q}}\Big[\frac{S_{t+\Delta t}}{S_t}\Big] = e^{r\Delta t}, \tag{1.3.7}$$

$$\mathbb{E}^{\mathbb{Q}}\Big[\Big(\frac{S_{t+\Delta t}}{S_t}\Big)^2\Big] = e^{(2r+\sigma^2)\Delta t}. \tag{1.3.8}$$

If we force the discrete process $(S_n)_{n=1}^N$ to match the risk-neutral dynamics in (1.2.12), we obtain:

$$\begin{cases} qu + (1-q)d = e^{r\Delta t}, \\ qu^2 + (1-q)d^2 = e^{(2r+\sigma^2)\Delta t}. \end{cases}$$

Since there are 2 equations for the 3 parameters (q, u, d), there are infinitely many parameter choices possible. One of the most popular ones is the Cox-Ross-Rubinstein (CRR) [13] specification in which we add the extra constraint $ud = 1$. This system of equations has solutions:

$$q = \frac{e^{r\Delta t} - d}{u - d}, \quad d = \frac{1}{u}, \quad u = \frac{e^{-r\Delta t}}{2}\left(1 + \nu^2 + \sqrt{(1+\nu^2)^2 - 4e^{2r\Delta t}}\right),$$

where $\nu^2 := e^{(2r+\sigma^2)\Delta t}$. However, when we perform a Taylor expansion of u in terms of $\sqrt{\Delta t}$, we notice that it agrees with the Taylor expansion of $e^{\sigma\sqrt{\Delta t}}$ up to the Δt term. That's why in practice in the CRR model one takes:

$$q = \frac{e^{r\Delta t} - d}{u - d}, \quad d = e^{-\sigma\sqrt{\Delta t}}, \quad u = e^{\sigma\sqrt{\Delta t}}.$$

The reason for this approximation is that this way the magnitude of the log return is constant in each step, namely $\sigma\sqrt{\Delta t}$. Now fix $T > 0$ and define $\Delta t := \frac{T}{N}$. It can then be proven that in the limit S_N converges in distribution to a log-normal random variable:

$$S_N \xrightarrow{dist.} S_0 \exp\left((r - \frac{\sigma^2}{2})T + \sigma W_T\right), \quad \text{as } N \uparrow \infty, \tag{1.3.9}$$

where $W = (W_t)_{t\geq 0}$ is a standard Wiener process. This means that when we take enough steps N, the discrete process $(S_n)_{n=1}^N$ is a good approximation of the continuous time, risk-neutral drift process described in (1.2.12). The idea is that for each end-node, the corresponding option price can be easily computed since it is equal to the payoff. By backward induction and taking the risk-neutral probability measure into account, a fair value for European option can be deduced. If we define $V^n := e^{-r(N-n)\Delta t}\mathbb{E}^{\mathbb{Q}}[g(S_N) \mid \mathcal{F}_n]$ as the time-n fair value of the European option with payoff $g(\cdot)$, the following recursive relationship follows from the Tower property:

$$V_n := \begin{cases} g(S_n), & n = N; \\ e^{-r\Delta t}\mathbb{E}^{\mathbb{Q}}[V_{n+1} \mid \mathcal{F}_n], & n = 0, 1, \ldots, N-1. \end{cases} \tag{1.3.10}$$

This method of tree pricing is not particularly interesting when used on vanilla European options, since in that case it will just be an approximation of the analytic value obtained with the Black-Scholes formula (1.2.16). However, since the method describes the evolution of the underlying during its life instead of only at maturity, it is very useful when dealing with options that contain discrete dividends payments and early exercising features. Similar to (1.3.10), one can derive that the time-n fair value V_n of an American option satisfies:

$$V_n := \begin{cases} g(S_n), & n = N; \\ \max\{g(S_n), e^{-r\Delta t}\mathbb{E}^{\mathbb{Q}}[V_{n+1} \mid \mathcal{F}_n]\}, & n = 0, 1, \ldots, N-1. \end{cases} \tag{1.3.11}$$

We see here clearly the benefits of the tree pricing method: in each node one can determine whether early exercising is optimal or not, and take that into account in the fair valuation of the American option using backwards induction.

1.3.4 Estimating forward

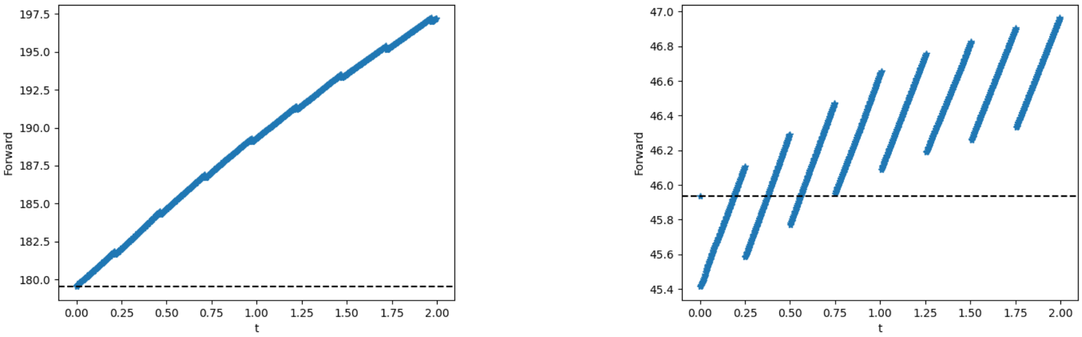
Estimating the forward from American option quotes alone is not recommended, since put-call parity doesn't hold: because of the early exercising feature we can't hedge a call (put) by selling a put (call) and selling (buying) a forward. However, using the tree pricing methods, an implied volatility estimate $\hat{\sigma}$ can still be made: it corresponds to the value of σ, such that the tree price with $u = e^{\sigma\sqrt{\Delta t}}$ matches the one found in the market. We can therefore argue that having an accurate forward estimate isn't that important as in the case of European options, where we needed it to calculate the implied volatility from Black's formula (1.2.24).

It is still interesting to visualize a forward estimate as function of time, since it gives us insight in the rational behaviour of an American option holder. Just as we used SPX data for European options, we fetched data of AAPL (on 2023/08/24) and C (on 2023/08/03) as the example for American options. We have chosen C on purpose because of its complexity: it has an ex-dividend date 2023/08/04, meaning that the next morning a (significant) dividend will be paid out to holders of the stock. On the other hand, AAPL represents a simpler case: its dividend payments are way smaller and the next dividend is only due in a few months.

Using estimates of upcoming dividend payments combined with the discount rate, we can employ (1.2.21) to provide a forward estimate. This leads to Figure 1.3, where we plotted the forward price F_t as a function of time t and denote the spot price S_0 with a horizontal dotted line. We clearly see that after each dividend payment, the forward drops by the (discounted) estimated dividend amount. From Figure 1.3b we notice that the forward price can become smaller than spot for short expirations. This ultimately results in a significant early exercise premium for calls, as we will see in Figure 3.1c.

1.4 No arbitrage conditions

We can summarize an arbitrage strategy (Definition 1.2.3) as a costless trading strategy that has a positive probability of earning risk-free profit [12]. In financial markets one usually assumes that there is no arbitrage, since whenever arbitrage arises it is quickly capitalized upon by specialized market participants after which the arbitrage disappears.



(a)

(b)

Figure 1.3: Forward prices of AAPL (left) and C (right).

The fact that arbitrage isn't possible, puts constraints on the prices of options on the same underlying. Moving forward, we consider portfolio's of call options, but similar relationships can be constructed for puts. Additionally, we assume products have one price (ignoring the bid-ask spread), no transaction costs are included and we can lend/borrow money at the same rate r. Finally, we only consider *static arbitrage*: an arbitrage exploitable by fixed positions in options and the underlying stock at initial time, while the position of underlying stock can be modified at at only a finite number of trading times in the future. Any other arbitrage is called *dynamic arbitrage*. The static arbitrage constraints can be viewed as the prerequisites that the price data must satisfy at time zero for admitting a dynamically arbitrage-free model [12].

Consider a filtered probability space $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t\geq 0}, \mathbb{P})$ that carries an adapted price process $\{(S_t, \mathbf{C}_t)\}_{t\geq 0}$, where $\mathbf{C}_t$ gives the price of the N considered call options at time t, and we observe $\mathbf{C}_0$. As a consequence of the *First Fundamental Theorem of Asset Pricing*, one arrives to [12]:

Theorem 1.4.1. *There is no static arbitrage if $\exists \mathbb{Q} \sim \mathbb{P}$ such that:*

$$C_0(T, K) = D_T \, \mathbb{E}^{\mathbb{Q}}[C_T(T, K)], \tag{1.4.1}$$

where $C_t(T, K)$ is the call price at time t, T is the expiration, and D_T is the discount factor.

In practice, we can summarize this as the simple idea that portfolios with guaranteed non-negative payout must have a non-negative price. Practically, this results in 4 main types of no-arbitrage constraints. Denote with C_K^T the call option with strike K and expiration T:

1. Call prices are non-negative:
$$C_K^T \geq 0$$

2. Absence of *vertical spread* arbitrage:
$$K_2 - K_1 \geq C_{K_1}^T - C_{K_2}^T \geq 0, \quad \text{where } K_1 < K_2$$

3. Absence of *butterfly* arbitrage:
$$\frac{C_{K_1}^T - C_{K_2}^T}{K_2 - K_1} + \frac{-C_{K_2}^T + C_{K_3}^T}{K_3 - K_2} \geq 0, \quad \text{where } K_1 < K_2 < K_3$$

4. Absence of *calendar spread* arbitrage:
$$C_K^{T_2} - C_K^{T_1} \geq 0, \quad \text{where } T_1 < T_2$$

Intuitively, the first three constraints imply that call portfolios with guaranteed non-negative future payout must have a positive price. The last constraint states that call options with longer time to expiration must cost more. In all cases, if this was not the case one could buy the underpriced asset, sell the overpriced one and hedge accordingly, this way achieving a costless trading strategy with a positive probability of earning a risk-free profit (i.e. an arbitrage strategy). We can combine these two ideas to define two additional types of no-arbitrage constraints, this time combining calls with both different expiries and strikes:

5. Absence of *calendar vertical spread* arbitrage:

$$C_{K_1}^{T_1} - C_{K_2}^{T_2} \geq 0, \quad \text{where } K_1 < K_2,\ T_1 > T_2$$

6. Absence of *calendar butterfly* arbitrage:

$$\frac{C_{K_1}^{T_1} - C_{K_2}^{T_2}}{K_2 - K_1} + \frac{-C_{K_2}^{T_2} + C_{K_3}^{T_3}}{K_3 - K_2} \geq 0, \quad \text{where } K_1 < K_2 < K_3,\ T_1 > T_2,\ T_3 < T_2$$

In [12] it is proven that when a set of call prices $\{C_K^T\}_{K \in \mathcal{K}, T \in \mathcal{T}}$ satisfies these constraints, there is no static arbitrage. Furthermore, directions to express these constraints as a matrix inequality $A\mathbf{c} \geq \mathbf{b}$ are given and a Python implementation is included [11]. This will prove to be very useful to check whether a vector of call prices $\mathbf{c}$ is arbitrage-free.

1.5 Implied volatility surface

The *implied volatility* of an option is the value of volatility that needs to be plugged in into an option pricing model to obtain the market price. In the case of European options, it is the value of σ that lets the Black-Scholes model price (1.2.16) match the market price of the option. For American options, it is the value of σ that lets the American option pricer (e.g. a binomial tree) match the market price. This idea allows to normalize options with different characteristics (different underlying, strike, maturity, etc.) and corresponds roughly to the markets expectation of realized volatility in the future (in practice implied volatility is usually higher than realized volatility, this is called the *volatility risk premium* [2]).

1.5.1 Shape of implied volatility surface

In the Black-Scholes model, there is one value of σ which represents the volatility of the stock, as in (1.2.12). However, when looking at the implied volatility of observed market prices, this is typically not the case: volatility changes both as a function of expiration T and strike K. First of all,

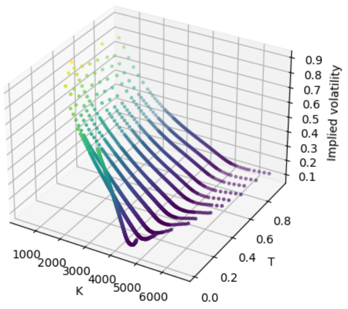


Figure 1.4: Implied volatility surface of SPX, generated using L1BA-PC (see Section 2.3)

we need to consider the concept of a *fair price*, the theoretical value that a rational investor would assign to the asset based on its fundamental characteristics, future cash flows, and other relevant factors. Most financial models assume an asset to have a single price, while usually two prices (i.e. bid price and ask price) can be observed for an asset. Methods to convert this information into an estimate of the fair price include the mid-price, the quantity-weighted price, the last traded price or the micro-prices of Stoikov [37]. In our framework (and in [12]) we choose the mid price as the fair price estimate because of the limited order book data available and the fact that more accurate price models are mostly useful in high frequency trading, which we don't cover here. However, other estimates could be considered and might be an interesting direction for future research.

When looking at the implied volatility of an equity option as a function of strike (for a constant expiration), we typically observe a *volatility smile* or *volatility smirk*. This is a phenomenon where options with strikes far from the money have significantly higher implied volatility than ATM options. There are multiple ways to explain why this phenomenon arises, which all boil down to the same idea:

- *Demand side:* Market participants that are net long want to protect again strong downside moves, and therefore buy relatively more OTM puts. Similarly, market participants that are net short want to protect against upside moves and therefore buy more OTM calls. Because of the greater demand, these options are priced relatively more expensive (in terms of implied volatility).
- *Supply side:* The implied volatility is a parameter that is part of the Black-Scholes model, which implicitly assumes that stock price change continuously and are log-normally distributed. However, in practice stock prices experience jumps. This results in OTM options paying out more often than would be expected in a log-normal model and sellers of these OTM options ask a premium to protect again these jumps. It turns out that an implied volatility smile results in a distribution that corresponds better to the heavy-tailed, asymmetric distribution of real returns.

The plot of $\sigma(T, K)$ as a function of expiration T and strike K is called the *implied volatility surface.*

1.5.2 Arbitrage in the implied volatility surface

In practice the implied volatility surface derived directly from the (market) mid prices is rarely arbitrage-free. In [21] a few reasons are given:

- While ideally the implied volatility would be derived from the *fair price* of an option, in practice we only have bid and ask prices. The fair price is therefore typically approximated by the average of the bid and ask (the *mid price*), which causes inaccuracies when the bid-ask spread is large. Moreover, research shows that the fair price doesn't always need to lie between bid and ask quotes: in [38] it is reported that around 4 % of option trades in a sample of CBOE options occur outside the last quoted spread.
- To understand the relationship between an option price V and its implied volatility σ, we can look at vega $\nu := \frac{\partial V}{\partial \sigma}$. By taking the derivative of the Black-Scholes formula, one observes that vega is large for ATM options but small for (deep) OTM and ITM options. The intuitive explanation is that that there is not much uncertainty left whether these deep OTM or ITM options will expiration OTM or ITM, which means that a change in implied volatility doesn't influence their price a lot. However, since $\nu \approx \frac{\Delta V}{\Delta \sigma}$, this means that for low vega options a small error in option price will result in a large error in implied volatility ($\Delta\sigma \approx \frac{\Delta V}{\nu}$).
- Unlike the idealized Black-Scholes markets, in actual markets the prices of options, underlying securities and interest rates can not move in infinitely small price increments, but are restricted by tick sizes. This introduces a measurement error, since the true price is not constrained to move in discrete steps.

However, the implied volatility surface is heavily relied upon by several market participants [22]:

- Traders use the volatility surface to trade volatility directly (i.e. express an opinion about future volatility).
- Traders use the volatility surface to price European options for strikes and expiries that are not quoted in a market (i.e. OTC trading).
- Traders price and hedge exotic options by using the implied volatilities observed on vanilla options. More realistic and advanced pricing models than Black-Scholes are calibrated against the observed implied volatility surface.

- From the implied volatility surface traders derive important signals such as market sentiment and available liquidity. Furthermore, the *risk-neutral density* $q(S_T)$ (density of the stock price at expiration under the risk-neutral drift (1.2.12)) can be derived:

  $$q(S_T = s) = D_T \, \frac{\partial^2 C}{\partial K^2}(K = s, T),$$

  where D_T is the discount factor and $C(K, T)$ is the call price derived from the implied volatility surface.

- Risk managers run stress scenarios on the implied volatility surface to assess the risk exposure of their options positions. The implied volatility surface can help assess tail risks — the potential for extreme market moves.

Since the implied volatility surface is so important in a trading operation, it is therefore essential to remove the arbitrage from the surface. Intuitively it doesn't make sense to use a model that contains arbitrage, since it is economically meaningless to have a model that has the potential to make risk-free profits. In practice, the presence of arbitrage leads to poor or even failed model calibration as well as incorrect estimations of the risk-neutral density. For example, the calibration of the *local volatility model* of [15] and [14] will fail since negative local volatilities or negative transition probabilities appear. This obstructs the convergence of the finite difference schemes solving the underlying generalized Black-Scholes PDE (1.2.10) [16].

1.5.3 Repairing the implied volatility surface

As mentioned earlier, the two most common methods to obtain an arbitrage-free volatility surface are smoothing (continuous interpolation of option quotes using some parametrization) and filtering (removal of low-quality data). In [12] they propose a different method: instead of smoothing, which changes nearly all data, or filtering, which loses information, they propose to *repair* data. In this approach the changes to the implied volatility surface are the minimal changes that are necessary to make the surface arbitrage-free. They achieve this by constructing a linear optimization problem, with an l^1 cost function and with constraints the no-arbitrage constraints $A\mathbf{c} \geq b$, explained in Section 1.4. There are a few advantages of this approach:

- By using an l^1-norm cost function, a more sparse solution is obtained, which means that most prices are unchanged.
- Liquidity considerations such as the bid-ask spread can easily be taken into account by changing the optimization function (adding them as soft constraints).
- The resulting volatility surface is intuitive, in the sense that it can be interpreted as the *best* approximation of the raw volatility surface.
- The method is quite fast, which makes it well-suited to online computations.
- Because the method doesn't rely on a parametrization, it doesn't have any problems with more exotic shapes of implied volatility surface (e.g. before earnings announcements, the implied volatility slices of some short-dated options will resemble a W-shape).
- When used as a pre-processing step before option price calibration, the calibration is more robust in the sense that there is less variation in the obtained parameters, see Section 4.1.
- It can be used as a post-processing step of prices predicted by deep learning algorithms (which have recently gained substantial popularity [31]) that themselves don't rule out arbitrage.
- The solution of the method detects whether executable arbitrages exist, see Section 4.2.

We will examine this method more deeply in Section 2.2.

The main focus of this dissertation is to test the approach in [12] and extend it to be used in a wider range of strikes, include American options and add handling of discrete dividends.

Chapter 2

Repairing the volatility surface of European options

In this chapter we describe a method to obtain arbitrage-free implied volatility data points from a set of European option prices quoted in the market. First, we give an introduction to the data we use and how to get market parameters in practice. Next, we shortly summarize linear programming. After this, we explain in detail the methodology that is currently used to repair implied volatility surfaces of European options [12], namely L1BA. Finally, in Section 2.3 we extend this methodology by adding put prices, which better capture the markets assessment of fair implied volatility for low strikes. We compare both methods to real market data and observe a significantly better fit for low strikes.

2.1 Linear programming

A linear program (LP) is an optimization problem in which both the objective function and the (equality and inequality) constraints are linear. It can generally be expressed in canonical form as:

$$\boxed{\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{c}^T\mathbf{x} \\ \text{s.t.} \quad & \mathrm{A}\mathbf{x} \geq \mathbf{b} \end{aligned}} \tag{2.1.1}$$

Note that the feasible region is a convex polytope, an intersection of finitely many half spaces. The goal of linear programming is to find the point in this region for which the affine objective function is minimized. Since a linear program can be seen as the simplest form of a convex optimization problem, every local minimum must be a global minimum. Moreover, when the feasible region is bounded and non-empty, a solution must exist and can be found on its boundary.

Linear programs appear in a wide variety of fields such as engineering, economics, transportation and manufacturing. Because of their ubiquity in applied mathematics and their relative simplicity, they have been studied extensively and many efficient algorithms have been devised to solve large linear programs. The simplest (and perhaps best known) algorithm is the simplex method, which solves linear programming problems by iteratively moving from one feasible solution to another in the direction of improving the objective function, after which it finally converges. A more modern approach are interior-point methods, which move through the interior of the feasible region, rather than through the boundary like the simplex method. They rely on a barrier function that penalizes violations of the constraints, allowing efficient convergence to the optimal solution. Without going further into the details of these methods, it is sufficient here to understand that describing optimization problems as linear programs is attractive because large-scale linear programs can be solved quickly and efficiently with modern algorithms and (under reasonable assumptions) are guaranteed to converge.

2.2 The L1BA method

As discussed earlier in Section 1.5, removing arbitrage from the implied volatility surface is absolutely necessary. Here, we introduced the most common approaches namely arbitrage-free *smoothing* and *filtering*. The problem with the former method is that nearly all data is changed, while the latter removes data (and therefore information). In [12] they propose to formulate the data repair as a linear program, where the constrains are the no-arbitrage relations and the objective function minimizes price changes (where perturbations outside the bid-ask spread are punished more harshly). They show that this method - which they call L1BA - is fast in real world large-scale problems (due to the LP formulation) and that the perturbations are sparse, in the sense that most data is unchanged.

As discussed in Section 1.4, the no-arbitrage relations can be written as a system of linear inequalities in the form A$\mathbf{c} \geq \mathbf{b}$. If we then define ϵ to be the vector of perturbations that we add to the call prices $\mathbf{c}$ in the hope of making them arbitrage-free, the most general description of an optimization problem that removes arbitrage is:

$$\boxed{\begin{aligned} \min_{\boldsymbol{\epsilon}} \quad & f(\boldsymbol{\epsilon}) \\ \text{s.t.} \quad & A\boldsymbol{\epsilon} \geq \mathbf{b} - A\mathbf{c} \end{aligned}} \tag{2.2.1}$$

where the objective function f measures how much the perturbation vector $\boldsymbol{\epsilon}$ deviates from the zero vector.

2.2.1 The L1 method

A very important question is now how to construct an optimization function f which satisfies our goal of adjusting just a few prices (i.e. keeping the perturbation sparse). Obvious candidates are:

- The L^0-norm: $f(\boldsymbol{\epsilon}) = \sum_{j=1}^{N} \mathbb{1}_{\{\epsilon_j=0\}}$
- The L^1-norm: $f(\boldsymbol{\epsilon}) = \sum_{j=1}^{N} |\epsilon_j|$
- The L^2-norm: $f(\boldsymbol{\epsilon}) = \sqrt{\sum_{j=1}^{N} \epsilon_j^2}$

In [12] they argue (using extensive referenced literature) to use the L^1-norm. This is because even though the L^0-norm captures the concept of sparsity best, it leads to a non-convex optimization problem which is NP-hard to solve. While the L^2-norm has been used extensively in data-smoothing problems, the authors choose not to use this here since it usually leads to small perturbations for all prices, while in our application a sparse solution where most prices are unperturbed is preferred. They finally note that the L^1-norm is more robust to outliers, since the L^2-norm squares outliers resulting in a larger contribution to the objective function.

When we define $\epsilon_j^+ := \max(\epsilon_j, 0)$, $\epsilon_j^- := -\min(\epsilon_j, 0)$, $\boldsymbol{\theta} := [\boldsymbol{\epsilon}^+ \; \boldsymbol{\epsilon}^-]^T$ and $B = [-A \; A]$, the repair problem with L^1-norm minimization can be written as the following LP:

$$\boxed{\begin{aligned} \min_{\theta} \quad & \mathbf{1}^T\theta \\ \text{s.t.} \quad & \mathrm{B}\theta \leq A\mathbf{c} - \mathbf{b} \\ & \theta \geq \mathbf{0} \end{aligned}} \tag{2.2.2}$$

After finding the optimal solution $\boldsymbol{\theta}^*$, the optimal perturbation vector is recovered as $\boldsymbol{\epsilon}^* = \boldsymbol{\epsilon}^{+*} - \boldsymbol{\epsilon}^{-*}$. We (and [12]) call this method the 'L1 method'.

2.2.2 The L1BA method

When solving the LP (2.2.2), no consideration of bid-ask prices is taken. Its input consists of the calls' normalized mid prices, which are perturbed by the solution $\boldsymbol{\epsilon}^*$ to produce arbitrage-free mid prices. In practice, however, we observe a bid-ask spread: the bid price is the highest price someone is willing to pay, while the ask price is the lowest price at which someone is willing to sell.

When we look at a snapshot of the order book of SPX, the bid price is always strictly smaller than the ask price, because if it is not, the buyer and seller are matched, the transaction goes through, and the order is removed from the order book. This difference is called the bid-ask spread and provides important information we want to include in the model. To be more specific, we want as many of the perturbed prices to be in the original bid-ask bounds as possible. This means that a price with wider spreads should be given more freedom to be perturbed.

Instead of adding the bid-ask constraints in the repair problem (*hard constraints*), the authors choose to allow perturbations outside of the bid-ask bounds, but to dis-encourage them by adding a penalty in the objective function (*soft constraints*). The main reason is that there might not exist a solution where all perturbations are inside the bid-ask bounds, and we want our optimization problem to be feasible in all cases.

An objective function of the form $f(\boldsymbol{\epsilon}) = \sum_{j=1}^{N} f_j(\epsilon_j)$ is chosen, where f_j has a non-negative image and $f_j(x)$ should be interpreted as the cost of perturbing the j-th option price by an amount x. Note that the L^1-norm objective stated as the LP (2.2.2) corresponds to $f_j(x) = |x|$. This function will now be slightly modified to include the bid and ask prices. In [12], they define δ_j^a,$\delta_j > 0$ as the normalised ask-fair and fair-bid spreads respectively, with the idea that the fair price estimate doesn't necessarily need to be the mid price. However, since we always use the mid price as the fair price, we will just write δ_j to mean the normalized ask-mid spread (same as mid-bid spread by definition). They argue that for the objective function to make sense, $f_j(x)$ should have the following properties for all j:

1. $f_j(0) = \inf_x f_j(x) = 0$. The minimum is attained when there is no perturbation, which is costless to the objective;

2. $f_j(x)$ is monotonically increasing (decreasing) for $x > 0$ ($x < 0$);

3. $f_j(-\delta_j) = f_j(\delta_j) = \delta_0$ where $\delta_0 \geq 0$ is a constant. The cost of perturbing a price to its bid or ask price is the same for all options;

4. $df_j(x)/d|x| = 1$ for $x \in (-\infty, -\delta_j) \cup (\delta_j, +\infty)$. The marginal cost for perturbing a price out of the bid-ask price bounds is the same for all options.

They therefore propose the following objective function that is very similar to the L^1-norm LP (2.2.2), satisfies all properties and allows to express the repair problem as a LP:

$$f_j(x) = \max\left(\frac{\delta_0}{\delta_j}|x|, \quad |x| - \delta_j + \delta_0\right), \tag{2.2.3}$$

where $\delta_0 \leq \min(\delta_j)$ such that the marginal cost of perturbing a price within the bid-ask price band (i.e. δ_0/δ_j) is always smaller than the marginal cost of perturbing mid prices outside the bid-ask price bounds. This means that a solution with a lot of perturbations inside the bid-ask spread is preferred to a solution with fewer perturbations, but with perturbation(s) outside the bid-ask spread. This will be important when we detect arbitraged in Section 4.2. The authors of [12] simply take $\delta_0 := \min(\delta_j)$, and we will use the same value as well moving forward. The L1BA

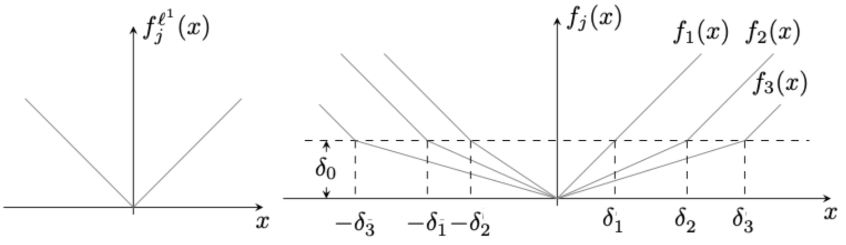


Figure 2.1: Comparison of the penalty function in L1 and the penalty function in L1BA, [12].

method therefore corresponds to the optimization problem:

$$\boxed{\begin{aligned} \min_{\boldsymbol{\epsilon}} \quad & f(\boldsymbol{\epsilon}) = \sum_{j=1}^{N} \underbrace{\max\left(\frac{\delta_0}{\delta_j}|\epsilon_j|,\ \ |\epsilon_j| - \delta_j + \delta_0\right)}_{=f_j(\epsilon_j)} \\ \text{s.t.} \quad & A\boldsymbol{\epsilon} \geq \mathbf{b} - A\mathbf{c} \end{aligned}} \tag{2.2.4}$$

The repair problem is rewritten into an LP using introducing auxiliary variables $\mathbf{t} := [t_1, \dots, t_N]^T$:

$$\boxed{\begin{aligned} \min_{\mathbf{t},\boldsymbol{\epsilon}} \quad & \sum_{j=1}^{N} t_j \\ \text{s.t.} \quad & -\epsilon_j - \delta_j + \delta_0 \leq t_j,\ \ \epsilon_j - \delta_j + \delta_0 \leq t_j \\ & -\frac{\delta_0}{\delta_j}\epsilon_j \leq t_j,\ \ \frac{\delta_0}{\delta_j}\epsilon_j \leq t_j, \\ & -A\boldsymbol{\epsilon} \leq -\mathbf{b} + A\mathbf{c} \end{aligned}} \tag{2.2.5}$$

where the benefit of expressing $f_j(x)$ as a maximum (instead of as a piecewise function) becomes clear.

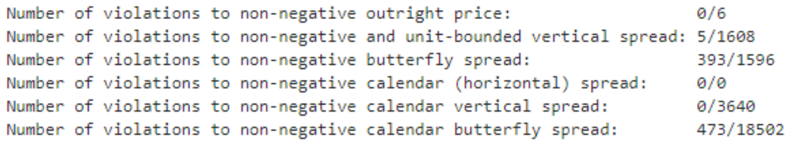
We now investigate the L1BA method to SPX options data, fetched on 13 July 2023. From Figure 2.2a we see that initially quite a lot of arbitrages are detected. This is because of the inaccuracy of the mid prices of calls with low strikes. These arbitrages are typically not executable, since we cannot buy and sell at these mid prices but need to pay half of the (large) bid-ask spread as well. In Figure 2.2b, we plot the optimal perturbation from L1BA to the option prices (expiring in 5 months). We see that perturbations within the bid-ask spread are preferred, resulting in larger perturbations for low strikes. Even though perturbations outside this spread are penalized, it is apparently necessary to put some quotes outside the spread in order to be arbitrage-free.

2.3 The L1BA-PC method

2.3.1 Problems with the L1BA method

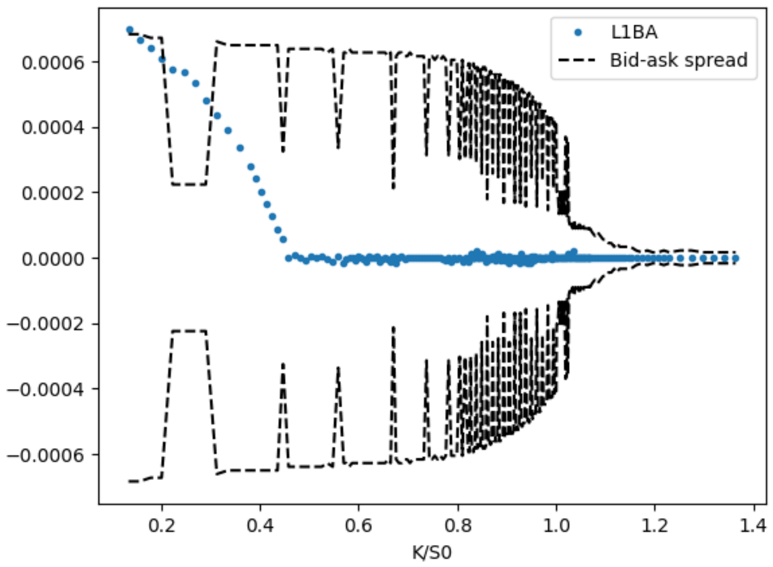
As demonstrated in [12], the L1BA method constructed by Cohen et. al. works well in some limited examples, but in our experience has some problems when applied to the large-scale dataset one encounters in practice. In their paper they usually repair only a small part of a volatility slice, e.g. only the strikes ±20% around the ATM value. However, we want to leverage our large dataset which contains price information on a way wider range of strikes (e.g. ±80% around the ATM value) so we can repair the implied volatility at these strikes as well, without resorting to extrapolation methods.

The main problem in doing so is that deep ITM options are not often traded, as can be seen on Figure 2.3a-2.3b. This is a consequence of the fact that OTM options are attractive to investors for hedging purposes, since they protect for downside portfolio risk. However, deep ITM options have a delta near one and therefore behave very similar to the underlying itself. Since one also foregoes the dividend payments when holding the option instead of the underlying, these deep ITM options are not particularly attractive to investors. In our order book data, we see that this manifests itself in low volume, wide spreads and therefore relatively inaccurate mid prices. Using these illiquid (and therefore inaccurate) prices to infer an implied volatility surface will typically not be meaningful. In the case of the L1BA method (where only call price data is used) this problem is apparent for low strikes, since these are the ITM calls. However, when looking at Figure 2.3, we observe that put options are liquid for these low strikes (since these are the OTM puts). It is for this reason that it is common practice to use call options to calculate implied volatility for strikes above ATM, while using put options to calculate implied volatility for strikes below ATM. The main question is now how exactly the put price data can be included in our optimization framework in a manner that is useful and produces a volatility slice that makes sense. Ideally, we would like to not impose arbitrary constraints (e.g. use only puts for strikes below ATM), but use



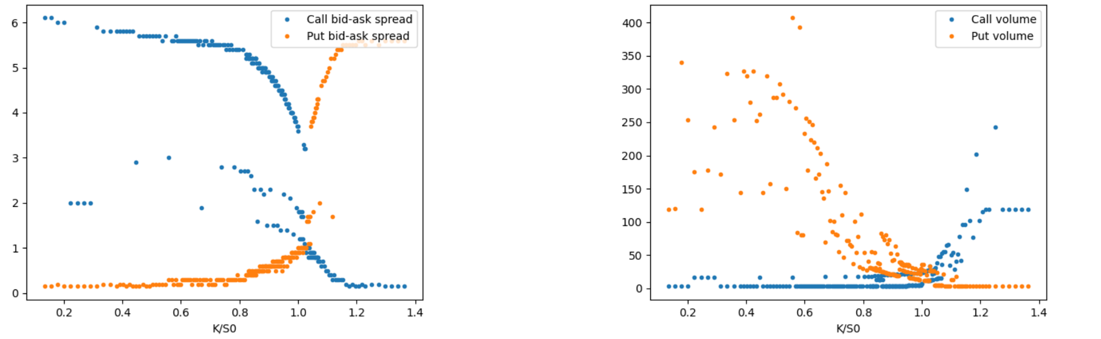
```
Number of violations to non-negative outright price:                     0/6
Number of violations to non-negative and unit-bounded vertical spread: 5/1608
Number of violations to non-negative butterfly spread:                   393/1596
Number of violations to non-negative calendar (horizontal) spread:       0/0
Number of violations to non-negative calendar vertical spread:           0/3640
Number of violations to non-negative calendar butterfly spread:          473/18502
```

(a) Initial arbitrages detected in the SPX options data



(b) The L1BA method applied to SPX data

Figure 2.2: Results of the L1BA method



(a) Bid-ask spread for SPX options in the order book

(b) Average volume for SPX options in the order book

Figure 2.3: Liquidity characteristics for SPX options

both price quotes and let the algorithm automatically weigh both quotes based on its liquidity (i.e. its bid-ask spread).

We will propose a new method, called L1BA-PC, which extends the L1BA method by adding the put option data to the repair problem. We will show that this allows us to obtain arbitrage-free implied volatility estimates that are close to the implied volatility slice that is provided to us by an undisclosed external datasource.

2.3.2 A naive approach

A first approach would be to construct the equivalent no-arbitrage constraints for puts and write it as a system of linear inequalities using a matrix $\bar{A}$ and vector $\bar{\mathbf{b}}$, where the (normalized) put prices $\mathbf{p} := [p_1, \ldots, p_N]^T$ are arbitrage free if $\bar{A}\mathbf{p} \geq \bar{\mathbf{b}}$. We can then plug in the puts' normalized mid-ask δ'_j to get an optimization problem that is equivalent to (2.2.4) but with puts instead of calls. We write this as:

$$\boxed{\begin{aligned} \min_{\boldsymbol{\epsilon}'} \quad & f(\boldsymbol{\epsilon}') = \sum_{j=1}^{N} \underbrace{\max\left(\frac{\delta_0}{\delta'_j}|\epsilon'_j|,\ |\epsilon'_j| - \delta'_j + \delta_0\right)}_{=f_j(\epsilon'_j)} \\ \text{s.t.} \quad & \bar{A}\boldsymbol{\epsilon}' \geq \bar{\mathbf{b}} - \bar{A}\mathbf{p} \end{aligned}} \tag{2.3.1}$$

Solving (2.2.4) and (2.3.1) then leads to optimal call prices $\mathbf{c}^*$ and optimal put prices $\mathbf{p}^*$. However, there are a few problems with this approach, mainly because the optimization problems are solved independently from each other:

- For a fixed maturity T and for each strike K, $\mathbf{c}^*$ and $\mathbf{p}^*$ imply a forward price from put-call parity (1.2.19). However, note that the optimal call prices are found independently from the optimal put prices. This means that in general the implied forward price for time T is not constant as a function of K, similar to Figure 1.1a. This is a violation of the no-arbitrage condition: we could buy the lowest priced (pseudo-)forward and sell the highest priced one, to generate an immediate profit while the payoff of the portfolio is always zero. Another way to state this, is that the no-arbitrage conditions outlined in Section 1.4 are sufficient when only considering calls. When we add puts to the mix, we need to add the constraint that put-call parity holds with an implied forward is constant for each maturity.
- For a fixed maturity T and for each strike K, the optimal prices $\mathbf{c}^*$ and $\mathbf{p}^*$ correspond to implied volatilities σ_c^* and σ_p^*. These are not necessarily equal to each other, and some (arbitrary) weighting scheme must be devised to obtain one implied volatility σ^* that will be considered the implied volatility in the volatility slice at time T
- The construction of $\bar{A}$ and $\bar{b}$ is not trivial and requires extra memory space (relevant since A can consist of a million rows and a few thousand columns).

2.3.3 A better approach: L1BA-PC

The second approach - the one we will use in our L1BA-PC method - is to convert put prices to 'synthetic call' prices using put call parity:

$$\mathbf{c}' = \mathbf{p} + \mathbf{1}_N - \mathbf{k}, \tag{2.3.2}$$

where N is the number of quotes, $\mathbf{k} := \mathbf{K}/\mathbf{F}$, $\mathbf{c}'$ and $\mathbf{p}$ are the normalized prices for a (synthetic) call and put with strike K and expiration T, and F is the forward at time T.

This results in the following optimization problem:

$$\boxed{\begin{aligned} \min_{\boldsymbol{\epsilon},\boldsymbol{\epsilon}'} \quad & f(\boldsymbol{\epsilon}, \boldsymbol{\epsilon}') = \sum_{j=1}^{N} f_j(\epsilon_j) + \sum_{j=1}^{N} f'_j(\epsilon'_j) \\ \text{s.t.} \quad & A\boldsymbol{\epsilon} \geq b - A\mathbf{c} \\ & A\boldsymbol{\epsilon}' \geq b - A\mathbf{c}' \\ & \mathbf{c} + \boldsymbol{\epsilon} = \mathbf{c}' + \boldsymbol{\epsilon}' \end{aligned}} \tag{2.3.3}$$

where f'_j is defined as in (2.2.3), but with the mid-ask spread δ'_j of the synthetic calls instead of δ_j, and ϵ'_j is the applied perturbation the the j-th synthetic call price c'_j. Note that from (2.3.2) it follows that the mid-ask spread of the synthetic calls δ'_j is the same as that of the puts. We see immediately that we don't need to construct nor save matrices representing the put no-arbitrage constraints, since we only use call prices in our program. Furthermore, the last constraint implies that the result will have a constant implied forward for each volatility slice, so that the forward arbitrage is removed as well.

Theorem 2.3.1. *The constraint $\mathbf{c} + \boldsymbol{\epsilon} = \mathbf{c}' + \boldsymbol{\epsilon}'$ corresponds to the put-call parity constraint.*

Proof. Define $\mathbf{c}_* := \mathbf{c} + \boldsymbol{\epsilon}_*$ and $\mathbf{c}'_* := \mathbf{c}' + \boldsymbol{\epsilon}'_*$, where $(\boldsymbol{\epsilon}_*, \boldsymbol{\epsilon}'_*)$ is the solution of the optimization problem (2.3.3). Substituting $\mathbf{c}'_*$ into the put-call parity equation (2.3.2), we get the corresponding optimal put prices $\mathbf{p}^* = \mathbf{c}'_* - \mathbf{1}_N + \mathbf{k}$. However, due to the last constraint in (2.3.3), we must have $\mathbf{c}_* = \mathbf{c}'_*$ for the solution to be feasible. This leads to:

$$\mathbf{c}_* - \mathbf{p}_* = \mathbf{1}_N - \mathbf{k},$$

meaning that this constraint implies put-call parity for the solution of the optimization problem. □

Finally, since after the optimization both the original calls and the synthetic calls will have the same value, there is no ambiguity over which implied volatility to use to construct the implied volatility surface.

After more inspection, we observe that it makes no sense to optimize both the original call prices as the synthetic call prices, as in the end they will have the same price due to the last constraint. We can reduce the dimension of the optimization problem (2.3.3) by substituting $\hat{\boldsymbol{\epsilon}} = \Delta c + \boldsymbol{\epsilon}$, where $\Delta c := \mathbf{c} - \hat{\mathbf{c}}$. This leads to:

$$\boxed{\begin{aligned} \min_{\boldsymbol{\epsilon}} \quad & g(\boldsymbol{\epsilon}) = \sum_{j=1}^{N} \underbrace{f_j(\epsilon_j) + \hat{f}_j(\Delta c_j + \epsilon_j)}_{:=g_j(\epsilon_j)} \\ \text{s.t.} \quad & A\boldsymbol{\epsilon} \geq b - A\mathbf{c} \end{aligned}} \tag{2.3.4}$$

This is the optimization problem we will solve in our L1BA-PC method. When comparing this to (2.2.4), we see that L1BA-PC is very similar to L1BA, but with the important difference that the cost of the j-th perturbation is $g_j(\epsilon_j)$ instead of $f_j(\epsilon_j)$. The difference between the two is that g_j adds information about the put prices through $\hat{f}_j$. Just as (2.2.4) was converted to the linear program (2.2.5), we will convert (2.3.4) to a linear program using the same technique.

2.3.4 Understanding L1BA-PC

We will now further investigate the L1BA-PC method, to better understand how it works. The main idea of this method is to convert put prices to synthetic call price using put-call parity (2.3.2). This way we obtain two option prices for each strike: the original one and the synthetic one. On Figure **??** we illustrate both prices and their corresponding spread from the perspective of the original calls c_j. The green line corresponds to the (normalized) mid price of the original calls (c_j), and the green covered area to the corresponding spread ($\pm\delta_j$). The red line illustrates the mid price of the synthetic calls ($\Delta c_j := c'_j - c_j$) and the covered area its spread ($\Delta c_j \pm \delta'_j$). We see that typically one of these prices will have a significantly smaller spread than the other one. Intuitively, the price with a smaller spread is the more accurate one, as a result of the higher liquidity. We therefore want our algorithm to take both prices into account, but to give a higher weighting to the more accurate one. This is what L1BA-PC achieves. To understand this, we need to investigate the optimization function g_j. First start with approximating $f_j(\epsilon_j)$ as $\frac{\delta_0}{\delta_j}|\epsilon_j|$. This is the case for $|\epsilon_j| < \delta_j$, i.e. perturbations inside the bid-ask spread, as is typical for most. We write:

$$g_j(\epsilon_j) \approx \frac{\delta_0}{\delta_j}|\epsilon_j| + \frac{\delta_0}{\delta'_j}|\Delta c_j + \epsilon_j| \tag{2.3.5}$$

If the spread on the original price is much lower than the one on the synthetic price, the first term dominates g_j. On the other hand, when this spread is much higher, the second term dominates. We can write:

$$\begin{aligned} g_j(\epsilon_j) &\approx \frac{\delta_0}{\delta_j}|\epsilon_j|, \quad \delta_j \ll \delta'_j \\ g_j(\epsilon_j) &\approx \frac{\delta_0}{\delta'_j}|\Delta c_j + \epsilon_j|, \quad \delta_j \gg \delta'_j. \end{aligned}$$

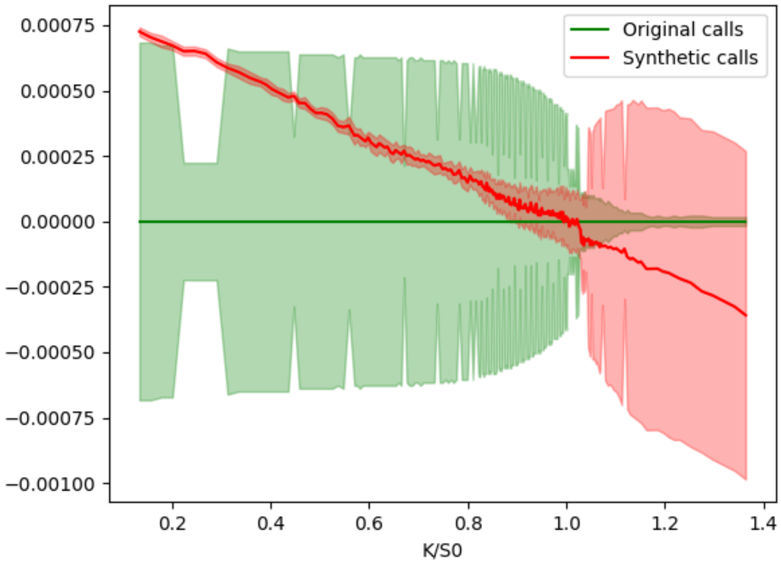


Figure 2.4: Analysis of the original and synthetic call prices and their spreads

In the first case (typically the case for strikes higher than ATM), ϵ_j will be close to zero, and the corresponding perturbed call price $c_j + \epsilon_j^*$ will be close to the original call price c_j. In the second case (strikes lower than ATM), $\Delta c_j + \epsilon_j^*$ will be close to zero, which means that $c_j + \epsilon_j^* \approx c_j'$, or that the perturbed call price will be close to the synthetic one. In a third case (typically around ATM) the spread for both call prices is similar, resulting in a perturbed price that will be close to the weighted average of two input prices. The method developed here clearly includes both call and put price information, and will automatically prefer put prices for low strikes and call prices for high strikes when constructing the volatility surface.

This is clearly illustrated in Figure 2.5, which plots the solution of the L1BA and L1BA-PC algorithm. Remember that this solution is the optimal perturbation ϵ_j^* from the original call prices c_j. We clearly see that in the case of L1BA-PC the put price information is preferred for low strikes. The synthetic spread δ_j' is way smaller and therefore $\epsilon_j^* \approx \Delta c_j$. For high strikes, the original spread is smaller resulting in $\epsilon_j^* \approx 0$. This is in contrast with the L1BA algorithm of [12], that doesn't take into account the information of the synthetic prices and therefore prefers a solution which has less deviations from the original call prices' point of view.

2.4 Comparison

It is now time to compare the different implied volatility slices that result from each of the methods. The data used here is again the option prices fetched on 2023/07/13 with underlying SPX and expiration 2023/12/15. We can compare the obtained implied volatility slices with an external benchmark (of which we are not allow to disclose the source). Raw implied volatilities are calculated by solving for the parameter σ in Black's formula (1.2.24).

First of all, we look at the implied volatilities of the raw option mid prices. One of the first things we notice is that for low strikes the implied volatility of raw call prices is zero. To explain this, notice that from a hedging argument if follows that an ask option price should always be higher than its intrinsic value $D(F - K)$: if the ask price would be lower than $D(F - K)$, we could buy the call and hedge by selling a forward and putting DK in a bank account, thereby locking in a risk-free profit. However, for low strikes, the bid-ask spread of call options is quite large (see Figure 2.3b) and the mid price might therefore be lower than intrinsic value. In that case, no value of implied volatility σ will return this mid price, resulting in a default implied volatility of zero as output. Similarly, for high strikes the raw puts have implied volatility of zero as well (again because

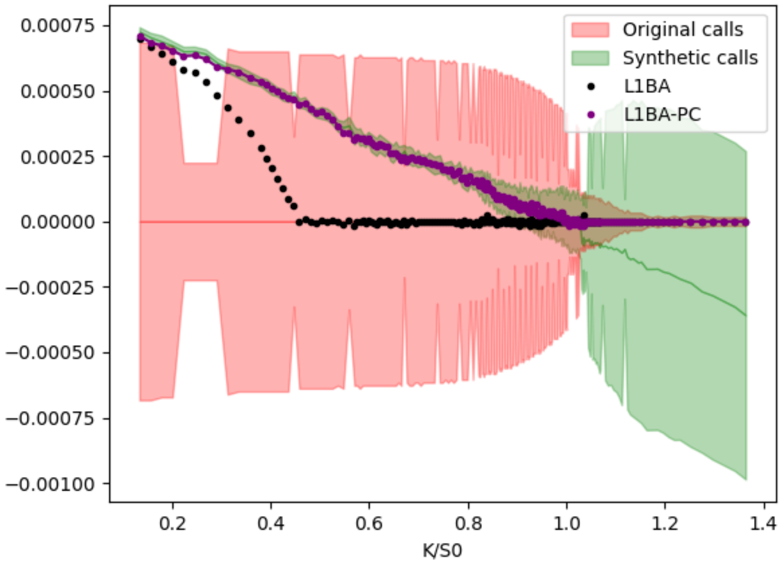


Figure 2.5: The solution of L1BA and L1BA-PC in the context of the call prices and their spreads.

inaccurate mid prices are smaller than intrinsic value). Because L1BA (the method developed in [12]) only uses call prices information, we clearly see that for low strikes L1BA deviates far from the benchmark's implied volatility slice. This is in contrast with L1BA-PC, which automatically uses put implied volatilities for low strikes and call implied volatilities for high strikes. We see that this results in an implied volatility slice that is much closer to the benchmark. However, also note that the benchmark isn't perfect either: it uses certain extrapolation methods on the wings. This explains the divergence of our method from the benchmark for low strikes.

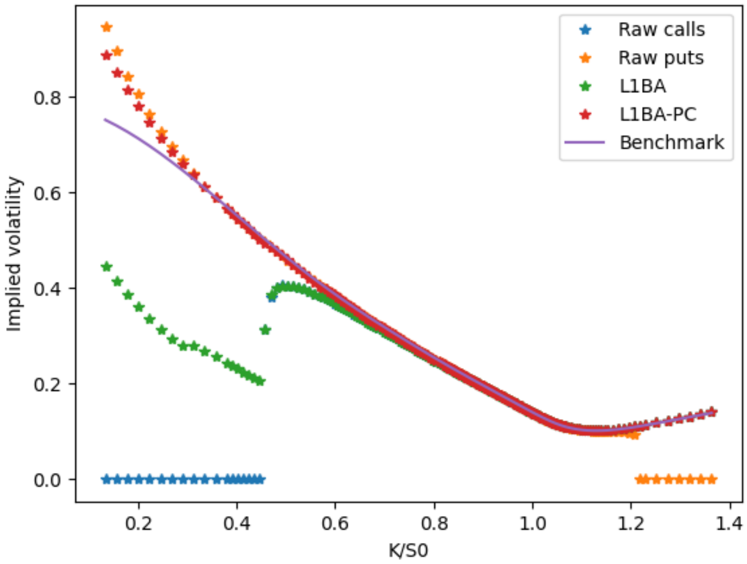


Figure 2.6: Implied volatility slice comparison

Chapter 3

Repairing the volatility surface of American options

The methods we covered previously, are based on European options, which can only be exercised at expiration. The options on index ETF's (e.g. SPX, QQQ, ...) are usually European. However, most options on single name stocks (e.g. AMZN, AAPL, LVMH,...) are American (even though the underlying isn't necessarily listed in America). The fact that American stocks are considerably more complicated means that we need a different framework to repair the volatility surface of American options. On top of that, we have to take into account the dividend payments that these stocks usually pay, which has an impact on their price and therefore on the forward and implied volatility as well.

First, we take a look at the difference in price between American and European options, defined as the *early exercise premium.* After this, we develop multiple methods to repair the volatility surface using American calls and puts. Finally, we compare the different methods to the implied volatility of an external data source.

3.1 The early exercise premium

The difference between an American option and the corresponding European option (i.e. an option on the same underlying, with the same expiration date and same strike) is that American options give the holder the right to exercise at any time, while European options only allow to do so at maturity. Since American options give holders the same rights as European options do, *plus* some extra ones, it intuitively makes sense that American options should always be worth as least as much as their European counterparts:

$$C^{AM} \geq C^{EU}, \qquad P^{AM} \geq P^{EU}. \tag{3.1.1}$$

The difference between the American option price and their European counterpart is called the *exercise premium* α:

$$\alpha_C := C^{AM} - C^{EU} \geq 0, \quad \alpha_P := P^{AM} - P^{EU} \geq 0 \tag{3.1.2}$$

This early exercise premium (EEP) can be explained in multiple ways:

- With American options we get extra *optionality* and optionality always has a non-negative price. The idea is that if a contract pays the holder money in the case that a certain event - with positive probability - happens, then this contract should be more expensive than an identical one which doesn't contain this. If this was not the case, one could buy the former, sell the latter, thereby locking in a profit when the event with positive probability happens while being hedged in all other cases.
- Another way to state this is that when comparing (1.2.11) and (1.3.1), we observe that the American option value is the risk-neutral expectation of the discounted intrinsic value using the optimal exercise policy, while for European options this expectation is only for a specific exercise policy (i.e. the one where you exercise at expiration) which is not necessarily the optimal one.

- In [10], an interpretation of the early exercise premium is that *"It corresponds to the compensation that the option holder would require in the stopping region in order to postpone exercise until the maturity date"*. The main idea is that for each t, an optimal stopping boundary B_t can be derived. For an American put, if $S_t < B_t$ it is optimal to early exercise, while for $S_t \geq B_t$ one should hold on. Following this argument, an American put can be converted into a European put by holding one American put when the stock price is above the exercise boundary, and duplicating the put's exercise value below the boundary (by selling short the stock and keeping K dollars in a bank account). Note that to keep a level position in the bank account, the interest on the K dollars must be siphoned off. At expiration, this strategy clearly matches the payoff of a European put, $(K - S_T)^+$, meaning that the present value of this strategy must be the European put price. This eventually implies that the early exercise premium can be interpreted as the present value of the interest earned when the stock price is below the boundary.

Using the fact that we can derive an arbitrage-free volatility surface using European options, an intuitive approach to obtain the same for American options would be to convert the American option quotes to *corresponding* European ones, after which the original L1BA-PC method can be used. Another way to state this is that we want to estimate the early exercise premium, to subtract it from the American option quotes.

3.1.1 EEP estimation

There are multiple approaches to the estimation of the early exercise premium. In the end, this boils down to the problem of pricing American options, for which (as mentioned in Section 1.3) one typically uses either tree pricing methods [13], finite difference methods [40], least-squares Monte Carlo [28], or analytical approximations [3], [5]. Once the pricing model is calibrated against the American options observed in the market, we can obtain an EEP estimate by plugging in the calibrated parameters in the corresponding model for European options and subtracting it from the Americans. This is one of the most common approaches in the financial industry and often called *de-Americanization* [9].

In our , we opted for a binomial tree pricer, but other pricing methods are possible. As we saw in Section 1.3, once the parameter u is defined (e.g. $u = e^{\sigma\sqrt{\Delta t}}$ as in the CRR model), the American option price is unambiguously defined. Since this parameter is a function of the implied volatility σ, we can state that there is a one-to-one correspondence between the implied volatility used and the American option price in the model. Now for each option quote, we find the σ which corresponds to the market price:

$$\text{Find } \sigma^* \text{ such that } \text{TreePricer}^{AM}(\sigma^*) = C^{AM}.$$

This problem corresponds to finding the root of $g(\sigma) := \text{TreePricer}^{AM}(\sigma) - C^{AM}$. Since American options prices are always larger than European ones, we know that the implied volatility σ^{AM} of the American option price interpreted as a European price must be an upper bound. We can then find the root with the bisection algorithm, starting with the interval $[0, \sigma^{AM}]$.

Once σ^* is found, we can immediately derive the corresponding European model price from the tree, or plug in this σ^* into Black's formula (1.2.24) directly. Either way, as N approaches infinity both lead to the same price, as postulated in (1.3.9). Subtracting the European price from the American one, we get an estimate of the EEP.

One of the factors that significantly complicates matters is the payment of dividends to holders of the stock. When a company issues a (cash) dividend, the stock price of that company is expected to drop by the dividend amount on the date the dividend is paid out (the *ex-dividend date*). The reason for this is simple: if this were not the case, a simple arbitrage would be possible where one buys the stock right before the ex-dividend date and sells it right after, thereby pocketing a risk-free profit (i.e. the dividend payment).

There are multiple ways to tackle this problem, and even 50 years after the publication of the Black-Scholes model, there is apparently no consensus on the appropriate generalization to the case of underlyings with cash dividends [25]. We opt here for the so-called *escrowed dividend* model, presented in [18]. The concept is rather straightforward: since dividends represent a risk-free element within the dynamics of stock prices, they can be subtracted from the stock price process, after which the usual Black-Scholes arguments can be applied. The resulting process (called the *capital process*) C_t is modelled as a geometric Brownian motion. This leads to:

$$dC_t = rC_t dt + \sigma C_t dW_t, \quad C_t := S_t - \sum_{t<t_i<T} I_{t_i} e^{-r(t_i - t)}, \quad S_T = C_T, \tag{3.1.3}$$

where I_{t_i} is the dividend payment paid out at time t_i. The payout of a call option is then $(C_T - K)^+$ and for European options the Black-Scholes formula can be applied to the process C_t with spot price $C_0 := S_0 - \sum_{0<t_i<T} I_{t_i} e^{-r(t_i - t)}$. Similarly for American options and using the tree-pricing method, we start from $C_0 := S_0 - \sum_{0<t_i<T} I_{t_i} e^{-r(t_i - t)}$ and discretize the continuous process as described in Section 1.3.

3.1.2 EEP estimation in practice

Before we look at an EEP estimation in practice, we prove a useful property of the early exercise premium: it is monotonic as a function of strike.

Theorem 3.1.1. *Assuming positive interest rates, the early exercise premium for American calls is decreasing as a function of strike:*

$$\alpha_C(K_1) \geq \alpha_C(K_2), \quad K_1 < K_2.$$

The early exercise premium for American puts is increasing as a function of strike:

$$\alpha_P(K_1) \leq \alpha_P(K_2), \quad K_1 < K_2.$$

Proof. We prove this for the case of puts, since the argument for calls is very similar. Consider the portfolio:

$$V_0 = P^{AM}_{0,K_2} - P^{EU}_{0,K_2} - (P^{AM}_{0,K_1} - P^{EU}_{0,K_1}), \quad K_1 < K_2,$$

where the subscript t, K denotes the price of the corresponding put at time t with strike K, and all puts have the same expiration. Until the start time $t = 0$ and expiration, either one of the following two happens:

1. The counterparty to which we sold short the American put with strike K_1 doesn't exercise their option early. In this case, we don't exercise the American put with strike K_2 early either, in which case the value of the American puts are equal to those of the European puts. In this case, the portfolio value at time T is clearly equal to zero:

$$V_T = 0.$$

2. The counterparty exercises the American put option with strike K_1 early. This means that we are forced to buy the underlying for a price of K_1. However, in this case we will exercise the American put that we are long at (with strike K_2) immediately. This means that we can sell the underlying for K_2, and we pocket a profit of $K_2 - K_1$. At expiration, the portfolio value is:

$$V_T = \underbrace{(K_2 - K_1)e^{r(T-t^*)}}_{\geq K_2 - K_1} - \underbrace{(P^{EU}_{T,K_2} - P^{EU}_{T,K_1})}_{\leq K_2 - K_1} \geq 0$$

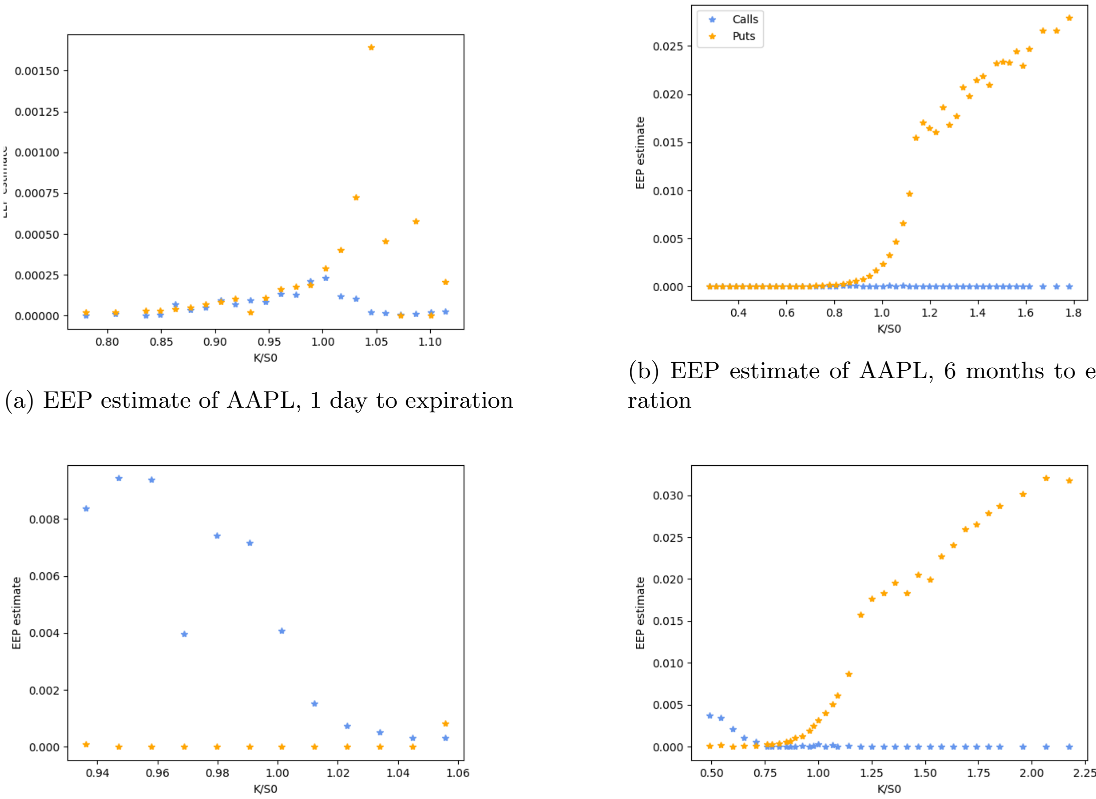
Note that our exercising policy is not necessarily optimal, but even with this suboptimal strategy it follows from the definition of an arbitrage strategy 1.2.3 that in a no-arbitrage world the portfolio at time 0 must have a positive price. This implies that $\alpha_P(K_2) \geq \alpha_P(K_1)$. □

To better understand the behaviour of these premiums, we plot the (normalized) early exercise premium estimates of AAPL and C in Figure 3.3. We notice a couple of interesting things.

First of all, in Figure 3.1b we observe the expected pattern: it is not optimal to exercise an

American call early if there is no nearby dividend payment, whereas for (deep) ITM American puts early exercise is attractive, resulting in a significant early exercise premium. Moreover, for very short times to maturity as in 3.1a, the early exercise premiums for both puts and calls is close to zero (notice the scale). This is because we are already that close to expiration that there is not much added value in the optionality to exercise early. We also notice that these EEP estimates are quite noisy: they are neither monotonic nor exactly equal to zero.

In the case of an upcoming dividend payment the picture changes dramatically. On Figure 3.1c we plot the early exercise premium estimate for C, one day to expiration but with a dividend payment in the meantime. As explained in Section 1.3.4 this dividend payment leads to a drop in the share price next morning, its size in expectation equal to the dividend amount. Intuitively, it is therefore logical to the holder of an American call to exercise early, before the dividend payment leads to a decline in the stock price. On the other hand, it is clearly not optimal for American put holders to exercise before this drop. This is evident in the EEP estimates: the call options generally have a significant EEP, while the put options do not. In Figure 3.1d we see that the upcoming dividend payment influences option prices with longer times to maturity as well, but to a lesser extent. There is still a significant EEP for deep ITM calls, which is otherwise not the case (as in Figure 3.1b).



(a) EEP estimate of AAPL, 1 day to expiration

(b) EEP estimate of AAPL, 6 months to expiration

(c) EEP estimate of C, 1 day to expiration

(d) EEP estimate of C, 6 months to expiration

Figure 3.1: Early exercise premium estimates in various cases

3.2 Repair methods

As explained in Section 2.3, the main idea of L1BA-PC is to convert put prices to (synthetic) call prices using put-call parity. This is fine when using European options (as is the case for index options), but unfortunately for American options (all single-name options) put-call parity doesn't hold anymore. This means that we can't easily convert put prices observed in the market to call prices anymore. Moreover, the no-arbitrage conditions $A\mathbf{c} \geq \mathbf{b}$ derived in Section 1.4 are valid for

European option prices and not necessarily for American ones.

The main insight of our proposed method(s), is that once a value for the early exercise premium is fixed, American put prices can be converted to European put prices, after which they can be converted to (synthetic) European call prices. Hereafter, we solve the same linear optimization problem as in 2.3 to obtain a repaired arbitrage-free volatility surface.

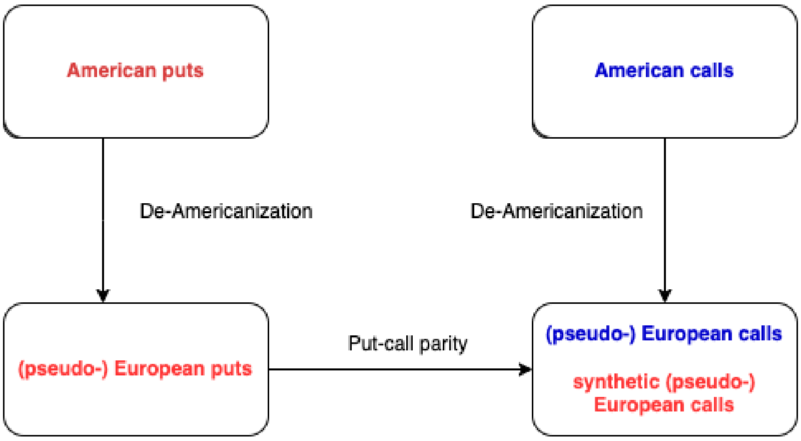


Figure 3.2: General schematic of our repair methods for American options

3.2.1 The L1BAA-PC method

The first method to construct the arbitrage-free implied volatility surface for American options is straightforward but rather naive. We simply use the early exercise estimate and subtract it from the American options to get (pseudo-)European options. These quotes are then passed directly to the L1BA-PC method described in Section 2.3.

As always, we start by normalizing the quotes and early exercise premium estimates by dividing by the discounted forward price. Writing **C** for normalized American calls and **P** for normalized American puts, we obtain normalized European calls **c** and puts **p** by subtracting the estimated (normalized) early exercise premium:

$$\begin{aligned}\mathbf{c} &= \mathbf{C} - \hat{\boldsymbol{\alpha}_C},\\ \mathbf{p} &= \mathbf{P} - \hat{\boldsymbol{\alpha}_P}.\end{aligned}$$

Note that for simplicity we assume that the early exercise premium is the same for the bid and ask price. We then convert the puts to synthetic calls using put call parity:

$$\mathbf{c}' = \mathbf{p} + 1 - \frac{\mathbf{K}}{\mathbf{F}}.$$

Finally, we define $\Delta c := \mathbf{c} - \mathbf{c}'$ and plug this in the L1BA-PC method, described in (2.3.4)

This method is relatively problematic, since it heavily depends on the early exercise premium estimates. These are typically quite noisy, as can be seen in Figure 3.3. Moreover, the implied forward (1.2.25) is not constant as a function of strike. In other words: put-call parity doesn't hold for its solution. Similar to how we handled this in the case of L1BA, we will solve this by adding the put-call parity constraint in the optimization problem.

3.2.2 The L1BAA-OPT method

The strategy we came up with to handle the noisy early exercise premiums is to include them as *free variables* in the optimization problem. This way we can easily force their monotonicity by adding it as a constraint. However, since the value of this early exercise premium isn't totally unknown (we have some estimate $\hat{\boldsymbol{\alpha}}$), we add a penalty in the optimization function that punishes large deviations from this estimate. Now, for simplicity, we construct an optimization problem using call prices only.

$$\begin{aligned}
\min_{\boldsymbol{\epsilon},\boldsymbol{\alpha}} \quad & f(\boldsymbol{\epsilon},\boldsymbol{\alpha}) = \sum_{j=1}^{N} f_j(\epsilon_j) + \lambda_1 \sum_{k=1}^{N} |\alpha_k - \hat{\alpha}_k| \frac{\delta_0}{\delta_k^a} \\
\text{s.t.} \quad & A(\boldsymbol{\epsilon} - \boldsymbol{\alpha}) \geq b - A\mathbf{C} \\
& \alpha_k \geq \alpha_{k+1}, \quad k = 1, ..., N, \text{ except when } k \text{ and } k+1 \text{ don't correspond to same expiration.}
\end{aligned}$$

First of all, we add a penalty term which punishes deviations from the EEP estimates. This term is scaled by a factor δ_0/δ_k, because (for a default value of $\lambda_1 = 1$) it makes this term roughly equal in scale to the corresponding term in the first sum (since $f_j(x) \approx \frac{\delta_0}{\delta_j^a}|x|$, see (2.2.3)). This also puts a greater confidence in estimates for which the bid-ask spread is small, similar to the what is done in the design of f_j.

Moreover, the no-arbitrage constraint $A\mathbf{c} \geq \mathbf{b}$ for European options can now be written as $A(\mathbf{C} - \boldsymbol{\alpha}) \geq \mathbf{b}$. Adding the EEP $\boldsymbol{\alpha}$ as a variable in the optimization problem allows us to use this matrix inequality, which only holds for European options, in the context of American options.

Finally, as proved in Section 3.1.2, for each expiration, α should be decreasing as a function of strike. Again, because we add the EEP as free variables in the optimization method, we can enforce this by adding it as a constraint. However, since $\boldsymbol{\alpha}$ contains EEP's for all expirations, we need to be careful and remove the constraints that would imply that the last estimate for an expiration must be greater than the first one for the next expiration.

One of the downsides of this method is that the resulting European call prices $(\mathbf{C} - \boldsymbol{\alpha} + \boldsymbol{\epsilon})$, and therefore also the repaired volatility surface, are dependent on the choice of an extra regularization parameter λ_1. However, we observe in practice that a value of $\lambda_1 = 1$ gives good result, since the scaling δ_0/δ_k^a makes the term $|\alpha_k - \hat{\alpha}_k|\frac{\delta_0}{\delta_k^a}$ roughly of the same size as $f_k(\epsilon_k)$. We can also interpret this as an advantage, since a larger value of λ_1 forces the early exercises premiums to be closer to the estimates, thereby giving us a way to tune the optimal solution.

3.2.3 The L1BAA-OPT-PC method

It is now time to add the put option quotes to the framework, like we have done in Section 2.3 to improve the L1BA method to the L1BA-PC method. As explained there, this allows for a much better fit for lower strikes, since calls for these strikes are very illiquid and the implied volatilities derived from the mid prices therefore inaccurate. This results in the following optimization problem:

$$\begin{aligned}
\min_{\boldsymbol{\epsilon},\boldsymbol{\epsilon}',\boldsymbol{\alpha},\boldsymbol{\beta}} \quad & f(\boldsymbol{\epsilon},\boldsymbol{\alpha}) = \sum_{j=1}^{N} \left(f_j(\epsilon_j) + f_j'(\epsilon_j')\right) + \lambda_1 \sum_{k=1}^{N} |\alpha_k - \hat{\alpha}_k| \frac{\delta_0}{\delta_k^a} + \lambda_2 \sum_{l=1}^{N} |\beta_l - \hat{\beta}_l| \frac{\delta_0}{\delta_l'^a} \\
\text{s.t.} \quad & A(\boldsymbol{\epsilon} - \boldsymbol{\alpha}) \geq \mathbf{b} - A\mathbf{C}, \\
& A(\boldsymbol{\epsilon}' - \boldsymbol{\beta}) \geq \mathbf{b} - A\mathbf{P} - \mathbf{1}_N + \mathbf{k} \\
& \alpha_k \geq \alpha_{k+1}, \quad k = 1, ..., N, \text{ except when } k \text{ and } k+1 \text{ don't correspond to same expiration.} \\
& \beta_k \leq \beta_{k+1}, \quad k = 1, ..., N, \text{ except when } k \text{ and } k+1 \text{ don't correspond to same expiration.} \\
& \mathbf{C} - \boldsymbol{\alpha} + \boldsymbol{\epsilon} = \mathbf{P} - \boldsymbol{\beta} + \boldsymbol{\epsilon}' + \mathbf{1}_N - \mathbf{k}
\end{aligned}$$

First of all, we have two extra penalty terms instead of one: deviations from the EEP estimates are punished, in proportion with the inverse of the bid-ask spread of the corresponding option. Using default values of 1 for both λ_1 and λ_2, this means that all three terms should be roughly equal in terms of their size.

The first constraint is exactly the same constraint as the one in LIBAA-OPT: it ensures that the (pseudo-)European calls $\mathbf{C} - \boldsymbol{\alpha}$ are arbitrage-free. Similarly, the constraint for the the synthetic (pseudo-) European calls $\mathbf{C}' - \boldsymbol{\alpha}'$ would be $A(\mathbf{C}' - \boldsymbol{\alpha}' + \boldsymbol{\epsilon}') \geq \mathbf{b}$. However, to decrease the number of free variables, we apply put-call parity to these synthetic (pseudo-) European calls so we can use our (pseudo-) European puts $\mathbf{P} - \boldsymbol{\beta}$ directly:

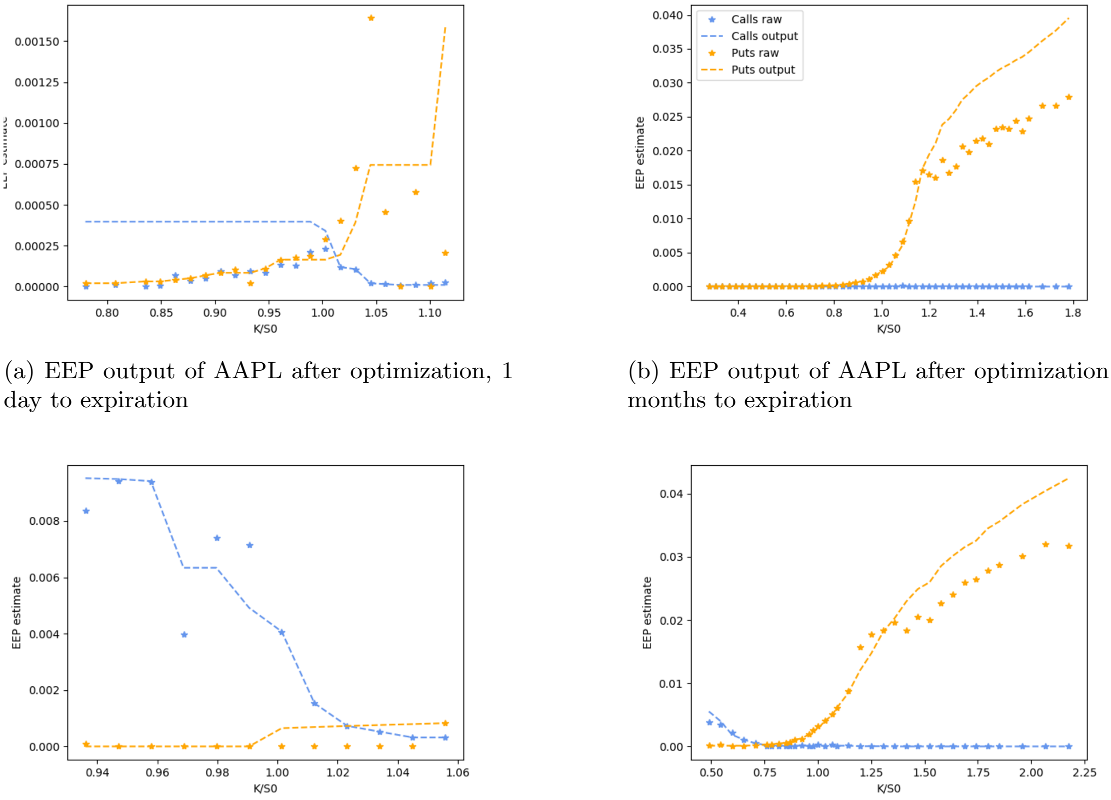
$$\mathbf{C}' - \boldsymbol{\alpha}' = \mathbf{P} - \boldsymbol{\beta} + \mathbf{1}_N - \mathbf{k}. \tag{3.2.1}$$

Doing this results in the second constraint.

The third and fourth constraint are a consequence of Theorem 3.1.1, enforcing the call EEP's to be decreasing and the put EEP's to be increasing.

The last constraint we need to enforce is that put-call parity needs to hold between the optimal (pseudo-) European calls and the optimal synthetic (pseudo-) European calls: $\mathbf{C} - \boldsymbol{\alpha} + \boldsymbol{\epsilon} = \mathbf{C}' - \boldsymbol{\alpha}' + \boldsymbol{\epsilon}'$. Again, we plug in the put-call parity (3.2.1), to limit the amount of variables, resulting in the last constraint. This is exactly the same as the last constraint in L1BA-PC (2.3.3). The difference however, is that for L1BA-PC we further reduced the dimension by substituting $\boldsymbol{\epsilon}'$, thereby removing that constraint. We won't do this here, mainly to keep the optimization problem clear and tractable. However, note that an equality constraint of the form $\mathbf{x} = \mathbf{y}$ can always be converted into two inequality constrains $\mathbf{x} \leq \mathbf{y}$ and $\mathbf{x} \geq \mathbf{y}$, meaning that this still corresponds to an LP.

Again, we have a similar disadvantage as previously mentioned: the solution is dependent on the choice of λ_1 and λ_2. However, due to the appropriate scaling it seems that both values taken to be one, gives good results in practice. As an illustration, in Figure 3.3 we investigate the output after optimization of the early exercise premiums for the AAPL and C options data. We clearly see that the obtained early exercise premiums are close to the estimates, while the monotonicity constraints result in monotonely decreasing and increasing call and put premiums respectively. Note however that this doesn't necessarily mean that these are *the* early exercise premiums. One could even argue that *the* early exercise premium doesn't even exist, since there is usually not a liquid market for both European and American options. We can view these obtained risk premiums as values that are convenient in the solution of the optimization problem and lead to acceptable implied volatility surfaces.



(a) EEP output of AAPL after optimization, 1 day to expiration

(b) EEP output of AAPL after optimization, 6 months to expiration

(c) EEP output of C after optimization, 1 day to expiration

(d) EEP output of C after optimization, 6 months to expiration

Figure 3.3: Early exercise premium output in various cases

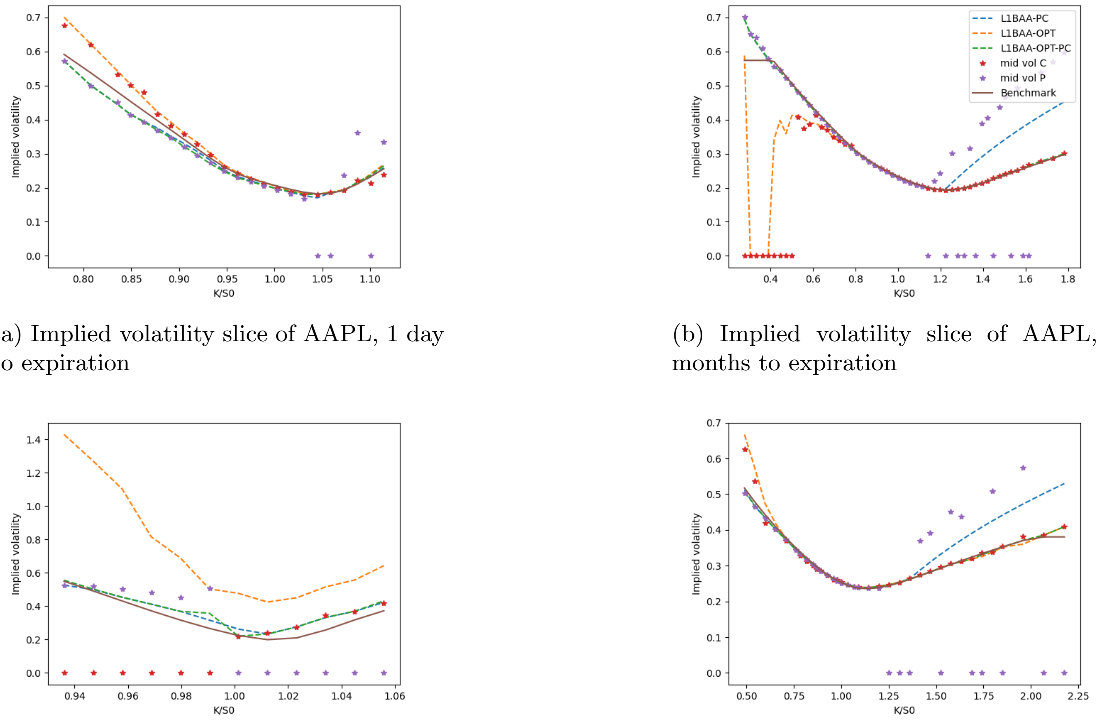
3.3 Comparison

In Figure 3.4, we compare these three methods in the four cases. We notice that our most sophisticated method, i.e. L1BAA-OPT-PC generally results in an implied volatility slice that is close to the one provided by the benchmark. Especially in Figure 3.4c, the situation is relatively complex because there is only one day to expiration with a dividend payment inbetween. We also note that in the case of only one day to expiration, vega is very low (as is always the case close to expiration) over the whole range of strikes, meaning that a seemingly large deviation in implied volatility still correspond to a very similar option price. Taking this into account, we conclude that the L1BAA-OPT-PC generally provides a good fit.

Additionally, we see that the L1BAA-PC method works well for close expirations, but fails to work well for further expirations. The explanation for this is that early exercise premiums are larger the longer to maturity, meaning that it becomes more important to have accurate values for the early exercise premiums. Even though the estimates are clearly noisy in Figure 3.1a and 3.1c, it doesn't matter that much since the premium is generally small and doesn't impact the solution that much.

Unsurprisingly, L1BAA-OPT doesn't work that well, especially for low strikes. This is essentially the same idea as to why L1BA doesn't work that well in Figure 2.6, namely that the calls for these strikes are illiquid, resulting in inaccurate mid prices and therefore inaccurate implied volatilities. This is not surprising, since the L1BAA-OPT method was developed as a theoretic intermediate step to better understand the L1BAA-OPT-PC method, rather than as a valid method on its own.

From the raw implied volatilities of calls and puts, we see that the mid prices for calls are inaccurate for low strikes, while those for puts are inaccurate for high strikes. Sometimes the mid prices are even smaller than intrinsic value, resulting in a default implied volatility of zero. The L1BAA-OPT-PC methods applies exactly the same principle as L1BA-PC: it prefers put's implied volatility for low strikes and call's implied volatility for high strikes.



(a) Implied volatility slice of AAPL, 1 day to expiration

(b) Implied volatility slice of AAPL, 6 months to expiration

(c) Implied volatility slice of C, 1 day to expiration

(d) Implied volatility slice of C, 6 months to expiration

Figure 3.4: Implied volatility slices in various cases

Chapter 4

Applications

In this chapter, we will shortly go over the applications of the repair methods we discussed. First we show that pre-processing options data to make it arbitrage free results in more robust model calibration. Thereafter, we investigate how the repair method can be used to detect executable arbitrage. The applications we discuss are similar to the ones mentioned in [12], but we extend their analysis by adding the L1BA-PC method (which takes puts into account as well) or pointing out some important details.

4.1 Robust model calibration

In [12] it is shown how removing arbitrage from option price data results in a more robust model calibration. Their methodology is the following:

1. Start with arbitrage-free call price data $\mathbf{c} \in \mathbb{R}^N$.
2. In each iteration $m \in [1, \dots, M]$, generate a noise vector $\boldsymbol{\xi}^{(m)}$ which contains i.i.d. noise $\xi_j^{(m)} \sim \mathcal{N}(0, \sigma_\xi)$. For a proportion $1-\gamma$, replace $\xi_j^{(m)}$ with 0 (so that a proportion γ of call prices will contain noise). Define the noise price $\tilde{\boldsymbol{c}}^{(m)} \in \mathbb{R}^N$ by:
$$\tilde{c}_j^{(m)} = c_j e^{\xi_j^{(m)}}, \quad \forall 1 \leq j \leq N \tag{4.1.1}$$
3. For each m, repair the noise contaminated prices $\tilde{\boldsymbol{c}}^{(m)}$ by using a repair method (such as L1BA), resulting in $\hat{\boldsymbol{c}}^{(m)} := \tilde{\boldsymbol{c}}^{(m)} + \boldsymbol{\epsilon}^{(m)}$.
4. Calibrate an option pricing model to $\tilde{\boldsymbol{c}}^{(m)}$ and $\hat{\boldsymbol{c}}^{(m)}$ separately, resulting in calibration parameters $\tilde{\boldsymbol{\Theta}}^{(m)}$ and $\hat{\boldsymbol{\Theta}}^{(m)}$. To be more specific, we define the calibration objective as $G(\Theta; c) = \sum_{j=1}^N (c_j^\Theta - c_j)^2/\delta_j$, where c_j^Θ is the model price, c_j the market price and δ_j the bid-ask spread. We therefore have:
$$\tilde{\boldsymbol{\Theta}}^{(m)} = \text{argmin}_\Theta G(\Theta; \tilde{\boldsymbol{c}}^{(m)}), \quad \hat{\boldsymbol{\Theta}}^{(m)} = \text{argmin}_\Theta G(\Theta; \hat{\boldsymbol{c}}^{(m)})$$

We can then compare the variation in the arbitrage-contaminated parameters $\tilde{\boldsymbol{\Theta}}^{(m)}$ with the variation observed in the arbitrage-free parameters $\hat{\boldsymbol{\Theta}}^{(m)}$. As expected, [12] reports that for the Heston model, the variation in the parameters is significantly smaller for arbitrage-free data. They conclude that pre-processing option price data by removing arbitrage will result in more robust model calibration.

A difference in our approach is that instead of creating arbitrage-free call prices $\mathbf{c}$ synthetically, we start from the SPX market data introduced in Section 1.2.5 and make it arbitrage-free using the L1 method (see Section 2.2.1). Moreover, to test the effectiveness of the L1BA-PC method, we want to include the put prices as well. We convert the raw put data to synthetic call data using put-call parity. We then apply the L1 method to obtain arbitrage-free synthetic call data, after which we apply noise following (4.1.1). The optimal perturbation $\boldsymbol{\epsilon}$ corresponding L1BA-PC is then obtained by applying L1BA-PC to the noise-contaminated call and noise-contaminated synthetic call data.

We now try to replicate the results in [12] and compare it with our own method L1BA-PC. Instead of calibrating the Heston model, we choose to calibrate the Merton jump-diffusion model, because of its faster calibration properties. This model extends the Black-Scholes model by allowing for occasional price jumps in the underlying. In this model the underlying asset follows a stochastic process that is a combination of geometric Brownian motion and a jump process:

$$S_t = \mu t + \sigma W_t + \sum_{i=1}^{N_t} Y_i, \tag{4.1.2}$$

where $N_t \sim \text{Poisson}(\lambda)$ representing the number of jumps of S_t up to time t and $Y_i \sim \mathcal{N}(\mu_J, \sigma_J^2)$ is the size of each jump. The strength of the model lies in the fact that it accounts for the possibility of extreme market events that cannot be explained by continuous diffusion alone. Similar to other more advanced option pricing models (Heston, Variance-Gamma,...) it is able to capture some of the volatility smile observed in the markets. This model consists of 4 parameters: the diffusion coefficient σ, the jump activity λ, the jump mean μ_J and the jump standard deviation σ_J Therefore, we can write: $\Theta := (\sigma, \lambda, \mu_J, \sigma_J)$.

The result of our calibration procedure is shown as a normalized histogram in Figure 4.1. First of all, we come the same conclusion as in [12]: removing arbitrage from the call price data results in significantly less variation in the obtained model parameters. Moreover, the sample distribution is clearly more centered around the parameter values of the original prices $\mathbf{c}$, indicated with a vertical dotted line. This suggest that using repair methods to remove arbitrage from option quotes is strongly recommended before applying pricing models;

From Figure 4.1, we can also observe that including put price data and using L1BA-PC instead of L1BA results in even less variation and improved calibration robustness. This is not surprising, since the put prices add extra information, especially relevant at low strikes where the call price information is inaccurate. Next to improving the repaired implied volatility surface (as outlined in Section 2.4), this shows the usefulness of the L1BA-PC method.

4.2 Detecting arbitrage

Another useful property of the repair methods is that it can be employed to detect *executable* arbitrages. An executable arbitrage is one of the arbitrage strategies outlined in Section 1.4 (e.g. vertical spread, butterfly, calendar spread,...), but where we take into account that in practice we need to sell at bid and buy at ask. Note that if we simply check $A\mathbf{c} \geq \mathbf{b}$, the arbitrage violations (as shown in Figure 2.2a) are arbitrages that are not necessarily executable, since it assumes that we can buy/sell options at the mid price. However, the arbitrages detected using the repair method are executable. The argument for this, outlined in [12], is the following.

4.2.1 Executable arbitrage opportunities

Define ϵ as the optimal perturbation after applying the L1BA method, the bid-ask spread as $E_j := [c_j - \delta_j, c_j + \delta_j]$ and $N^{\epsilon,\delta}$ as the number of *effectively perturbed prices*, i.e. the number of perturbations resulting in prices outside of the bid-ask spread:

$$N^{\epsilon,\delta} := \sum_{j=1}^{N} \mathbf{1}_{\{|\epsilon_j| > \delta_j\}}. \tag{4.2.1}$$

In [12] the following Theorem is stated.

Theorem 4.2.1. *The presence of effective perturbations (i.e. $N^{\epsilon,\delta} > 0$) indicates that there exist executable arbitrages.*

In other words: $N^{\epsilon,\delta} > 0$ is a sufficient condition to guarantee the existence of executable arbitrage. However, we note that this is only the case if the following critical assumption holds.

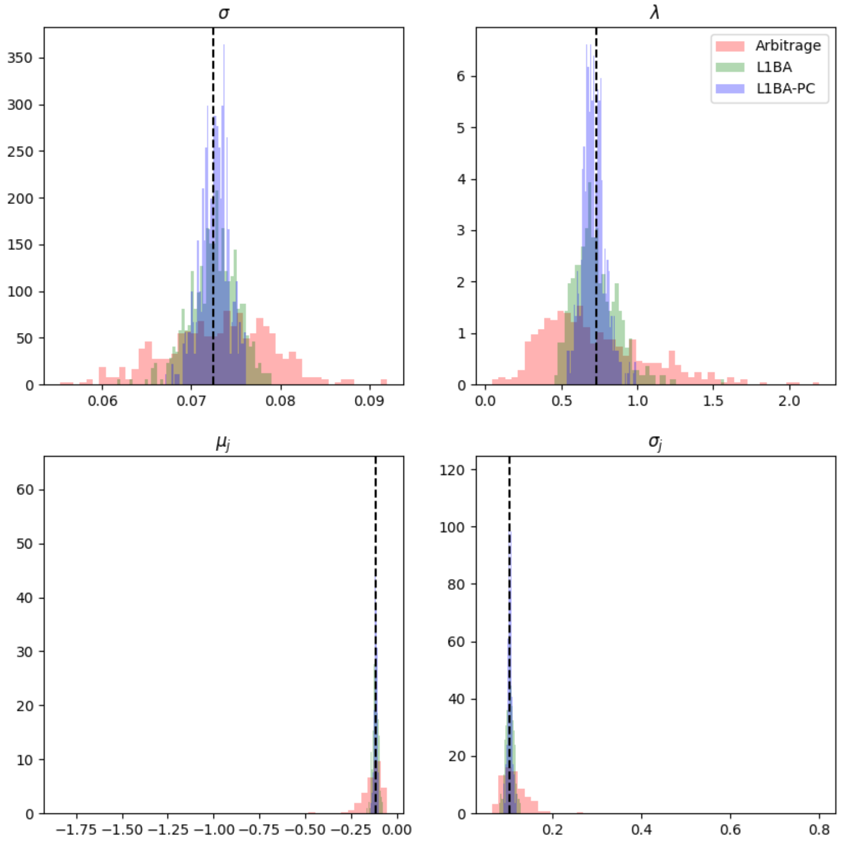


Figure 4.1: Calibration procedure for $M = 500, \sigma_\xi = 0.05, \gamma = 0.8$

Assumption 4.2.2. The repair method L1BA will always prefer a solution with all options inside the bid-ask spread over a solution with at least one option outside the bid-ask spread.

In practice this doesn't always hold, as we will show. More specifically, the value of δ_0 needs to be chosen infinitesimally small to ensure this property. In defense of [12], they later mention that Theorem 4.2.1 holds for *"sufficiently small values of δ_0"*, which is correct. The problem is that the value of δ_0 that they propose doesn't always guarantee Assumption 4.2.2.

To prove Theorem 4.2.1, they give the following argument [12]. First of all, note that $N^{\epsilon,\delta} > 0$ means that for any perturbation $\hat{\mathbf{c}} := \mathbf{c} + \boldsymbol{\epsilon}$:

$$\text{if } \forall i \in [1, M], \sum_{j=1}^{N} a_{ij}\hat{c}_j \geq b_i, \text{then } \exists j \in [1, N] \text{ s.t. } \hat{c}_j \notin E_j. \tag{4.2.2}$$

In other words, the fact that for the optimal perturbations at least one of the quotes is outside the bid-ask spread ($N^{\epsilon,\delta} > 0$), must mean that in order for any perturbed quotes $\hat{\mathbf{c}}$ to be arbitrage-free, at least one of its prices must be outside of the bid-ask spread as well. If this were not the case, the perturbation $\boldsymbol{\epsilon}$ corresponding to $\hat{\mathbf{c}}$ would have been preferred by L1BA (see Assumption 4.2.2) resulting in $N^{\epsilon,\delta} = 0$, thereby reaching a contradiction. The contra-positive statement of (4.2.2) is:

$$\text{if } \forall j \in [1, N], \hat{c}_j \in E_j, \text{then } \exists i^* \in [1, M] \text{ s.t. } \sum_{j=1}^{N} a_{i^*j}\hat{c}_j < b_{i^*}. \tag{4.2.3}$$

This statement expresses that if $N^{\epsilon,\delta} > 0$, but for some perturbation $\hat{\mathbf{c}}$ all quotes are inside the bid-ask spread, then the perturbation $\hat{\mathbf{c}}$ cannot be arbitrage-free. Again, this is because if it were not the case, the perturbation $\boldsymbol{\epsilon}$ corresponding to $\hat{\mathbf{c}}$ would have been preferred by L1BA (see the assumption) resulting in $N^{\epsilon,\delta} = 0$, thereby reaching a contradiction.

The main insight from (4.2.3) is now that in the case of $N^{\epsilon,\delta} > 0$, even if we push all quotes c_j to either its bid $c_j - \delta_j$ or its ask $c_j + \delta_j$, there must still be arbitrage in that perturbed price data. Therefore, it holds that:

$$\sum_{j=1}^{N} a_{i^*j} \left[(c_j + \delta_j)\mathbf{1}_{a_{i^*j} \geq 0} + (c_j - \delta_j)\mathbf{1}_{a_{i^*j} < 0} \right] < b_{i^*}. \tag{4.2.4}$$

This represents an executable arbitrage: we can go long the left-hand side (option positions where we buy at ask and sell at bid) and short the right hand side. This is a portfolio that makes immediate positive profit, while having only non-negative future payoffs. This means that as a byproduct of repairing the volatility surface, we can immediately see whether there is executable arbitrage present as well.

4.2.2 Arbitrage opportunities in practice

However, note again that Assumption 4.2.2 is essential in the argument. In practice, the choice of the parameter δ_0 is crucial. From the definition of the cost function f_j (see (2.2.3)) we see that this parameter punishes the perturbations outside of the bid-ask spread: the smaller we choose δ_0, the harder that perturbations outside of the bid-ask spread are penalized.

In [12] they opt for:

$$\delta_0^{(h)} := \frac{1}{N} \wedge \min_{j=1,\cdots,N} \delta_j. \tag{4.2.5}$$

This ensures that for each f_j, the marginal cost inside the bid-ask spread (δ_0/δ_j) is smaller than outside (1), and [12] considers this sufficient for the Assumption 4.2.2 to hold. However, we will show that this isn't always true. In some edge cases, $\delta_0^{(h)}$ will result in a solution which has perturbations outside of the bid-ask spread, while an arbitrage-free solution with all perturbations inside the bid-ask spread exists. This is clearly a contradiction of Assumption 4.2.2. We propose the following value of δ_0, which is considerably smaller:

$$\delta_0^{(l)} := \left(\min_{j=1,\cdots,N} \delta_j \right) / N \tag{4.2.6}$$

This value of δ_0 gives better guarantees that a the L1BA method will prefer a solution with all perturbations inside the bid-ask spread. However, this value doesn't guarantee that Assumption 4.2.2 holds either. Only if we take δ_0 infinitesimally small, we are theoretically guaranteed for this Assumption to hold. However, this is obviously not feasible in practice, because we are limited by the finite-precision floating point arithmetic of our computer. Also note that we can't take δ_0 to be equal to zero, because in that case the L1BA optimization problem (2.2.4) becomes infeasible.

The only way that we can know for sure that there exists a perturbation within bid-ask that is arbitrage-free, is by adding it as a *hard constraint* instead of a soft constraint. In the development of L1BA (see Section 2.2) it was argued that solutions outside of the bid-ask spread should be possible, because otherwise the optimization problem might be infeasible. The bid-ask spread was added as a soft constraint: perturbations outside of it were allowed, but were punished marginally harder than inside the bid-ask spread. However, we can also choose to only allow solutions inside the bid-ask spread $\boldsymbol{\delta}$ (hard constraint). The extension of L1 with the hard constraint is as follows:

$$\boxed{\begin{aligned} \min_{\boldsymbol{\epsilon}} \quad & \sum_{j=1}^{N} |\epsilon_j| \\ \text{s.t.} \quad & A(\mathbf{c} + \boldsymbol{\epsilon}) \geq \mathbf{b} \\ & -\boldsymbol{\delta} \leq \boldsymbol{\epsilon} \leq \boldsymbol{\delta} \end{aligned}} \tag{4.2.7}$$

This might make the problem infeasible: in some cases it is impossible to repair the call options without putting some quotes outside the bid-ask spread. If this is the case, it means that all perturbations within bid-ask must contain arbitrage. Therefore, executable arbitrages must exist (since in such a strategy we sell at bid and buy at ask, making the resulting perturbation within bid-ask). Also note that in the limit (δ_0 infinitesimally small), L1BA (2.2.4) and 4.2.7 are equivalent, because perturbations outside bid-ask are made infinitely expensive.

We now use SPX options data gathered between 2023/06/16 and 2023/07/13 to check whether executable arbitrage is detected accurately. Each hour during market open, a snapshot is made and the following procedure is executed:

- We repair the option quotes using L1BA with parameter $\delta_0^{(h)}$
- We repair the option quotes using L1BA with parameter $\delta_0^{(l)}$
- We solve the optimization problem 4.2.7, thereby checking whether an arbitrage-free solution within bid-ask exists (resulting in an executable arbitrage opportunity)

The result is shown in Figure 4.2. For each method, we show when the method flags an executable arbitrage opportunity (i.e. $N^{\epsilon,\delta} > 0$). We clearly observe here that L1BA using $\delta_0^{(1)}$ often flags executable arbitrage when it doesn't exist (i.e. a *false positive*). As discussed, this is because for high values of δ_0, Assumption 4.2.2 doesn't necessarily hold, meaning that L1BA doesn't necessarily always choose a solution inside the bid-ask spread, even though there might be such a solution. Because of this, the statement '$N^{\epsilon,\delta} > 0$' doesn't always imply the existence of executable arbitrage. In the second plot, we see that decreasing δ_0 results in less false positives. However, in one case it still resulted in a solution outside the bid-ask spread, while a solution inside exists. In order to overcome this, δ_0 should be chosen even smaller. In Figure 4.3 we show an example of an edge case in which L1BA with insufficiently small δ_0 results in a violation of Assumption

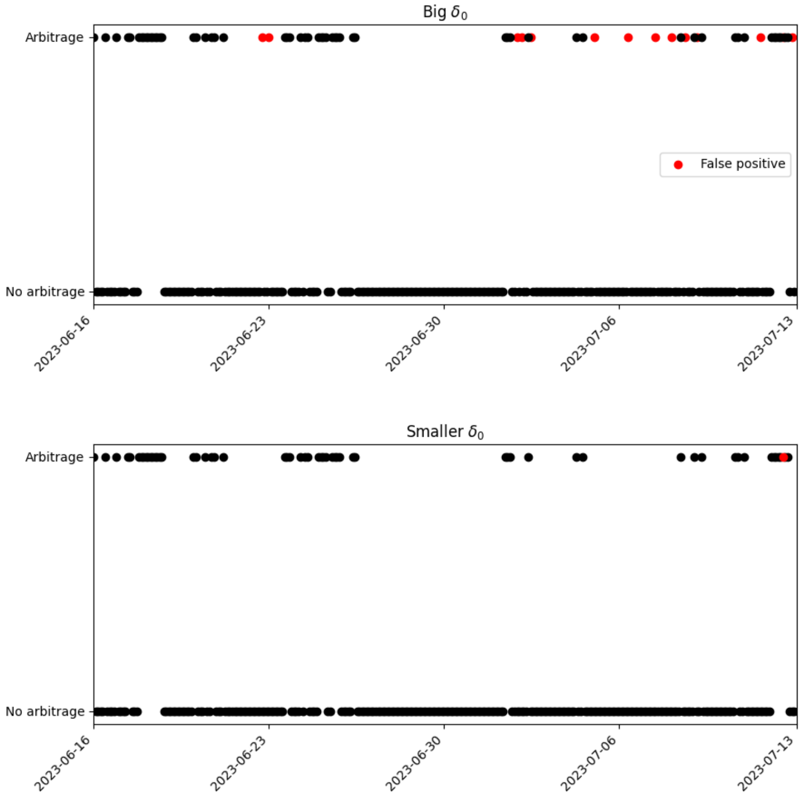


Figure 4.2: Effect of δ_0 on the capability of detecting executable arbitrage

4.2.2: a solution outside the bid-ask spread is preferred (black dotted line), while a solution inside the bid-ask spread exists. In this case, using the $\delta_0^{(h)}$ results in a flagging of executable arbitrage, while this is not true. However, we see that by decreasing δ_0, a solution inside the bid-ask spread is found, thereby preventing the false positive signal. We also observe that by taking δ_0 to be small enough, L1BA is basically equivalent to the hard-constrained problem (4.2.7), which corresponds to the fact that both are the same for infinitesimally small δ_0. We conclude that the L1BA method

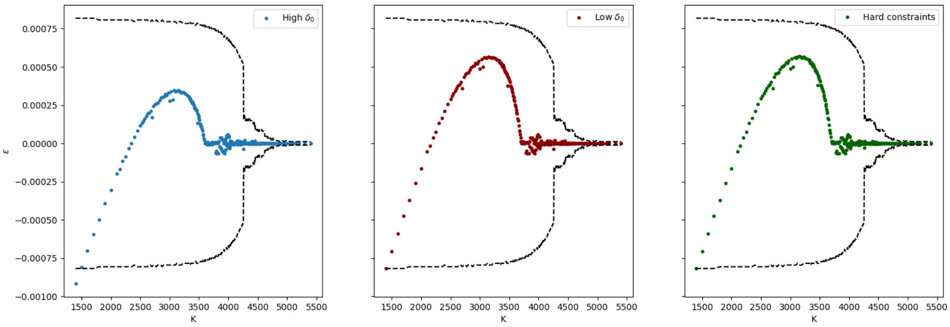


Figure 4.3: Example in which a parameter value of δ_0 that is too large results in a (false positive) detection of executable arbitrage

can therefore be used as a reliable indicator to check whether executable arbitrage exists. However, in order to do so, δ_0 needs to be chosen small enough to prevent false positives.

Bibliography

[1] Y. Ait-Sahalia and J. Duarte. Nonparametric option pricing under shape restrictions. *Journal of Econometrics*, 116(1-2):9–47, 2003.

[2] G. Bakshi and N. Kapadia. Volatility risk premiums embedded in individual equity options: Some new insights. *The Journal of Derivatives*, 11(1):45–54, 2003.

[3] G. Barone-Adesi and R. E. Whaley. Efficient analytic approximation of american option values. *the Journal of Finance*, 42(2):301–320, 1987.

[4] A. Bensoussan. On the theory of option pricing. *Acta Applicandae Mathematica*, 2:139–158, 1984.

[5] P. Bjerksund and G. Stensland. Closed form valuation of american options. 2002.

[6] F. Black. The pricing of commodity contracts. *Journal of financial economics*, 3(1-2):167–179, 1976.

[7] F. Black and M. Scholes. The pricing of options and corporate liabilities. *Journal of political economy*, 81(3):637–654, 1973.

[8] D. Brigo. Lecture notes on interest rate models, 2023.

[9] O. Burkovska, M. Gaß, K. Glau, M. Mahlstedt, W. Schoutens, and B. Wohlmuth. Calibration to american options: numerical investigation of the de-americanization method. *Quantitative finance*, 18(7):1091–1113, 2018.

[10] P. Carr, R. Jarrow, and R. Myneni. Alternative characterizations of american put options. *Mathematical Finance*, 2(2):87–106, 1992.

[11] S. N. Cohen, C. Reisinger, and S. Wang. arbitragerepair. `https://github.com/vicaws/arbitragerepair`, 2020. DOI: 10.5281/zenodo.5338299.

[12] S. N. Cohen, C. Reisinger, and S. Wang. Detecting and repairing arbitrage in traded option prices. *Applied Mathematical Finance*, 27(5):345–373, 2020.

[13] J. C. Cox, S. A. Ross, and M. Rubinstein. Option pricing: A simplified approach. *Journal of financial Economics*, 7(3):229–263, 1979.

[14] E. Derman and I. Kani. Riding on a smile. *Risk*, 7(2):32–39, 1994.

[15] B. Dupire et al. Pricing with a smile. *Risk*, 7(1):18–20, 1994.

[16] M. R. Fengler. Arbitrage-free smoothing of the implied volatility surface. *Quantitative Finance*, 9(4):417–428, 2009.

[17] J.-P. Fouque, G. Papanicolaou, and K. R. Sircar. From the implied volatility skew to a robust correction to black-scholes american option prices. *International Journal of Theoretical and Applied Finance*, 4(04):651–675, 2001.

[18] V. Frishling. A discrete question. *Risk Magazine*, January 2002.

[19] J. Gatheral and A. Jacquier. Arbitrage-free svi volatility surfaces. *Quantitative Finance*, 14(1):59–71, 2014.

[20] E. G. Haug and N. N. Taleb. Why we have never used the black-scholes-merton option pricing formula. *Social Science Research Network Working Paper Series*, 1(4), 2008.

[21] L. Hentschel. Errors in implied volatility estimation. *Journal of Financial and Quantitative analysis*, 38(4):779–810, 2003.

[22] C. Homescu. Implied volatility surface: Construction methodologies and characteristics. `https://arxiv.org/abs/1107.1834`, 2011.

[23] A. Ivanovas. Option data, missing tails, and the intraday variation of implied moments. 2014.

[24] M. S. Joshi. *The concepts and practice of mathematical finance*, volume 1. Cambridge University Press, 2003.

[25] T. Klassen. Pricing vanilla options with cash dividends. In *Pricing Vanilla Options with Cash Dividends: Klassen, Timothy.* [Sl]: SSRN, 2015.

[26] D. P. Leisen and M. Reimer. Binomial models for option valuation-examining and improving convergence. *Applied Mathematical Finance*, 3(4):319–346, 1996.

[27] H. Lim. Improved methods for implied volatility surface and implied distributions. `https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3561100`, 2020.

[28] F. A. Longstaff and E. S. Schwartz. Valuing american options by simulation: a simple least-squares approach. *The review of financial studies*, 14(1):113–147, 2001.

[29] P. Meier. *Essays on pricing kernel estimation, option data filtering and risk-neutral density tail estimation.* PhD thesis, 2015.

[30] R. C. Merton. Theory of rational option pricing. *The Bell Journal of economics and management science*, pages 141–183, 1973.

[31] J. Ruf and W. Wang. Neural networks for option pricing and hedging: a literature review. `https://arxiv.org/abs/1911.05620`, 2019.

[32] C. Salvi. Lecture notes on numerical methods in finance, 2023.

[33] M. Sander. *Bondesson's Representation of the Variance Gamma Model and Monte Carlo Option Pricing.* Lund University, 2009.

[34] S. Shreve. *Stochastic calculus for finance I: the binomial asset pricing model.* Springer Science & Business Media, 2005.

[35] S. E. Shreve et al. *Stochastic calculus for finance II: Continuous-time models*, volume 11. Springer, 2004.

[36] B. M. Smith. *A history of the global stock market: from ancient Rome to Silicon Valley.* University of Chicago press, 2004.

[37] S. Stoikov. The micro-price: a high-frequency estimator of future prices. *Quantitative Finance*, 18(12):1959–1966, 2018.

[38] A. M. Vijh. Liquidity of the cboe equity options. *The Journal of Finance*, 45(4):1157–1179, 1990.

[39] D. Williams. *Probability with martingales.* Cambridge university press, 1991.

[40] L. Wu and Y.-K. Kwok. A front-fixing finite difference method for the valuation of american options. *Journal of Financial Engineering*, 6(4):83–97, 1997.