# Simulation of Arbitrage-Free Implied Volatility Surfaces

Rama Cont and Milena Vuletić
Mathematical Institute, University of Oxford

December 2022.

**Abstract**

We present a computationally tractable method for simulating arbitrage free implied volatility surfaces. We illustrate how our method may be combined with a factor model for the implied volatility surface to generate dynamic scenarios for arbitrage-free implied volatility surfaces. Our approach conciliates static arbitrage constraints with a realistic representation of statistical properties of implied volatility co-movements. We then introduce VolGAN, a nonparametric generative model for implied volatility surfaces.

## Contents

# 1 Introduction

Market prices of options are quoted in terms of their Black-Scholes implied volatilities, obtained by inverting the Black Scholes formula given the market price of the option. It has been empirically documented across almost all options markets that the implied volatility $\Sigma_t(K,T)$ associated with a call option with exercise price $K$ and maturity date $T$ actually depends on $(K,T)$ [9, 12, 13, 17, 14, 17]. The function $\Sigma_t : (K,T) \to \Sigma_t(K,T)$ which represents this dependence, called the *implied volatility surface* at date $t$, provides a snapshot of the state of the options market at date $t$. An example is given in Figure 1 for SPX index options.

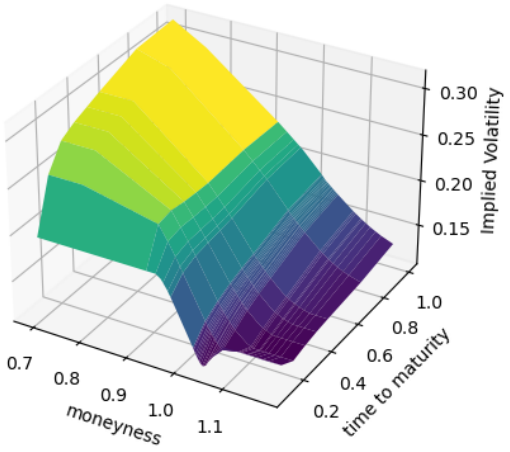



Figure 1: SPX implied volatility surface on 01/11/2021.

Two features of this surface have captured the attention of researchers in financial modelling. First, the non-flat instantaneous profile of the surface, whether it be a "smile", "skew" or the existence of a term structure, point out to the insufficiency of the Black Scholes model for matching a set of option prices at a given time instant and have led to various generalisations of the Black-Scholes model which aim at reproducing realistic instantaneous profiles for the surface $\Sigma_t(K,T)$. Second, the fact that the surface itself changes randomly with time as a result of supply and demand in the options market means that a good risk management model must not only fit the shape of the surface at a given date but also, give realistic dynamics for co-movements of implied volatilities across strikes and maturities.

Market models of implied volatility [3, 5, 7, 9, 10, 15, 23, 22] attempt to directly model the cross-section and dynamics of implied volatilities. One of the challenges in modelling implied volatility surfaces is satisfying arbitrage conditions. Indeed, the profile of the implied volatility surface cannot be ar-

bitrary: static arbitrage constraints on the values of call and put options [11] put restrictions on the possible shape of the implied volatility surface. Analytical modeling has focused on obtaining parameterisations of implied volatility surfaces which guarantee that such arbitrage constraints are satisfied [5, 23, 7]. Such models, however, are computationally challenging to implement, and even more challenging to calibrate to obtain realistic surface dynamics.

We present a computationally tractable method for simulating arbitrage-free implied volatility surfaces, which correctly captures the co-movements of implied volatilities across a range of strikes and maturities. We first illustrate how our method may be combined with a factor model for the implied volatility surface to generate dynamic scenarios for arbitrage-free implied volatility surfaces. We then introduce VolGAN, a nonparametric generative model for implied volatility surfaces. Our approach conciliates static arbitrage constraints with a realistic representation of statistical properties of implied volatility co-movements.

**Outline** Section 2 defines some notation for implied volatility surfaces and recalls some desired properties that market models of implied volatility have attempted to capture. Section 3 recalls static arbitrage constraints on the implied volatility surface and introduces a penalty function for quantifying static arbitrage violations. We propose a Weighted Monte Carlo approach [1] in Section 4 which prunes scenarios generated from a base model using this penalty function. We illustrate in Section 5 how this approach may be applied to a factor model for the implied volatility surface [9]. Finally, in Section 6 we introduce VolGAN, a a nonparametric generative model for implied volatility surfaces.

# 2 Implied volatility surfaces

Consider a market where (European) call and put options are traded on an underlying asset whose price we shall denote by $S_t$, across a range of strikes $K$ and maturity dates $T$. The Black Scholes formula for the value of a call option with time to maturity $\tau = T - t$ and moneyness $m = K/S_t$ is:

$$C_{BS}(S_t, K, \tau, \sigma) = S_t N(d_1) - K e^{-r\tau} N(d_2) \tag{1}$$

$$d_1 = \frac{-\ln m + \tau(r + \frac{\sigma^2}{2})}{\sigma\sqrt{\tau}} \qquad d_2 = \frac{-\ln m + \tau(r - \frac{\sigma^2}{2})}{\sigma\sqrt{\tau}} \tag{2}$$

where $N(u) = (2\pi)^{-1/2} \int_{-\infty}^{u} \exp(-\frac{z^2}{2}) dz$.

Conversely, given the (observed) market price $C_t^*(K, T)$ of such a call option, we the Black-Scholes implied volatility $\Sigma_t(K, T)$ is defined as the value of the volatility parameter which equates the market price with the Black-Scholes value:

$$\exists! \quad \Sigma_t(K, T) > 0, \qquad C_{BS}(S, K, T - t, \Sigma_t(K, T)) = C_t^*(K, T) \tag{3}$$

From the implicit function theorem, one expects that in general $\Sigma$ will depend on $t, S, T, K$ (and of course on the randomness $\omega$!). For fixed $(K, T)$, $\Sigma_t(K, T)$

is in general a stochastic process and, for fixed $t$, its value depends on the characteristics of the option: the maturity $T$ and the strike level $K$. The function $\Sigma_t : (K, T) \to \Sigma_t(K, T)$ is called the *implied volatility surface* at date $t$. Using the *moneyness* $m = K/S_t$ of the option, one can also represent the implied volatility surface as a function of moneyness and maturity:

$$\sigma_t(m, \tau) \quad = \quad \Sigma_t(mS_t, t + \tau) \tag{4}$$

This representation is convenient since there is usually a range of moneyness around $m = 1$ for which the options are most liquid and therefore the empirical data are most readily available. The implied volatility surface today gives a snapshot of today's market prices of vanilla options: given the current term structure of interest rates and dividends, specifying the implied volatility surface is equivalent to specifying prices of all vanilla options quoted on the market.

A plethora of models have been proposed to model the instantaneous profile in $(m, \tau)$ of the implied volatility surface: local volatility models, jump-diffusion models and stochastic volatility models with or without jumps [14]. These "smile" models are defined in terms of stochastic differential equations whose parameters describe the *infinitesimal* evolution of the asset price: since this evolution is not directly observed, *calibration* of model parameters to market prices of options turns out to be an ill-posed problem whose numerical solution is not trivial. However, even in cases where perfect calibration to today's option prices is achievable by a non-parametric model (for example, a local volatility model), getting a perfect fit of the implied volatility surfaces does not mean the model will generate realistic future scenarios. This problem can be seen in the shape of the future smile (that is, the smile for forward-start options) generated by the model: many of these models, while giving good fits to today's implied volatility/ call prices generate unrealistic forms for future smiles, thus leading to a bias in prices of forward options.

Empirical studies of the behaviour of implied volatilities of exchange-traded options on various market indices (SP500, FTSE, DAX and others) point to many common statistical properties across markets [2, 9], which we summarise here:

1. The implied volatility surface has a non-flat profile and exhibits both strike and term structure.

2. The shape of the implied volatility surface undergoes deformation in time.

3. Implied volatilities display high (positive) autocorrelation and mean-reverting behaviour.

4. The variance of the daily log-variations in implied volatility can be satisfactorily explained in terms of a small number of principal components.

5. The first principal component reflects an overall shift in the level of all implied volatilities.

6. The second principal component reflects opposite movements in (out of the money) call and put implied volatility.

7. The third principal component reflects changes in the convexity and term structure of the surface.

8. Global level shifts in implied volatility are negatively correlated with the returns of the underlying asset.

9. Relative movements of implied volatilities have low correlation with the underlying returns.

10. The projections of the surface on its principal components ("principal component processes") exhibit high (positive) autocorrelation and mean reversion over a time scale close to a month [9]. This autocorrelation structure is well represented by an AR(1)/ Ornstein Uhlenbeck process [9].

These dynamical properties of co-movements of implied volatilities and the underlying have important implications for hedging and should be reflected in any model used for risk management.

The possible shapes of implied volatility surfaces are limited by the arbitrage constraints on option prices. Call prices should be:

- increasing in time to maturity: $\partial_\tau C_{BS}(S_t, K, \tau, \sigma_t(m, \tau)) \geq 0$,
- decreasing in moneyness: $\partial_m C_{BS}(S_t, K, \tau, \sigma_t(m, \tau)) \leq 0$,
- convex in moneyness: $\partial_m^2 C_{BS}(S_t, K, \tau, \sigma_t(m, \tau)) \geq 0$.

These constraints translate to nonlinear inequalities involving $\sigma_t$, $\partial_m \sigma_t$, $\partial_m^2 \sigma_t$, $\partial_\tau \sigma_t$ [10]. The resulting constraints on $\sigma_t$ and the appropriate derivatives impose restrictions on the possible shapes.

Any option pricing model implies a dynamic model for the implied volatility surface. However, the corresponding shapes and dynamics are often intractable. "Market models" of implied volatility aim to model implied volatility directly. The goal of such models has been to correctly capture the co-movements of implied volatilities of options across different strikes and maturities while satisfying the no-arbitrage conditions, a challenging task, which has been capturing the attention of researchers for over 20 years. Statistical models of implied volatility dynamics [9, 2] have focused on correctly capturing the statistical properties of the market data and the co-movements across different strikes and maturities. These models are tractable and have been adopted for risk management applications -such as margin computations- but may lead to scenarios which are not compatible with arbitrage constraints. In parallel, analytical models have been developed with the goal of satisfying static [15] and dynamic arbitrage constraints [23, 26, 5, 7]. These models are computationally challenging to implement, simulate or estimate.

We now describe an approach which aims to conciliate computational tractability, arbitrage constraints and realistic dynamics for the surface.

# 3 Static arbitrage constraints

We are interested in a realistic setting where only a finite number of options are available. We fix a grid in moneyness and time to maturity $(\boldsymbol{m}, \boldsymbol{\tau}) = (m_i, \tau_j)_{i=1,\dots,N_m; j=1,\dots N_\tau}$, with $m_i < m_{i+1}$ and $\tau_j < \tau_{j+1}$ for all $i, j$. Using the notation introduced in Section 2, denote by

$$c(m, \tau) := \frac{1}{S} C_{BS}(S, K, \tau, \sigma) = N(d_1) - m e^{-r\tau} N(d_2)$$

the relative call price, which is a dimensionless quantity with $0 \leq c(m, \tau) \leq 1$.

## 3.1 Arbitrage constraints and arbitrage penalty

As shown by Davis and Hobson (Corollaries 4.2 and 4.3 in [11]), absence of static arbitrage among options with strikes and maturities defined by $(\boldsymbol{m}, \boldsymbol{\tau})$ is equivalent to the following three conditions:

1. Absence of calendar spread arbitrage:

$$\tau_j \frac{c(m_i, \tau_j) - c(m_i, \tau_{j+1})}{\tau_{j+1} - \tau_j} \leq 0, \tag{5}$$

   for $j = 1, \dots, N_\tau - 1$ and $i = 1, \dots, N_m$.

2. Absence of call spread arbitrage:

$$\frac{c(m_{i+1}, \tau_j) - c(m_i, \tau_j)}{m_{i+1} - m_i} \leq 0 \tag{6}$$

   for $i = 1, \dots, N_m - 1$ and $j = 1, \dots, N_\tau$.

3. Absence of butterfly spread arbitrage:

$$\frac{c(m_i, \tau_j) - c(m_{i-1}, \tau_j)}{m_i - m_{i-1}} - \frac{c(m_{i+1}, \tau_j) - c(m_i, \tau_j)}{m_{i+1} - m_i} \leq 0 \tag{7}$$

   for $i = 2, \dots, N_m - 1$ and $j = 1, \dots, N_\tau$.

Conversely, a non-zero positive part of the left-hand side of these inequalities indicates the presence of static arbitrage. We investigate whether an implied volatility surface $\boldsymbol{\sigma}_t(\boldsymbol{m}, \boldsymbol{\tau})$ is arbitrage-free by considering the inequalities (5), (6) and (7). Hence, for an implied volatility surface $\boldsymbol{\sigma}_t(\boldsymbol{m}, \boldsymbol{\tau})$, we define the **arbitrage penalty** $\Phi\left(\boldsymbol{\sigma}_t(\boldsymbol{m}, \boldsymbol{\tau})\right)$ as

$$\Phi\left(\boldsymbol{\sigma}_t(\boldsymbol{m}, \boldsymbol{\tau})\right) = p_1(\boldsymbol{\sigma}_t(\boldsymbol{m}, \boldsymbol{\tau})) + p_2(\boldsymbol{\sigma}_t(\boldsymbol{m}, \boldsymbol{\tau})) + p_3(\boldsymbol{\sigma}_t(\boldsymbol{m}, \boldsymbol{\tau})), \tag{8}$$

where

$$p_1(\boldsymbol{\sigma}_t(\boldsymbol{m}, \boldsymbol{\tau})) = \sum_{i=1}^{N_m} \sum_{j=1}^{N_\tau} \left( \tau_j \frac{c(m_i, \tau_j) - c(m_i, \tau_{j+1})}{\tau_{j+1} - \tau_j} \right)^+, \tag{9}$$

$$p_2(\boldsymbol{\sigma}_t(\boldsymbol{m},\boldsymbol{\tau}) = \sum_{i=1}^{N_m}\sum_{j=1}^{N_\tau}\left(\frac{c(m_{i+1},\tau_j)-c(m_i,\tau_j)}{m_{i+1}-m_i}\right)^+, \tag{10}$$

$$p_3(\boldsymbol{\sigma}_t(\boldsymbol{m},\boldsymbol{\tau})) = \sum_{i=1}^{N_m}\sum_{j=1}^{N_\tau}\left(\frac{c(m_i,\tau_j)-c(m_{i-1},\tau_j)}{m_i-m_{i-1}} - \frac{c(m_{i+1},\tau_j)-c(m_i,\tau_j)}{m_{i+1}-m_i}\right)^+. \tag{11}$$

The quantities $p_1, p_2, p_3$ correspond to deviations from the calendar, call, and butterfly spread arbitrage constraints, respectively. They are the positive parts of the left-hand sides of the inequalities (5), (6), and (7). If $p_1, p_2, p_3$ are all equal to zero, there is no arbitrage. Conversely, if any of $p_1, p_2, p_3$ are non-zero, then there is arbitrage present in $\boldsymbol{\sigma(m,\tau)}$. Therefore,

$$\Phi\left(\boldsymbol{\sigma}(\boldsymbol{m},\boldsymbol{\tau})\right) = 0 \iff \boldsymbol{\sigma(m,\tau)} \text{ is arbitrage-free.}$$

For better visualisation, we introduce the corresponding penalty matrices $P_1, P_2, P_3$ of size $N_m \times N_\tau$ such that

$$\begin{aligned}(P_1)_{i,j} = \left(\frac{c(m_i,\tau_j)-c(m_i,\tau_{j+1})}{\tau_{j+1}-\tau_j}\right)^+, \quad (P_2)_{i,j} = \left(\frac{c(m_{i+1},\tau_j)-c(m_i,\tau_j)}{m_{i+1}-m_i}\right)^+ \\ (P_3)_{i,j} = \left(\frac{c(m_i,\tau_j)-c(m_{i-1},\tau_j)}{m_i-m_{i-1}} - \frac{c(m_{i+1},\tau_j)-c(m_i,\tau_j)}{m_{i+1}-m_i}\right)^+,\end{aligned} \tag{12}$$

with the appropriate endpoints being set to zero: $(P_1)_{i,N_\tau} = 0$ for $i = 1, \dots N_m$, $(P_2)_{N_m,j} = 0$, $(P_3)_{N_m,j} = (P_3)_{0,j} = 0$ for $j = 1, \dots, N_\tau$. The arbitrage penalty may then be expressed as the 1-norm of the matrix $P_1 + P_2 + P_3$:

$$\Phi\left(\boldsymbol{\sigma}(\boldsymbol{m},\boldsymbol{\tau})\right) = \sum_{i=1}^{N_m}\sum_{j=1}^{N_\tau}(P_1+P_2+P_3)_{i,j} = \|P_1+P_2+P_3\|_1.$$

**Remark 1** (Extension to swaption implied volatility cube)**.** *When working with swaptions, for every tenor $T^a$ there is an implied volatility surface $\boldsymbol{\sigma}^a(m,\tau)$. Hence, one could calculate the arbitrage penalty for each possible surface (across all of the available tenors) in order to reach an aggregated penalty for the swaption implied volatility cube. That is, suppose that we have available tenors $T^1, ..., T^A$. Then we may define the arbitrage penalty for the swaption implied volatility cube $\{\boldsymbol{\sigma}_t^a(\boldsymbol{m},\boldsymbol{\tau})\}_{a=1\dots A}$ by*

$$\Phi^1(\{\boldsymbol{\sigma}_t^a(\boldsymbol{m},\boldsymbol{\tau})\}_{a=1\dots A}) = \sum_{a=1}^{A}\Phi(\boldsymbol{\sigma}_t^a(\boldsymbol{m},\boldsymbol{\tau})), \tag{13}$$

*or by*

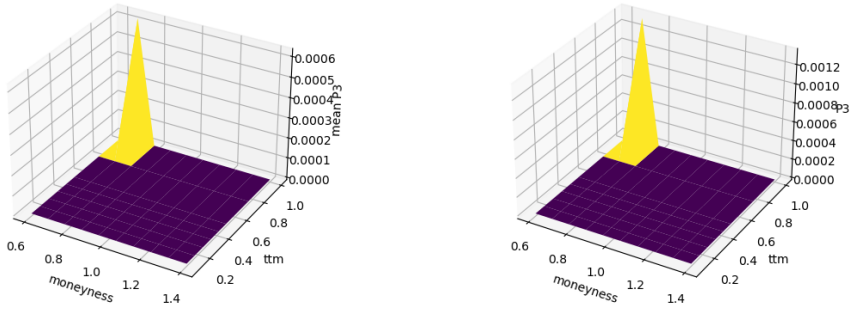
$$\Phi^\infty(\{\boldsymbol{\sigma}_t^a(\boldsymbol{m},\boldsymbol{\tau})\}_{a=1\dots A}) = \max_{a=1,\dots,A}\Phi(\boldsymbol{\sigma}_t^a(\boldsymbol{m},\boldsymbol{\tau})). \tag{14}$$

### 3.2 Behaviour of arbitrage penalty under perturbations

To gain intuition about the properties of arbitrage penalty (8) we investigate its behaviour under perturbations of an arbitrage-free implied volatility surface by IID noise and parallel shifts.

Throughout the paper, we consider a grid $(\boldsymbol{m}, \boldsymbol{\tau})$ with 10 equispaced moneyness values between 0.6 and 1.4, and time-to-maturity values of 30, 60, 91, 122, 152, 182, 273, 365 calendar days. We use time series of implied volatility from SP500 options from the OptionMetrics SPX Implied Volatility Surface File. Surfaces are linearly first in moneyness, and then in time to maturity to yield values on the grid $(\boldsymbol{m}, \boldsymbol{\tau})$. Other methods of interpolation are also possible, such as Kahale's method [19]. We note that the Implied Volatility Surface File is based on a previous interpolation of listed option prices so may not necessarily be arbitrage-free, as already noted in [7]. In the numerical results below, the initial implied volatility surface is taken to be the SPX implied volatility surface on 31/12/2021.

**Addition of IID noise** We sample $10,000$ implied volatility surfaces by adding IID noise (i.e. independent across strike and maturity) with a standard deviation of $\epsilon = 0.001$ to the initial arbitrage-free SPX implied volatility surface. We observe that 23% of generated surfaces exhibit butterfly spread arbitrage. The mean butterfly arbitrage penalty matrix $P_3$ is displayed in Figure 2a. We note that violations occur only for far from the money, long-dated options.




(a) Mean effect of adding noise.

(b) Sample parallel shift effect.

Figure 2: Butterfly penalty matrices (P3) arising from noise and parallel shifts.

**Parallel shifts** Rogers and Tehranchi [21] showed that moving implied volatility surfaces by parallel shifts will eventually result in configurations with static arbitrage. We explore this phenomenon numerically by adding a parallel shift to an initial arbitrage-free implied volatility surface and testing for static arbitrage. The absolute value of the largest negative shift is taken to be smaller than the smallest implied volatility value, guaranteeing non-negativity. The effect of parallel shifts on the arbitrage penalty are displayed in Figure 3: arbitrage

constraints are violated only for large enough positive shifts, and the arbitrage penalty grows linearly thereafter. For such large shifts, the only constraint which is violated is convexity. A sample butterfly arbitrage penalty matrix $P_3$ is displayed in Figure 2b.

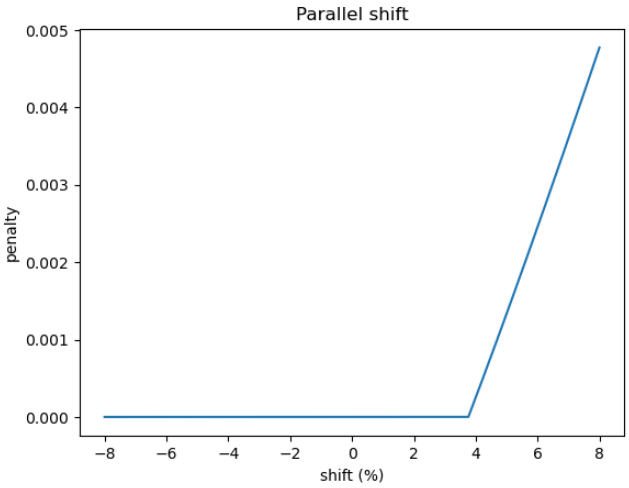



Figure 3: Arbitrage violations induced by parallel shifts.

### 3.3 Arbitrage penalty in market data

Data sources on implied volatility, such as OptionMetrics, are often interpolated from actual market quotes, a procedure which may itself introduce static arbitrage. This has been previously noted by several studies, see e.g. [7]. We study this phenomenon using daily SPX implied volatility surfaces from 2000 to end 2021. We observe non-zero arbitrage penalties, with a decomposition displayed in Figure 4 and Table 1. 90.5% of dates display no calendar arbitrage, 97.3% display no call spread arbitrage and 84.9% no butterfly arbitrage. Overall 80.2% of the observations correspond to abitrage-free surfaces.

We observe a number of spikes in arbitrage penalties. The two largest spikes happen on 29/09/2008 (during the 2008 financial crisis) and on 13/03/2020 (the start of the Covid-19 pandemic). Figure 4 shows that the majority of arbitrage violations happen during the 2008 financial crisis and during the start of the Covid-19 pandemic. For comparisons, the calendar, call, and butterfly arbitrage penalty matrices $P_1, P_2, P_3$ on dates 29/09/2008 and 13/03/2020 are displayed in Figures 5, 6, and 7, respectively. This also shows that before using such data as input for model calibration, it needs to be 'cleaned'. Our observations concur with those of Cohen, Reisinger and Wang [6], who proposed an 'Arbitrage Repair' Python package for this purpose.

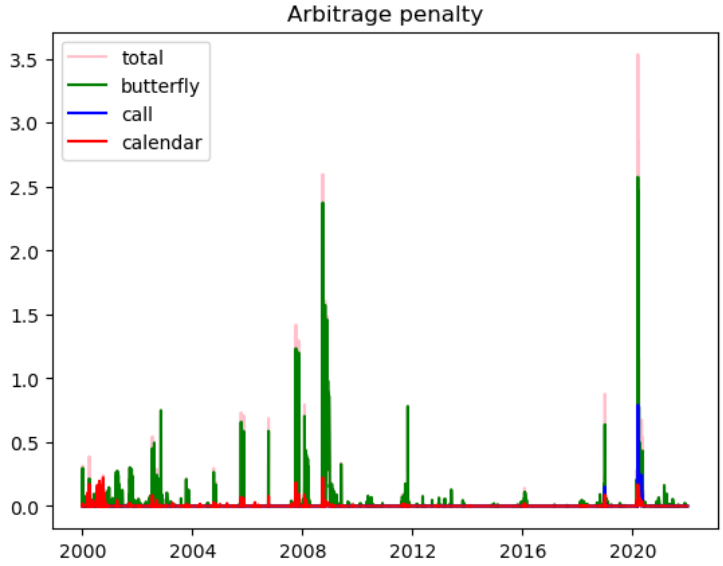
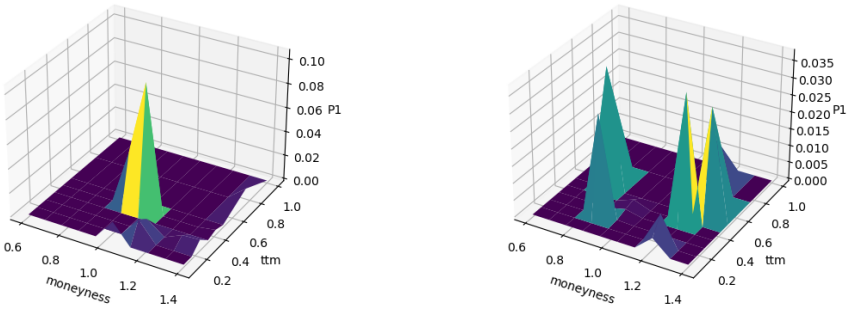
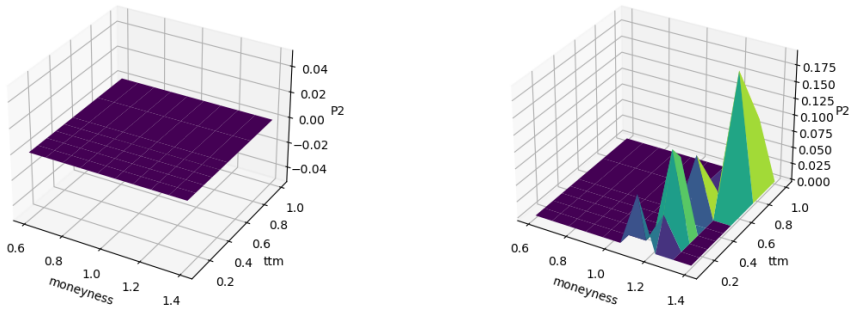



Figure 4: Arbitrage penalty decomposition for SPX options.



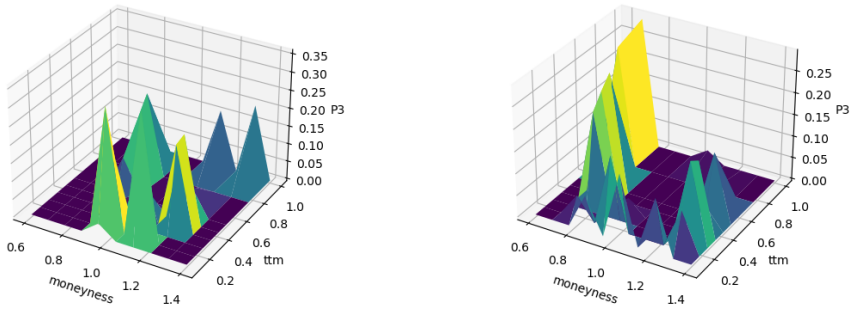

(a) 29.09.2008. (b) 13.03.2020.

Figure 5: Calendar spread arbitrage (P1) for SPX options.




(a) 29.09.2008. (b) 13.03.2020.

Figure 6: Call spread arbitrage (P2) for SPX options.




(a) 29.09.2008. (b) 13.03.2020.

Figure 7: Butterfly spread arbitrage (P3) for SPX options.

| **Penalty** | **Median** | **90th quantile** | **95th quantile** | **99th quantile** |
|---|---|---|---|---|
| **Total** $\Phi$ | 0 | 0.075 | 0.13 | 0.5 |
| **Calendar spread** $p_1$ | 0 | 0 | 0.002 | 0.038 |
| **Call spread** $p_2$ | 0 | 0 | 0 | 0.009 |
| **Butterfly** $p_3$ | 0 | 0.01 | 0.06 | 0.458 |

Table 1: Quantiles of arbitrage penalties for SPX options.

# 4 Penalising static arbitrage

## 4.1 Penalisation via scenario reweighting

Our starting point is a baseline model $\mathbb{P}_0$ for implied volatility surface dynamics, which correctly captures the co-movements and statistical properties of implied volatilities and the underlying asset, but may not necessarily be arbitrage-free. For example, this may be a factor model based on PCA, such as [2, 9].

We are interested in generating market scenarios over a time grid $\mathbb{T} = \{0, \ldots, t_{max}\}$. $\mathbb{P}_0$ may be a discrete-time or continuous-time model. For ease of notation, we will continue to denote by $\mathbb{P}_0$ the joint law of the variables $(S_t, \sigma_t(\boldsymbol{m}, \tau), t \in \mathbb{T})$ under $\mathbb{P}_0$.

Our idea is to penalise arbitrage along the paths generated by $\mathbb{P}_0$ by 'tilting' the probabilities associated with such paths. We choose $\beta > 0$ and define a new probability $\mathbb{P}_\beta$ on the space of market scenarios by

$$\frac{d\mathbb{P}_\beta}{d\mathbb{P}_0}(\omega) = \frac{\exp\left(-\beta \sum_{t\in\mathbb{T}} \Phi(\sigma_t(\boldsymbol{m}, \boldsymbol{\tau}; \omega))\right)}{Z(\beta)} \tag{15}$$

where $Z(\beta)$ is a normalisation factor:

$$Z(\beta) = \mathbb{E}^{\mathbb{P}_0}\left[\exp\left(-\beta \sum_{t\in\mathbb{T}} \Phi(\sigma_t(\boldsymbol{m}, \boldsymbol{\tau})\ )\right)\right]. \tag{16}$$

If the baseline model $\mathbb{P}_0$ is arbitrage-free then $\Phi(\sigma_t(\boldsymbol{m}, \boldsymbol{\tau})) = 0$ $\mathbb{P}_0$-almost surely so $Z(\beta) = 1$ and $\mathbb{P}_\beta = \mathbb{P}_0$. If, however, $\mathbb{P}_0$ generates surfaces which violate static arbitrage constraints, then the change of measure (15) penalises such scenarios, and may be thought of as an importance sampling method which penalises static arbitrage violations. This penalisation increases if we take large $\beta$ and as $\beta \to \infty$ we reject all scenarios violating static arbitrage constraints. Thus one may think of $1/\beta$ as a 'tolerance' for static arbitrage.

Note that $\mathbb{P}_\beta$ is absolutely continuous with respect to $\mathbb{P}_0$ so we are keeping the same paths but re-weighting them. In the case where the dynamics of variables are given by stochastic differential equations driven by Brownian motion, Girsanov's theorem implies that the re-weighting will impact the drift, but not the quadratic covariation of the variables. However, our approach does not assume that variables are driven by Brownian motion factors, and may be applied in a more general setting. Indeed, the whole procedure also makes sense for a discrete-time time series model.

## 4.2 A 'Weighted Monte Carlo' approach

We now propose a method for sampling from $\mathbb{P}_\beta$, using the idea of Weighted Monte Carlo as outlined by Avellaneda et al.[1]. We proceed as follows:

- Simulate $N$ independent paths $(\omega_i, i = 1, \ldots, N)$ from $\mathbb{P}_0$. Each path corresponds to the joint evolution of the underlying asset and the implied

volatility surface:

$$\omega_i = (S_t(\omega_i), \boldsymbol{\sigma}_t(\boldsymbol{m}, \boldsymbol{\tau}; \omega_i); t \in \{0, \dots, t_{max}\})$$

- Compute the arbitrage penalty $\varphi(\omega_i)$ along each path:

$$\varphi(\omega_i) = \sum_{t \in \{0,\dots,t_{max}\}} \Phi(\boldsymbol{\sigma}_t(\boldsymbol{m}, \boldsymbol{\tau}; \omega_i)). \tag{17}$$

- Associate a weight $w_i(\beta)$ with each path:

$$w_i(\beta) = \frac{\exp\left(-\beta\varphi(\omega_i)\right)}{\sum_{j=1}^{N} \exp\left(-\beta\varphi(\omega_j)\right)}. \tag{18}$$

- Sample from the **weighted** model $\mathbb{P}_\beta^N$ defined as

$$\mathbb{P}_\beta^N(\omega_i) = w_i(\beta) \qquad i = 1 \dots N. \tag{19}$$

  That is, instead of sampling each simulated path $\omega_i$ with probability $\frac{1}{N}$, we sample it with probability $w_i(\beta)$.

This step-by-step procedure is summarised in Table 2.

Note that we keep the same paths but modify their weight. Thus, all quantities computed along the path, such as realised volatility and realised covariances will remain the same.

As $\beta \to \infty$, $w_i(\beta) \to 0$ as soon as $\varphi(\omega_i) > 0$ so only paths with arbitrage-free implied volatility surfaces survive for large $\beta$. Hence, $\dfrac{1}{\beta}$ can be viewed as an *arbitrage tolerance* parameter.

If the model $\mathbb{P}_0$ is arbitrage-free, then the re-weighting will have no influence as for every $\beta > 0$ we will have $w_i(\beta) = \dfrac{1}{N}$, implying that $\mathbb{P}_\beta^N$ is simply the empirical distribution associated with the $N$ simulated paths. In the general case, the relative entropy of $\mathbb{P}_\beta^N$ with respect to this empirical distirbution (i.e. the uniform distirbution on $\{\omega_i, i = 1, \dots, N\}$ is an indicator of the 'distance to no-abitrage':

$$\mathcal{E}_N(\beta) = -N \ln N - N \sum_{i=1}^{N} w_i(\beta) \ln w_i(\beta). \tag{20}$$

When there is no static arbitrage in the scenarios $\omega_i$ generated by $\mathbb{P}_0$, then relative entropy is zero: $\mathcal{E}_N(\beta) = 0$. On the other hand, the model $\mathbb{P}_0$ is far from being arbitrage-free, the arbitrage penalties $\varphi(\omega_i)$ are large and the relative entropy $\mathcal{E}_N(\beta)$ will be large.

Our approach is more efficient than rejection sampling, as we sample a fixed number of paths, regardless of the initial model $\mathbb{P}_0$, so the complexity is of order $O(N)$. If the scenarios generated by $\mathbb{P}_0$ are likely to admit static arbitrage, even

INGREDIENTS

- 'Baseline model' $\mathbb{P}_0$ for implied volatility surface dynamics.
- Time grid $\mathbb{T} = \{0 = t_0 < t_1 < \cdots < t_{max}\}$
- Moneyness and time to maturity grid $(\boldsymbol{m}, \boldsymbol{\tau}) = (m_i, \tau_j)_{i=1,\ldots,N_m; j=1,\ldots N_\tau}$
- Number of paths $N$
- Arbitrage penalty parameter $\beta > 0$

**Step 1:** Simulate $N$ independent scenarios $\omega_i, i = 1, \ldots, N$ from the baseline model $\mathbb{P}_0$. Each scenario $\omega_i$ represents a joint evolution of the underlying asset $S_t$ and the implied volatility surface $\boldsymbol{\sigma}_t(\boldsymbol{m}, \boldsymbol{\tau})$ for $t \in \{0, \ldots, t_{max}\}$:

$$\omega_i = (S_t(\omega_i), \boldsymbol{\sigma}_t(\boldsymbol{m}, \boldsymbol{\tau}; \omega_i); t \in \{t_0, \ldots, t_{max}\})$$

**Step 2:** For each simulated path $\boldsymbol{\sigma}(\boldsymbol{m}, \boldsymbol{\tau})^i$, compute the arbitrage penalty

$$\varphi(\omega_i) = \sum_{t \in \mathbb{T}} \Phi(\sigma_t(\boldsymbol{m}, \boldsymbol{\tau}; \omega_i)\ )$$

**Step 3:** If $\varphi(\omega_i) = 0$ for all $i = 1..N \rightarrow$ **STOP**, else.
**Step 4:** Compute the weights

$$w_i(\beta) = \frac{\exp(-\beta\varphi(\omega_i))}{\sum_{j=1}^{N} \exp(-\beta\varphi(\omega_j))}$$

**Step 5:** Calculate the relative entropy $\mathcal{E}_N(\beta)$.
**Step 6:** Sample the scenarios $\omega_i, i = 1..N$ with probability $w_i(\beta)$:

$$\mathbb{P}_\beta^N(\omega_i) = w_i(\beta)$$

Table 2: Weighted Monte Carlo for implied volatility scenarios

if the penalty is small and arises from interpolation, rejection sampling may result in an infinite loop.

The following result, which we state for completeness, clarifies the relation between the various probability measures involved. We use the notation of Section 4.1.

**Proposition 1.**

*(i)* $\mathbb{P}_\beta^N$ *weakly converges to* $\mathbb{P}_\beta$ *as the number of scenarios* $N \to \infty$.

*(ii) Let* $U = \{\omega \in \Omega,\ \varphi(\omega) = 0\}$ *be the set of scenarios free of static arbitrage. If* $\mathbb{P}_0(U) > 0$ *then, as* $\beta \to \infty$, *then support of* $\mathbb{P}_\beta$ *concentrates on* $U$*:*

$$\forall \varepsilon > 0, \qquad \mathbb{P}_\beta\left(\{\varphi > \varepsilon\}\right) \overset{\beta\to\infty}{\to} 0. \tag{21}$$

*Proof.* (i) is a consequence of the weak law of large numbers. To show (ii), first note that if $\mathbb{P}_0$ is supported on arbitrage-free scenarios, then so is $\mathbb{P}_\beta$. Hence, suppose that $\mathbb{P}_0$ is not supported on the (closed) set $U = \{\omega \in \Omega,\ \varphi(\omega) = 0\}$. The arbitrage penalty

$$\varphi : \omega \in \Omega \mapsto \sum_{t\in\mathbb{T}} \Phi(\sigma_t; \omega)$$

defines a random variable on scenario space and $U = \{\varphi = 0\} \in \mathcal{F}_{t_{\max}}$. Note that since $\Phi$ defined by (8) is bounded, so is $\varphi$. Define

$$A_n = \{\omega \in \Omega,\ \varphi(\omega) > 1/n\} \in \mathcal{F}_{t_{\max}}.$$

Then $A = U^c = \{\varphi > 0\} = \cap_{n\geq 1} A_n$. If $\mathbb{P}_0$ is not supported on $U$, there exists $n \geq 1$ such that $\mathbb{P}_0(A_n) > 0$.

$$\mathbb{P}_\beta(A_n) = \frac{\int_{A_n} \exp\left(-\beta\varphi(\omega)\right) d\mathbb{P}_0(\omega)}{Z(\beta)}.$$

Since $\varphi = 0$ on $U \subset A_n^c$, we have

$$Z(\beta) = \int_{U^c} \exp\left(-\beta\varphi(\omega)\right) d\mathbb{P}_0(\omega) + \int_U d\mathbb{P}_0(\omega) \geq \mathbb{P}_0(U)$$

Also, since $\varphi > 1/n$ on $A_n$, we have

$$\int_{A_n} \exp\left(-\beta\varphi(\omega)\right) d\mathbb{P}_0(\omega) \leq \mathbb{P}_0(A_n) \exp(-\beta/n) \to 0 \text{ as } \beta \to \infty.$$

Hence,

$$\mathbb{P}_\beta(A_n) \leq \frac{\mathbb{P}_0(A_n)\exp(-\beta/n)}{\mathbb{P}_0(U)} \to 0 \text{ as } \beta \to \infty.$$

Taking $n > 1/\varepsilon$ yields the result. □

# 5 A factor model for implied volatility dynamics

In order to illustrate our approach, we consider a three-factor for implied volatilty dynamics introduced in [9], based on a Karhunen-Loeve decomposition of co-movements in implied volatilities.

The evolution of implied volatility surface paths in this model is given by

$$\sigma_t(m,\tau) = \sigma_0(m,\tau)\exp\left(x_t^1 f_1(m,\tau) + x_t^2 f_2(m,\tau) + x_t^3 f_3(m,\tau)\right) \tag{22}$$

where the factors $x_t^1, x_t^2, x_t^3$ correspond to *level, skew* and *curvature*. Their dynamics are modelled as independent Ornstein-Uhlenbeck processes:

$$dx_t^i = -\lambda^i x_t^i dt + \gamma^i dW_t^i, \tag{23}$$

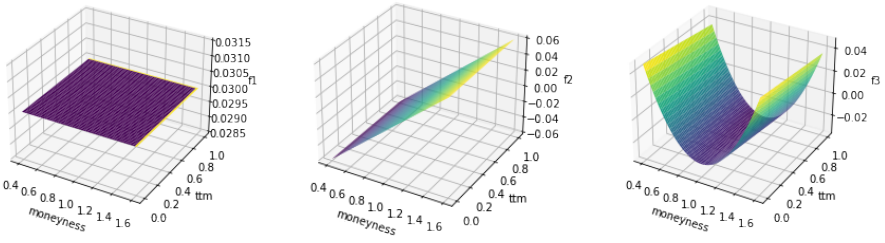
where $W_t^i$ are independent Brownian motions. The basis functions $f_1, f_2, f_3$ are the first three principal components of the log-implied volatility surface, shown in Figure 8. The price of the underlying asset $S$ is modelled as a diffusion with stochastic volatility $\sigma_t(1,0)$, which corresponds to the short-term at-the-money implied volatility:

$$dS_t = \sigma_t(1,0)dW_t^0, \qquad W^0 = \rho W^1 + \sqrt{1-\rho^2}\, B$$

where $\rho < 0$ and $B$ is a Brownian motion independent from $W^i, i = 1\dots 3$. The increments of the first factor $x_t^1$ are negatively correlated with the returns of the underlying asset: $\mathrm{cov}(W_t^0, W_t^1) = \rho t < 0$. We use $\rho = -0.5$ as an example.

Given that this model is based on a principal component analysis of market data, the simulated paths correctly capture implied volatility co-movements. Furthermore, the functional form (22) of the implied volatility surface guarantees smoothness of the surface and continuity of simulated paths.

For the simulation, we use the coefficients given in [9] for SPX implied volatilities to drive the level, skew and curvature processes $x_t^1, x_t^2, x_t^3$. The basis functions are based on the first three principal components of the market data and displayed in Figure 8. The initial condition was taken to be the arbitrage-free SPX implied volatility surface on 31/12/2021.




(a) $f_1(m,\tau)$ (b) $f_2(m,\tau)$ (c) $f_3(m,\tau)$

Figure 8: Basis functions $f_1, f_2, f_3$ corresponding to level, skew and curvature.

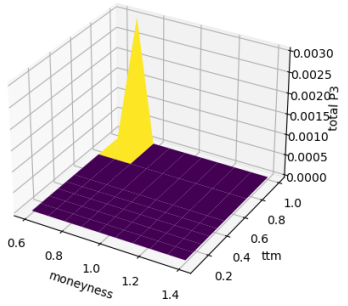



Figure 9: Three-factor model: sample butterfly penalty.

It is not clear whether the model (22)-(23) respects static arbitrage constraints, so we apply the reweighting procedure described in Table 2. We simulate $N = 100,000$ 3-month scenarios from the factor model (22)-(23). Among these scenarios, 64.8% were arbitrage-free. However, even when the arbitrage penalty of a simulated path was non-zero, it was much lower than that of market data. Quantiles of arbitrage penalties across different $\mathbb{P}_\beta$ are displayed in Table 3. When comparing the arbitrage penalty quantiles to those of market data displayed in Table 1, it is important to note that the factor model arbitrage penalties are calculated for paths, whereas the SPX market data arbitrage penalties are for individual surfaces only. In the scenarios generated by the factor model (22)-(23), only butterfly arbitrage was observed. All simulated implied volatility surfaces satisfied the absence of calendar and call spreads. A sample non-zero path butterfly penalty matrix $P_3$ is displayed in Figure 9. The pattern of violations in the butterfly constraint resemble the violations induced by the addition of IID noise perturbations in Section 3.

Table 3 displays quantiles of the arbitrage penalty $\varphi$ defined by (17) under $\mathbb{P}_\beta^N$. To compute the $q$th quantile of the arbitrage penalty under $\mathbb{P}_\beta^N$, we sort the scenarios in increasing order of arbitrage penalties $\varphi(\omega_{(1)}) \leq \varphi(\omega_{(2)}) \leq \cdots \leq \varphi(\omega_{(N)})$. The $q$th quantile is then estimated as:

$$F_\varphi^{-1}(q) = \varphi(\omega_{(k)}), \qquad k = \min\{j \in \{1, \ldots, N\} : \sum_{i=1}^{j} w(\omega_{(i)}) \geq q\}. \tag{24}$$

| $\beta$ | **0** | $\mathbf{10^2}$ | $\mathbf{10^3}$ | $\mathbf{10^4}$ | $\mathbf{10^5}$ |
|---|---|---|---|---|---|
| **90th quantile** | 0.044 | 0.001 | 0 | 0 | 0 |
| **95th quantile** | 0.09 | 0.004 | 0.0001 | 0 | 0 |
| **99th quantile** | 0.206 | 0.014 | 0.001 | 0.0004 | 0 |

Table 3: Quantiles of arbitrage penalty under $\mathbb{P}_\beta^N$.

As we increase $\beta$, we are less likely to sample paths exhibiting a non-zero arbitrage penalty. Table 3 shows that with $\beta = 10^5$, 99% of scenarios are arbitrage-free. In this example, when $\beta = 10^{10}$, we are left with arbitrage-free paths only. That is, under $\mathbb{P}_\beta$, with $\beta = 10^{10}$, we sample arbitrage-free paths with probability one.

Figure 10 shows the relative entropy $\mathcal{E}(\beta)$, as a function of $\beta$. We observe a sharp transition around $\beta = 100$, suggesting that for $\beta \gg 10^2$ the penalisation eliminates scenarios with arbitrage.

The histogram of weights $w_i(10^2)$ (Figure 11) illustrates the clustering of weights into two groups: those corresponding to arbitrage-free scenarios, which are equally weighted, and those corresponding to scenarios with arbitrage penalty $\varphi(\omega_i) > 0$ whose weights are driven very close to zero. We note that even with $\beta = 10^2$, some of the weights are already very close to zero.

We conclude that for the factor model (22)-(23) the impact of the penalty step is small in terms of entropy distance.

We therefore have a scalable, explainable model for implied volatility surfaces, which correctly captures the co-movements, is tractable, and when combined with our Weighted Mote Carlo approach, is arbitrage-free.

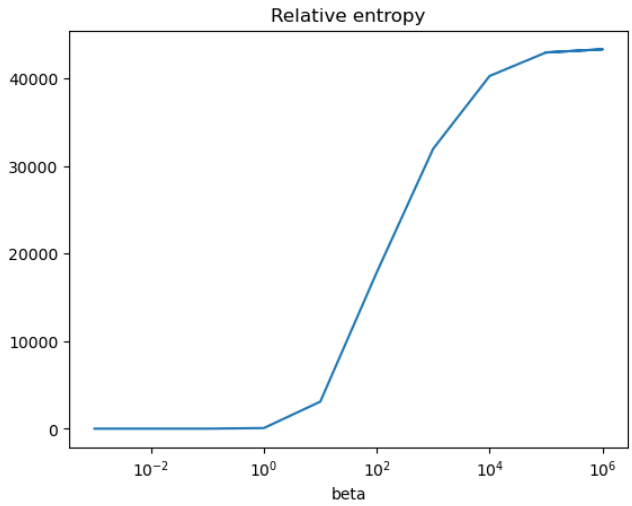



Figure 10: Relative entropy $H(\mathbb{P}_\beta^N|\mathbb{P}_0^N)$ as a function of $\beta$.

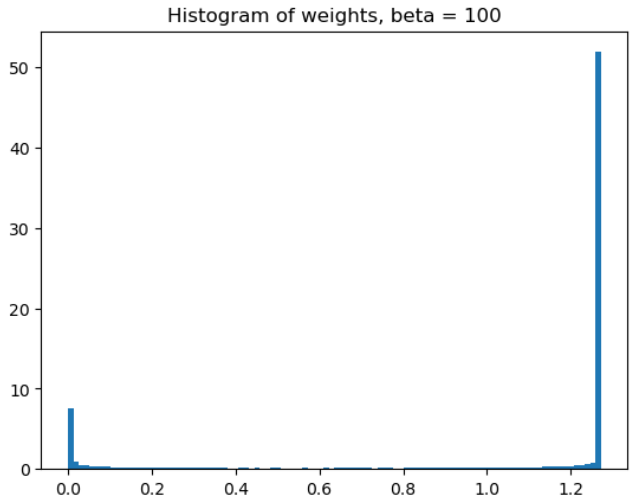



Figure 11: Histogram of $Nw(\beta)$ with $\beta = 10^2$.

# 6 VolGAN: a generative model for arbitrage-free implied volatility surfaces

We now apply our scenario re-weighting approach for penalising arbitrage to design a non-parametric generative model for implied volatility surfaces.

Generative Adversarial Networks [16] are increasingly used for synthetic data generation, especially for photo and video generation, where they are state-of-the-art. GANs have recently been employed in various financial settings: Quant-GAN [25] and FIN-GAN [24] for generating financial time series; TailGAN [8] for generating tail risk scenarios; and for Deep Hedging [4].

We use here a Conditional GAN [20], which consists of two neural networks: a *generator*, and a *discriminator*. The generator receives noise input and a deterministic condition and outputs a simulation which is supposed to mimic the conditional distribution of the training data, given the condition. The discriminator is a classifier, whose goal is to determine whether the output of the generator is a plausible sample drawn from the target conditional distribution. The generator learns iteratively from the feedback it receives from the discriminator. A diagram of a conditional GAN is shown in Figure 12.

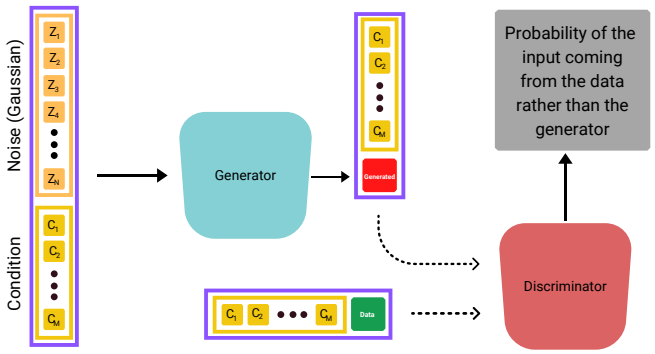



Figure 12: Conditional GAN.

## 6.1 A customised GAN for implied volatility surfaces

We design a customised conditional GAN, which we call VolGAN, to simulate daily log-returns of the underlying asset and implied volatilities over a fixed moneyness/maturity grid $(\mathbf{m}, \boldsymbol{\tau})$. The inputs (the condition) of VolGAN are the previous day's log implied volatility surface, the previous two (daily) returns of the underlying, and the realised volatility on the previous day. VolGAN learns the joint conditional distribution of the movements of the log implied volatility surface and the underlying returns, given the previous two returns of the underlying, previous log implied volatility surface, and the last realised volatility.

**Architecture** The generator $G$ is a three-layer feedforward dense neural network, with the first two activations being softplus, and the final layer is an affine layer. The noise is (standard) Gaussian of dimension 16. The first layer consists of 8 neurons, whereas the second layer contains 16 neurons. Similarly, the discriminator $D$ is a two-layer feedforward neural network, with softplus and sigmoidal activation functions and layer sizes of 16 and 1, respectively. The discriminator has a simpler architecture than the generator, as it is of the utmost importance to keep the two neural networks in balance. If the discriminator is too skilled at classifying its inputs, the generator cannot learn from the feedback it receives due to weak gradients. The architecture of the generator is displayed in Figure 13, and the architecture of the discriminator is shown in Figure 14.

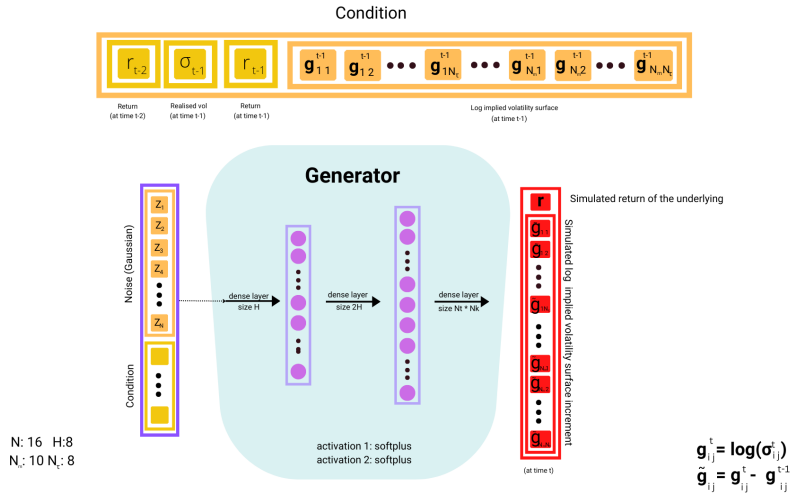



Figure 13: VolGAN generator architecture.

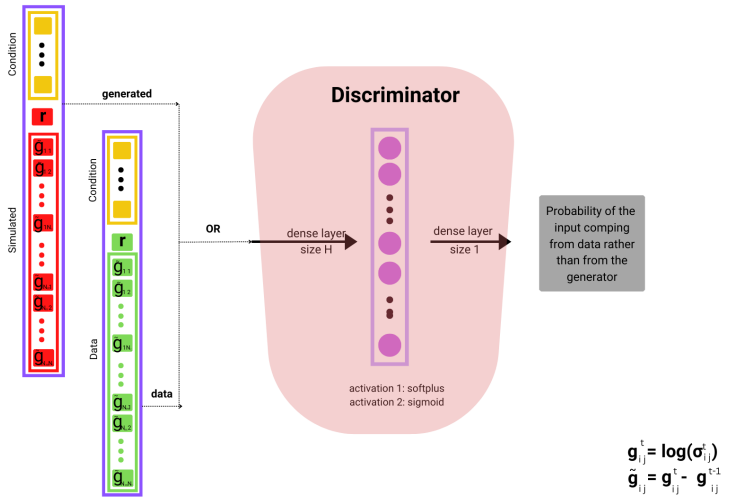



Figure 14: VolGAN discriminator architecture.

**Loss function** The core component of VolGAN is its loss function. Using the standard binary cross-entropy (BCE) loss would result in irregular surfaces. In order to generate smooth surfaces, we include a smoothness penalty (discrete Sobolev semi-norm) in the objective $J^{(G)}(\theta_d, \theta_g)$ for the generator:

$$\begin{aligned} J^{(G)}(\theta_d, \theta_g) = &- \frac{1}{2}\mathbb{E}\left[\log[D(G(\boldsymbol{z}; \theta_g); \theta_d)]\right] \\ &+ \alpha\mathbb{E}\left[L_m(G(\boldsymbol{z}; \theta_g); \theta_d)\right] + \beta\mathbb{E}\left[L_\tau(G(\boldsymbol{z}; \theta_g); \theta_d)\right], \end{aligned} \tag{25}$$

where the smoothness terms in $m$ and $\tau$, $L_m$ and $L_\tau$ are defined as

$$L_m(\boldsymbol{\sigma}) = \sum_{i,j} \frac{\left(\boldsymbol{\sigma}(m_{i+1}, \tau_j) - \boldsymbol{\sigma}(m_i, \tau_j)\right)^2}{\left|m_{i+1} - m_i\right|^2}, \tag{26}$$

$$L_\tau(\boldsymbol{\sigma}) = \sum_{i,j} \frac{\left(\boldsymbol{\sigma}(m_i, \tau_{j+1}) - \boldsymbol{\sigma}(m_i, \tau_j)\right)^2}{\left|\tau_{j+1} - \tau_j\right|^2}, \tag{27}$$

$\alpha > 0$ and $\beta > 0$; $\boldsymbol{z}$ is the $\mathcal{N}(0, 1)$ noise; $\theta_g$ and $\theta_d$ are the parameters of the generator and the discriminator networks, respectively. In the equations above, $\boldsymbol{\sigma}(m, \tau)$ is the generated implied volatility surface. That is, let

$$\boldsymbol{g}_{t-1}(m, \tau) = \log\left(\boldsymbol{\sigma}_{t-1}(m, \tau)\right)$$

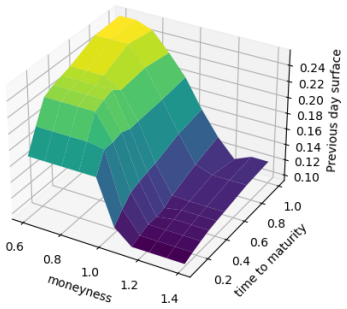
be the logarithm of the implied volatility surface on the day $t - 1$. We are interested in simulating the next-day implied volatility surface. Hence, we take the log implied vol surface increment output of the generator $\tilde{\boldsymbol{g}}(m, \tau)$ given the condition corresponding to the day $t - 1$, and simulate $\boldsymbol{\sigma}(m, \tau)$ as

$$\boldsymbol{\sigma}(m, \tau) = \exp\left(\boldsymbol{g}_{t-1}(m, \tau) + \tilde{\boldsymbol{g}}(m, \tau)\right).$$

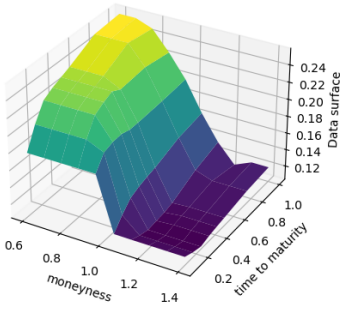
## 6.2 Results

We use SPX implied volatility data and the same $(\boldsymbol{m}, \boldsymbol{\tau})$ grid as described in Section 3. The period used for training is from 03/01/2000 to 08/08/2017, and data from 09/08/2017 to 21/10/2019 is used for testing.

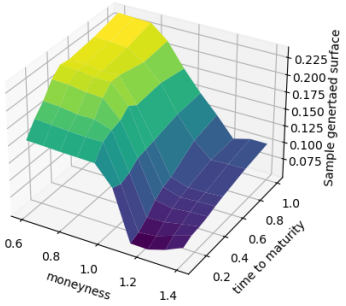
The smoothness penalty in the objective function plays two key roles: producing smooth surfaces and reducing arbitrage penalties of its outputs. Figure 15 compares the shapes of generated surfaces of VolGAN and of a classical GAN trained using Binary Cross-Entropy (BCE) loss, given the same input condition. We note that training without the smoothness penalty results in irregular surfaces, which is undesirable and unrealistic.



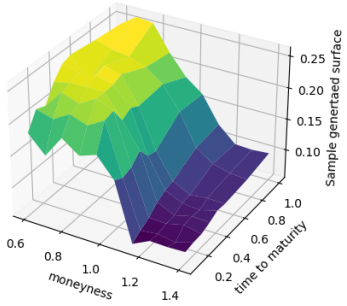

(a) Input surface (market data).




(b) SPX surface on the target day (market data).




(c) VolGAN output surface.




(d) BCE GAN output surface.

Figure 15: Sample generated surfaces.

Table 4 displays the mean, the standard deviation, as well as the median of arbitrage penalties associated with generated surfaces. We observe that their values are low, compared with typical values observed in input market data, as shown in Figure 16.

Table 4 gives the statistics for the time series shown in Figure 16. We note the similarity in arbitrage penalties exhibited by raw VolGAN outputs (i.e.

before reweighting of scenarios) compared to the SPX implied vol market (input) data, as well as a major difference in arbitrage penalties produced by VolGAN outputs compared to that of the outputs of a BCE GAN. VolGAN outputs (after reweighting) have lower arbitrage penalties than that of the input (market) data. The standard BCE GAN, which is the GAN with the same architecture as VolGAN, but trained without the smoothness penalty, leads to significantly higher arbitrage penalties than the (input) market data, due to the roughness of the generated surfaces. Raw VolGAN outputs have similar arbitrage penalties to market inputs, as seen from Table 4 and from the blue path (generated data) being close to the red path (market data) in Figure 16. Finally, we note a reduction in arbitrage penalties in VolGAN, after applying the re-weighting procedure, which is visible in Table 4. Furthermore, the green line corresponding to the reweighted scenarios in Figure 16 is always below the blue line (VolGAN outputs) and the red (input data).

| | **Mean** | **Std** | **Median** |
|---|---|---|---|
| **Market data** | 0.0058 | 0.0419 | 0 |
| **BCE GAN** | 0.027 | 0.083 | 0.012 |
| **Raw VolGAN** (before reweighting) | 0.0058 | 0.034 | $2 \times 10^{-7}$ |
| **VolGAN** | 0.0046 | 0.02 | $2 \times 10^{-7}$ |

Table 4: Arbitrage penalties in SPX implied volatility market data (test set) vs generated data via GANs trained using (i) BCE loss only (ii) VolGAN loss (iii) VolGAN reweighted scenarios ($\beta = 1000$). Standard deviation and median for GAN outputs correspond to the standard deviation and the median of (reweighted) mean outputs given 100 samples.

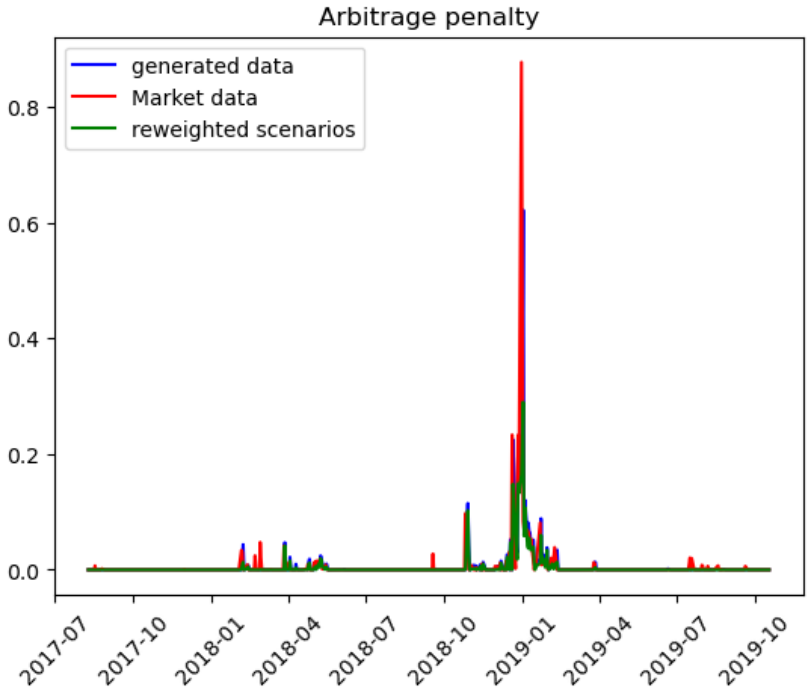
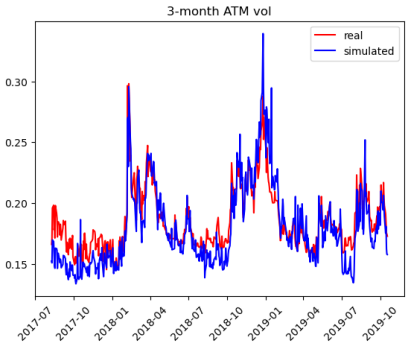
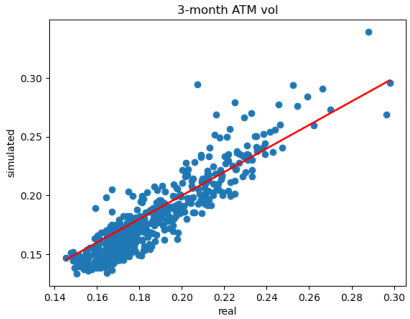



Figure 16: Arbitrage penalty in SPX implied volatility market data (input: red), vs mean arbitrage penalty of surfaces generated via VolGAN, before (blue) and after reweighting with $\beta = 1000$ (green).

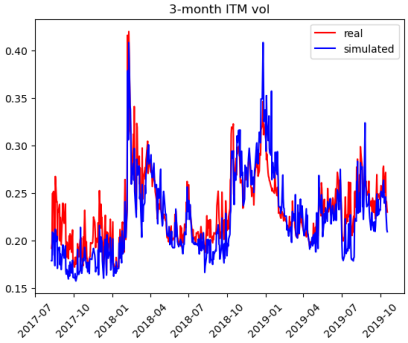
In Figure 17 we compare the 3-month market data SPX implied volatilities with the VolGAN one-day-ahead simulated ones. We notice that VolGAN is able to replicate the market data well. It is able to respond to shocks appropriately, and the relationship between the simulated values and the realised ones is clustered around the line $x = y$. The same holds for spot volatility $\sigma_t(1, 0)$, and for the movements in the underlying, displayed in Figure 18. The correlation between the simulated returns of the underlying and the simulated returns of the spot volatility is $-52\%$. We observe that the correlation between underlying returns and *relative* movements in implied vols is low, as observed in empirical data [9].



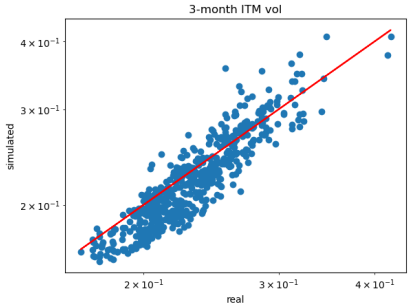

(a) 3 month ATM.



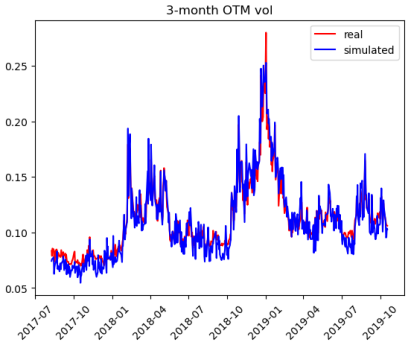

(b) 3 month ATM.



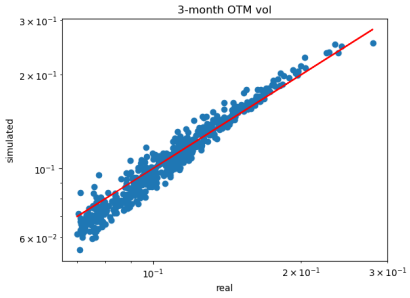

(c) 3 month $m = 0.75$.




(d) 3 month $m = 0.75$




(e) 3 month $m = 1.25$.




(f) 3 month $m = 1.25$.

Figure 17: Simulated (one-day-ahead) vs actual SPX implied volatility, with $\tau = 0.25, m \in \{0.75, 1, 1.25\}$

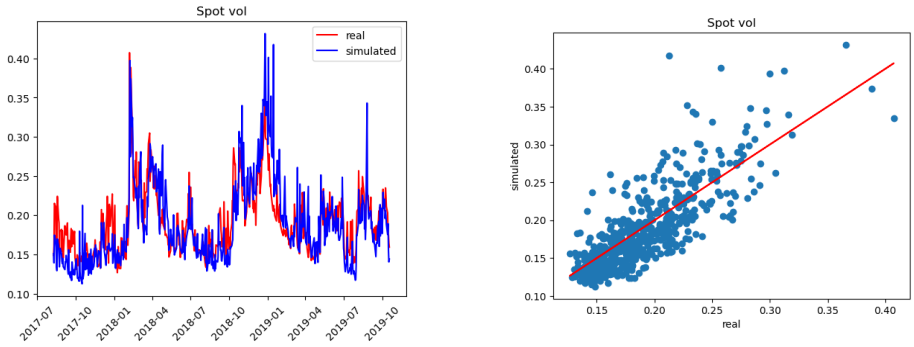



Figure 18: Simulated (one-day-ahead) vs SPX spot volatility $\sigma_t(1,0)$ for SPX.

As VolGAN is trained on inputs which are not always arbitrage-free (see discussion in Section 3.3), it is not realistic to expect outputs to have zero arbitrage penalties, so the scenario reweighting step is necessary. However, we note that our GAN-based approach produces arbitrage penalties similar to that of the input data and, when combined with our re-weighting procedure, it outputs implied volatility surfaces with lower arbitrage penalties than that of the (input) market data. VolGAN is furthermore able to generate smooth, realistic-looking implied volatility surfaces, which resemble market data, undergo deformation in time, and have the appropriate (negative) correlation with the returns of the underlying.

**Remark 2** (Further training details)**.** *The hyperparameters $\alpha, \beta > 0$ are chosen by matching the norms of gradients. For the first* 100 *epochs of training, we train the network based on the BCE loss only ($-\frac{1}{2}\mathbb{E}_{p_{\boldsymbol{z}}}[\log[D(G(\boldsymbol{z};\theta_g);\theta_d)]]$), and at each update, we calculate the gradient norms with respect to each of the three loss function terms: BCE, $L_m$, $L_\tau$. We then set $\alpha$ and $\beta$ to be the means of observed ratios of the gradient norms with respect to the BCE term and the gradient norms with respect to $L_m$ and $L_\tau$, respectively. We then continue to train* VolGAN *with the loss function defined by Equation 25 for* 10000 *epochs, using an alternating direction method i.e. one discriminator update for each generator update. The optimiser used is RMSProp [18] and the learning rates of both networks are set to* 0.0001*.*

# 7 Conclusion

We introduced a simple and computationally tractable method for simulating arbitrage-free implied volatility surfaces. Our approach offers flexibility with respect to the underlying model for implied volatility dynamics, whilst preserving the co-movements across strikes and maturities.

We have shown how a PCA-based multifactor model such as [9] may be combined with a Weighted Monte Carlo method to conciliate static arbitrage

constraints with a statistically realistic representation of co-movements of implied volatilities.

We furthermore introduced VolGAN, a non-parametric method for generating arbitrage-free implied volatility surfaces, based on a customised conditional Generative Adversarial Network which learns from market data to simulate realistic movements in the implied volatility surface. This model is fully non-parametric and allows for straightforward extensions to multi-asset cases.


## Funding

Milena Vuletić's research is supported through the *EPSRC Centre for Doctoral Training in Mathematics of Random Systems: Analysis, Modelling and Simulation* (ESPRC Grant EP/S023925/1).

## Acknowledgements

We thank Katia Babbar, Andrey Chirikhin, Samuel N. Cohen, Bruno Dupire, Terry Lyons, Fabio Mercurio, Christoph Reisinger, Justin Sirignano and seminar participants at QuantMinds 2022 for helpful comments and remarks.


# References


[1] M. Avellaneda, R. Buff, C. Friedman, N. Grandechamp, L. Kruk, and J. Newman, *Weighted Monte Carlo: a new technique for calibrating asset-pricing models*, International Journal of Theoretical and Applied Finance, 4 (2001), pp. 91–119.

[2] M. Avellaneda, B. Healy, A. Papanicolaou, and G. Papanicolaou, *PCA for Implied Volatility Surfaces*, The Journal of Financial Data Science, 2 (2020), pp. 85–109.

[3] K. A. Babbar, *Aspects of stochastic implied volatility in financial markets*, PhD thesis, Imperial College London, 2001.

[4] H. Buehler, L. Gonon, J. Teichmann, and B. Wood, *Deep hedging*, Quantitative Finance, 19 (2019), pp. 1271–1291.

[5] R. Carmona, Y. Ma, and S. Nadtochiy, *Simulation of Implied Volatility Surfaces via Tangent Lévy Models*, SIAM Journal on Financial Mathematics, 8 (2017), pp. 171–213.

[6] S. N. Cohen, C. Reisinger, and S. Wang, *Detecting and Repairing Arbitrage in Traded Option Prices*, Applied Mathematical Finance, 27 (2020), pp. 345–373.

[7] S. N. Cohen, C. Reisinger, and S. Wang, *Arbitrage-free neural-SDE market models*, 2021.

[8] R. Cont, M. Cucuringu, R. Xu, and C. Zhang, *Tail-gan: Nonparametric scenario generation for tail risk estimation*, arXiv preprint arXiv:2203.01664, (2022).

[9] R. Cont and J. da Fonseca, *Dynamics of implied volatility surfaces*, Quant. Finance, 2 (2002), pp. 45–60.

[10] R. Cont, J. d. Fonseca, and V. Durrleman, *Stochastic models of implied volatility surfaces*, Economic Notes, 31 (2002), pp. 361–377.

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

[12] B. Dumas, J. Fleming, and R. E. Whaley, *Implied volatility functions: Empirical tests*, The Journal of Finance, 53 (1998), pp. 2059–2106.

[13] B. Dupire, *Pricing with a smile*, Risk, 7 (1994), pp. 18–20.

[14] J. Gatheral, *The volatility surface: a practitioner's guide*, John Wiley & Sons, 2011.

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

[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, *Generative adversarial nets*, in Advances in Neural Information Processing Systems, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, eds., vol. 27, Curran Associates, Inc., 2014.

[17] R. Heynen, *An empirical investigation of observed smile patterns*, Review of Futures Markets, 13 (1994), pp. 317–317.

[18] G. Hinton, N. Srivastava, and K. Swersky, *Neural Networks for Machine Learning, Lecture 6*. Coursera, 2012.

[19] N. Kahale, *An Arbitrage-free Interpolation of Volatilities*, Risk, 17 (2003).

[20] M. Mirza and S. Osindero, *Conditional Generative Adversarial Nets*, 2014.

[21] L. C. G. Rogers and M. R. Tehranchi, *Can the implied volatility surface move by parallel shifts?*, Finance Stoch., 14 (2010), pp. 235–248.

[22] P. J. Schönbucher, *A market model for stochastic implied volatility*, Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences, 357 (1999), pp. 2071–2092.

[23] M. Schweizer and J. Wissel, *Arbitrage-free market models for option prices: The multi-strike case*, Finance and Stochastics, 12 (2008), pp. 469–505.

[24] S. Takahashi, Y. Chen, and K. Tanaka-Ishii, *Modeling financial time-series with generative adversarial networks*, Physica A: Statistical Mechanics and its Applications, 527 (2019), p. 121261.

[25] M. Wiese, R. Knobloch, R. Korn, and P. Kretschmer, *Quant GANs: deep generation of financial time series*, Quantitative Finance, 20 (2020), pp. 1419–1440.

[26] J. S. Wissel, *Arbitrage-free market models for liquid options*, PhD thesis, ETH Zurich, 2008.