Detecting and repairing arbitrage in traded option prices

Preprint · August 2020

3 authors, including:

Detecting and repairing arbitrage in traded option prices

Samuel N. Cohen   Christoph Reisinger
Sheng Wang
Mathematical Institute, University of Oxford
{samuel.cohen, christoph.reisinger, sheng.wang}
@maths.ox.ac.uk

August 24, 2020

Abstract

Option price data are used as inputs for model calibration, risk-neutral density estimation and many other financial applications. The presence of arbitrage in option price data can lead to poor performance or even failure of these tasks, making pre-processing of the data to eliminate arbitrage necessary. Most attention in the relevant literature has been devoted to arbitrage-free smoothing and filtering (i.e. removing) of data. In contrast to smoothing, which typically changes nearly all data, or filtering, which truncates data, we propose to repair data by only necessary and minimal changes. We formulate the data repair as a linear programming (LP) problem, where the no-arbitrage relations are constraints, and the objective is to minimise prices' changes within their bid and ask price bounds. Through empirical studies, we show that the proposed arbitrage repair method gives sparse perturbations on data, and is fast when applied to real world large-scale problems due to the LP formulation. In addition, we show that removing arbitrage from prices data by our repair method can improve model calibration with enhanced robustness and reduced calibration error.

1 Introduction

Price data of vanilla options are widely used in various financial applications such as calibrating models for pricing and hedging, and computing risk-neutral densities (RND) of the underlying. The presence of arbitrage in option price data can lead to poor or even failed model calibration, as well as erroneous RND estimation. Derivative pricing models in nearly all applications are constructed to be arbitrage-free, as it is economically meaningless to have a model that has the potential to make risk-free profits. Exact calibration is impossible for any arbitrage-free model or RND function with any size of input data if the price data contain arbitrage. For example, calibration of the local volatility model of Dupire [15] and Derman and Kani [12] would fail given arbitrageable data. It is also not possible to have any arbitrage-free interpolation such as Kahale [26] and Wang et al. [36], because the data to be interpolated are not arbitrage-free from the beginning. Though inexact calibration methods are available for many models, it seems natural to expect that removing arbitrage from input data can improve calibration of arbitrage-free models, such as enhancing robustness or reducing calibration error.

Therefore, it is important to remove arbitrage (if present) from option price data. Most attention in the relevant literature has been devoted to the *smoothing* and *filtering* of data. Notable works on smoothing include Aït-Sahalia and Duarte [1], Fengler et al. [16] [17] [18], Gatheral and Jacquier [19], and Lim [29]. In fact, the calibration of many pricing models, such as stochastic volatility models, is essentially arbitrage-free smoothing. Arbitrage-free data is only a byproduct of smoothing, since the main goal of smoothing is to produce a $C^{1,2}$ call price function $(T, K) \mapsto C(T, K)$ (or equivalently implied volatility function $(T, K) \mapsto \sigma_{\mathrm{imp}}(T, K)$). For smoothing, usually an ℓ^2-norm optimisation is used when searching over polynomial, spline or kernel parameters that produce values as close to the given price data as possible. This method leads to changes for nearly *all* data. Though liquidity considerations can be included by adding weights in the optimisation, it remains unclear what should be an effective way to set weight values for different options. The filtering of data refers to simply removing suspiciously low-quality price data according to criteria in terms of moneyness, expiry, trading volume, intra-day activity, etc. A good survey of popular empirical filtering criteria can be found in Ivanovas [25] and Meier [31]. Filtering can be quite subjective, can cause information loss, and might not even be feasible as many criteria are based on order-book level data, which are not always available (for example, in OTC markets).

In contrast to smoothing, which typically changes nearly all data, or filtering, which truncates data, we propose to *repair* data in the sense that only necessary and minimal changes are made to the given data in the presence of arbitrage. If arbitrages in data are mainly consequences of infrequent price updates of illiquid options rather than noncompetitive market, it is better to only perturb as few data points as possible. In addition, when making changes, we use bid and ask prices as soft bounds such that liquidity profiles of different options are considered in an objective way. Bid-ask spread is a measure of liquidity, i.e. the

2

narrower the spread, the easier a market order can be matched and executed. Since the "fair" price could lie anywhere within the bid-ask price bounds, the width of the bid-ask spread represents the degree of certainty in the market prices. Empirically, deep out-of-the-money (OTM) and in-the-money (ITM) options are thinly traded with wide bid-ask spreads, leading to less trustworthy price data compared with more liquid options. We therefore formulate the data repair as a constrained optimisation problem, where the no-arbitrage relations are written as constraints, and the objective is to minimise price changes within soft bounds. By carefully choosing the objective function, we can rewrite the formulation as a linear programming (LP) problem, so that we can take advantage of efficient solution techniques and software for large-scale LPs.

Our method is to repair single-price data. At any moment during the trading day, each tradable asset has multiple prices, i.e. bid price and ask price. However, most applications require single-price inputs. There is a need to construct some "fair" reference price from the market-quoted multiple prices. Examples of a reference price are the mid-price, the quantity-weighted price, the last trade price or the micro-price by Stoikov [35]. In this article, we do not discuss the construction of reference price, and use the mid-price by default, however other reference prices could easily be considered.

We envisage further applications of our methodology in repairing data generated by models which do not themselves rule out arbitrage. Included in this class are prices predicted by deep learning methods, which have gained substantial popularity recently, as documented by the survey paper by Ruf and Wang [33]. Typically, there is no guarantee for arbitrage-free predicted option prices even if the training set is arbitrage free; see also a more detailed discussion of this point in the introduction of Dixon, Crépey and Chataigner [13], which goes on to use the local volatility code book for arbitrage free vanilla prices as a means of guaranteeing arbitrage-free interpolation of prices. The arbitrage repair method from our paper can provide a simple post-processing step of potentially arbitrageable learned prices. By repairing a discrete set of input prices directly without extra assumptions, using linear constraints only, the method distinguishes itself by versatility, transparency, and speed, making it particularly well-suited to online computations.

The rest of the paper is structured as follows. We derive a set of empirically verifiable model-independent, static arbitrage constraints in Section 2. Our derivation is mainly based on Carr, Géman, Madan and Yor [6] [7], Davis and Hobson [10], and Cousot [8] [9]. In Section 3, we formulate data repair as a constrained LP problem, and the design of the objective function is carefully discussed[1]. Finally in Section 4, we apply our arbitrage repair method to FX option data to justify why arbitrage repair is needed for real data, and demonstrate how our method performs empirically on various metrics, especially on the improvement of model calibration. We also show an example of how we can use our repair method for identifying the formation and disappearance of

---

[1]Our implementation of this algorithm in Python is available in the repository https://github.com/vicaws/arbitragerepair.

executable arbitrage in intra-day price data.

2 Arbitrage constraints

We consider a finite collection of traded European call options[2] written on the
same asset. These options can have arbitrary expiry and strike parameters
rather than a rectangular grid of parameters, a restrictive prerequisite for many
arbitrage detection [7] and spline-type smoothing methods [16] [26] to work.
In practice, it is uncommon to have price data on a rectangular grid, see, e.g.
Figure 1.

Consider N European call options that have expiries $0 < T_1 < T_2 < \cdots <
T_m$. For a given expiry T_i, available strikes are $0 < K_1^i < K_2^i < \cdots < K_{n_i}^i$. The
(i,j)-th option has reference price C_j^i at present time 0, and terminal payoff
$(S_{T_i} - K_j^i)^+$, where S_t denotes the price of the underlying asset at time t. Hence
$N = \sum_{i=1}^m n_i$. Denote $\mathcal{T}^e = \{T_i\}_{1 \le i \le m}$ and $\mathcal{P}^{T,K} = \{(T_i, K_j^i)\}_{1 \le j \le n_i, 1 \le i \le m}$.
In Figure 1, we show how (T, K) are distributing for traded call options on
a typical trading day. A detailed description of the data used can be seen in
Section 4.
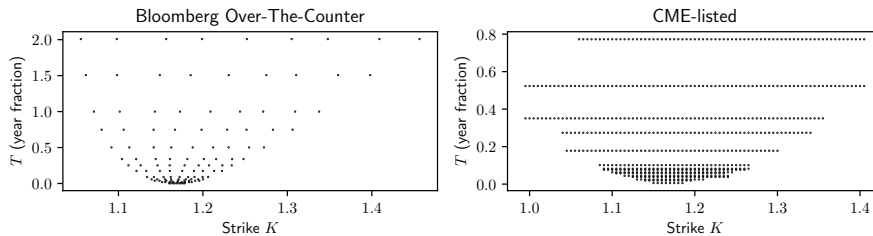
Figure 1: Distributions of (K, T) for traded EURUSD call options in the OTC market
(Bloomberg data) and at the CME market, observed as of 31st May, 2018.

2.1 Assumptions

The arbitrage constraints that price data should satisfy are derived under a
frictionless market assumption. As a consequence, when the price data break
these constraints, it may not be possible in practice to exploit the apparent
arbitrage, given practical market barriers and transaction costs. However, the
assumption that prices should be arbitrage free is justified by the fact that the
single-price data are not executable prices in the market, but are designed to
be reference or benchmark prices for tradable assets, which are useful inputs to

---

[2]We focus on European style vanilla options in this study. Specifically, we only consider
call options, since the static arbitrage constraint between call and put options is the put-call
parity, which can be easily incorporated in our approach. The framework of our arbitrage
repair method is applicable to a mixture of a wider range of options, as long as their arbitrage
constraints can be defined by feasible linear inequalities of prices.

a variety of models. Nevertheless, we will use bid and ask prices as soft bounds for guiding our arbitrage repair mechanism.

We allow for non-zero but deterministic interest and dividends[3]. At present time 0, we use $D(T)$ to denote the market discount factor for time T, and $\Gamma(T)$ to denote the number of shares which will be owned by time T if dividend income is invested in shares. Then there is a model-independent, arbitrage-free forward price, $F(T) = S_0/(\Gamma(T)D(T))$, for delivery of the asset at T.

We assume that zero-coupon bonds and forward contracts on the risky asset, with the same expiries as the options, are traded in the market. In addition, they are sufficiently liquid that we can neglect their bid-ask spreads (e.g. usually one or two ticks). Therefore, we observe market discount factors $D_i := D(T_i)$ and forward prices $F_i := F(T_i)$ for $1 \le i \le m$. However, when the underlying (spot or forward) trades at a sufficiently large bid-ask spread, then any arbitrage strategy can become impossible (see the discussion by Gerhold and Gülüm [20]).

2.2   Static arbitrage

Arbitrage refers to a costless trading strategy that has a positive probability of earning risk-free profit. A *static arbitrage* is an arbitrage exploitable by fixed positions in options and the underlying stock at initial time, while the position of the underlying stock can be modified at a finite number of trading times in the future. Any other arbitrage is called *dynamic arbitrage*. As an example of a static arbitrage, it must hold the condition that $C_1^1 \ge C_2^1$ for $K_1^1 < K_2^1$, otherwise by going long one (T_1, K_1^1) option and short one (T_1, K_2^1) option, we make immediate profit of $C_2^1 - C_1^1$ with non-negative terminal payoff. An example of dynamic arbitrage is a continuously delta-hedged short position on an over-priced option in the perfect Black–Scholes world.

Dynamic arbitrage relies on dynamics and path properties of the tradable assets. From the data repair perspective, we should minimise model dependence, because the repaired data are to be used in more generic applications. Hence, data should only be adjusted by model-independent constraints, so we restrict ourselves to static arbitrage in which no dynamics need to be modelled. Static arbitrage constraints establish the prerequisites that the price data have to satisfy at time zero for admitting a dynamically arbitrage-free model.

A *model* $\mathbb{M}$ is a filtered probability space $(\Omega, \mathscr{F}, \{\mathscr{F}_t\}_{t \in \mathcal{T}}, \mathbb{P})$, that carries an adapted price process $\{(S_t, \mathbf{C}_t)\}_{t \in \mathcal{T}}$, where $\mathbf{C}_t$ gives the prices of the N options at time t, and we observe $\mathbf{C}_0$. Here $\mathcal{T}$ denotes the set of times at which the asset can be traded so that $0 \in \mathcal{T}$ and $\mathcal{T}^e \subset \mathcal{T}$, and $\mathscr{F}_0 = \{\Omega, \emptyset\}$ augmented with all null sets of $\mathscr{F}_{T_m}$.

The First Fundamental Theorem of Asset Pricing (FFTAP) establishes an equivalence relation between no-arbitrage (static and dynamic) and the existence of an equivalent martingale measure (EMM). After the landmark work of Harrison and Kreps [22], there are various versions of the FFTAP and extensions of the no-arbitrage concept (e.g. no free lunch by Kreps [28], no free

---

[3]When applying our method to other asset classes, dividends of stock shares are comparable to foreign currency interest rates for FX rates, or convenience yields for commodities.

lunch with vanishing risk by Delbaen and Schachermayer [11]). In this article, we work with a simplified version of FFTAP as follows. Given a model $\mathbb{M}$, there is no arbitrage if and only if $\exists \mathbb{Q} \sim \mathbb{P}$, such that

$$\forall (T,K) \in \mathcal{P}^{T,K} \cup (\mathcal{T}^e \times \{0\}), \ D(t)C_t(T,K) = D(s)\mathbb{E}^{\mathbb{Q}}\left[C_s(T,K)|\mathscr{F}_t\right] \quad (1)$$

for all $t < s \leq T$ where $t, s \in \mathcal{T}$. No static arbitrage corresponds to a much smaller set of conditions, since the path dynamics governed by $\mathbb{Q}$ no longer matter. As discussed by Carr, Géman, Madan and Yor [6], [7] and Davis [10], static arbitrage is present if no $\mathbb{Q}$ exists such that $C_0(T,K) = D(T)\mathbb{E}^{\mathbb{Q}}[C_T(T,K)|\mathscr{F}_0]$. Therefore, static arbitrage constraints are consequences of relations between terminal payoffs, projected to the present time.

2.3 Shape constraints of the call price surface

Let us define $M_{T_i} = S_{T_i}/F_i$, $k_j^i = K_j^i/F_i$, $c_j^i = C_j^i/(D_i F_i)$, for all i, j. To have no static arbitrage, there must exist $\mathbb{Q}$ such that

$$c_j^i = \mathbb{E}^{\mathbb{Q}}\left[\left.\left(\frac{S_{T_i}}{F_i} - \frac{K_j^i}{F_i}\right)^+ \right| \mathscr{F}_0 \right] = \mathbb{E}^{\mathbb{Q}}\left[\left.\left(M_{T_i} - k_j^i\right)^+ \right| \mathscr{F}_0 \right], \quad \forall i, j.$$

We will work on "normalised" quantities M, k, c in the rest of this section. We define the *normalised call function* $c(T,k)$ as

$$c(T,k) := \mathbb{E}^{\mathbb{Q}}\left[\left. (M_T - k)^+ \right| \mathscr{F}_0 \right], \text{ where } T \in \mathbb{R}_{>0}, \ k \in \mathbb{R}_{\geq 0}. \quad (2)$$

Given the specific structure in (2), a probability measure $\mathbb{Q}$ exists only when the call function satisfies some shape constraints. For arbitrary but fixed T, using Breeden and Litzenberger's analysis [4], the marginal measure $\mathbb{Q}_T := \mathbb{Q}(\cdot|\mathscr{F}_T)$ exists if

$$\forall \ k_3 > k_2 > k_1 \geq 0, \ \ -1 \leq \frac{c(T,k_2) - c(T,k_1)}{k_2 - k_1} \leq \frac{c(T,k_3) - c(T,k_2)}{k_3 - k_2} \leq 0.$$

If a family of marginal measures $\{\mathbb{Q}_T\}_{T \in \mathcal{T}^e}$ on $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$ exists with time-independent mean, and $\mathbb{Q}_{T_1} \geq_{\mathrm{cx}} \mathbb{Q}_{T_2}$ whenever $T_1 > T_2$, then, by Kellerer's theorem [27], there exists a Markov martingale measure with these marginals. Here we write $\mathbb{Q}_{T_1} \geq_{\mathrm{cx}} \mathbb{Q}_{T_2}$ if $\int_{\mathbb{R}} \phi \, d\mathbb{Q}_{T_1} \geq \int_{\mathbb{R}} \phi \, d\mathbb{Q}_{T_2}$ for each convex function $\phi : \mathbb{R} \to \mathbb{R}$, and we say $\{\mathbb{Q}_T\}_{T \in \mathcal{T}^e}$ is Non-Decreasing in Convex Order (NDCO). The convex order can be equivalently characterised in terms of the call function [34]:

$$\mathbb{Q}_{T_1} \geq_{\mathrm{cx}} \mathbb{Q}_{T_2} \Longleftrightarrow \begin{cases} \mathbb{Q}_{T_i} \text{ and } \mathbb{Q}_{T_j} \text{ have equal means;} \\ \displaystyle\int_{\mathbb{R}} (x-k)^+ \, d\mathbb{Q}_{T_1} \geq \int_{\mathbb{R}} (x-k)^+ \, d\mathbb{Q}_{T_2} \quad \forall x \in \mathbb{R}. \end{cases}$$

Given that $\mathbb{E}^{\mathbb{Q}_T}[M_U] = \mathbb{E}^{\mathbb{Q}_T}[S_U/F_T(U)] = 1$ is time-independent for any $T < U$ where $T, U \in \mathcal{T}^e$, it is then sufficient to conclude that $\{\mathbb{Q}_T\}_{T \in \mathcal{T}^e}$ is NDCO if

$c(T, \cdot) \leq c(U, \cdot)$. Also note that $\lim_{k\downarrow 0} c(T, k) = \mathbb{E}^{\mathbb{Q}}[M_T | \mathscr{F}_0] = M_0 = 1$ for any T, and by monotonicity we have $0 \leq c(T, k) \leq 1$. Hence, if we define a set of functions $s(x, y) : X \times Y \to \mathbb{R}$, where $X, Y \subseteq \mathbb{R}_{\geq 0}$ are compact sets, by

$$
\mathcal{S}(X \times Y) = \left\{ (x, y) \mapsto s(x, y) : \forall\ x_1 < x_2 \in X,\ y_1 < y_2 < y_3 \in Y, \right.
$$
$$
0 \leq s \leq 1,\ s(x_1, \cdot) \leq s(x_2, \cdot), \tag{3}
$$
$$
\left. -1 \leq \frac{s(\cdot, y_2) - s(\cdot, y_1)}{y_2 - y_1} \leq \frac{s(\cdot, y_3) - s(\cdot, y_2)}{y_3 - y_2} \leq 0 \right\},
$$

then no arbitrage can be constructed on the static surface $(T, k) \mapsto c(T, k)$ if $c \in \mathcal{S}(\mathbb{R}_{>0} \times \mathbb{R}_{\geq 0})$. Consequently, no static arbitrage can be constructed from the finite collection of prices if

$$
\exists c \in \mathcal{S}\left(\mathcal{T}^e \times [0, \max_{i,j} k_j^i] \right),\ \text{s.t.}\ \forall (T_i, k_j^i) \in \mathcal{P}^{T,k},\ c(T_i, k_j^i) = c_j^i, \tag{4}
$$

where $\mathcal{P}^{T,k} = \{ (T_i, k_j^i) \}_{1 \leq j \leq n_i, 1 \leq i \leq m}$.

Condition (4) can be characterised by practically verifiable constraints of prices $\mathbf{c}$. We slightly revise Cousot's construction (Definition 2.1 – 2.3 in [9]). We augment the given price data with the price that corresponds to a call struck at 0 for each expiry. This means $\forall i \in \{1, \cdots, m\}$ we add $K_0^i = 0$ and $C_0^i = F_i$, or equivalently $k_0^i = 0$ and $c_0^i = 1$. This augmentation is necessary to check arbitrage relationships between call options and forwards. Define, for any $k_{j_1}^{i_1} > k_{j_2}^{i_2}$, where $1 \leq i_1, i_2 \leq m$, $0 \leq j_1 \leq n_{i_1}$, and $0 \leq j_2 \leq n_{i_2}$,

$$
\beta(i_1, j_1; i_2, j_2) := \frac{c_{j_1}^{i_1} - c_{j_2}^{i_2}}{k_{j_1}^{i_1} - k_{j_2}^{i_2}}, \tag{5}
$$

which can be viewed as the slope of the straight line passing through the two points $(k_{j_1}^{i_1}, c_{j_1}^{i_1})$ and $(k_{j_2}^{i_2}, c_{j_2}^{i_2})$, if we plot all prices on the (k, c) plane. We will employ $\beta(\cdot)$ to define the price of some *test strategies*.

Definition 1. *A test spread strategy is defined* $\forall 1 \leq i_1 \leq i_2 \leq m$, *and* $\forall 0 \leq j_1 \leq n_{i_1}$, $0 \leq j_2 \leq n_{i_2}$ *such that* $k_{j_1}^{i_1} \geq k_{j_2}^{i_2}$, *by*

$$
S_{j_1, j_2}^{i_1, i_2} = \begin{cases} -\beta(i_1, j_1; i_2, j_2) & \textit{if } k_{j_1}^{i_1} > k_{j_2}^{i_2}, \\ c_{j_2}^{i_2} - c_{j_1}^{i_1} & \textit{if } k_{j_1}^{i_1} = k_{j_2}^{i_2}. \end{cases}
$$

In particular, there are three types of test spread strategies:

(1) *Vertical spread:* $VS_{j_1, j_2}^{i} = S_{j_1, j_2}^{i, i}$ *with* $k_{j_1}^{i} > k_{j_2}^{i}$.

(2) *Calendar spread:* $CS_{j}^{i_1, i_2} = S_{j_1, j_2}^{i_1, i_2}$ *with* $k_{j_1}^{i_1} = k_{j_2}^{i_2}$ *and* $i_1 < i_2$.

(3) *Calendar vertical spread:* $CVS_{j_1, j_2}^{i_1, i_2} = S_{j_1, j_2}^{i_1, i_2}$ *with* $k_{j_1}^{i_1} > k_{j_2}^{i_2}$ *and* $i_1 < i_2$.

Definition 2. *A test butterfly strategy is defined $\forall i, i_1, i_2 \in [1, m]$ s.t. $i \leq i_1$ and $i \leq i_2$, $\forall j \in [0, n_i]$, $j_1 \in [0, n_{i_1}]$, $j_2 \in [0, n_{i_2}]$ such that $k_{j_1}^{i_1} < k_j^i < k_{j_2}^{i_2}$, by*

$$B_{j,j_1,j_2}^{i,i_1,i_2} = -\beta(i, j; i_1, j_1) + \beta(i_2, j_2; i, j).$$

In particular, there are two types of test butterfly strategies:

(1) Vertical butterfly: $VB_{j,j_1,j_2}^i = B_{j,j_1,j_2}^{i,i,i}$.

(2) Calendar butterfly: $CB_{j,j_1,j_2}^{i,i_1,i_2} = B_{j,j_1,j_2}^{i,i_1,i_2}$ where i, i_1 and i_2 are not all equal.

Based on these definitions of test strategies, we restate Cousot's constraints for no-arbitrage in the following proposition.

Proposition 1 (Cousot [8], [9])**.** *All test strategies are non-negative, and all test vertical spreads are not greater than 1, if and only if there exist m risk-neutral measures $\{\mathbb{Q}_{T_i}\}_{1 \leq i \leq m}$ corresponding to all option expiries, that are NDCO. In addition, all their means are equal to $M_0 = 1$.*

Together with Kellerer's theorem [27], Proposition 1 gives sufficient conditions for the existence of a $\mathbb{Q}$-martingale (thus no static arbitrage), in terms of constraints on prices of the test strategies. Those constraints are also necessary for no static arbitrage if semi-static strategies are allowed to exploit arbitrage opportunities, as proved by Cousot in Appendix A of [8].

2.4 Constraints reduction

Cousot's constraints contain redundancies. For instance, if two vertical spreads VS_{j_2,j_1}^i and VS_{j_3,j_2}^i (where $k_{j_1}^i < k_{j_2}^i < k_{j_3}^i$) are non-negative, then $VS_{j_3,j_1}^i \geq 0$ holds automatically. We will reduce the number of constraints from $\mathcal{O}(N^3)$ to $\mathcal{O}(m^2 N)$ by localisation on the surface. Localisation can successfully reduce the amount of constraints because the shape constraints specified in (3) include only boundedness, positivity, monotonicity and convexity, which are all local properties. The reduced set of constraints is listed in Table 1, where the (order of the) number of constraints in each category is also indicated.

We give details of the localisation method in Appendix A. We claim that the reduced set of constraints listed in Table 1 are sufficient to imply Cousot's constraints, thus are sufficient and necessary to guarantee no-arbitrage, as stated in Proposition 2.

Proposition 2. *If the constraints $C1 - C6$ are satisfied, then all test strategies are non-negative, and all test vertical spreads are not greater than 1.*

Proof. See Appendix B. □

3 Arbitrage repair

The static arbitrage constraints in Table 1 are linear inequalities of at most three call prices. Therefore, we can write these constraints in the form $A\mathbf{c} \geq \mathbf{b}$, where

Category	Constraints	Number
C1 Outright	$\forall i \in [1,m],\ c_{n_i}^i \geq 0$	m
C2 Vertical spread	$\forall i \in [1,m],\ j \in [1,n_i],$ $\mathrm{VS}_{j,j-1}^i \geq 0$ and $\mathrm{VS}_{1,0}^i \leq 1$	$N+m$
C3 Vertical butterfly	$\forall i \in [1,m],\ j \in [1,n_i-1],\ \mathrm{VB}_{j,j-1,j+1}^i \geq 0$	$N-m$
C4 Calendar spread	$\forall 1 \leq i_1 < i_2 \leq m,\ j_1 \in [0,n_{i_1}],\ j_2 \in [0,n_{i_2}],$ $\mathrm{CS}_{j_1,j_2}^{i_1,i_2} \geq 0$	$\mathcal{O}(mN)$
C5 Calendar vertical spread	$\forall i^* \in [1,m],\ j^* \in [1,n_{i^*}],$ define $\mathcal{I} := \{i,j : T_i > T_{i^*},\ k_{j^*-1}^{i} < k_j^i < k_{j^*}^{i}\},$ then $\forall i,j \in \mathcal{I},\ \mathrm{CVS}_{j^*,j}^{i^*,i} \geq 0$	$\mathcal{O}(mN)$
C6.1 Calendar butterfly I (Absolute location convexity)	$\forall i^* \in [1,m],\ j^* \in [1,n_{i^*}-1],$ define $\mathcal{I} := \{i,j : T_i > T_{i^*},\ k_{j^*-1}^{i} < k_j^i < k_{j^*}^{i}\},$ then $\forall i,j \in \mathcal{I},\ \mathrm{CB}_{j^*,j,j^*+1}^{i^*,i,i^*} \geq 0;$ $\forall i^* \in [1,m],\ j^* \in [2,n_{i^*}],$ define $\mathcal{I} := \{i,j : T_i > T_{i^*},\ k_{j^*-1}^{i} < k_j^i < k_{j^*}^{i}\},$ then $\forall i,j \in \mathcal{I},\ \mathrm{CB}_{j^*-1,j^*-2,j}^{i^*,i^*,i} \geq 0;$ $\forall i^* \in [1,m],$ define $\mathcal{I} := \{i,j : T_i > T_{i^*},\ k_j^i > k_{n_{i^*}}^{i^*}\},$ then $\forall i,j \in \mathcal{I},\ \mathrm{CB}_{n_{i^*},n_{i^*}-1,j}^{i^*,i^*,i} \geq 0$	$\mathcal{O}(m^2N)$
C6.2 Calendar butterfly II (Relative location convexity)	$\forall i^* \in [1,m],\ j^* \in [1,n_{i^*}-1],$ define $\mathcal{I}_1 := \{i,j : T_i > T_{i^*},\ k_{j^*-1}^{i} < k_j^i < k_{j^*}^{i}\},$ $\mathcal{I}_2 := \{i,j : T_i > T_{i^*},\ k_{j^*}^{i} < k_j^i < k_{j^*+1}^{i}\},$ $\forall i_1,j_1 \in \mathcal{I},\ \forall i_2,j_2 \in \mathcal{I}_2,\ \mathrm{CB}_{j^*,j_1,j_2}^{i^*,i_1,i_2} \geq 0;$ $\forall i^* \in [1,m],$ define $\mathcal{I}_1 := \{i,j : T_i > T_{i^*},\ k_{n_{i^*}-1}^{i} < k_j^i < k_{n_{i^*}}^{i}\},$ $\mathcal{I}_2 := \{i,j : T_i > T_{i^*},\ k_j^i > k_{n_{i^*}}^{i}\},$ $\forall i_1,j_1 \in \mathcal{I},\ \forall i_2,j_2 \in \mathcal{I}_2,\ \mathrm{CB}_{n_{i^*},j_1,j_2}^{i^*,i_1,i_2} \geq 0$	$\mathcal{O}(m^2N)$

Table 1: The reduced set of static arbitrage constraints

$\mathbf{c} = [c_1^1\ \cdots\ c_{n_1}^1\ \cdots\ c_{n_m}^m]^\top \in \mathbb{R}^N$, and $A = (a_{ij}) \in \mathbb{R}^{R \times N}$ and $\mathbf{b} = (b_j) \in \mathbb{R}^R$ are a constant matrix and a vector corresponding to coefficients and bounds of the inequalities, respectively, that are completely determined by the expiries and strikes of observed options. Here, R is the number of no-arbitrage constraints, where $R \sim \mathcal{O}(m^2N)$. These constraints are *feasible* by construction, i.e. $\{\mathbf{x} \in \mathbb{R}^N : A\mathbf{x} \geq \mathbf{b}\} \neq \emptyset$, because $\mathcal{S}(\mathcal{T}^e \times [0, \max_{i,j} k_j^i]) \neq \emptyset$, (for example, the prices under a Black–Scholes model satisfy the requirements).

When some row of the system of inequalities $A\mathbf{c} \geq \mathbf{b}$ is not satisfied, there is arbitrage. We define $\boldsymbol{\varepsilon}$ to be the vector of perturbations added to the vector of call prices $\mathbf{c}$ such that the perturbed prices are arbitrage-free, i.e. $A(\mathbf{c}+\boldsymbol{\varepsilon}) \geq \mathbf{b}$. Hence, to remove arbitrage from the call price data, we seek the "minimal" repair subject to no-arbitrage constraints:

$$\min_{\boldsymbol{\varepsilon} \in \mathbb{R}^N} f(\boldsymbol{\varepsilon}), \quad \text{subject to } A\boldsymbol{\varepsilon} \geq \mathbf{b} - A\mathbf{c}, \tag{6}$$

where the objective $f : \mathbb{R}^N \to \mathbb{R}$ measures how much the perturbation deviates from zero. The formulation (6) is *feasible* because its constraints are feasible.

3.1 Design of the objective without liquidity consideration

We start from the simple case where there is no liquidity difference among options. It seems natural to use the ℓ^2-norm for measuring the size of perturbations due to its convexity and computational efficiency when optimising by gradient-based methods. The ℓ^2-norm has been widely used in data smoothing algorithms, such as [1], [16], [17] and [18].

However, the ℓ^2-norm usually leads to small perturbations for *all* prices, while in our application *sparse* perturbation is desirable. An alternative is the ℓ^0-norm[4], which is a natural way of comparing difference, and produces sparse solutions. Nevertheless, the ℓ^0-norm is nonconvex and in general leads to an NP-hard [32] optimisation problem. Hence, it is natural to consider the ℓ^1-norm, which is well known as a convex relaxation of the ℓ^0-norm. In fact, optimal solutions of the ℓ^0 and ℓ^1 norms objectives are equivalent under certain conditions, see [5], [21] and [14].

Choosing the ℓ^1-norm has other benefits. When minimising a convex continuous objective function like the ℓ^1-norm, every local minimum is a global minimum, see Chapter 4 of [3]. In addition, our repair problem is a Linear Programming (LP) problem with the ℓ^1-norm objective, which can be solved fairly quickly even for large-scale problems. Finally, compared with the ℓ^2-norm, the ℓ^1-norm is more robust to outliers because the ℓ^2-norm squares values, which increases the cost of outliers quadratically, see Huber [24].

Consequently, the ℓ^1-norm is a natural candidate for the objective function. Blacque-Florentin and Missaoui [2] also choose the ℓ^1-norm as objective when fitting tensor polynomials to sparse data, as inspired by the compressed sensing framework. The differences between our work and theirs are that they are concerned with smoothing data rather than repairing data, and assume a rectangular grid of strikes and expiries. The ℓ^1-norm optimisation with linear constraints can be expressed as an LP problem. We write the objective function as $f(\boldsymbol{\varepsilon}) := ||\boldsymbol{\varepsilon}||_{\ell^1} = \sum_{i=j}^{N} |\varepsilon_j| = \sum_{i=j}^{N} \left(\varepsilon_j^+ + \varepsilon_j^- \right)$, where $\varepsilon_j^+ = \max(\varepsilon_j, 0), \quad \varepsilon_j^- = -\min(\varepsilon_j, 0)$ for each j. We denote $\boldsymbol{\varepsilon}^+ = [\varepsilon_1^+ \cdots \varepsilon_N^+]$ and $\boldsymbol{\varepsilon}^- = [\varepsilon_1^- \cdots \varepsilon_N^-]$ so that $\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^+ - \boldsymbol{\varepsilon}^-$. We define $B = [-A\ A]$ and $\boldsymbol{\theta} = [\boldsymbol{\varepsilon}^+ \boldsymbol{\varepsilon}^-]^\top$. Hence, the repair problem with the ℓ^1-norm minimisation is equivalent to the following LP in canonical form:

$$\min_{\boldsymbol{\theta}} \mathbf{1}^\top \boldsymbol{\theta}, \quad \text{subject to } B\boldsymbol{\theta} \leq A\mathbf{c} - \mathbf{b}, \ \boldsymbol{\theta} \geq \mathbf{0}. \tag{7}$$

---

[4]Note that the ℓ^0-norm is not actually a "norm" as it violates the homogeneity and triangle inequality properties that a vector norm must satisfy.

After solving for an optimal $\hat{\boldsymbol{\theta}} = [\hat{\boldsymbol{\varepsilon}}^+ \ \hat{\boldsymbol{\varepsilon}}^-]^\top$, the optimal perturbation vector is recovered by $\hat{\boldsymbol{\varepsilon}} = (\hat{\boldsymbol{\varepsilon}}^+ - \hat{\boldsymbol{\varepsilon}}^-)^\top$.

3.2 Inclusion of bid and ask prices

The reference prices will typically lie within their corresponding bid-ask price bounds. In the presence of arbitrage, we not only want minimal repair, but also wish to have as many perturbed prices falling within the bid-ask price bounds as possible. Specifically, a reference price with wider bid-ask spread shall be given more freedom to be perturbed. The sparsity of the solution of the ℓ^1-norm optimisation is less desirable if perturbing a larger number of prices can keep more perturbed prices within the bid-ask price bounds.

Design of the objective with bid and ask prices

We consider using the best bid/ask prices for data repair. To incorporate bid-ask price constraints into the repair problem, we revise the objective function f rather than adding extra constraints. In other words, we treat bid-ask price bounds as *soft constraints* rather than *hard constraints* like the arbitrage constraints. There may not be arbitrage-free prices within the bid-ask price bounds, and adding bid-ask price bounds as hard constraints may cause the repair problem to be infeasible.

We choose an objective function of the form $f(\boldsymbol{\varepsilon}) = \sum_{j=1}^{N} f_j(\varepsilon_j)$ with $f_j(x) \geq 0$ for $x \in \mathbb{R}$. Then $f_j(x)$ can be naturally interpreted as the cost of perturbing the j-th option price, and $\mathrm{d}f_j(x)/\mathrm{d}|x| > 0$ (if defined) gives the marginal cost. The ℓ^1-norm objective sets $f_j(x) = |x| = \max(-x, x)$ and any perturbation x, where $|x| > 0$, has marginal cost 1 for all j. Let $\delta_j^a, \delta_j^b > 0$ be ask-reference spread and bid-reference spread for the j-th price, respectively. To incorporate these spreads into the objective, we require that $f_j(x)$ should have the following properties, for all $j \in [1, N]$:

(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) $f_j(-\delta_j^b) = f_j(\delta_j^a) = \delta_0$, 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) $\mathrm{d}f_j(x)/\mathrm{d}|x| = 1$ for $x \in (-\infty, -\delta_j^b) \cup (\delta_j^a, +\infty)$. The marginal cost of perturbing a price out of the bid-ask price bounds is the same for all options.

We therefore propose the following objective that meets all the properties, and, with particular merit, retains the ability to be expressed as an LP:

$$f_j(x) = \max\left(-x - \delta_j^b + \delta_0, \ -\frac{\delta_0}{\delta_j^b}x, \ \frac{\delta_0}{\delta_j^a}x, \ x - \delta_j^a + \delta_0\right),$$

with $\delta_0 \leq \min(\delta_j^a, \delta_j^b)$ for all $j \in [1, N]$, as such the marginal cost of perturbing a price within the bid-ask price band is not greater than the marginal cost of perturbing mid prices outside the bid-ask price bounds.

(a) $f_j^{\ell^1}(x)$ (b) $f_j(x)$
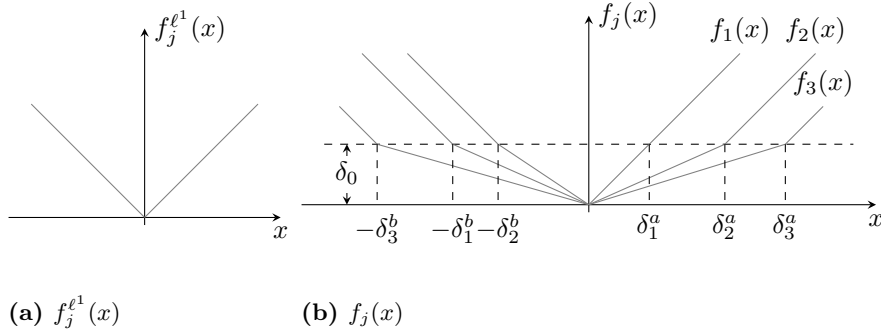
Figure 2: Plot of the objective function component $f_j^{\ell^1}(x)$ and $f_j(x)$

Denote $f_j^{\ell^1}$ as the j-th component of the ℓ^1-norm objective. We visualise the difference between $f_j^{\ell^1}$ and f_j in Figure 2. Note that $f_j^{\ell^1}$ is a special case of f_j when $\delta_j^a = \delta_j^b = \delta_0 > 0$ for all j. Choosing smaller δ_0 makes it relatively more costly to move prices outside of their bid-ask price bounds. Nevertheless, letting $\delta_0 = 0$ causes the optimisation problem to be ill-posed as it admits infinitely many solutions. For example, if $\varepsilon_j^* = 0$ is optimal, then so is $\varepsilon_j^* = \omega \min(\delta_j^a, \delta_j^b)$ for all $\omega \in [0, 1]$. In practice, we will choose

$$\delta_0 = \frac{1}{N} \wedge \min_{j=1,\ldots,N} \left(\delta_j^a \wedge \delta_j^b \right). \tag{8}$$

This means we prefer to move all options (by $\boldsymbol{\varepsilon}$) within the bid-ask, rather than moving one option outside its bid-ask bounds.

Hence, the objective function taking into account bid-ask spread is

$$f(\boldsymbol{\varepsilon}) = \sum_{j=1}^{N} \max \left(-\mathbf{e}_j^\top \boldsymbol{\varepsilon} - \delta_j^b + \delta_0, \ -\frac{\delta_0}{\delta_j^b} \mathbf{e}_j^\top \boldsymbol{\varepsilon}, \ \frac{\delta_0}{\delta_j^a} \mathbf{e}_j^\top \boldsymbol{\varepsilon}, \ \mathbf{e}_j^\top \boldsymbol{\varepsilon} - \delta_j^a + \delta_0 \right), \tag{9}$$

where $\mathbf{e}_j$ is the standard basis vector for $\mathbb{R}^N$ with its j-th element being 1 and others being 0. With objective (9), we can rewrite the repair problem (6) as the following LP by introducing auxiliary variables $\mathbf{t} = [t_1 \ \cdots \ t_N]^\top$:

$$
\begin{aligned}
\underset{\boldsymbol{\varepsilon}, \mathbf{t}}{\text{minimise}} \quad & \sum_{j=1}^{N} t_j \\
\text{subject to} \quad & -\varepsilon_j - \delta_j^b + \delta_0 \leq t_j, \ \varepsilon_j - \delta_j^a + \delta_0 \leq t_j, \quad \forall j \in [1, N], \\
& -\frac{\delta_0}{\delta_j^b} \varepsilon_j \leq t_j, \ \frac{\delta_0}{\delta_j^a} \varepsilon_j \leq t_j, \quad \forall j \in [1, N], \\
& -A\boldsymbol{\varepsilon} \leq -\mathbf{b} + A\mathbf{c}.
\end{aligned}
\tag{10}
$$

After solving for the optimal perturbation vector $\hat{\boldsymbol{\varepsilon}}$, we get the arbitrage-free normalised call price $\hat{\mathbf{c}} = \mathbf{c} + \hat{\boldsymbol{\varepsilon}}$. For each i, j, the arbitrage-free call price is $\widehat{C}_j^i = \hat{c}_j^i D_i F_i$.

Executable arbitrage opportunities

We refer to the objective function taking into account bid-ask spread with δ_0 as in (8) as the ℓ^1-BA objective. We define the *effectively perturbed prices* as those that are perturbed outside of the bid-ask price bounds. We denote the number of perturbed (resp. effectively perturbed) prices by N^ε (resp. $N^{\varepsilon,\delta}$), thus

$$N^\varepsilon = \sum_{j=1}^N \mathbb{1}_{\{|\varepsilon_j|>0\}}, \quad N^{\varepsilon,\delta} = \sum_{j=1}^N \mathbb{1}_{\{\varepsilon_j > \delta_j^a\} \cup \{\varepsilon_j < -\delta_j^b\}}. \tag{11}$$

We say an arbitrage is *executable* if we can realise it by buying and selling its components at their ask and bid quotes, respectively. The arbitrage detected in options' reference prices is not necessarily executable. However, if the ℓ^1-BA repair results in effective perturbations, i.e. $N^{\varepsilon,\delta} > 0$, then there must exist executable arbitrages. To see this, let $E_j = [c_j - \delta_j^b, c_j + \delta_j^a]$, and we can characterise $N^{\varepsilon,\delta} > 0$ as

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

Equivalently, its contrapositive statement is

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

Therefore, it holds that

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

By going long on the left-hand side and going short on the right side of the inequality, we construct a portfolio that makes immediate positive profit, while the portfolio has non-negative future payoffs. The left-hand side of the inequality consists of positions in options, for which we buy at ask price $(c_j + \delta_j^a)$ and sell at bid price $(c_j - \delta_j^b)$.

4 Empirical studies

We carry out a series of empirical studies. We show that arbitrage is frequently present in historical price data, so repairing data is important. We also demonstrate empirical performances of the repair method in terms of sparsity, speed

and improvement to model calibration. Last, we use the ℓ^1-BA repair for identifying the formation and disappearance of arbitrage in the intra-day S&P 500 options market on a day when the market underwent a regime switch.

4.1 Presence of arbitrage in historical option price data

We collect daily close (bid, ask and mid) prices from 1st November, 2007 to 31st May, 2018 for OTC FX options from Bloomberg. Bloomberg provides price quoted as implied volatility given in terms of delta. We choose 13 benchmark tenors (expiries) from overnight (one-day) to two-year. For each tenor, a list of standard *instruments* are available: at-the-money (ATM), risk-reversal (RR) and butterfly (BF). We choose the liquid 10-delta, 15-delta, 25-delta and 35-delta instruments, and construct a vanilla volatility smile of 9 moneynesses for each tenor. Following the OTC FX market conventions [37], we compute strike and time-to-expiry for each IV mid quote, and generate vanilla IV spreads from the bid/ask quotes for the instruments[5]. Thereafter, we calculate mid call prices and vanilla call price spreads using the mid vanilla IVs and generated vanilla IV spreads, together with Bloomberg FX mid forward curves. There are $117 = 13$ (tenors) $\times$ 9 (deltas and ATM) data points on the call price surface for each day. In Figure 3, we show an example of OTC call option mid-prices and IV mid-quotes on one day.
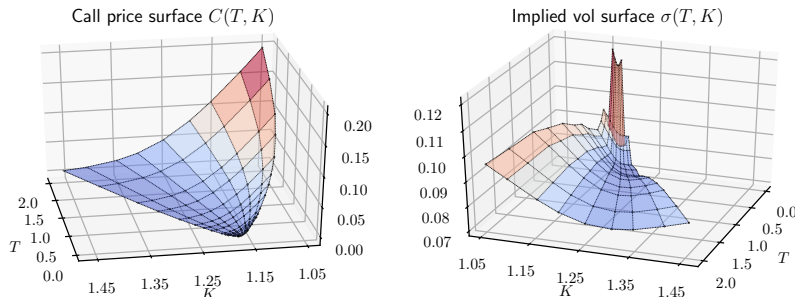
Figure 3: An example of observed OTC-traded call option prices. These are end of day prices settled by Bloomberg for EURUSD European call options as of 31st May, 2018.

We count violations of arbitrage constraints in raw daily close mid-prices over time for some major currencies and emerging market (EM) currencies. In Figure 4, we see that there are more arbitrages in the EM currency markets. We also see persistent clustering of (mild) arbitrages from early 2007 to mid 2012 in major currency markets. Further investigation suggests that these are caused

---

[5]Given the *instrument bid-ask spreads* for ATM, RR and BF, one cannot uniquely determine the corresponding *vanilla spreads* without specifying some rule. For example, in practice, trading desks may estimate vanilla spreads only using ATM spreads, which makes the spread of each option at the same expiry equal, see Section 4.2.1 of [37]. Since vanilla IVs are linear transformations of instrument IVs, we conservatively assume that vanilla spreads are weighted sums of instrument spreads. This does not take into account that delta-symmetric vanilla spreads are dependent on each other, and generates the widest possible bid-ask spreads for vanilla IVs.

by over-priced 1-day options, which result in calendar arbitrages with longer-dated options. We conjecture that the systematic appearance of the same type of arbitrage is due to Bloomberg's legacy data cleansing method.
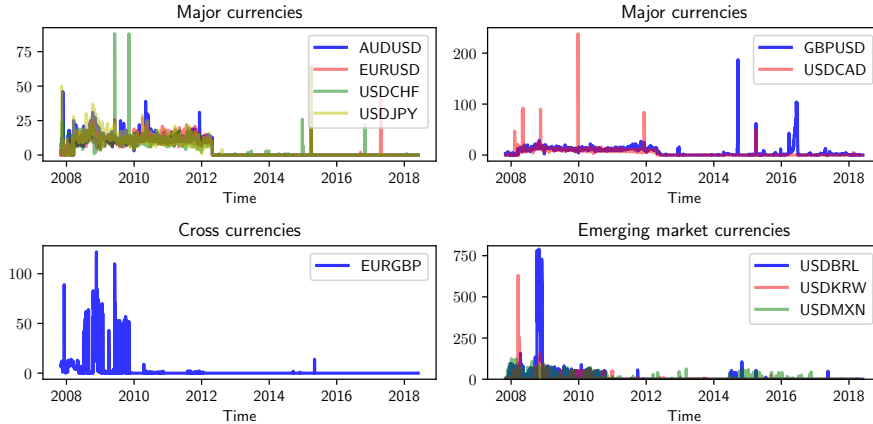
Figure 4: Time series of number of daily violated arbitrage constraints in OTC FX option market, during the period from 1st November, 2007 to 31st May, 2018.

Calendar arbitrage (especially CVS C5 and CBS C6) is more difficult and costly to exploit than non-calendar arbitrages, as it requires rebalancing the hedging portfolio over time. Most arbitrage-free smoothing algorithms in the literature only remove calendar arbitrage of C4 type, because they assume a rectangular grid of expiries and strikes. However, calendar arbitrage can be a major source of arbitrage. In Figure 5, we consider what fraction of the arbitrages are of calendar type for different currency pairs. Comparing medians (and overall distributions), as shown in the plot, the proportion of calendar arbitrages for major currencies (AUD, EUR, GBP, CAD, CHF, and JPY) is larger than that for EM currencies (BRL, KRW, and MXN), though the cross pair EURGBP is an exception. In fact, the medians are very close to 100% for almost all major currencies except sterling. In other words, nearly all arbitrages in major currencies' option markets are calendar ones.
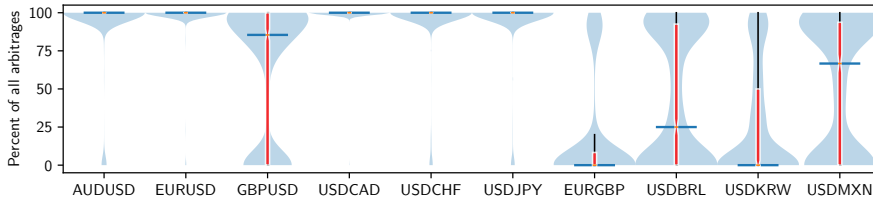
Figure 5: Fraction of calendar arbitrages on a given day, for different currency pairs during the period from 1st November, 2007 to 31st May, 2018. The light blue shadow is a violin plot which indicates the kernel density of the percentages, and the red notched box is a box plot. The horizontal short bar shows the median of each sample.

We examine the day when the EURUSD option price data have the most occurrences of calendar arbitrages over our observation period, and plot the call

price curves for the first three expiries in Figure 6. There is no non-calendar arbitrage on that day since each curve is non-increasing and convex. After the repair, the T_1-curve is pushed downwards until it does not lie beyond the other two curves, which ensures NDCO marginal risk-neutral measures.
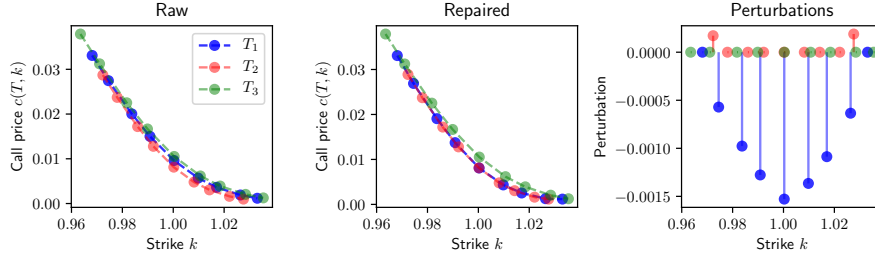
Figure 6: An example of arbitrage repair for EURUSD call options on 2nd April, 2015. *Left* – raw call price curves for the first three expiries. *Middle* – repaired arbitrage-free call price curves. *Right* – perturbations added to each data point.

However, when calendar and non-calendar arbitrages are mixed, the perturbations added to ensure no arbitrage tend to be more varied in signs. For instance, in Figure 7 we plot the call price curves for the first four expiries on the day when USDBRL options had the most occurrences of calendar arbitrage, however, there are also many non-calendar arbitrages. Unlike the above EURUSD example, the repair does not simply translate any curve. Therefore, the perturbations are not systematically negative.
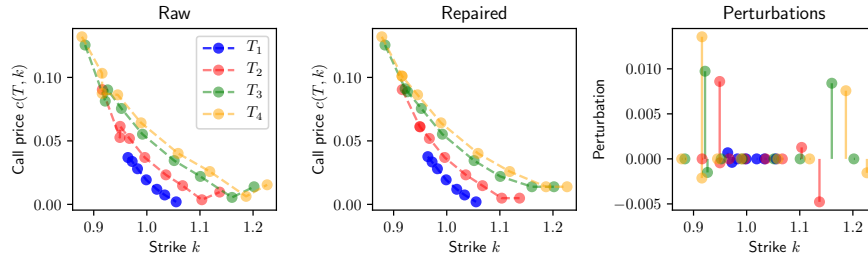
Figure 7: An example of arbitrage repair for USDBRL call options on 28th October, 2008.

4.2 Properties of the repair method

Sparse solution of the ℓ^1-norm objective

The ℓ^1-norm objective leads to sparse perturbations. We show the fraction of perturbed prices in Figure 8. The medians are very close to zero for all currency pairs, indicating that very few data points need to be perturbed on average to remove arbitrage. This is especially true for major currencies, as their distributions collapse almost entirely to zero.
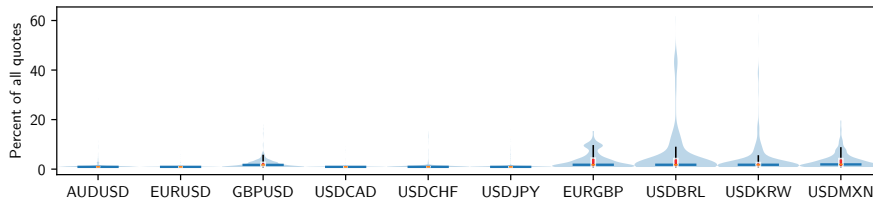
Figure 8: Number of perturbed prices as a percentage of all prices, for different currency pairs during the period from 1st November, 2007 to 31st May, 2018.

Computational time

Our data repair method is designed to be fast due to the LP formulation. In addition, the reduction of arbitrage constraints shrinks the scale of the LP and speeds up the repair. We investigate the computational time of our repair method when applied to a few practical cases. All of the following studies were carried out on a quadcore Intel Core i7-8650U CPU with 32GB RAM. All LPs are solved using the GLPK (GNU Linear Programming Kit) solver wrapped by the CVXOPT [30] Python package.

In Figure 9, we plot histograms of (1) the number of constraints $R \sim \mathcal{O}(m^2 N)$, (2) the fraction of violated constraints, and (3) the elapsed times for constructing the constraints (Table 1) and solving the LP (7). We take EURUSD and USDBRL as representatives of major currencies and EM currencies, respectively. The number of constraints rarely exceeds 4000, while it takes less than 0.4 seconds to transform them to the matrix form in most cases. Solving the LP with $N = 117$ variables and $R < 4000$ constraints takes less than 0.05 seconds for EURUSD options and less than 0.1 seconds for USDBRL options. More violated constraints lead to higher computational time.

To see how the repair method performs for larger-scale problems, we collect daily settled prices for all *traded* EURUSD call options listed by CME from 1st January, 2013 to 31st December, 2018. The number of traded options varies from one day to another, see Figure 10a. We show the distribution of traded expiry and strikes on a typical day in Figure 10b. In Figure 10c, we plot similar repair statistics to those for the OTC data. On average, there are 500 call prices per day, which result in on average 25000 arbitrage constraints to verify. Though the number of constraints is observed as high as 90000, it takes less than 1 second to construct them in the matrix form. Solving the LP now can take up to 6 seconds, but on average it only takes 1.44 seconds.

Stress testing the ℓ^1-norm objective repair

We test how our repair method works in hypothetical extreme scenarios when there is massive arbitrage. First, we collect arbitrage-free call prices for a day, denote these data by $\mathbf{c} \in \mathbb{R}^N$, and let $\mathcal{I} = \{1, \ldots, N\}$ be the set of its indices. Next, we simulate noises and add them to a portion $\lambda \in (0, 1]$ of the price data, where we denote $\mathcal{I}_\xi \subset \mathcal{I}$ as the set of indices of those polluted prices. Here, $\mathcal{I}_\xi$ is
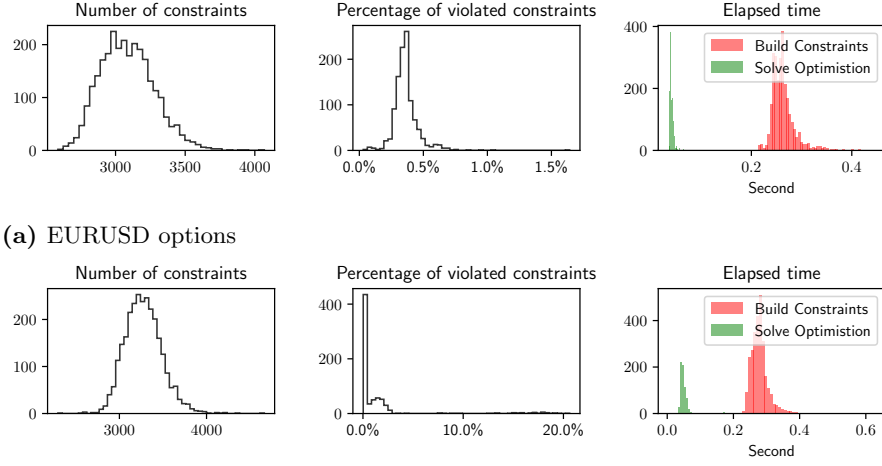
(a) EURUSD options

(b) USDBRL options

Figure 9: Histograms of various statistics for repairing data of EURUSD options and USDBRL options, during the period from 1st November, 2007 to 31st May, 2018.
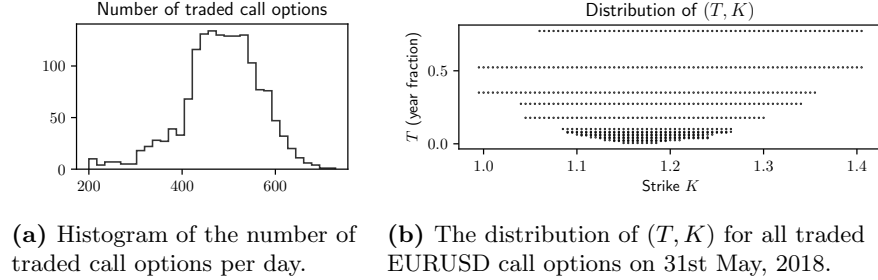
(a) Histogram of the number of traded call options per day.

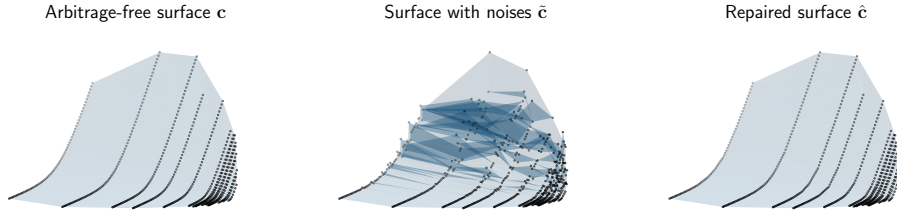**(b)** The distribution of (T, K) for all traded EURUSD call options on 31st May, 2018.

(c) Histograms of various statistics.

Figure 10: Statistics for repairing data of CME-listed EURUSD options during the period from 1st January, 2013 to 31st December, 2018.

randomly sampled without replacement such that $|\mathcal{I}_\xi| = \lceil \lambda |\mathcal{I}| \rceil$. Constructing the noise $\boldsymbol{\xi} = (\xi_j) \in \mathbb{R}^N$ by taking $\xi_j = \zeta_j \mathbb{1}_{\{j \in \mathcal{I}_\xi\}}$ where $\boldsymbol{\zeta} = (\zeta_j)$ are i.i.d., we then define the noisy price $\tilde{\mathbf{c}} \in \mathbb{R}^N$ by

$$\tilde{c}_j = c_j e^{\xi_j}, \ \forall 1 \le j \le N.$$

The noisy price vector contains arbitrage in general, which we repair by seeking an optimal perturbation $\varepsilon \in \mathbb{R}^N$. The perturbed arbitrage-free price vector is $\hat{\mathbf{c}} = \tilde{\mathbf{c}} + \varepsilon$. An example of $\mathbf{c}$, $\tilde{\mathbf{c}}$, and $\hat{\mathbf{c}}$ is given in Figure 11.

Arbitrage-free surface c   Surface with noises c̃   Repaired surface ĉ

Figure 11: An example used for stress testing the repair method. There are $N = 591$ prices in total. The data used are CME-traded EURUSD options' prices on 31st May, 2018.

We assess how well $\mathbf{c}$ is recovered by examining the quantities $\ln(\hat{c}_j/c_j)$ and $\hat{\lambda} := \frac{1}{N} \sum_{j=1}^{N} \mathbb{1}_{\{\hat{c}_j \neq c_j\}}$. Note that $\ln(\hat{c}_j/c_j) \approx (\hat{c}_j - c_j)/c_j$ (when $\hat{c}_j \approx c_j$). In addition, $\hat{\lambda}$ counts the portion of different prices per call surface. It is unrealistic to expect any repair method to fully recover a price as it is unlikely to know the exact marginal that generates the price. However, given the ground truth that a portion λ of the surface prices has been polluted by noise, a desirable data repair method should leave as many unpolluted prices unchanged as possible, i.e. $\hat{\lambda} - \lambda$ should be small.

Assuming Gaussian noises $\boldsymbol{\zeta} \sim \mathcal{N}(\mathbf{0}, \sigma_\xi I)$, we simulate noises M times, and compute the average value of $\hat{\lambda}$ and plot the histograms of $\ln(\hat{c}_j/c_j)$, conditional on non-zero values, as shown in Figure 12. For a fixed noise magnitude σ_ξ, the gap between $\hat{\lambda}$ and λ widens as λ increases, i.e. the repair method adjusts a larger number of prices to remove arbitrage if there are more noisy prices. The same observation holds for different values of σ_ξ, though larger noise magnitude σ_ξ results in more arbitrages. Note that taking $\sigma_\xi = 1$ and $\lambda = 25\%$ already results in, on average, $\hat{\lambda} = 30.80\%$ of the price data being perturbed, an extremely large fraction that has rarely been seen in our data, see Figure 8. Hence, in practice our repair method seems to only perturb a few additional (i.e. $\hat{\lambda} - \lambda \approx 5\%$ for $\lambda = 25\%$) prices to ensure no arbitrage.

Comparing the objectives: ℓ^1-norm and ℓ^1-BA

The ℓ^1-BA repair is designed to perturb more prices (larger N^ε) than the ℓ^1-norm repair does, but fewer of them are effective (smaller $N^{\varepsilon,\delta}$), if possible. To verify this, we apply the ℓ^1-BA repair method to the same OTC FX option price data. In Figure 13, we show the histograms of the difference in these two statistics N^ε and $N^{\varepsilon,\delta}$ that are produced by the two repair methods.

A detailed example showing how the two repair methods work in reality is given in Figure 14. From left to right, the displayed data are ordered by increasing strikes, grouped by expiry. The light blue areas are confined by bid-ask spread as a percentage of option prices (green lines). We see that ITM and
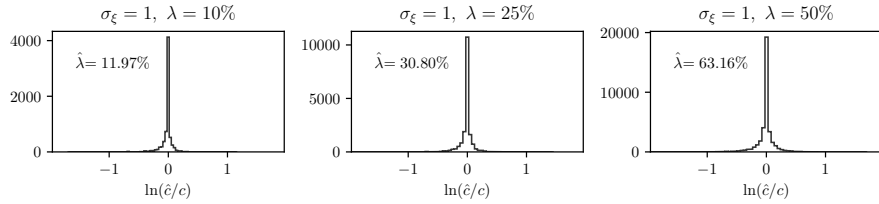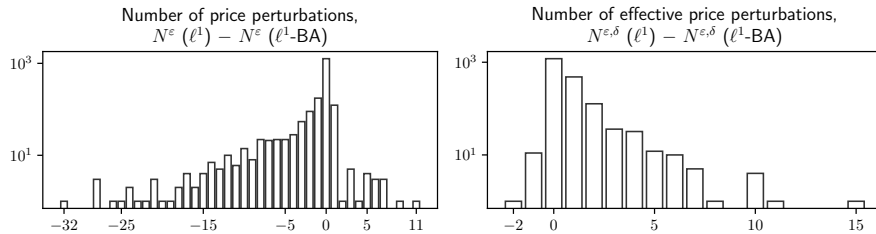
(a) $\sigma_\xi = 1$.
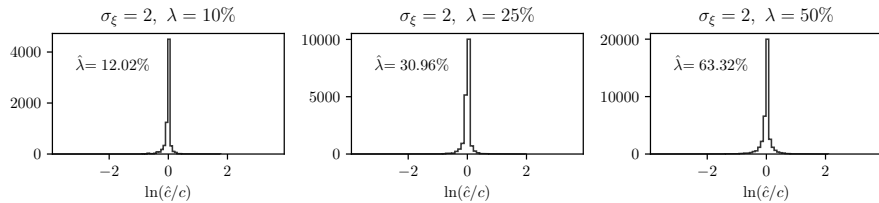
(b) $\sigma_\xi = 2$.

Figure 12: Histograms of $\ln(\hat{c}_j/c_j) \approx (\hat{c}_j - c_j)/c_j$, conditional on non-zero values, computed under differently valued noise simulation parameters (λ, σ_ξ). We simulate $M = 100$ times.

Figure 13: Histograms of the difference in N^ε and $N^{\varepsilon,\delta}$ that are produced by the ℓ^1-norm repair method and the ℓ^1-BA repair method. These two methods are separately applied to the same set of OTC FX data as in Figure 4, and the histograms are plotted by stacking data of all ten currency pairs and historical dates.

OTM options have wider bid-ask spreads than ATM options do. The ℓ^1-BA repair method results in fewer effective perturbations. First, there is one less effective perturbation of 1M option prices, at the cost of perturbing a few 2W, 3W and 1M option prices to their bid or ask prices. Second, all four effective perturbations of 4M option prices by the ℓ^1-norm repair are replaced by six ineffective perturbations of 6M option prices by the ℓ^1-BA repair.

For a given set of prices, if none of the perturbations is effective, then the bid and ask quotes given by the market admit some arbitrage-free prices that fall within the bid-ask price bounds. In contrast, effective perturbations imply the existence of *executable* arbitrages that are exploitable through matching existing bid or ask orders in the market, see Section 3.2. In Table 2, we count the number of days when there is arbitrage in mid-prices ($N^\varepsilon > 0$) and the
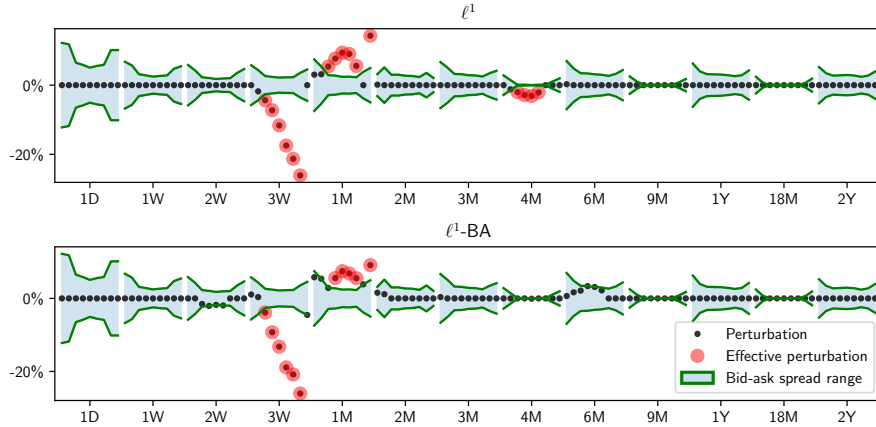
Figure 14: Perturbations (as percentages of the raw price data) resulted from the ℓ^1-norm and the ℓ^1-BA objectives. Data used are bid, ask and mid prices for OTC-traded USDBRL options on 18th September, 2008.

number of days when there is executable arbitrage ($N^{\varepsilon,\delta} > 0$) in historical data for the four currency pairs that have been seen to have the most occurrences of arbitrages.

Currency pair	EURGBP	USDBRL	USDKRW	USDMXN
#days when $N^{\varepsilon} > 0$	470	708	623	577
#days when $N^{\varepsilon,\delta} > 0$	285	89	163	144

Table 2: Number of days when there is arbitrage in mid-prices ($N^{\varepsilon} > 0$) and when there is executable arbitrage ($N^{\varepsilon,\delta} > 0$).

4.3   Improvement to model calibration

We verify that our repair method improves model calibration with more robust parameter estimates and smaller calibration error.

Test framework

Let Θ be model parameters. We specify $\Theta = \overline{\Theta}$ and generate model prices $\mathbf{c}$ for call options on a set of expiries and strikes. Then we carry out the following steps M times. For the m-th time:

(1) Simulate noises to create synthetic arbitrageable price data $\tilde{\mathbf{c}}^{(m)}$, following the method in Section 4.2. Recall that $\lambda \in (0,1]$ portion of prices are polluted by Gaussian noises of variance σ_{ξ}^2.

(2) Repair arbitrage in $\tilde{\mathbf{c}}^{(m)}$ to get arbitrage-free data $\hat{\mathbf{c}}^{(m)}$.

(3) Calibrate model parameters Θ to $\tilde{\mathbf{c}}^{(m)}$ and $\hat{\mathbf{c}}^{(m)}$ separately[6], and get calibrated parameters $\widetilde{\Theta}^{(m)}$ and $\widehat{\Theta}^{(m)}$, respectively. Defining the calibration objective as $G(\Theta; \mathbf{c}) = \sum_{j=1}^{N} (c_j^{\Theta} - c_j)^2$ where c_j^{Θ} is the model price for the j-th option, we have

$$\widetilde{\Theta}^{(m)} = \arg\min_{\Theta} G(\Theta; \tilde{\mathbf{c}}^{(m)}), \quad \widehat{\Theta}^{(m)} = \arg\min_{\Theta} G(\Theta; \hat{\mathbf{c}}^{(m)}).$$

We measure model calibration performance by two metrics, which are (a) the *robustness* defined by variations in the parameter estimates, and (b) the calibration error defined as the square root of the minimal objective value. Since we have parameter estimates $\{\widetilde{\Theta}^{(m)}\}_{1 \leq m \leq M}$ and $\{\widehat{\Theta}^{(m)}\}_{1 \leq m \leq M}$, we can compare the variations in them for assessing robustness. For each m, we define the (relative) reduction of calibration error as

$$\Delta G^{(m)} = 1 - \sqrt{\frac{G(\widehat{\Theta}^{(m)}; \hat{\mathbf{c}}^{(m)})}{G(\widetilde{\Theta}^{(m)}; \tilde{\mathbf{c}}^{(m)})}}.$$

Heston model calibration

We carry out a test on calibration of the Heston model [23]. Recall that the Heston model is described by the SDEs with model parameters $\Theta = (\nu_0, \theta, k, \sigma, \rho)$:

$$\mathrm{d}S_t = r_t S_t \, \mathrm{d}t + \sqrt{\nu_t} S_t \, \mathrm{d}W_t^S,$$
$$\mathrm{d}\nu_t = k(\theta - \nu_t) \, \mathrm{d}t + \sigma \sqrt{\nu_t} \, \mathrm{d}W_t^{\nu}, \quad \mathrm{d}\langle W_t^S, W_t^{\nu} \rangle = \rho \, \mathrm{d}t,$$

where the Feller condition $2k\theta > \sigma^2$ is sufficient to ensure strict positivity of the instantaneous variance process ν_t.

We specify a typical set of expiries and strikes that is observed on a day in the OTC market, such as the one shown in Figure 3. Other simulation parameters and ground truth model parameters[7] are listed in Table 3.

|	Heston model					Simulation			
Parameter	ν_0	θ	k	σ	ρ	N	M	λ	σ_ξ
Value	0.003	0.008	2.32	0.38	0.36	117	500	0.25	0.1 or 1

Table 3: Parameter values

Next, we follow the test framework and evaluate $\widetilde{\Theta}^{(m)}$, $\widehat{\Theta}^{(m)}$, $\Delta G^{(m)}$ and $\Delta t^{(m)}$ for $m = 1, \ldots, M$. In Figure 15, we plot and compare the normed histograms of calibrated Heston parameters $\widehat{\Theta}$ (using noisy data) and $\widetilde{\Theta}$ (using

---

[6]Note that we must apply exactly the same numerical procedure for these two separate calibrations, i.e. the same optimisation algorithm, terminal criteria, lower and upper bounds, and initial values.

[7]Heston model parameters are chosen as those that reproduce a typical call price surface for USDBRL options. Noise simulation parameters λ and σ_ξ are chosen to mimic severe but not extreme arbitrage scenarios (measured by the fraction of perturbed prices by the repair method) observed in real world data.
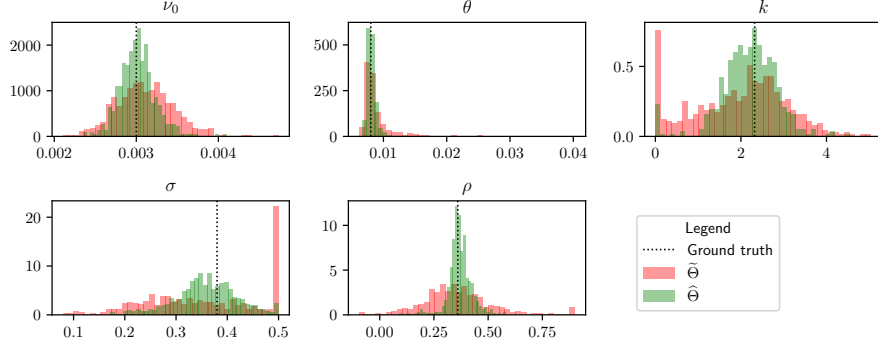
(a) $\sigma_\xi = 0.1$.
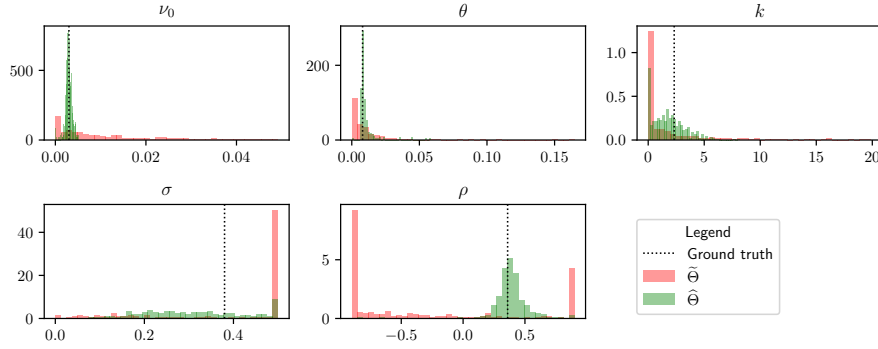

(b) $\sigma_\xi = 1$.

Figure 15: Sample (normed) histograms of $\widetilde{\Theta}$ and $\widehat{\Theta}$, where $\Theta = (\nu_0, \theta, k, \sigma, \rho)$.

repaired data) given different choices of σ_ξ. The ground truth parameter values are also indicated by vertical dotted lines. Repairing data does make the model calibration more robust, as supported by two types of evidence. First, there are apparently more variations in $\widetilde{\Theta}$ than in $\widehat{\Theta}$. Second, $\widetilde{\Theta}$ tends to hit the bounds set in the numerical optimisation procedure (e.g. 0 for k, 0.5 for σ, 1 for ρ) much more often than $\widehat{\Theta}$ does. Moreover, when the price data are more noisy (larger σ_ξ) so that more prices with arbitrage are present, the robustness improvement of model calibration by the repair method becomes more significant.

In Figure 16, we plot the histograms of ΔG and indicate their means by vertical dotted lines. Repairing arbitrage in data reduces the calibration errors in all M simulations with no exception. Moreover, the more noisy the raw data are, the arbitrage repair method reduces relatively more calibration errors. On average, repairing data can reduce the calibration error by more than 70% for $\sigma_\xi = 0.1$, and more than 95% for $\sigma_\xi = 1$.

Hence, for model calibration task, there is more benefit of repairing data by removing arbitrage when the data contain larger noise.
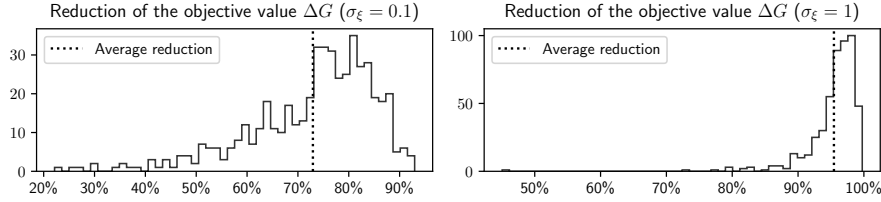
Figure 16: Sample histograms of (relative) reductions in calibration error.

4.4 Identifying intra-day executable arbitrage

We can use the ℓ^1-BA repair method on order book data for identifying executable arbitrage. An example is given in Figure 17. We collect the order book data for all E-mini S&P 500 monthly European call options from 12:00 ET to 16:10 ET on 12th June, 2020.
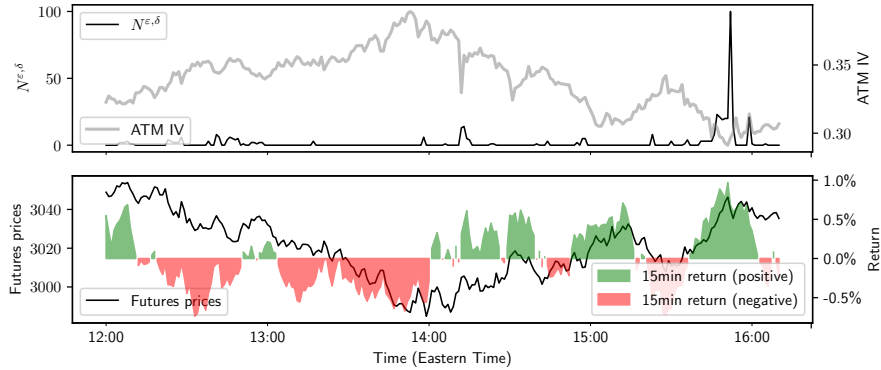
Figure 17: *Top* – The formation and disappearance of intra-day executable arbitrage opportunities in the E-mini S&P 500 monthly European call option market on 12th June, 2020. *Bottom* – front-month futures' prices and 15-minute return.

We extract the active best ask and best bid prices for all quoted call options from the order book at the end of every minute. Then we compute mid prices, apply the ℓ^1-BA repair method to the mid prices, and count the number of effective perturbations $N^{\varepsilon,\delta}$. Recall from Section 3.2 that, given δ_0 small, there exists executable arbitrage if the ℓ^1-BA repair method results in effective perturbations. In the top plot of Figure 17, the black line gives $N^{\varepsilon,\delta}$ over time, while we also indicate the ATM implied volatility of the front-month option (which would expire on 19th June) by the grey line. The bottom plot gives the prices and the 15-minute returns of the front-month futures contract. The downward trend of the futures market was inverted around 14:00 ET, after when the implied volatility also falls gradually from its peak.

There is a large spike of $N^{\varepsilon,\delta}$ at around 15:52 ET, a few minutes before the close of the S&P 500 index market at 16:00 ET. This spike coincided with rallies in the futures market, while the IV maintained its relatively low level. There are some clusters of smaller spikes of $N^{\varepsilon,\delta}$ outside of the US trading

hours. Apart from this arbitrage outbreak preceding the close of the underlying market, which lasted for around 15 minutes, there seems to be trivial executable arbitrage during the rest of the afternoon trading hours, even when the market underwent regime switch (from downward trend to upward trend) at around 14:00 ET.

A   Localisation of static arbitrage constraints

To localise calendar butterfly constraints, we use a sequential build-up of local constraints from the shortest expiry to the longest expiry. Define $\mathcal{D}_i := \{(k_j^i, c_j^i) : 1 \leq j \leq n_i\}$ as price data for options of expiry $T_i \in \mathcal{T}^e$. Given arbitrage-free $\mathcal{D}_{i^*}$, we construct constraints such that adding price data of any longer-expiry option should not introduce arbitrage. This is done locally in two steps, where we scan a neighbourhood of each $k_j^{i^*}$. The first step, we call "absolute location convexity" C6.1, finds constraints ensuring that adding *any single* data point (k_j^i, c_j^i) where $i > i^*$ will not introduce arbitrage. In Figure 18 we indicate the regions where adding a single data point will not introduce arbitrage for four types of strike neighbourhood. In the second step "relative location convexity" C6.2, we find constraints making sure that adding *all* data points (k_j^i, c_j^i) where $i > i^*$ will not introduce arbitrage for two types of strike neighbourhood. As shown in Figure 19, if we draw line segments by linking each added point and the reference point $o = (k_{j^*}^{i^*}, c_{j^*}^{i^*})$, we require the slope of any line on the left $\{l_i\}$ to be not greater than the slope of any line on the right $\{r_j\}$.

(a) $j^* = 1$ **(b)** $j^* \in [2, n_{i^*} - 1]$ **(c)** $j^* = n_{i^*}$

(d) $j^* = n_{i^*}$, $k > k_{j^*}^{i^*}$
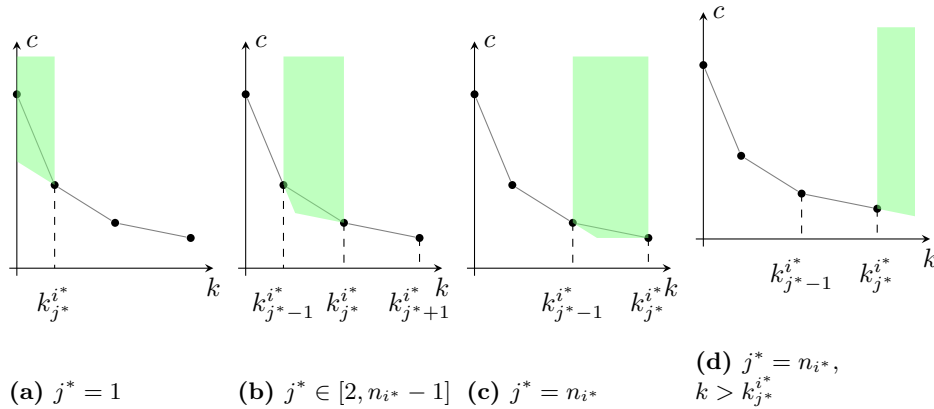
Figure 18: Absolute location convexity constraint, discussed in four cases. Points falling in the green region satisfy the absolute location convexity constraint.
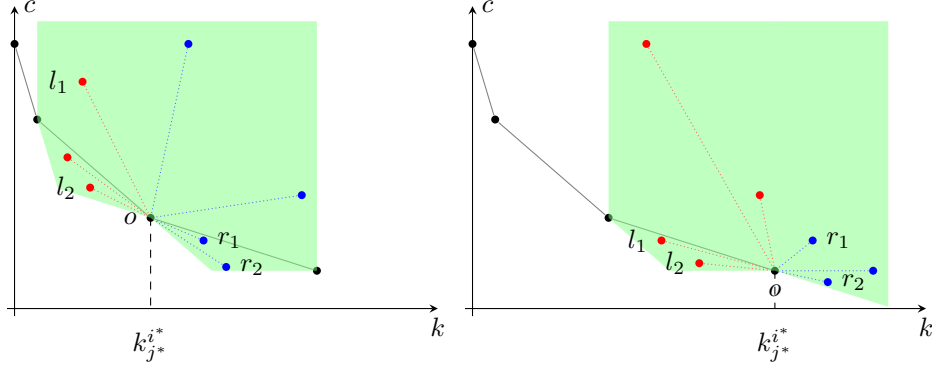
(a) $j^* \in [1, n_{i^*} - 1]$ **(b)** $j^* = n_{i^*}$

Figure 19: Relative location convexity constraint, discussed in two cases. Points within the green region satisfy the absolute location convexity constraint.

B   Proof of Proposition 2

We prove Proposition 2 by establishing Lemma 1, 2 and 3.

Lemma 1. *If C1, C2 and C3 are satisfied, then all outrights, vertical spreads and vertical butterflies are non-negative. In addition, all test vertical spreads are not greater than 1.*

Proof. We consider the prices of call options with the same expiry T_i where $i \in [1, m]$.

First, we prove that any vertical spread is non-negative, i.e. $\forall 0 \leq j_1 < j_2 \leq n_i$, $c_{j_1}^i \geq c_{j_2}^i$. This is true by the vertical spread constraint C2, as $c_{j_1}^i \geq c_{j_1+1}^i \geq \cdots \geq c_{j_2}^i$.

Second, we show that all outrights are non-negative, i.e. $\forall j \in [0, n_i]$, $c_j^i \geq 0$. Given the outright constraint C1 and that any vertical spread is non-negative, we have $c_j^i \geq c_{n_i}^i \geq 0$.

Next, we show that any vertical butterfly is non-negative, i.e. $\forall 0 \leq j_1 < j < j_2 \leq n_i$, $\beta(i, j_2; i, j) - \beta(i, j; i, j_1) \geq 0$. To do that, we claim

$$\beta(i, j_1 + 1; i, j_1) \leq \beta(i, j; i, j_1 + 1), \quad \text{if } j_1 < j - 1, \tag{12a}$$

$$\beta(i, j_2; i, j_2 - 1) \geq \beta(i, j_2 - 1; i, j), \quad \text{if } j_2 > j + 1. \tag{12b}$$

These two claims can be proved by induction. Here we only show the proof for (12a). It is true that $\beta(i, j - 1; i, j - 2) \leq \beta(i, j; i, j - 1)$ (the $j_1 = j - 2$ case for (12a)) by the vertical butterfly constraint C3. Assume that (12a) holds for

26

$j_1 = l < j - 2$, i.e.

$$c_j^i \geq c_{l+1}^i + (k_j^i - k_{l+1}^i)\frac{c_{l+1}^i - c_l^i}{k_{l+1}^i - k_l^i} = c_{l+1}^i + \left[(k_j^i - k_l^i) - (k_{l+1}^i - k_l^i)\right]\frac{c_{l+1}^i - c_l^i}{k_{l+1}^i - k_l^i}$$
$$= c_l^i + (k_j^i - k_l^i)\beta(i, l+1; i, l).$$

This leads to $\beta(i, j; i, l) \geq \beta(i, l+1; i, l)$. Again by C3 we have $\beta(i, l+1; i, l) \geq \beta(i, l; i, l-1)$. Hence, $\beta(i, j; i, l) \geq \beta(i, l; i, l-1)$, which is the stated inequality by setting $j_1 = l - 1$. Therefore, (12a) holds by induction in reverse order from $j_1 = j - 2$ to $j_1 = 0$. Thereafter, (12a) implies

$$-c_{j_1}^i \leq -c_{j_1+1}^i + (k_{j_1+1}^i - k_{j_1}^i)\frac{c_j^i - c_{j_1+1}^i}{k_j^i - k_{j_1+1}^i}$$
$$= -c_{j_1+1}^i + \left[(k_{j_1+1}^i - k_j^i) - (k_{j_1}^i - k_j^i)\right]\frac{c_j^i - c_{j_1+1}^i}{k_j^i - k_{j_1+1}^i}$$
$$= -c_j^i + (k_{j_1+1}^i - k_j^i)\beta(i, j; i, j_1+1),$$

which leads to $\beta(i, j; i, j_1) \leq \beta(i, j; i, j_1 + 1)$. Similarly we have $\beta(i, j; i, j_1) \leq \beta(i, j; i, j_1 + 1) \leq \cdots \leq \beta(i, j; i, j - 1)$. In similar fashion we can prove (12b) by induction from $j_1 = j + 2$ to $j_1 = n_i$, and deduce $\beta(i, j_2; i, j) \geq \beta(i, j_2 - 1; i, j) \geq \cdots \geq \beta(i, j+1; i, j)$. Therefore, with C3, we can conclude $\beta(i, j_2; i, j) \geq \beta(i, j; i, j_1)$.

Finally, we show that any vertical spread is bounded by 1, i.e. $0 \leq j_1 < j_2 \leq n_i$, $-\beta(i, j_2; i, j_1) \leq 1$. For the case $j_1 > 1$, given that any butterfly spread is non-negative, we have $-\beta(i, j_2; i, j_1) \leq -\beta(i, j_1; i, 1) \leq -\beta(i, 1; i, 0) \leq 1$, where the last inequality holds due to the vertical spread constraint C2. If $j_1 = 1$, then $-\beta(i, j_2; i, j_1 = 1) \leq -\beta(i, 1; i, 0) \leq 1$. Otherwise $j_1 = 0$, applying (12a) by assigning $j_1 = 0$, $j = j_2$ yields $-\beta(i, j_2; i, j_1 = 0) \leq -\beta(i, 1; i, 0) \leq 1$. □

Lemma 2. *If C2, C4 and C5 are satisfied, then any calendar spread or calendar vertical spread is non-negative.*

Proof. We would like to prove that $\forall 0 \leq i_1 < i_2 \leq m$ and $\forall j_1 \in [0, n_{i_1}]$, $j_2 \in [0, n_{i_2}]$ where $k_{j_1}^{i_1} \geq k_{j_2}^{i_2}$, we have $c_{j_1}^{i_1} \leq c_{j_2}^{i_2}$.
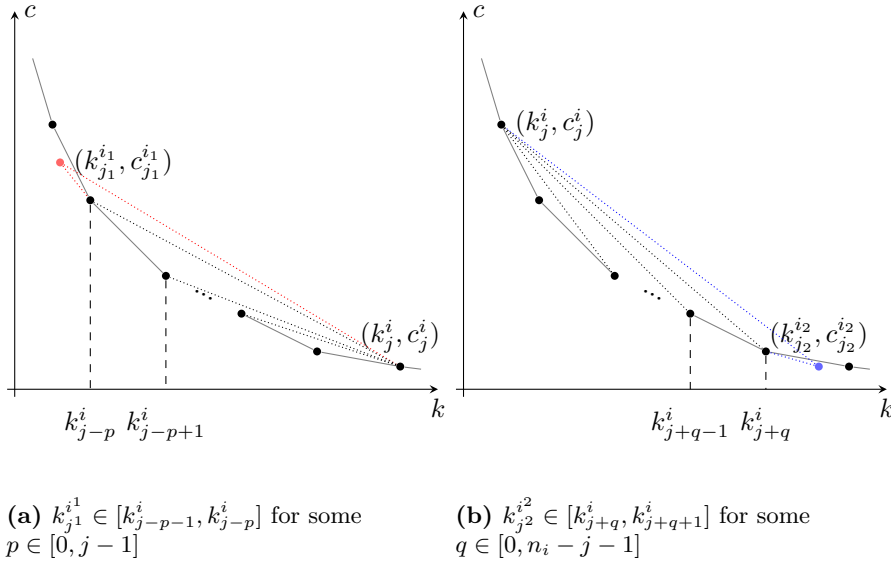
First consider the calendar spread case when $k_{j_1}^{i_1} = k_{j_2}^{i_2}$. The calendar spread constraint C4 immediately leads to $c_{j_1}^{i_1} \leq c_{j_2}^{i_2}$.

Otherwise $k_{j_1}^{i_1} > k_{j_2}^{i_2}$, which implies that j_1 must be greater than 0. Given $i_1 \in [1, m]$, $j_1 \in [1, n_{i_1}]$, there must be $k_{j_2}^{i_2} \in [k_{j_1-p-1}^{i_1}, k_{j_1-p}^{i_1})$ for some $p \in [0, j_1 - 1]$. By the calendar vertical spread constraint C5, we have $c_{j_2}^{i_2} \geq c_{j_1-p}^{i_1}$. In addition, $c_{j_1-p}^{i_1} \geq c_{j_1}^{i_1}$ due to the vertical spread constraint C2. Hence, $c_{j_2}^{i_2} \geq c_{j_1}^{i_1}$. □

Lemma 3. *If C3 and C6 are satisfied, then any calendar butterfly is non-negative.*

27

Proof. We would like to prove that $\forall i, i_1, i_2 \in [1, m]$ where $i \leq i_1, i \leq i_2$ and $\forall j \in [1, n_i]$, $j_1 \in [0, n_{i_1}]$, $j_2 \in [0, n_{i_2}]$ where $k_{j1}^{i^1} < k_j^i < k_{j2}^{i^2}$, we have $\beta(i, j; i_1, j_1) \leq \beta(i_2, j_2; i, j)$.

Given $i \in [1, m]$, $j \in [1, n_i]$, it must be that $k_{j1}^{i^1} \in [k_{j-p-1}^i, k_{j-p}^i]$ for some $p \in [0, j-1]$ and either $k_{j2}^{i^2} \in [k_{j+q}^i, k_{j+q+1}^i]$ for some $q \in [0, n_i - j - 1]$ or $k_{j2}^{i^2} \in (k_{n_i}^i, \infty)$. See Figure 20.

(a) $k_{j1}^{i^1} \in [k_{j-p-1}^i, k_{j-p}^i]$ for some $p \in [0, j-1]$

(b) $k_{j2}^{i^2} \in [k_{j+q}^i, k_{j+q+1}^i]$ for some $q \in [0, n_i - j - 1]$

Figure 20: Locations of $k_{j1}^{i^1}$ and $k_{j2}^{i^2}$ relative to k_j^i.

Let us consider the case when $k_{j2}^{i^2} \leq k_{n_i}^i$ (which implies that $j < n_i$). If $p = q = 0$, then by the calendar butterfly relative location constraints C6.2 we conclude $\beta(i, j; i_1, j_1) \leq \beta(i_2, j_2; i, j)$. Otherwise, we claim that if $p > 0$

$$\beta(i, j; i_1, j_1) \leq \beta(i, j; i, j - p), \tag{13a}$$

$$\beta(i, j; i, j - p) \leq \beta(i, j; i, j - p + 1) \leq \cdots \leq \beta(i, j; i, j - 1); \tag{13b}$$

and similarly if $q > 0$

$$\beta(i_2, j_2; i, j) \geq \beta(i, j + q; i, j), \tag{14a}$$

$$\beta(i, j + q; i, j) \geq \beta(i, j + q - 1; i, j) \geq \cdots \geq \beta(i, j + 1; i, j). \tag{14b}$$

We will show the proof for the four claims later. If $p > 0$ and $q > 0$, the four claims and the vertical butterfly constraint C3 lead to the stated result. If $p > 0$ but $q = 0$, then (13) and the calendar butterfly absolute location convexity constraint C6.1 lead to the stated result. If $p = 0$ but $q > 0$, then (14) and C6.1 lead to the stated result.

Next we would like to prove the claims (13) and (14). First of all, (13b) and (14b) hold because of the convexity of the set of points $\{(k_l^i, c_l^i)\}_{l \in [j-p, j+q]}$ resulted from the vertical butterfly constraint C3. The calendar butterfly absolute location convexity constraint C6.1 results in $\beta(i, j-p; i_1, j_1) \leq \beta(i, j-p+1; i, j-p)$. In addition, the vertical butterfly constraint C3 results in $\beta(i, j; i, j-p+1) \geq \beta(i, j-p+1; i, j-p)$, then

$$c_j^i \geq c_{j-p+1}^i + (k_j^i - k_{j-p+1}^i) \frac{c_{j-p+1}^i - c_{j-p}^i}{k_{j-p+1}^i - k_{j-p}^i}$$

$$= c_{j-p+1}^i + \left[(k_j^i - k_{j-p}) - (k_{j-p+1}^i - k_{j-p})\right] \frac{c_{j-p+1}^i - c_{j-p}^i}{k_{j-p+1}^i - k_{j-p}^i}$$

$$= c_{j-p}^i + (k_j^i - k_{j-p})\beta(i, j-p+1; i, j-p).$$

Hence $\beta(i, j, i, j-p) \geq \beta(i, j-p+1, i, j-p) \geq \beta(i, j-p, i_1, j_1)$. Then

$$c_{j_1}^{i_1} \leq c_{j-p}^i + (k_{j_1}^{i_1} - k_{j-p}^i) \frac{c_j^i - c_{j-p}^i}{k_j^i - k_{j-p}^i} = c_{j-p}^i + \left[(k_{j_1}^{i_1} - k_j^i) - (k_{j-p}^i - k_j^i)\right] \frac{c_j^i - c_{j-p}^i}{k_j^i - k_{j-p}^i}$$

$$= c_j^i + (k_{j_1}^{i_1} - k_j^i)\beta(i, j; i, j-p),$$

which indicates that $\beta(i, j; i_1, j_1) \leq \beta(i, j; i, j-p)$, i.e. (13a).

The calendar butterfly spread absolute location convexity constraint C6.1 results in $\beta(i_2, j_2; i, j+q) \geq \beta(i, j+q; i, j+q-1)$. In addition, the vertical butterfly constraint C3 results in $\beta(i, j+q-1; i, j) \leq \beta(i, j+q; i, j+q-1)$, then

$$-c_j^i \leq -c_{j+q-1}^i + (k_{j+q-1}^i - k_j^i) \frac{c_{j+q}^i - c_{j+q-1}^i}{k_{j+q}^i - k_{j+q-1}^i}$$

$$= -c_{j+q-1}^i + \left[(k_{j+q-1}^i - k_{j+q}^i) - (k_j^i - k_{j+q}^i)\right] \frac{c_{j+q}^i - c_{j+q-1}^i}{k_{j+q}^i - k_{j+q-1}^i}$$

$$= -c_{j+q}^i + (k_{j+q}^i - k_j^i)\beta(i, j+q-1; i, j+q).$$

Hence $\beta(i, j+q; i, j) \leq \beta(i, j+q; i, j+q-1) \leq \beta(i_2, j_2; i, j+q)$. Then

$$c_{j_2}^{i_2} \geq c_{j+q}^i + (k_{j_2}^{i_2} - k_{j+q}^i) \frac{c_j^i - c_{j+q}^i}{k_j^i - k_{j+q}^i} = c_{j+q}^i + \left[(k_{j_2}^{i_2} - k_j^i) - (k_{j+q}^i - k_j^i)\right] \frac{c_j^i - c_{j+q}^i}{k_j^i - k_{j+q}^i}$$

$$= c_j^i + (k_{j_2}^{i_2} - k_j^i)\beta(i, j; i, j+q),$$

which indicates that $\beta(i_2, j_2; i, j) \geq \beta(i, j+q; i, j)$, i.e. (14a).

Now we consider the case when $k_{j_2}^{i_2} > k_{n_i}^i$. The same proof as above applies if we let $q = n_i - j$ and allow j to take the value n_i.

$\square$

Acknowledgements

References

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

[2] P. M. Blacque-Florentin and B. Missaoui. Nonparametric and arbitrage-free construction of call surfaces using ℓ^1-recovery, 2015.

[3] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

[4] D. T. Breeden and R. H. Litzenberger. Prices of state-contingent claims implicit in option prices. *The Journal of Business*, 51(4):621–51, 1978.

[5] E. Candes, M. Rudelson, T. Tao, and R. Vershynin. Error correction via linear programming. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, pages 668–681, 2005.

[6] P. Carr, H. Geman, D. Madan, and M. Yor. Stochastic volatility for Lévy processes. *Mathematical Finance*, 13(3):345–382, 2003.

[7] P. Carr and D. B. Madan. A note on sufficient conditions for no arbitrage. *Finance Research Letters*, 2(3):125–130, 2005.

[8] L. Cousot. Conditions on option prices for absence of arbitrage and exact calibration. *Journal of Banking & Finance*, 31(11):3377–3397, 2007.

[9] L. Cousot and M. Street. Necessary and sufficient conditions for no static arbitrage among European calls. 2004.

[10] M. H. A. Davis and D. G. Hobson. The range of traded option prices. *Mathematical Finance*, 17(1):1–14, 2007.

[11] F. Delbaen and W. Schachermayer. A general version of the fundamental theorem of asset pricing. *Mathematische Annalen*, 300(1):463–520, 1994.

[12] E. Derman and I. Kani. Riding on a smile. *Risk*, 7, 1994.

[13] M. Dixon, S. Crépey, and M. Chataigner. Deep local volatility. *arxiv preprint arxiv:2007.10462*, 2020.

[14] D. L. Donoho and M. Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via l1 minimization. *Proceedings of the National Academy of Sciences*, 100(5):2197–2202, 2003.

[15] B. Dupire. Pricing with a smile. *Risk Magazine*, 7:18–20, 1994.

[16] M. Fengler. Arbitrage-free smoothing of the implied volatility surface. *Quantitative Finance*, 9:417–428, 06 2009.

[17] M. Fengler. Option data and modeling BSM implied volatility. *Handbook of Computational Finance*, pages 117–142, 2012.

[18] M. R. Fengler and L.-Y. Hin. Semi-nonparametric estimation of the call-option price surface under strike and time-to-expiry no-arbitrage constraints. *Journal of Econometrics*, 184(2):242–261, 2015.

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

[20] S. Gerhold and I. C. Glm. Consistency of option prices under bidask spreads. *Mathematical Finance*, 30(2):377–402, 2020.

[21] R. Gribonval and M. Nielsen. Sparse representations in unions of bases. *IEEE Transactions on Information Theory*, 49(12):3320–3325, 2003.

[22] J. M. Harrison and D. Kreps. Martingales and arbitrage in multiperiod securities markets. *Journal of Economic Theory*, 20(3):381–408, 1979.

[23] S. L. Heston. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *Review of Financial Studies*, 6:327–343, 1993.

[24] P. Huber, J. Wiley, and W. InterScience. *Robust statistics*. Wiley New York, 1981.

[25] A. Ivanovas. *Option data, missing tails, and the intraday variation of implied moments*. PhD thesis, University of St. Gallen, 2015.

[26] N. Kahale. An arbitrage-free interpolation of volatilities. *Risk Magazine*, 17:102–106, 2004.

[27] H. G. Kellerer. Markov-Komposition und eine Anwendung auf Martingale. *Mathematische Annalen*, 198:99–122, 1972.

[28] D. Kreps. Arbitrage and equilibrium in economies with infinitely many commodities. *Journal of Mathematical Economics*, 8(1):15–35, 1981.

[29] H. Lim. Improved methods for implied volatility surface and implied distributions. *SSRN preprint 3561100*, 2020.

[30] L. V. Martin S. Andersen, J Dahl. Cvxopt: A python package for convex optimization. [Version 1.2.5; available at cvxopt.org].

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

[32] B. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM Journal on Computing*, 24(2):227–234, 1995.

[33] J. Ruf and W. Wang. Neural networks for option pricing and hedging: a literature review. *SSRN preprint 3486363*, 2019.

[34] M. Shaked and J. Shanthikumar. *Stochastic Orders.* Springer Series in Statistics. Springer New York, 2007.

[35] S. Stoikov. The micro-price: a high-frequency estimator of future prices. *Quantitative Finance*, 18:1–8, 2018.

[36] Y. Wang, H. Yin, and L. Qi. No-arbitrage interpolation of the option price function and its reformulation. *Journal of Optimization Theory and Applications*, 120(3):627–649, 2004.

[37] U. Wystup. *FX options and structured products.* John Wiley & Sons, Ltd, 2017.