FAST EXACT JOINT S&P 500/VIX SMILE CALIBRATION IN DISCRETE AND CONTINUOUS TIME

FLORIAN BOURGEY[1] AND JULIEN GUYON[2]

Abstract. We introduce a novel discrete-time-continuous-time exact calibration method: we first build an S&P 500/VIX jointly calibrated discrete-time model that is later extended to continuous time by martingale interpolation. The benefit is that both steps can be made much faster than the known methods that directly calibrate a continuous-time model. We propose Newton–Sinkhorn and implied Newton algorithms that are much faster than the Sinkhorn algorithm that (Guyon, Risk, April 2020) [11] used to build the first arbitrage-free model exactly consistent with S&P 500 and VIX market data. Using a (purely forward) Markov functional model, we then quickly build an arbitrage-free continuous-time extension of this discrete-time model. Additionally, new model-free bounds on S&P 500 options emphasize the value of the VIX smile information. Extensive numerical tests are conducted.

1. Introduction

In [11, 14], it was shown how to build a nonparametric discrete-time arbitrage-free model that perfectly matches market data on Standard & Poor's 500 index value (SPX) futures, SPX options, Chicago Board Options Exchange Volatility Index (VIX) futures, and VIX options. The probability distribution is built by minimizing the relative entropy with respect to a reference probability measure, and this Schrödinger problem is numerically solved using an extended Sinkhorn algorithm. This provided the first exact solution to this *joint calibration problem*, a difficult problem (especially for short maturities) that had eluded quants for many years. Jointly calibrating to SPX and VIX futures and options is important to prevent arbitrage and ensure accurate pricing of liquid hedging instruments; calibrating to VIX derivatives means incorporating market information on SPX forward volatilities. Figure 1.1 showcases the additional information contributed by the VIX by quantifying how model-free bounds for SPX path-dependent payoffs tighten when the prices of VIX futures and VIX options are included. The extra information is also seen in Table 2 where we compare the prices of several options in models that are all calibrated to SPX smiles but not all calibrated to VIX futures and smiles. Table 2 shows that to avoid mispricing some payoffs, in particular forward-starting payoffs, which are sensitive to forward volatilities, it is important that the model also fits VIX futures and VIX options, even when the payoff depends only on SPX prices.

The aim of this paper is twofold:

(1) **Speed up the construction of the discrete-time model** by turning to Newton-type methods for solving the Schrödinger system. In Section 3, we numerically show that a mixed Newton–Sinkhorn method and an implied Newton method converge much faster than the Sinkhorn algorithm.

(2) **Quickly build a continuous-time extension of the model**, allowing the pricing of options depending on SPX values at any date t, while ensuring calibration to the market smiles of SPX and VIX (Section 4).

Part (1) draws inspiration from De March [6], who explored the entropic approximation for discrete-time multidimensional martingale optimal transport, excluding VIX, using Newton's method. In Part (2), our continuous-time construction may appear similar to the Bass local volatility of [4] but ct fundamentally different. First and foremost, unlike in [4], our purely forward Markov functional construction does not require solving a fixed-point problem; it is thus much faster. This is because we first build an arbitrage-free *multimarginal* discrete-time model consistent with market data. Second, our continuous-time interpolated model fits not only SPX option prices but also VIX market data.

Date: February 2, 2024

[1] Bloomberg L.P., Quantitative Research, 731 Lexington Ave, New York, NY 10022, USA, fbourgey@bloomberg.net.

[2] CERMICS, École des Ponts ParisTech, julien.guyon@enpc.fr. Julien Guyon is supported by the BNP Paribas chair *Futures of Quantitative Finance*.

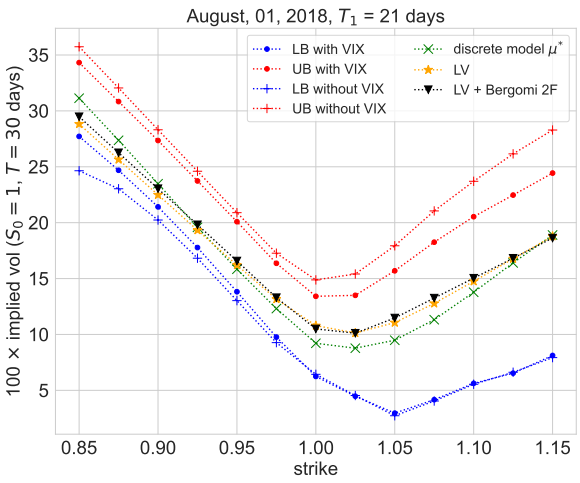

FIGURE 1.1. Model-free bounds of forward-starting calls $(S_{T_2}/S_{T_1} - K)_+$, with and without VIX options data. LB (resp., UB) denotes the lower bound (resp., upper bound) along with prices obtained with the local volatility model (LV), the two-factor Bergomi stochastic local volatility model (LV + Bergomi 2F), and the discrete minimum-entropy jointly calibrated model μ^* [11] that we build in the next section. We refer to [18] for details on how to compute model-free bounds.

Combining Parts (1) and (2), we thus quickly (in less than a minute) build a continuous-time model that is, by construction, exactly calibrated to SPX and VIX smiles and futures. By contrast, other known continuous-time exact solutions to the joint calibration problem [10, 12] are more involved and demand significantly more computation time. Approximate parametric continuous-time solutions, including those based on rough or rough-like path-dependent volatility models, [9, 15, 16, 19], classical stochastic volatility models [1], or signature-based models [5], are also costly in terms of computation time. This is the main benefit of our novel *discrete-time-continuous-time* calibration method: both steps (1) and (2) are much faster than the known methods that directly calibrate a continuous-time model.

A natural practical application of our continuous-time model is the pricing and hedging of structured products by exotics desks, see Table 2. With our model, the pricing and hedging of structured products on the SPX indeed take into account the whole information given by SPX smiles (the risk-neutral distributions of future SPX values) as well as the whole information brought by VIX futures and VIX smiles (the risk-neutral distributions of some future SPX implied volatilities). Once the model is calibrated (this takes less than a minute per VIX expiry), it is straightforward to implement and use, as it is a Markov functional model that involves simulating only one Brownian motion, along with the VIX at VIX future expiries. The model can also be used for computing reserves and other valuation adjustments, and for assessing model risk.

After a brief reminder on martingale optimal transport in Section 2, Section 3 deals with Part (1), while Section 4 is devoted to Part (2), i.e., the continuous-time extension.

2. A BRIEF REMINDER ON (DISPERSION-CONSTRAINED) MARTINGALE OPTIMAL TRANSPORT

2.1. **Setting and notations.** Let $T_1 > 0$ denote a VIX future maturity and set $T_2 = T_1 + \tau$ where $\tau = 30$ days. For simplicity, assume zero interest rates, repos, and dividends. We take as given the full market smiles of the SPX index S at T_1 and T_2, i.e., the full continuum of SPX call prices $C_i(K)$ of maturity T_i for any $i \in \{1, 2\}$ and all strikes $K \geq 0$, as well as the full market smile of the VIX index V at T_1, i.e., the full continuum of VIX call prices $C_V(K)$ for all strikes $K \geq 0$. For $i \in \{1, 2\}$, we use the short-hand notation $S_i := S_{T_i}$. We use the term forward-starting log contract (FSLC) to denote the financial derivative that pays $-\frac{2}{\tau}\ln(S_2/S_1)$ at T_2. From the VIX definition (substituting the strip of out-of-the-money options with the log contract for simplicity), the price at T_1 of the FSLC is V^2.

For each maturity $T_i, i \in \{1, 2\}$, the absence of static SPX arbitrage (or butterfly arbitrage) is equivalent to the existence of a risk-neutral measure $\mu_i := \frac{\partial^2 C_i}{\partial K^2}, i \in \{1, 2\}$, such that the price of any vanilla option $u_i(\cdot)$

written on S_i is the expectation $\mathbb{E}^i[u_i(S_i)] := \mathbb{E}^{\mu_i}[u_i(S_i)]$ of the payoff under μ_i. Similarly, by the absence of static VIX arbitrage, there exists a risk-neutral measure $\mu_V := \frac{\partial^2 C_V}{\partial K^2}$ such that the price of any vanilla option $u_V(\cdot)$ written on V is the expectation $\mathbb{E}^V[u_V(V)] := \mathbb{E}^{\mu_V}[u_V(V)]$ of the payoff under μ_V.

From the absence of dynamic SPX arbitrage (or calendar arbitrage), μ_1 and μ_2 are in the convex order, i.e., $\mathbb{E}^1[f(S_1)] \leq \mathbb{E}^2[f(S_2)]$ for any convex function $f : \mathbb{R}_{>0} \to \mathbb{R}$, even if we allow trading in the FSLC at T_1. By the absence of arbitrage, the price of S_i at time 0 is the initial SPX spot value $S_0 > 0$, i.e., $\mathbb{E}^i[S_i] = S_0$. Furthermore, $\mathbb{E}^V[V] = F_V \geq 0$ where F_V is the value at time 0 of the VIX future maturing at T_1. Finally, for the log contracts and the VIX squared to have finite prices, the following assumption is in force in what follows.

Assumption 1. *The given marginals μ_1, μ_V, μ_2 satisfy, for any $i \in \{1,2\}$, $\mathbb{E}^i[S_i] = S_0$, $\mathbb{E}^i[|\ln S_i|] < \infty$ and $\mathbb{E}^V[V] = F_V$, $\mathbb{E}^V\left[V^2\right] < \infty$.*

Let $\mathcal{X} := \mathbb{R}_{>0} \times \mathbb{R}_{\geq 0} \times \mathbb{R}_{>0}$ and define the strictly convex function $L : x \in \mathbb{R}_{>0} \mapsto -\frac{2}{\tau}\ln x$. For a probability distribution ρ on $\mathbb{R}$, we denote the associated cumulative distribution as F_ρ, i.e., $F_\rho(x) = \rho((-\infty, x])$ for every $x \in \mathbb{R}$. Let $\mathcal{P}\left(\mathbb{R}^2_{>0}\right)$ (resp., $\mathcal{P}(\mathcal{X})$) denote the set of all probability measures on $\mathbb{R}^2_{>0}$ (resp., $\mathcal{X}$).

Let $\mathcal{U}^V$ be the set of all measurable functions $u_1, u_2 : \mathbb{R}_{>0} \to \mathbb{R}$, $u_V : \mathbb{R}_{\geq 0} \to \mathbb{R}$, $\Delta_S, \Delta_L : \mathbb{R}_{>0} \times \mathbb{R}_{\geq 0} \to \mathbb{R}$ satisfying $u_i \in \mathrm{L}^1(\mu_i)$ for $i \in \{1, V, 2\}$, and Δ_S, Δ_L bounded. We use the shorthand notation

$$\Delta_S^{(S)}(s_1, v, s_2) := \Delta_S(s_1, v)(s_2 - s_1), \qquad \Delta_L^{(L)}(s_1, v, s_2) := \Delta_L(s_1, v)\left(L(s_2/s_1) - v^2\right),$$

to denote the P&Ls from delta-hedging at time T_1 in the SPX and the log-contract, respectively. Finally, let $\mathcal{M}_c(\mu_1, \mu_V, \mu_2)$ denote the set of all VIX-constrained martingale probability measures:

$$\mathcal{M}_c(\mu_1, \mu_V, \mu_2) := \Big\{\mu \in \mathcal{P}(\mathcal{X}) : S_1 \overset{\mu}{\sim} \mu_1, \quad V \overset{\mu}{\sim} \mu_V, \quad S_2 \overset{\mu}{\sim} \mu_2, \\ \mathbb{E}^\mu[S_2|\, S_1, V] = S_1, \quad \mathbb{E}^\mu[L(S_2/S_1)|S_1, V] = V^2\Big\}.$$

2.2. **Martingale Optimal Transport (MOT).** We first consider a market with two trading days $T_0 = 0$ and T_1, where the financial instruments are the SPX (tradable at T_0 and T_1), the vanilla options on it with maturities T_1 and T_2 (tradable at T_0).

From the theory of MOT [18], the model-free upper bound price for a payoff $\psi(s_1, s_2)$ is the smallest price at time 0 of a super-replication portfolio:

$$P_\psi := \inf_{(u_1, u_2, \Delta_S) \in \mathcal{U}_\psi} \left\{\mathbb{E}^1[u_1(S_1)] + \mathbb{E}^2[u_2(S_2)]\right\} \tag{2.1}$$

where $\mathcal{U}_\psi \subset \mathcal{U}$ is the set of all integrable super-replicating portfolios, i.e., the portfolios satisfying the super-replication constraint:

$$\forall (s_1, s_2) \in \mathbb{R}^2_{>0}, \qquad u_1(s_1) + u_2(s_2) + \Delta_S(s_1)(s_2 - s_1) \geq \psi(s_1, s_2).$$

This is known as the primal problem, which corresponds to the "physical" or portfolio problem, hence the notation P. At time T_1, delta-hedging in the SPX is allowed; as the price of S_2 at time T_1 is precisely S_1, the price of the superreplicating portfolio is $\mathbb{E}^{\mu_1}[u_1(S_1)] + \mathbb{E}^{\mu_2}[u_2(S_2)]$. The delta $\Delta_S(s_1)$ may depend on the SPX value at time T_1.

The problem P_ψ corresponds to a linear program which can be solved using the simplex algorithm method by discretizing (s_1, s_2) on a two-dimensional grid. When the number of discretization points is high and/or when dealing with a multi-dimensional asset a cutting-plane algorithm can be used, see [17] for details. The lower bound is obtained similarly by replacing inf by sup and $F(s_1, s_2) \geq \psi(s_1, s_2)$ by $F(s_1, s_2) \leq \psi(s_1, s_2)$.

The dual problem (or "measure problem") of super-replicating the payoff $\psi(S_1, S_2)$ is one of maximizing the expected payoff $\mathbb{E}^\mu[\psi(S_1, S_2)]$ over all probability measures $\mu \in \mathcal{M}(\mu_1, \mu_2)$:

$$D_\psi := \sup_{\mu \in \mathcal{M}(\mu_1, \mu_2)} \mathbb{E}^\mu[\psi(S_1, S_2)]$$

where $\mathcal{M}(\mu_1, \mu_2)$ denotes the set of all martingale probability measures μ on $\mathbb{R}_{>0} \times \mathbb{R}_{>0}$:

$$\mathcal{M}(\mu_1, \mu_2) = \left\{\mu \in \mathcal{P}(\mathbb{R}^2_{>0}) : S_1 \overset{\mu}{\sim} \mu_1, \quad S_2 \overset{\mu}{\sim} \mu_2, \quad \mathbb{E}^\mu[S_2|\, S_1] = S_1\right\}. \tag{2.2}$$

2.3. **Dispersion-constrained MOT.** When we add VIX information, we allow for static hedging in VIX vanilla options and dynamic hedging at T_1 in the FSLC, and the hedges at T_1 can now depend on the VIX. The theory of VIX-constrained MOT was studied in [11, 14] and the model-free upper bound price for a payoff $\psi(s_1, v, s_2)$ becomes

$$P_\psi^V := \inf_{(u_1,u_V,u_2,\Delta_S,\Delta_L)\in\mathcal{U}_\psi^V} \left\{\mathbb{E}^1\left[u_1\left(S_1\right)\right] + \mathbb{E}^V\left[u_V\left(V\right)\right] + \mathbb{E}^2\left[u_2\left(S_2\right)\right]\right\}, \tag{2.3}$$

where $\mathcal{U}_\psi^V \subset \mathcal{U}^V$ is the set of all integrable super-replicating portfolios, i.e., the portfolios satisfying the super-replication constraint:

$$u_1\left(s_1\right) + u_V\left(v\right) + u_2\left(s_2\right) + \Delta_S(s_1, v)\left(s_2 - s_1\right) + \Delta_L(s_1, v)\left(L(s_2/s_1) - v^2\right) \geq \psi\left(s_1, v, s_2\right)$$

for all $(s_1, v, s_2) \in \mathbb{R}_{>0} \times \mathbb{R}_{\geq 0} \times \mathbb{R}_{>0}$. Observe now that at time T_1, delta-hedging in the SPX and the FSLC is allowed. The respective deltas $\Delta_S(s_1, v)$ and $\Delta_L(s_1, v)$ may depend on the values s_1 and v of the SPX and the VIX at T_1.

Similarly, the linear program P_ψ^V can be solved using the simplex algorithm by discretizing (s_1, v, s_2) on a three-dimensional grid, and the dual problem (or "measure problem") becomes

$$D_\psi^V := \sup_{\mu\in\mathcal{M}_c(\mu_1,\mu_V,\mu_2)} \mathbb{E}^\mu\left[\psi\left(S_1, V, S_2\right)\right]$$

where $\mathcal{M}_c\left(\mu_1, \mu_V, \mu_2\right)$ denotes the set of all VIX-constrained martingale probability measures μ on $\mathbb{R}_{>0} \times \mathbb{R}_{\geq 0} \times \mathbb{R}_{>0}$:

$$\begin{aligned}\mathcal{M}_c\left(\mu_1, \mu_V, \mu_2\right) := \Big\{\mu \in \mathcal{P}\left(\mathbb{R}_{>0} \times \mathbb{R}_{\geq 0} \times \mathbb{R}_{>0}\right) : S_1 \overset{\mu}{\sim} \mu_1, \quad V \overset{\mu}{\sim} \mu_V, \quad S_2 \overset{\mu}{\sim} \mu_2,\\ \mathbb{E}^\mu\left[S_2 \middle| S_1, V\right] = S_1, \mathbb{E}^\mu\left[L\left(S_2/S_1\right) \middle| S_1, V\right] = V^2\Big\}.\end{aligned} \tag{2.4}$$

As shown in [11, 14], the absence of joint SPX/VIX arbitrage is equivalent to $\mathcal{M}_c\left(\mu_1, \mu_V, \mu_2\right) \neq \emptyset$ and to $P_0^V > -\infty$.

3. Fast exact joint calibration in discrete time

3.1. **Entropy minimization.** Solving the joint calibration problem is equivalent to building a probability measure $\mu \in \mathcal{M}_c\left(\mu_1, \mu_V, \mu_2\right)$. In the absence of joint SPX/VIX arbitrage, there may exist an infinite number of models μ within the convex set $\mathcal{M}_c\left(\mu_1, \mu_V, \mu_2\right)$ of jointly calibrated models. To build a specific, meaningful jointly calibrated model, in the spirit of [2], [11, 14] suggests a minimum entropy approach: choose a reasonable (though not jointly calibrated) model $\bar{\mu}$ on $\mathcal{X}$, possibly one derived from a model already in use at the financial institution, and build the probability measure $\mu \in \mathcal{M}_c\left(\mu_1, \mu_V, \mu_2\right)$ that is closest to $\bar{\mu}$ in the entropic sense, i.e., μ has minimum relative entropy with respect to $\bar{\mu}$:

$$D_{\bar{\mu}} := \inf_{\mu\in\mathcal{M}_c(\mu_1,\mu_V,\mu_2)} H\left(\mu\,|\bar{\mu}\right), \qquad H\left(\mu\,|\bar{\mu}\right) := \begin{cases} \mathbb{E}^\mu\left[\ln\left(\frac{\mathrm{d}\mu}{\mathrm{d}\bar{\mu}}\right)\right] = \mathbb{E}^{\bar{\mu}}\left[\frac{\mathrm{d}\mu}{\mathrm{d}\bar{\mu}}\ln\left(\frac{\mathrm{d}\mu}{\mathrm{d}\bar{\mu}}\right)\right] & \text{if } \mu \ll \bar{\mu}, \\ +\infty & \text{otherwise.}\end{cases} \tag{M}$$

From [14], if the minimum entropy problem is finite, there exists a unique minimizer $\mu^* \in \mathcal{M}_c\left(\mu_1, \mu_V, \mu_2\right)$, and it is of the form

$$\frac{\mathrm{d}\mu^*}{\mathrm{d}\bar{\mu}} = e_u(S_1, V, S_2), \qquad e_u(s_1, v, s_2) := e^{u_1(s_1)+u_V(v)+u_2(s_2)+\Delta_S^{(S)}(s_1,v,s_2)+\Delta_L^{(L)}(s_1,v,s_2)}$$

where $u := (u_1, u_V, u_2, \Delta_S, \Delta_L)$ are maximizers (called Schrödinger potentials), if they exist, of the dual problem

$$P_{\bar{\mu}} := \sup_{u\in\mathcal{U}^V} J_{\bar{\mu}}(u), \qquad J_{\bar{\mu}}(u) := \mathbb{E}^1[u_1(S_1)] + \mathbb{E}^V[u_V(V)] + \mathbb{E}^2[u_2(S_2)] - \mathbb{E}^{\bar{\mu}}[e_u(S_1, V, S_2)] + 1. \tag{P}$$

Additionally, in any case, $D_{\bar{\mu}} = P_{\bar{\mu}}$. Note that (P) is an *unconstrained* concave maximization problem. Both problems are dual to each other; following a terminology proposed by Dupire, (M) is a measure problem while (P) is a portfolio problem.

As in practice, only a finite number of SPX and VIX vanilla options are available for trading, we consider vanilla payoffs u_1, u_V, and u_2 that are linear combinations of finitely many call options, along with one position in the bond, one position in S_1, and one position in the VIX futures. Therefore we consider a

market data $\mathcal{K}$ composed of call options on S_1, V, and S_2 denoted $\left(C_K^1\right)_{K\in\mathcal{K}_1}$, $\left(C_K^V\right)_{K\in\mathcal{K}_V}$, and $\left(C_K^2\right)_{K\in\mathcal{K}_2}$ with respective strikes $\mathcal{K}_1, \mathcal{K}_V$, and $\mathcal{K}_2$, and we build a model of the form

$$\frac{\mathrm{d}\mu_{\mathcal{K},\theta}}{\mathrm{d}\bar{\mu}} = e_\theta(S_1, V, S_2)$$
$$e_\theta(s_1, v, s_2) := e^{c+\Delta_S^0 s_1+\Delta_V^0 v+\sum_{\mathcal{K}_1} a_K^1(s_1-K)_+ +\sum_{\mathcal{K}_V} a_K^V(v-K)_+ +\sum_{\mathcal{K}_2} a_K^2(s_2-K)_+ +\Delta_S^{(S)}(s_1,v,s_2)+\Delta_L^{(L)}(s_1,v,s_2)},$$

where for $i \in \{1, V, 2\}$, $\sum_{\mathcal{K}_i}$ is a shorthand notation for $\sum_{K\in\mathcal{K}_i}$ and θ is an element of the set Θ of all

$$\theta := \left(c, \Delta_S^0, \Delta_V^0, a^1, a^V, a^2, \Delta_S, \Delta_L\right)$$

such that $c, \Delta_S^0, \Delta_V^0 \in \mathbb{R}$, $a^1 \in \mathbb{R}^{\mathcal{K}_1}$, $a^V \in \mathbb{R}^{\mathcal{K}_V}$, $a^2 \in \mathbb{R}^{\mathcal{K}_2}$, and $\Delta_S, \Delta_L : \mathbb{R}_{>0} \times \mathbb{R}_{\geq 0} \to \mathbb{R}$ are bounded measurable functions of (s_1, v). The measure $\mu_{\mathcal{K},\theta}$ is then a consistent, arbitrage-free model that jointly calibrates to the market prices of SPX/VIX futures and options if and only if θ solves the so-called $\mathcal{K}$-Schrödinger system

$$\text{(3.1)}\quad \begin{cases} \mathbb{E}^{\bar{\mu}}\left[\frac{\mathrm{d}\mu_{\mathcal{K},\theta}}{\mathrm{d}\bar{\mu}}\right] = 1, \\ \mathbb{E}^{\bar{\mu}}\left[S_1\frac{\mathrm{d}\mu_{\mathcal{K},\theta}}{\mathrm{d}\bar{\mu}}\right] = S_0, \\ \mathbb{E}^{\bar{\mu}}\left[V\frac{\mathrm{d}\mu_{\mathcal{K},\theta}}{\mathrm{d}\bar{\mu}}\right] = F_V, \\ \mathbb{E}^{\bar{\mu}}\left[(S_1-K)_+\frac{\mathrm{d}\mu_{\mathcal{K},\theta}}{\mathrm{d}\bar{\mu}}\right] = C_K^1, & \forall K \in \mathcal{K}_1, \\ \mathbb{E}^{\bar{\mu}}\left[(V-K)_+\frac{\mathrm{d}\mu_{\mathcal{K},\theta}}{\mathrm{d}\bar{\mu}}\right] = C_K^V, & \forall K \in \mathcal{K}_V, \\ \mathbb{E}^{\bar{\mu}}\left[(S_2-K)_+\frac{\mathrm{d}\mu_{\mathcal{K},\theta}}{\mathrm{d}\bar{\mu}}\right] = C_K^2, & \forall K \in \mathcal{K}_2, \\ \mathbb{E}^{\bar{\mu}}\left[(S_2-S_1)\frac{\mathrm{d}\mu_{\mathcal{K},\theta}}{\mathrm{d}\bar{\mu}}\middle| S_1 = s_1, V = v\right] = 0, & \forall s_1 > 0, v \geq 0, \\ \mathbb{E}^{\bar{\mu}}\left[\left(L\left(\frac{S_2}{S_1}\right) - V^2\right)\frac{\mathrm{d}\mu_{\mathcal{K},\theta}}{\mathrm{d}\bar{\mu}}\middle| S_1 = s_1, V = v\right] = 0, & \forall s_1 > 0, v \geq 0. \end{cases}$$

The first equation states that $\mu_{\mathcal{K},\theta}$ is a probability measure while the others that it belongs to the set $\mathcal{M}_{c,\mathcal{K}}(\mu_1, \mu_V, \mu_2)$ of probability measures μ satisfying

$$\mathbb{E}^\mu[S_1] = S_0, \quad \mathbb{E}^\mu[V] = F_V, \quad \forall K \in \mathcal{K}_1,\ \mathbb{E}^\mu[(S_1-K)_+] = C_K^1, \quad \forall K \in \mathcal{K}_V,\ \mathbb{E}^\mu[(V-K)_+] = C_K^V,$$
$$\forall K \in \mathcal{K}_2,\ \mathbb{E}^\mu[(S_2-K)_+] = C_K^2, \quad \mathbb{E}^\mu[S_2|S_1, V] = S_1, \quad E^\mu[L(S_2/S_1)|S_1, V] = V^2.$$

Remark 2 (On the choice of the reference measure $\bar{\mu}$). The prior measure $\bar{\mu}$ is up to the modeler's choice. Examples include:

(1) lognormal prior: $\bar{\mu}(\mathrm{d}s_1, \mathrm{d}v, \mathrm{d}s_2) = \nu(\mathrm{d}s_1, \mathrm{d}v)T(s_1, v, \mathrm{d}s_2)$ where $\nu = \mu_1 \otimes \mu_V$ and $T(s_1, v, \mathrm{d}s_2)$ is the distribution of $s_1 \exp(v\sqrt{\tau}G - \frac{1}{2}v^2\tau)$ where $G \sim \mathcal{N}(0, 1)$ That is, S_1 and V are independent with their respective marginals μ_1 and μ_V, and conditional on S_1 and V, S_2 is lognormal with mean S_1 and annualized lognormal volatility V with a probability density function of the form[1]

$$T(s_1, v, s_2) = \frac{1}{v\sqrt{\tau}\sqrt{2\pi}s_2} e^{-\frac{1}{2v^2\tau}\left(\ln\left(\frac{s_2}{s_1}\right)+\frac{v^2\tau}{2}\right)^2} \mathbf{1}_{s_2>0}.$$

(2) independent prior (product measure): $\bar{\mu} = \mu_1 \otimes \mu_V \otimes \mu_2$. This is the standard choice made in the entropic optimal transport literature. However, financially speaking, this reference measure is not natural as it does not satisfy both the martingality and consistency conditions.

[1]Assuming that S_2 is lognormal conditioned on S_1 and V is financially natural but may not be the best choice in practice. Indeed, in this case, $\mathbb{E}^{\bar{\mu}}[e^{\delta S_2}|S_1, V] = +\infty$ for $\delta > 0$, so the integrals are not well defined for many reasonable values of the parameters $u_1, u_V, u_2, \Delta_S, \Delta_L$. In practice, we avoid those integrability issues by working with a finite support approximation of $\bar{\mu}$ stemming from the Gaussian quadrature approximation of the integrals (see Remark 3).

Remark 3 (Quadrature). To evaluate the expectations arising in (3.1), we use a Gauss–Legendre quadrature when integrating with respect to s_1 (resp., v) with grid $\mathcal{G}_1 := \{s_1^{(1)} \leq \cdots \leq s_1^{(n_1)}\}$ (resp., $\mathcal{G}_V := \{v^{(1)} \leq \cdots \leq v^{(n_V)}\}$). For s_2, in the case of the lognormal prior, we use a Gauss–Hermite quadrature with knots $\{z^{(1)}, \dots, z^{(n_2)}\}$. For example, in the log-normal case, we use the approximation

$$\begin{aligned}\mathbb{E}^{\bar{\mu}}[e_\theta(S_1, V, S_2)] &= \int_{\mathbb{R}^3} e_\theta(s_1, v, s_2)\bar{\mu}(\mathrm{d}s_1, \mathrm{d}v, \mathrm{d}s_2) \\ &\approx \sum_{i=1}^{n_1}\sum_{j=1}^{n_v}\sum_{k=1}^{n_2} \omega_{\mathrm{Le}}^{(i)}\omega_{\mathrm{Le}}^{(j)}\omega_{\mathrm{He}}^{(k)} e_\theta(s_1^{(i)}, v^{(j)}, s_1 e^{v^{(j)}\sqrt{\tau}z^{(k)} - \frac{1}{2}(v^{(j)})^2\tau})\mu_1(s_1^{(i)})\mu_V(v^{(j)}) \end{aligned} \tag{3.2}$$

where the ω_{He} (resp., ω_{Le}) denote the Hermite (resp., Legendre) quadrature weights.

3.2. Solving the Schrödinger system (3.1).

3.2.1. *The Sinkhorn algorithm.* The classical method for solving Schrödinger systems is the Sinkhorn algorithm, an iterative method that sequentially solves the individual equations in (3.1) to converge to the optimizer θ^* of the whole system. This algorithm has recently gained popularity in machine learning where it is used to quickly compute Wasserstein distances, and more generally solve optimal transport problems, via a small entropic penalty. It has also been applied to martingale optimal transport problems [6] and in particular in quantitative finance to quickly build arbitrage-free smiles [7], see also [18] and references therein for a large overview of MOT. In [14] the Sinkhorn algorithm was extended to accommodate the martingality and consistency constraints in (3.1), and shown to converge toward a jointly calibrated model. However, the convergence was somewhat slow (see Section 3.2.4). In the following sections, we present two faster alternatives for numerically solving (3.1); both rely on solving the portfolio problem (P).

3.2.2. *The Newton–Sinkhorn algorithm.* Observe that if we define the concave function

$$J_{\bar{\mu},\mathcal{K}}(\theta) := c + \Delta_S^0 S_0 + \Delta_V^0 F_V + \sum_{i\in\{1,V,2\}}\sum_{K\in\mathcal{K}_i} a_K^i C_K^i - \mathbb{E}^{\bar{\mu}}[e_\theta(S_1, V, S_2)] + 1,$$

solving the $\mathcal{K}$-Schrödinger system is equivalent to canceling the gradient of $J_{\bar{\mu},\mathcal{K}}$. Hence, to solve the system and build μ^*, one can directly solve the portfolio problem

$$P_{\bar{\mu},\mathcal{K}} := \sup_{\theta\in\Theta} J_{\bar{\mu},\mathcal{K}}(\theta), \tag{3.3}$$

which is the finitely-many-payoff version of (P). To this end, we suggest the following *Newton–Sinkhorn algorithm*. Each iteration involves a Newton step followed by a Sinkhorn step.

Newton step. Starting from an initial guess $\theta^{(0)}$, we first solve for every iteration $n \in \mathbb{N}$ the portfolio problem (3.3)

$$\theta^{-\Delta,(n+1)} = \operatorname*{arg\,max}_{\theta^{-\Delta}\in\Theta^{-\Delta}} J_{\bar{\mu},\mathcal{K}}(\theta^{-\Delta}, \Delta_S^{(n)}(s_1, v), \Delta_L^{(n)}(s_1, v)), \qquad \forall\,(s_1, v) \in \mathcal{G}_1 \times \mathcal{G}_V, \tag{3.4}$$

where $\theta^{-\Delta} := \left(c, \Delta_S^0, \Delta_V^0, a^1, a^V, a^2\right)$. Since the Hessian of $J_{\bar{\mu},\mathcal{K}}(\theta^{-\Delta}, \Delta_S^{(n)}(s_1, v), \Delta_L^{(n)}(s_1, v))$ is known in closed form, this Newton step is extremely fast. We solve (3.4) using the function `scipy.optimize.minimize (method="trust-exact")` from the `scipy` library.

Sinkhorn step. Then, for all $(s_1, v) \in \mathcal{G}_1 \times \mathcal{G}_V$, we jointly solve for $\Delta_S^{(n+1)}(s_1, v), \Delta_L^{(n+1)}(s_1, v)$ the two-dimensional nonlinear system

$$\begin{cases} f_{s_1,v}\left(\Delta_S^{(n+1)}(s_1, v), \Delta_L^{(n+1)}(s_1, v), a^{2,(n+1)}\right) &= 0, \quad (3.5)\\ g_{s_1,v}\left(\Delta_S^{(n+1)}(s_1, v), \Delta_L^{(n+1)}(s_1, v), a^{2,(n+1)}\right) &= 0, \quad (3.6)\end{cases}$$

where $a^{2,(n+1)}$ is the optimal vector a^2 from the previous step (3.4) and for all $(x, y) \in \mathbb{R}^2$,

$$\begin{aligned} f_{s_1,v}\left(x, y, a^2\right) &:= \int_{\mathbb{R}} (s_2 - s_1)\, e^{\sum_{K\in\mathcal{K}_2} a_K^2(s_2-K)_+ + x(s_2-s_1) + y\left(L\left(\frac{s_2}{s_1}\right) - v^2\right)} \bar{\mu}\left(s_1, v, \mathrm{d}s_2\right), \\ g_{s_1,v}\left(x, y, a^2\right) &:= \int_{\mathbb{R}} \left(L\left(\frac{s_2}{s_1}\right) - v^2\right) e^{\sum_{K\in\mathcal{K}_2} a_K^2(s_2-K)_+ + x(s_2-s_1) + y\left(L\left(\frac{s_2}{s_1}\right) - v^2\right)} \bar{\mu}\left(s_1, v, \mathrm{d}s_2\right). \end{aligned}$$

This second step is a Sinkhorn step, where the last two equations of (3.1) are jointly solved, hence the name Newton–Sinkhorn. We use the Levenberg–Marquardt algorithm via `scipy.optimize.root(method="lm")` to solve (3.5)-(3.6).

Remark 4 (Full Newton algorithm and parametrization of the deltas). As an alternative approach, we could use a full Newton algorithm on the vector θ. However, the number of parameters for the deltas $(\Delta_S(s_1,v),\Delta_L(s_1,v))_{(s_1,v)\in\mathcal{G}_1\times\mathcal{G}_V}$ is typically too large for the Newton algorithm to be fast: the inversion of the Hessian matrix required in Newton-type algorithms is too costly. For example, taking $n_1 = n_v = 45$ knots for each Legendre quadrature leads to $2\times 45\times 45 = 4050$ values for $\Delta_S(\cdot,\cdot)$ and $\Delta_L(\cdot,\cdot)$ while only a few dozens parameters for $\theta^{-\Delta}$ is required for the algorithm to converge (see section 3.2.4 for further details). This is the reason why we mix a Newton solver for $\theta^{-\Delta}$ with a Sinkhorn solver for $\Delta_S(\cdot,\cdot)$ and $\Delta_L(\cdot,\cdot)$.

The previous algorithms are nonparametric in the sense that we find the values of Δ_S and Δ_L on the two given quadrature grids $\mathcal{G}_1$ and $\mathcal{G}_V$. To reduce the number of parameters, it is natural to try parametrizing the functions $(s_1,v)\mapsto\Delta_S(s_1,v)$ and $(s_1,v)\mapsto\Delta_L(s_1,v)$ and see if we can make the algorithm converge even faster. We tried multivariate polynomials, i.e., for two-dimensional bases $\Psi^S(\cdot,\cdot),\Psi^L(\cdot,\cdot)$ and degrees $d_S,d_L\in\mathbb{N}_{>0}$, we set

$$\Delta_S(s_1,v)=\sum_{|\ell|\le d_S} b_\ell^S\Psi_\ell^S(s_1,v),\qquad \Delta_L(s_1,v)=\sum_{|\ell|\le d_L} b_\ell^L\Psi_\ell^L(s_1,v).$$

Here, the martingality and consistency constraints of (3.1) (last two equations) rewrite as

$$\begin{aligned}\mathbb{E}^{\bar\mu}\left[\Psi_\ell^S(S_1,V)(S_2-S_1)\frac{\mathrm{d}\mu_{\mathcal{K},\theta}}{\mathrm{d}\bar\mu}\right]&=0,\qquad \forall\,|\ell|\le d_S,\\ \mathbb{E}^{\bar\mu}\left[\Psi_\ell^L(S_1,V)\left(L\left(\frac{S_2}{S_1}\right)-V^2\right)\frac{\mathrm{d}\mu_{\mathcal{K},\theta}}{\mathrm{d}\bar\mu}\right]&=0,\qquad \forall\,|\ell|\le d_L.\end{aligned}$$

They are weaker than the original constraints as we are only projecting the random variables onto the space generated by our basis functions. In practice, we tried many different choices of basis (Legendre, Hermite, monomials, Fourier) but in each case, the martingale and consistency conditions were not satisfied with good enough accuracy, even with large degrees such as $d_S=d_L=20$. We also tried to parametrize Δ_S,Δ_L with two feed-forward neural networks. We tried many different architectures (different numbers of hidden layers, neurons and activation functions). Again, the algorithm converged but the martingale and consistency conditions were never satisfied with sufficient accuracy.

3.2.3. *The implied Newton algorithm.* Inspired by [6], we observe that

$$\theta^*=\operatorname*{arg\,max}_{\theta\in\Theta}J_{\bar\mu,\mathcal{K}}(\theta)=\operatorname*{arg\,max}_{\theta^{-\Delta}\in\Theta^{-\Delta}}\tilde J_{\bar\mu,\mathcal{K}}\left(\theta^{-\Delta},\Delta_S^*\left(\cdot,\cdot,a^2\right),\Delta_L^*\left(\cdot,\cdot,a^2\right)\right)$$

where $\left(\Delta_S^*\left(\cdot,\cdot,a^2\right),\Delta_L^*\left(\cdot,\cdot,a^2\right)\right)$ solves the two-dimensional nonlinear system

$$\begin{cases} f_{s_1,v}(\Delta_S^*(\cdot,\cdot),\Delta_L^*(\cdot,\cdot),a^2) &= 0,\\ g_{s_1,v}(\Delta_S^*(\cdot,\cdot),\Delta_L^*(\cdot,\cdot),a^2) &= 0.\end{cases}$$

That is, for each $\theta^{-\Delta}$, we first optimize over $\Delta_S(\cdot,\cdot)$ and $\Delta_L(\cdot,\cdot)$, and then we optimize over $\theta^{-\Delta}$. Note that the inner optimization depends on $\theta^{-\Delta}$ only through a^2.

Like for $J_{\bar\mu,\mathcal{K}}$, the gradient and Hessian of $\tilde J_{\bar\mu,\mathcal{K}}$ are known in closed form. In fact, $\tilde J_{\bar\mu,\mathcal{K}}$ has the same gradient and Hessian as $J_{\bar\mu,\mathcal{K}}$, except for the terms involving differentiation with respect to a^2. Namely, for every $m,n\in\mathcal{K}_2$, we have

$$\begin{aligned}\partial_{a_m^2}\tilde J_{\bar\mu,\mathcal{K}}(\lambda)=\int_{\mathbb{R}^3}\Big[&(s_2-m)_+ +\partial_{a_m^2}\Delta_S^*(s_1,v,a^2)(s_2-s_1)\\ &+\partial_{a_m^2}\Delta_L^*(s_1,v,a^2)(L(s_2/s_1)-v^2)\Big]e^*_{\theta^{-\Delta}}(s_1,v,s_2)\bar\mu(\mathrm{d}s_1,\mathrm{d}v,\mathrm{d}s_2)-C_m^2\end{aligned}$$

and

$$\begin{aligned}\partial_{a_m^2,a_n^2}\tilde{J}_{\bar{\mu},\mathcal{K}}(\lambda) = \int_{\mathbb{R}^3}\Big\{&\partial_{a_m^2,a_n^2}\Delta_S^*(s_2-s_1)+\partial_{a_m^2,a_n^2}\Delta_L^*(L(s_2/s_1)-v^2)\\&+\Big[(s_2-m)_+ +\partial_{a_m^2}\Delta_S^*(s_2-s_1)+\partial_{a_m^2}\Delta_L^*(L(s_2/s_1)-v^2)\Big]\times\Big[(s_2-n)_+ +\partial_{a_n^2}\Delta_S^*(s_2-s_1)\\&+\partial_{a_n^2}\Delta_L^*(L(s_2/s_1)-v^2)\Big]\Big\}e^*_{\theta^{-\Delta}}(s_1,v,s_2)\bar{\mu}(\mathrm{d}s_1,\mathrm{d}v,\mathrm{d}s_2),\end{aligned}$$

where $e^*_{\theta^{-\Delta}}(s_1,v,s_2) := e_{\left(\theta^{-\Delta},\Delta_S^*(\cdot,\cdot,a^2),\Delta_L^*(\cdot,\cdot,a^2)\right)}(s_1,v,s_2)$.

The derivatives $\partial_{a_m^2}\Delta_S^*$ and $\partial_{a_m^2}\Delta_L^*$ are computed as follows. Taking the derivative $\partial_{a_m^2}$ in

$$f_{s_1,v}\left(\Delta_S^*\left(s_1,v,a^2\right),\Delta_L^*\left(s_1,v,a^2\right),a^2\right)=0,$$

the following two-dimensional linear system holds

$$\left\{\begin{array}{rcl}\partial_{a_m^2}\Delta_S^*\partial_x f(\Delta_S^*,\Delta_L^*,a^2)+\partial_{a_m^2}\Delta_L^*\partial_y f(\Delta_S^*,\Delta_L^*,a^2) &=& -\partial_{a_m^2}f(\Delta_S^*,\Delta_L^*,a^2),\\ \partial_{a_m^2}\Delta_S^*\partial_x g(\Delta_S^*,\Delta_L^*,a^2)+\partial_{a_m^2}\Delta_L^*\partial_y g(\Delta_S^*,\Delta_L^*,a^2) &=& -\partial_{a_m^2}g(\Delta_S^*,\Delta_L^*,a^2),\end{array}\right. \tag{3.7}$$

for every $m\in\mathcal{K}_2$. Introducing the short-hand notations

$$\partial_\ell h^* := \partial_\ell h(\Delta_S^*,\Delta_L^*,a^2),\qquad \ell\in\{x,y,a_m^2,xx,xy,\dots\},\quad h\in\{f,g\},$$

we easily get

$$\begin{pmatrix}\partial_{a_m^2}\Delta_S^*\\ \partial_{a_m^2}\Delta_L^*\end{pmatrix}=\frac{1}{\partial_x f^*\partial_y g^*-\partial_y f^*\partial_x g^*}\begin{pmatrix}-\partial_y g^*\partial_{a_m^2}f^*+\partial_y f^*\partial_{a_m^2}g^*\\ \partial_x g^*\partial_{a_m^2}f^*-\partial_x f^*\partial_{a_m^2}g^*\end{pmatrix} \tag{3.8}$$

where

$$\partial_x f^*=\int(s_2-s_1)^2 e_*\bar{\mu}(s_1,v,\mathrm{d}s_2),\quad \partial_y f^*=\int(s_2-s_1)\left(L\left(\frac{s_2}{s_1}\right)-v^2\right)e_*\bar{\mu}(s_1,v,\mathrm{d}s_2),$$

$$\partial_x g^*=\partial_y f^*,\quad \partial_y g^*=\int\left(L\left(\frac{s_2}{s_1}\right)-v^2\right)^2 e_*\bar{\mu}(s_1,v,\mathrm{d}s_2),$$

$$\partial_{a_m^2}f^*=\int(s_2-s_1)(s_2-m)_+e_*\bar{\mu}(s_1,v,\mathrm{d}s_2),\quad \partial_{a_m^2}g^*=\int\left(L\left(\frac{s_2}{s_1}\right)-v^2\right)(s_2-m)_+e_*\bar{\mu}(s_1,v,\mathrm{d}s_2),$$

with $e_* := e^{\sum_{K\in\mathcal{K}_2}a_K^2(s_2-K)_+ +\Delta_S^*(s_2-s_1)+\Delta_L^*\left(L\left(\frac{s_2}{s_1}\right)-v^2\right)}$. As for the second order derivatives $\left(\partial_{a_m^2,a_n^2}\Delta_S^*,\partial_{a_m^2,a_n^2}\Delta_L^*\right)$, taking the derivative of (3.7) with respect to a_n^2, we again obtain a two-dimensional linear system

$$\left\{\begin{array}{rcl}\partial_{a_m^2,a_n^2}\Delta_S^*\partial_x f^*+\partial_{a_m^2,a_n^2}\Delta_L^*\partial_y f^* &=& -Z_f^*,\\ \partial_{a_m^2,a_n^2}\Delta_S^*\partial_x g^*+\partial_{a_m^2,a_n^2}\Delta_L^*\partial_y g^* &=& -Z_g^*,\end{array}\right. \tag{3.9}$$

where for $h\in\{f,g\}$,

$$\begin{aligned}Z_h^* := {}&\partial_{a_m^2}\Delta_S^*\left(\partial_{a_n^2}\Delta_S^*\,\partial_{xx}h^*+\partial_{a_n^2}\Delta_L^*\,\partial_{xy}h^*+\partial_{x,a_n^2}h^*\right)+\partial_{a_m^2}\Delta_L^*\left(\partial_{a_n^2}\Delta_S^*\,\partial_{yx}h^*+\partial_{a_n^2}\Delta_L^*\,\partial_{yy}h^*+\partial_{ya_n^2}h^*\right)\\&+\partial_{a_n^2}\Delta_S^*\,\partial_{a_m^2x}h^*+\partial_{a_n^2}\Delta_L^*\,\partial_{a_m^2y}h^*+\partial_{a_m^2,a_n^2}h^*.\end{aligned}$$

The solution $\left(\partial_{a_m^2,a_n^2}\Delta_S^*,\partial_{a_m^2,a_n^2}\Delta_L^*\right)$ is the same as (3.8) replacing $\partial_{a_m^2}f^*$ (resp., $\partial_{a_m^2}g^*$) with Z_f^* (resp., Z_g^*).

3.2.4. *Comparison of the different algorithms.* All the numerical tests were performed on a MacBook Pro laptop with a 2.6 GHz 6-Core Intel Core i7 processor and 32 GB memory using the programming language Python 3.9.6.

To compare the various algorithms, we plot the logarithm (in base 10) of the calibration error for the SPX smiles at T_1,T_2 and the VIX smile at T_1 as a function of the computational time as of August 1, 2018 ($T_1=21$ days), April 20, 2020 ($T_1=30$ days), and April 25, 2022, ($T_1=23$ days), see Figure 3.1. The calibration error is computed as

$$\sum_{\ell\in\{1,V,2\}}\left\{\frac{1}{|\mathcal{K}_\ell|}\sum_{K\in\mathcal{K}_\ell}\frac{\left|\sigma_{\mathrm{BS}}^{\ell,\mu^*}(K)-\sigma_{\mathrm{BS}}^{\ell,\mathrm{mkt}}(K)\right|}{\sigma_{\mathrm{BS}}^{\ell,\mathrm{mkt}}(K)}\right\}+\sum_{A\in\{S_1,V,S_2\}}\frac{\left|\mathbb{E}^{\bar{\mu}}[A\frac{\mathrm{d}\mu^*}{\mathrm{d}\bar{\mu}}]-F_A\right|}{F_A}+\left|\mathbb{E}^{\bar{\mu}}\left[\frac{\mathrm{d}\mu^*}{\mathrm{d}\bar{\mu}}\right]-1\right|, \tag{3.10}$$

where, for any $\ell\in\{1,V,2\}$, $\sigma_{\mathrm{BS}}^{\ell,\mu^*}(K)$ corresponds to the Black–Scholes implied volatility computed with one of the different algorithms (Sinkhorn, Newton–Sinkhorn, or implied Newton) and $\sigma_{\mathrm{BS}}^{\ell,\mathrm{mkt}}(K)$ is the market

implied volatility. The calibration error (3.10) is thus defined as the sum of the absolute relative errors of the three futures, the mean of the three smiles, and the total mass of μ^* (recall the constraint that μ^* must be a probability measure).

We choose the lognormal prior for the reference measure $\bar{\mu}$ (see Remark 2) and we take $n_1 = n_v = 45$ knots for the integration with respect to s_1 and v, and $n_2 = 25$ knots for the integration with respect to s_2. We set $s_1^{(1)} = F_{\mu_1}^{-1}(q)$, $v^{(1)} = F_{\mu_V}^{-1}(q)$ and $s_1^{(n_1)} = F_{\mu_1}^{-1}(1-q)$, $v^{(n_V)} = F_{\mu_V}^{-1}(1-q)$ with $q = 10^{-3}$ for the lowest and highest values of the quadrature grids of s_1 and v.

We compare the calibration speed of the Sinkhorn (S), Newton–Sinkhorn (NS), and implied Newton (IN) algorithms. For IN, we initialize the algorithm with 10 iterations of a (pure) S algorithm (warm-start procedure). A warm-start initialization of NS yielded no improvement. The fact that a Sinkhorn warm-start speeds up the convergence of IN was already reported in [6, section 7.1] where the author states: "*We notice that the Bregman projection* [i.e., Sinkhorn] *algorithm is more effective at the beginning, to find the optimal region, and then it converges slower. In contrast, the Newton algorithm is slow at the beginning when it is searching the neighborhood of the optimum, but when it finds this neighborhood, the convergence gets very fast. Then it makes sense to apply a hybrid algorithm that starts with Bregman projections, and concludes with the Newton method.*"

In Figure 3.1, we observe that IN is the fastest, and S the slowest. To emphasize the performance of IN, we report in Table 1 the ratio of the calibration error obtained with either S or NS over the calibration error obtained with IN after 30 and 60 seconds. For example, after 60 seconds the implied Newton is 17 (resp., 12) times more accurate than the Newton–Sinkhorn (resp., Sinkhorn) as of August 01, 2018.

In Figure 3.2, we plot the individual calibration errors on the futures, implied volatilities, and total masses of measure defined (3.10). Interestingly, it seems that the calibration error is mainly due to errors on the smile for S_2 for all maturities. A penalty term on the errors on the smile for S_2 could be added in the objective function $J_{\bar{\mu},\mathcal{K}}$; we leave this for further research.

In Figure 5.1, we plot the futures and smiles for S_1, V, S_2 obtained with the implied Newton algorithm after 60 seconds and the market smiles for the three calibration dates. The fits are perfect. We also show in Figure 5.2 the functions Δ_S, Δ_L and martingality and consistency conditions with respect to (s_1, v) as of August 01, 2018. The constraints are perfectly satisfied. Similar plots are obtained for the other two calibration dates.

Maturity	Accuracy IN/NS	Accuracy IN/S
August 01, 2018 (30 seconds)	2	1
August 01, 2018 (60 seconds)	17	12
April 20, 2020 (30 seconds)	4	36
April 20, 2020 (60 seconds)	2	38
April 25, 2022 (30 seconds)	7	5
April 25, 2022 (60 seconds)	2	5

TABLE 1. Ratio of the Newton–Sinkhorn (NS) and Sinkhorn (S) calibration errors over the implied Newton (IN) calibration error after 30 and 60 seconds. The calibration error is defined in (3.10).

We also tested the three methods when the reference measure is the product measure (independent prior). The calibration errors as of August, 01, 2018 (with a lognormal prior and independent prior) are reported in Figure 5.3 and the functions Δ_S, Δ_L along with the martingality and consistency checks are reported (with an independent prior) in Figure 5.4. Observe that the functions Δ_S and Δ_L again have similar shapes, but those shapes are significantly different from the ones obtained with a lognormal prior, see Figure 5.2 for a comparison. We observe that the three methods seem to be less stable with the independent prior and typically require a higher number of nodes; we chose $n_1 = n_V = n_2 = 45$. With the independent prior, IN needs more steps to converge.

4. CONTINUOUS-TIME EXTENSION

We now extend the discrete-time model μ^* (and the probability space) to build a continuous-time model for $(S_t)_{t\in[0,T_2]}$. The model will also include the VIX at T_1, V. That is, we build a probability $\mathbb{P}$ on $C([0,T_2],\mathbb{R})\times$

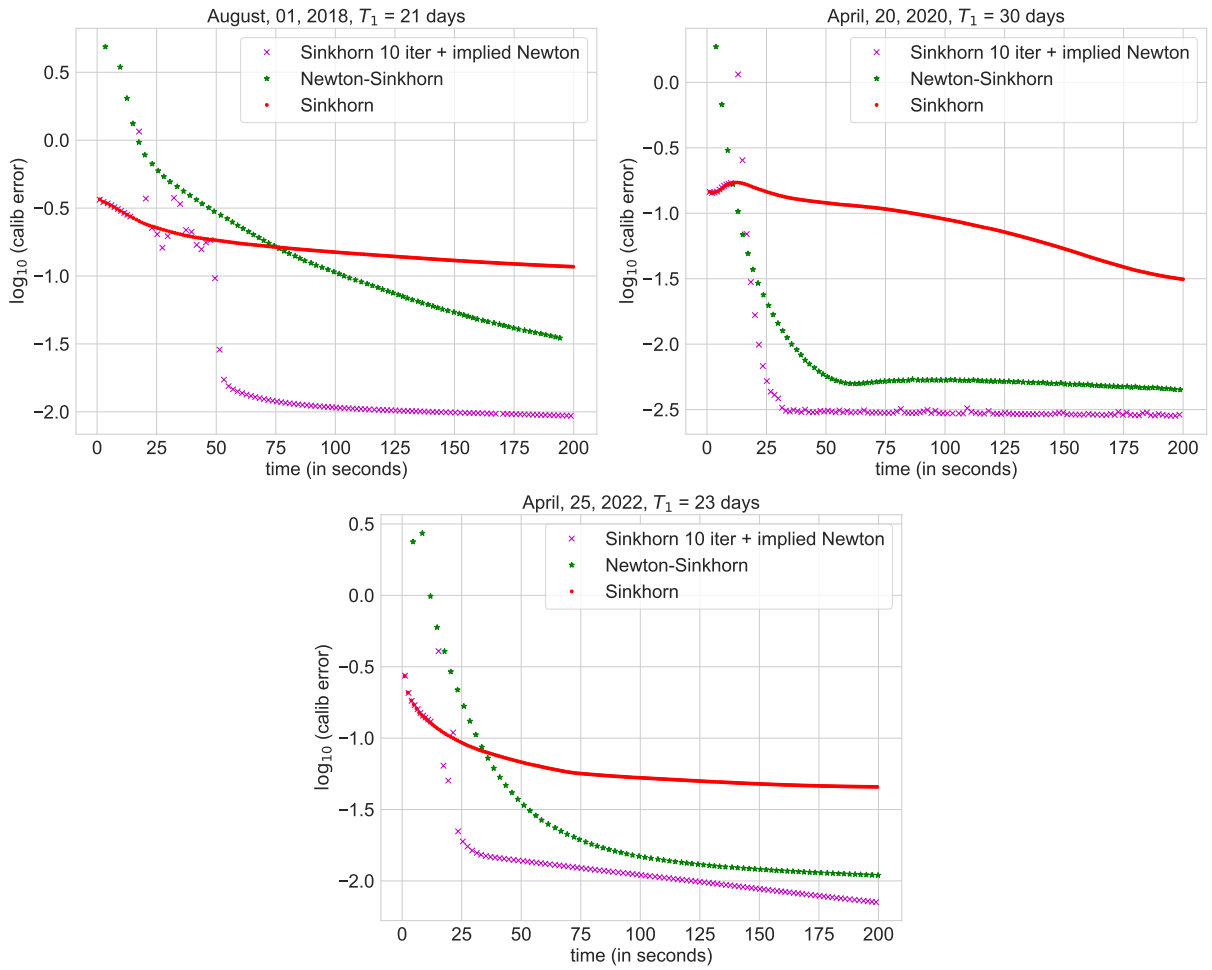

FIGURE 3.1. Comparison of the performance between the Newton–Sinkhorn, the implied Newton (with a 10-iteration pure Sinkhorn warm start), and the Sinkhorn algorithms. We plot the decadic logarithm of the calibration error as a function of the computational time as of August 1, 2018, with $T_1 = 21$ days, April 20, 2020, with $T_1 = 30$ days, and April 25, 2022, with $T_1 = 23$ days.

$\mathbb{R}_+$ representing the distribution of $((S_t)_{t\in[0,T_2]}, V)$. V plays the role of a discrete-time stochastic volatility, representing the stochastic volatility anticipated at T_1 for the $[T_1, T_2]$ period, but our model involves no continuous-time stochastic volatility process. Similar to [4], our model is computationally efficient as it only requires simulating one Brownian motion (and V).

The key advantage of our construction, compared with the Bass local volatility of [4], is that it directly starts from a *joint* discrete distribution of (S_1, S_2), making the continuous-time interpolation a purely forward construction with no need for solving a fixed-point problem. Moreover, it includes a stochastic volatility component V for the calibration to VIX futures and VIX smiles in addition to SPX smiles. As a result, in contrast with [4] and in line with the path-dependency observed in financial markets [15], our model is path-dependent: the SPX dynamics after T_1 depends on both S_1 and V.

4.1. **Step 1: Simulation of $(S_t)_{t\in[0,T_1]}$.** We want $(S_t)_{t\in[0,T_1]}$ to be a $\mathbb{P}$-martingale and S_1 to have distribution μ_1 under $\mathbb{P}$. To achieve this, one possible choice is to use a Markov functional model $S_t = u(t, W_t)$, $t \in [0, T_1]$, where W is a $\mathbb{P}$-Brownian motion and u satisfies the heat equation

$$\partial_t u + \frac{1}{2}\partial_x^2 u = 0.$$

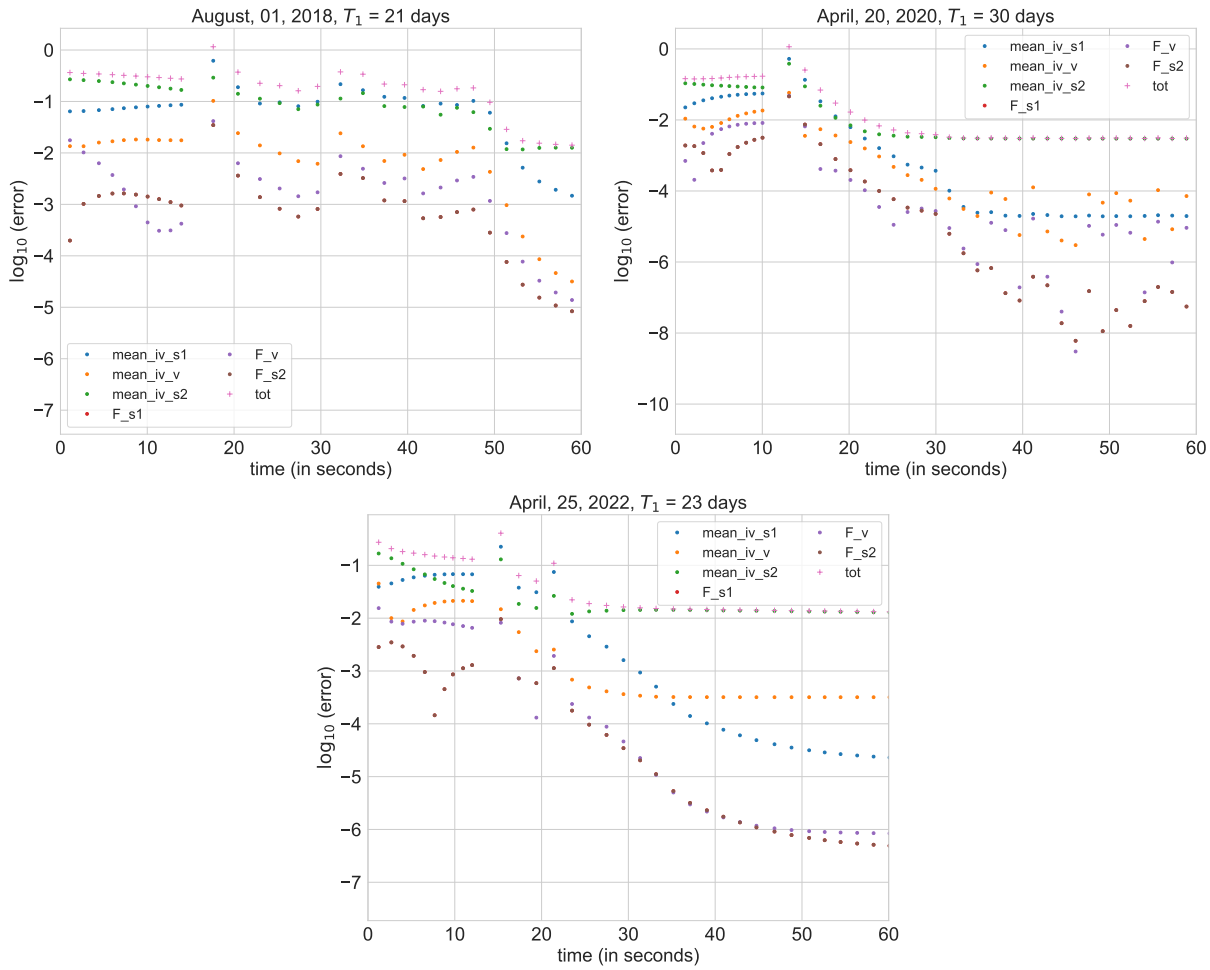

FIGURE 3.2. Distribution of the different calibration errors (3.10) (implied volatilities, futures, probability measure). We plot the logarithm of the errors as a function of the computational time as of August 1, 2018, with $T_1 = 23$ days, April 20, 2020, with $T_1 = 30$ days, and April 25, 2022, with $T_1 = 23$ days.

Further, define the filtration

$$\mathcal{F}_t := \begin{cases} \sigma\left((W_s)_{s\in[0,t]}\right) & \text{if } t \in [0, T_1), \\ \sigma\left((W_s)_{s\in[0,t]}, V\right) & \text{if } t \in [T_1, T_2]. \end{cases}$$

In such a case, $(S_t)_{t\in[0,T_1]}$ is an $((\mathcal{F}_t), \mathbb{P})$-martingale and the terminal condition $u(T_1, \cdot) = g$ is determined via quantiles so that $u(T_1, W_{T_1})$ has distribution μ_1, i.e., for all $x \in \mathbb{R}$, we set

$$g(x) := F_{\mu_1}^{-1}\left(F_{\mathcal{N}(0,1)}\left(\frac{x}{\sqrt{T_1}}\right)\right).$$

The solution u to the heat equation is explicit and given by

$$u(t,x) = \mathbb{E}\left[g(W_{T_1}) \middle| W_t = x\right] = \mathbb{E}\left[g\left(x + W_{T_1} - W_t\right)\right] = \left(g * K_{T_1 - t}\right)(x), \tag{4.1}$$

where $*$ is the convolution operator, and for $t > 0$, $K_t : x \in \mathbb{R} \mapsto \frac{e^{-\frac{x^2}{2t}}}{\sqrt{2\pi t}}$ is the heat kernel.

Remark 5 (Other approaches). To generate $(S_t)_{t\in[0,T_1]}$ with the given constraints, we could have also calibrated a local volatility model

$$\mathrm{d}S_t = S_t \ell(t, S_t)\,\mathrm{d}W_t, \qquad t \in [0, T_1],$$

to the SPX smile at T_1. For instance, when SPX smiles at all maturities before T_1 are known, simply use the market local volatility given by Dupire's formula [8]. Another possibility could have been to use any stopping time ν solution of the Skorokhod embedding problem for the distribution μ_1 and define $S_t := W_{\nu \wedge \frac{t}{T_1 - t}}$, where the Brownian motion W starts at S_0.

4.2. **Step 2: Simulation of V given $(S_t)_{t\in[0,T_1]}$.** At this stage, we have simulated $(S_t)_{t\in[0,T_1]}$ such that $(S_t)_{t\in[0,T_1]}$ is an $((\mathcal{F}_t),\mathbb{P})$-martingale, and S_1 has distribution μ_1 under $\mathbb{P}$. Now, we simulate V given $\sigma((W_s)_{s\in[0,T_1]})$ under $\mathbb{P}$ as follows: the distribution of V given $\sigma((W_s)_{s\in[0,T_1]})$ under $\mathbb{P}$ is assumed to depend only on S_1, and is taken equal to the distribution of V given S_1 under μ^*. Since S_1 has distribution μ_1 under both μ^* and $\mathbb{P}$, this means that the distribution of (S_1, V) is the same under μ^* and $\mathbb{P}$; in particular, V has distribution μ_V under $\mathbb{P}$.

4.3. **Step 3: Simulation of $(S_t)_{t\in[T_1,T_2]}$ given $\mathcal{F}_{T_1}$.** In this last step, we build dynamics for $(S_t)_{t\in[T_1,T_2]}$ conditional on $\mathcal{F}_{T_1}$ such that $(S_t)_{t\in[T_1,T_2]}$ is an $((\mathcal{F}_t),\mathbb{P})$-martingale starting from S_1, and S_2 has distribution μ_2. We use once again a Markov functional construction. Given $S_1 = s$ and $V = v$, we consider

$$g_{s,v}(x) := F^{-1}_{\mu_{2|s,v}}\left(F_{\mathcal{N}(0,1)}\left(\frac{x}{\sqrt{\tau}}\right)\right)$$

for all $x \in \mathbb{R}$ and where $\mu_{2|s,v}$ is the distribution of S_2 given $S_1 = s$ and $V = v$ under μ^*. Then, given $\mathcal{F}_{T_1}$, we define for $t \in (T_1, T_2]$,

$$S_t := u_{S_1,V}(t, W_t - W_{T_1})$$

where for every $s, v > 0$,

$$u_{s,v}(t,x) = \mathbb{E}\left[g_{s,v}\left(W_{T_2} - W_{T_1}\right) \middle| W_t - W_{T_1} = x\right] = \mathbb{E}\left[g_{s,v}\left(x + W_{T_2} - W_t\right)\right] = \left(g_{s,v} * K_{T_2 - t}\right)(x). \tag{4.2}$$

It is easy to check that $(S_t)_{t\in[T_1,T_2]}$ is an $((\mathcal{F}_t),\mathbb{P})$-martingale starting from S_1. Moreover, the distribution of S_2 given (S_1, V) is the same under μ^* and $\mathbb{P}$. As the distribution of (S_1, V) is the same under μ^* and $\mathbb{P}$, we conclude that the distribution of (S_1, V, S_2) is the same under μ^* and $\mathbb{P}$. In particular S_1, V, and S_2 have distributions μ_1, μ_V, and μ_2 under $\mathbb{P}$. We have thus built a model $\mathbb{P}$ on $((S_t)_{t\in[0,T_2]}, V)$ such that (a) S_1, V, and S_2 have distributions μ_1, μ_V, and μ_2 under $\mathbb{P}$; (b) $(S_t)_{t\in[0,T_2]}$ is an $((\mathcal{F}_t),\mathbb{P})$-martingale; and (c) V is the VIX at T_1, since by construction

$$\mathbb{E}^{\mathbb{P}}\left[L\left(\frac{S_2}{S_1}\right)\middle|\mathcal{F}_{T_1}\right] = \mathbb{E}^{\mathbb{P}}\left[L\left(\frac{S_2}{S_1}\right)\middle|S_1, V\right] = \mathbb{E}^{\mu^*}\left[L\left(\frac{S_2}{S_1}\right)\middle|S_1, V\right] = V^2.$$

Remark 6 (Extension to several VIX maturities). Note that this approach can easily be iterated on intervals $[T_i, T_{i+1}]$. For example, after Step 3, we will have generated $S_2 \sim \mu_2$. Then, we just need to generate $V_{T_2} \sim \mu_{V_{T_2}}$ and repeat the same procedure. Here, we disregard the Wednesday/Friday issue and are making the approximation that the VIX future maturities are exactly 30 days apart.

4.4. **Numerical implementation.** In this section, we detail how to implement the continuous-time extension. Let $\mathcal{T}_N = \{t_0 = 0, \ldots, T_1, \ldots, t_N = T_2\}$ be a given time-grid with $N \in \mathbb{N}_{>0}$ and let us simulate M Brownian motions $(W^{(i)}_{t\in\mathcal{T}_N})_{i=1,\ldots,M}$ where $M \in \mathbb{N}_{>0}$ denotes the number of Monte Carlo paths. For every Monte Carlo path $i \in \{1, \ldots, M\}$, we perform the following steps:

Step 1: Simulation of $(S^{(i)}_t)_{t\in[0,T_1]}$, see section 4.1. For every time-step $t \in \mathcal{T}_N$ such that $0 \le t \le T_1$, we set $S^{(i)}_t = u(t, W^{(i)}_t)$ where the function $u(t,\cdot)$ (computed using an Hermite quadrature) has been defined in (4.1). Note that we use the (true) market inverse c.d.f. $F^{-1}_{\mu_1}$.

Step 2: Simulation of $V^{(i)}$ given $(S^{(i)}_t)_{t\in[0,T_1]}$, see section 4.2. In general, we define $V^{(i)} = h_i(U^{(i)})$ where h_i is the linear interpolation of the two inverse c.d.f. $F^{-1}_{V|S_1=s}$ of V given S_1 under μ^* calculated at the two nearest points s of $S^{(i)}_1$ in the grid $\mathcal{G}_1$, where $U^{(i)} \sim \mathcal{U}([0,1])$ are i.i.d. However, if $S^{(i)}_1 < s^{(1)}_1$ (resp., $S^{(i)}_1 > s^{(n_{S_1})}_1$), we set $V^{(i)} = F^{-1}_{V|S_1=s^{(1)}_1}(U^{(i)})$ (resp., $V^{(i)} = F^{-1}_{V|S_1=s^{(n_{S_1})}_1}(U^{(i)})$).

Step 3: Simulation of $(S^{(i)}_t)_{t\in[T_1,T_2]}$ given $(S^{(i)}_{T_1}, V^{(i)})$, see section 4.3. For every time-step $t \in \mathcal{T}_N$ such that $T_1 < t \le T_2$, we set $S^{(i)}_t = u_{S^{(i)}_1,V^{(i)}}(t, W^{(i)}_t - W^{(i)}_{T_1})$ where for every $s, v > 0$, the function $u_{s,v}(\cdot,\cdot)$ (computed using an Hermite quadrature) was defined in (4.2). To compute $g_{S^{(i)}_1,V^{(i)}}(\cdot)$, we use a bilinear

interpolation in (s, v) of the inverse c.d.f. $F^{-1}_{\mu_{2|s,v}}$ when $(S_1^{(i)}, V^{(i)})$ lies within the range of the grid points; for all other $(S_1^{(i)}, V^{(i)})$, we use their projection onto their nearest point in the grid.

Algorithm 1 Continuous-time extension: pseudo-code for the simulation of $(S_t)_{t\in[0,T_2]}$.

Require: $\theta^\star$ from implied Newton algorithm, $M \in \mathbb{N}_{>0}$ (MC paths), $N \in \mathbb{N}_{>0}$ (discretization points), time-grid $\mathcal{T}_N = \{t_0 = 0 < \cdots < T_1 < \cdots < t_N = T_2\}$.

1: **for** $i = 1, \dots, M$ **do**
2: Simulate Brownian motion $W_t^{(i)}$ and set $S_t^{(i)} = u(t, W_t^{(i)})$ for every $t \in \mathcal{T}_N$ and $0 \le t \le T_1$.
3: Simulate $U_1^{(i)} \sim \mathcal{U}([0,1])$ and set $V^{(i)} = F^{-1}_{V|S^{(i)}_{T_1}}(U_1^{(i)})$.
4: Set $S_t^{(i)} := u_{S^{(i)}_{T_1}, V^{(i)}}(t, W_t^{(i)} - W_{T_1}^{(i)})$ for every $t \in \mathcal{T}_N$ and $T_1 < t \le T_2$.
5: **end for**

Remark 7 (General remarks on the algorithm). The different sources of errors in the proposed algorithm come from the quadrature grid, approximation of the inverse c.d.f., Monte Carlo simulation, and numerical integration for the functions (4.1) and (4.2). Once the discrete-time calibration has been achieved, the whole procedure is extremely fast. It essentially only requires the simulation of a one-dimensional Brownian motion along with two one-dimensional integrations.

The resulting smiles (market, discrete, and continuous time) for the SPX at T_1 and T_2 and that of the VIX at T_1 are displayed in Figure 4.1; they were computed by Monte Carlo simulation with 10^5 paths.

4.5. **Pricing.** Our continuous-time model can be used to price path-dependent options on the SPX with the guarantee that the model exactly matches the SPX smiles at T_1 and T_2 and the VIX future and VIX smile at T_1, thus taking into account information about the forward volatility at T_1 not included in the SPX smiles. As a pricing exercise, we compare it with models commonly used by practitioners: the Dupire local volatility model [8] and the local stochastic version of the two-factor Bergomi model [3]; those models are calibrated to the full SPX implied volatility surface, but not to VIX smiles. We consider spot-starting, forward-starting, and mixed versions of lookback and Asian options. The prices are reported in Table 2 along with their 95% confidence interval. We used 10^5 Monte Carlo paths and a trapezoidal rule to approximate the time integral for the Asian option. Note that for forward-starting options, the price in our jointly calibrated model is always larger than in the other two models: ignoring the VIX information leads to underpricing these forward-starting payoffs.

Price	LV	LV + Bergomi 2F	Our continuous-time model
$(M_{0,T_2} - S_0)_+$	76.81 ± 0.30	73.13 ± 0.28	75.37 ± 0.36
$(M_{T_1,T_2} - S_0)_+$	70.53 ± 0.32	67.20 ± 0.30	68.92 ± 0.38
$(M_{T_1,T_2} - S_{T_1})_+$	61.21 ± 0.28	56.16 ± 0.23	62.99 ± 0.34
$100 \times \frac{M_{T_1,T_2}}{S_{T_1}}$	101.896 ± 0.011	101.707 ± 0.09	101.941 ± 0.012
$(A_{T_1,T_2} - S_0)_+$	32.02 ± 0.22	31.96 ± 0.22	33.22 ± 0.26
$(A_{T_1,T_2} - S_{T_1})_+$	19.51 ± 0.16	19.05 ± 0.15	23.14 ± 0.17
$(\frac{A_{T_1,T_2}}{S_{T_1}} - 1)_+$	$(7.05 \pm 0.06) \times 10^{-3}$	$(6.87 \pm 0.05) \times 10^{-3}$	$(8.27 \pm 0.06) \times 10^{-3}$

TABLE 2. Prices of various options in the Dupire local volatility model, local Bergomi model, and our continuous-time model. We choose the same parameters as [13, Table 4] for the local two-factor Bergomi, jointly calibrated to the term-structures of SPX ATM skew and VIX^2 implied volatility: $k_1 = 21.91, k_2 = 1.04, \rho_{XY} = 1, \rho_{SX} = -1, \rho_{SY} = -1, \theta_1 = 0.77, \omega = 6.64$. We have defined: $M_{t,T} := \max_{t\le u\le T} S_u$, $A_{t,T} := \frac{1}{T-t}\int_t^T S_u \mathrm{d}u$

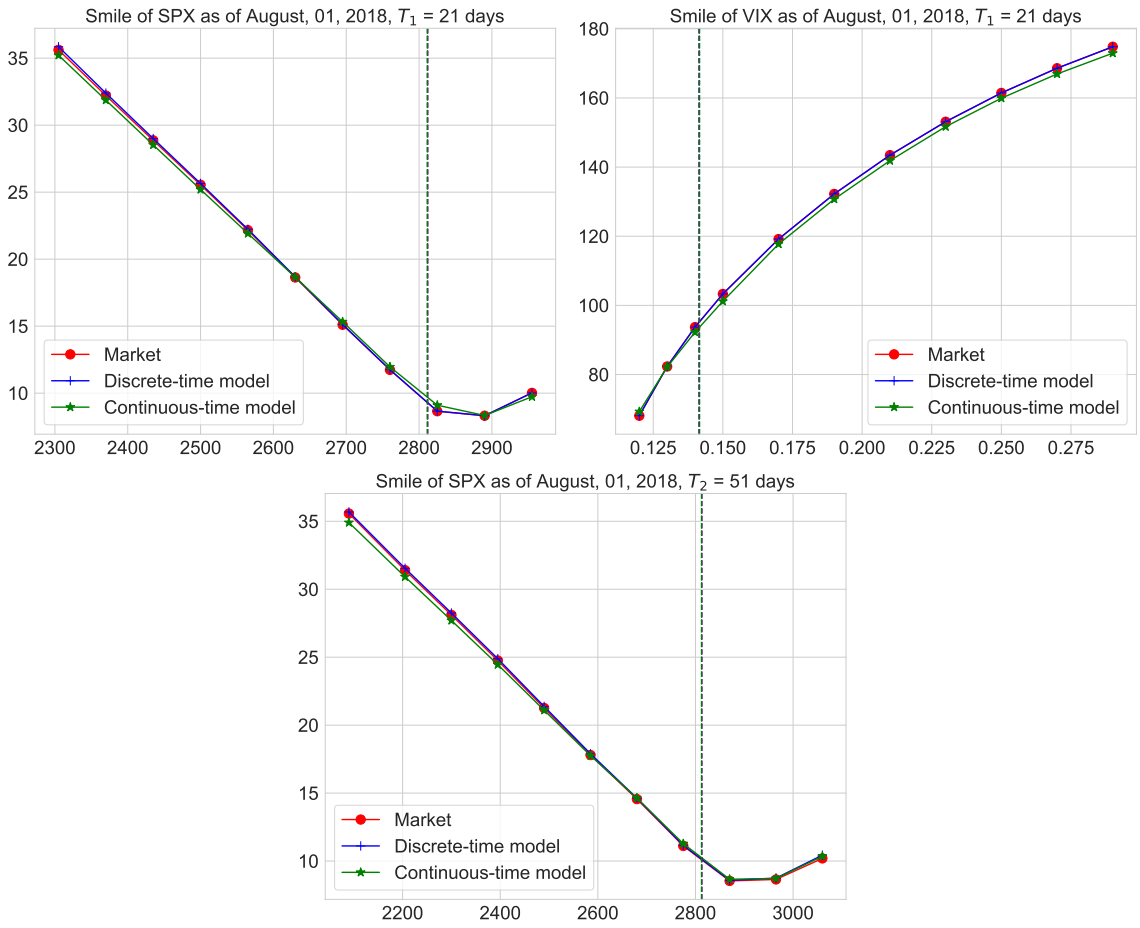

FIGURE 4.1. SPX smiles at T_1 and T_2 and VIX smile at T_1 (market, discrete-time model computed using the implied Newton algorithm, and continuous-time extension), as of August 1, 2018.

5. Conclusion

In this article, we have:

- improved model-free bounds on SPX options by incorporating VIX options data;
- built the minimum-entropy jointly calibrated discrete-time model μ^* very fast using an implied Newton method;
- seamlessly extended this discrete-time model to continuous time in a *purely forward* fashion, using Markov functionals.

Thus, we have established a swift process for creating an arbitrage-free continuous-time model for SPX that accurately calibrates to SPX smiles, VIX futures, and VIX smiles. Such a model can be used for pricing and hedging exotic options, computing reserves or valuation adjustments, and assessing model risk. Our main methodological contribution is that we first build a jointly calibrated *discrete-time* model (S_{T_i}), where the T_i are the calibrated maturities, that is later extended to *continuous time* using an arbitrage-free martingale time-interpolation. Since the discrete-time model can be exactly calibrated much faster than continuous-time models, and since extremely fast extrapolations exist, this novel approach seems to be a promising new avenue for calibrating models.

Acknowledgements. We appreciate the valuable feedback from the anonymous referees.

References

[1] Eduardo Abi Jaber, Camille Illand, and Shaun Li. The quintic Ornstein-Uhlenbeck model for joint SPX and VIX calibration. *Risk*, June 2023.

[2] Marco Avellaneda, Craig Friedman, Richard Holmes, and Dominick Samperi. Calibrating volatility surfaces via relative-entropy minimization. *Applied Mathematical Finance*, 4(1):37–64, 1997.

[3] Lorenzo Bergomi. Smile Dynamics II. Risk Magazine, 2005.

[4] Antoine Conze and Pierre Henry-Labordère. A New Fast Local Volatility Model. *Risk*, April 2022.

[5] Christa Cuchiero, Guido Gazzani, Janka Möller, and Sara Svaluto-Ferro. Joint calibration to SPX and VIX options with signature-based models. *arXiv preprint arXiv:2301.13235*, 2023.

[6] Hadrien De March. Entropic approximation for multi-dimensional martingale optimal transport. *arXiv preprint arXiv:1812.11104*, 2018.

[7] Hadrien De March and Pierre Henry-Labordère. Building arbitrage-free implied volatility: Sinkhorn's algorithm and variants. *Available at SSRN 3326486*, 2019.

[8] Bruno Dupire. Pricing with a smile. *Risk*, 7(1):18–20, 1994.

[9] Jim Gatheral, Paul Jusselin, and Mathieu Rosenbaum. The quadratic rough Heston model and the joint S&P 500/Vix smile calibration problem. *Risk*, 2020.

[10] Ivan Guo, Grégoire Loeper, Jan Obłój, and Shiyi Wang. Joint Modeling and Calibration of SPX and VIX by Optimal Transport. *SIAM Journal on Financial Mathematics*, 13(1):1–31, 2022.

[11] Julien Guyon. The joint S&P 500/VIX smile calibration puzzle solved. *Risk, April*, 2020.

[12] Julien Guyon. Dispersion-Constrained Martingale Schrödinger Bridges: Joint Entropic Calibration of Stochastic Volatility Models to S&P 500 and VIX Smiles. *Available at SSRN: https://ssrn.com/abstract=3853237*, 2022.

[13] Julien Guyon. The VIX future in Bergomi models: Fast approximation formulas and joint calibration with S&P 500 skew. *SIAM Journal on Financial Mathematics*, 13(4):1418–1485, 2022.

[14] Julien Guyon. Dispersion-Constrained Martingale Schrödinger Problems and the Exact Joint S&P 500/VIX Smile Calibration Puzzle. *Finance and Stochastics*, 28(1):27–79, 2024.

[15] Julien Guyon and Jordan Lekeufack. Volatility is (mostly) path-dependent. *Quantitative Finance*, 23(9):1221–1258, 2023.

[16] Julien Guyon and Scander Mustapha. Neural joint S&P 500/VIX smile calibration. *Risk*, December 2023.

[17] Pierre Henry-Labordere. Automated option pricing: Numerical methods. *International Journal of Theoretical and Applied Finance*, 16(08):1350042, 2013.

[18] Pierre Henry-Labordère. *Model-free hedging: A martingale optimal transport viewpoint*. Chapman and Hall/CRC, 2017.

[19] Mathieu Rosenbaum and Jianfei Zhang. Deep calibration of the quadratic rough Heston model. *Risk*, 2022.

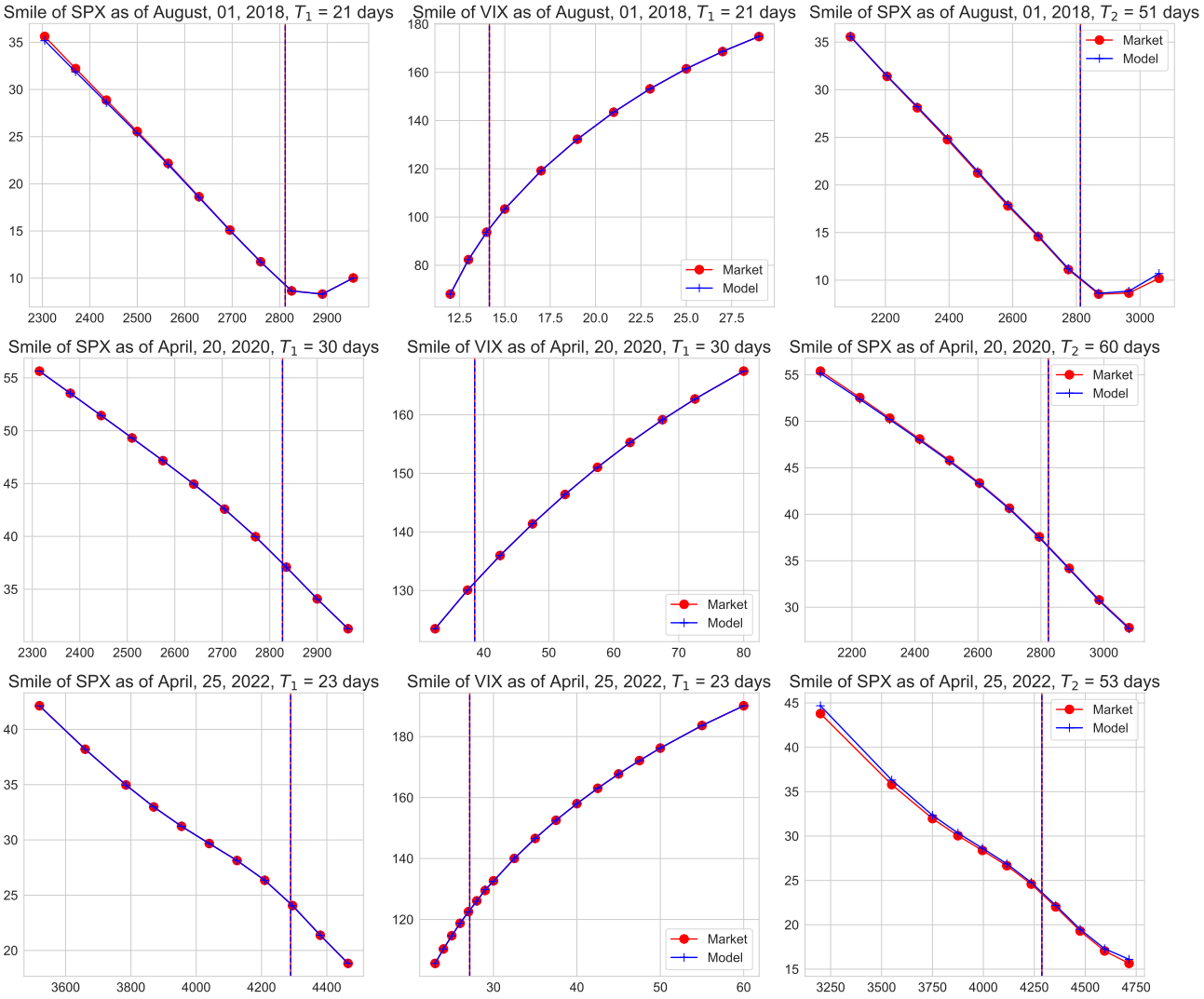

FIGURE 5.1. Futures and smiles of S_1, V, S_2 after 60 seconds in the calibrated model with the Newton-implied algorithm versus market smiles as of August 1, 2018, with $T_1 = 21$ days, April 20, 2020, with $T_1 = 30$ days, and April, 2022, with $T_1 = 23$ days

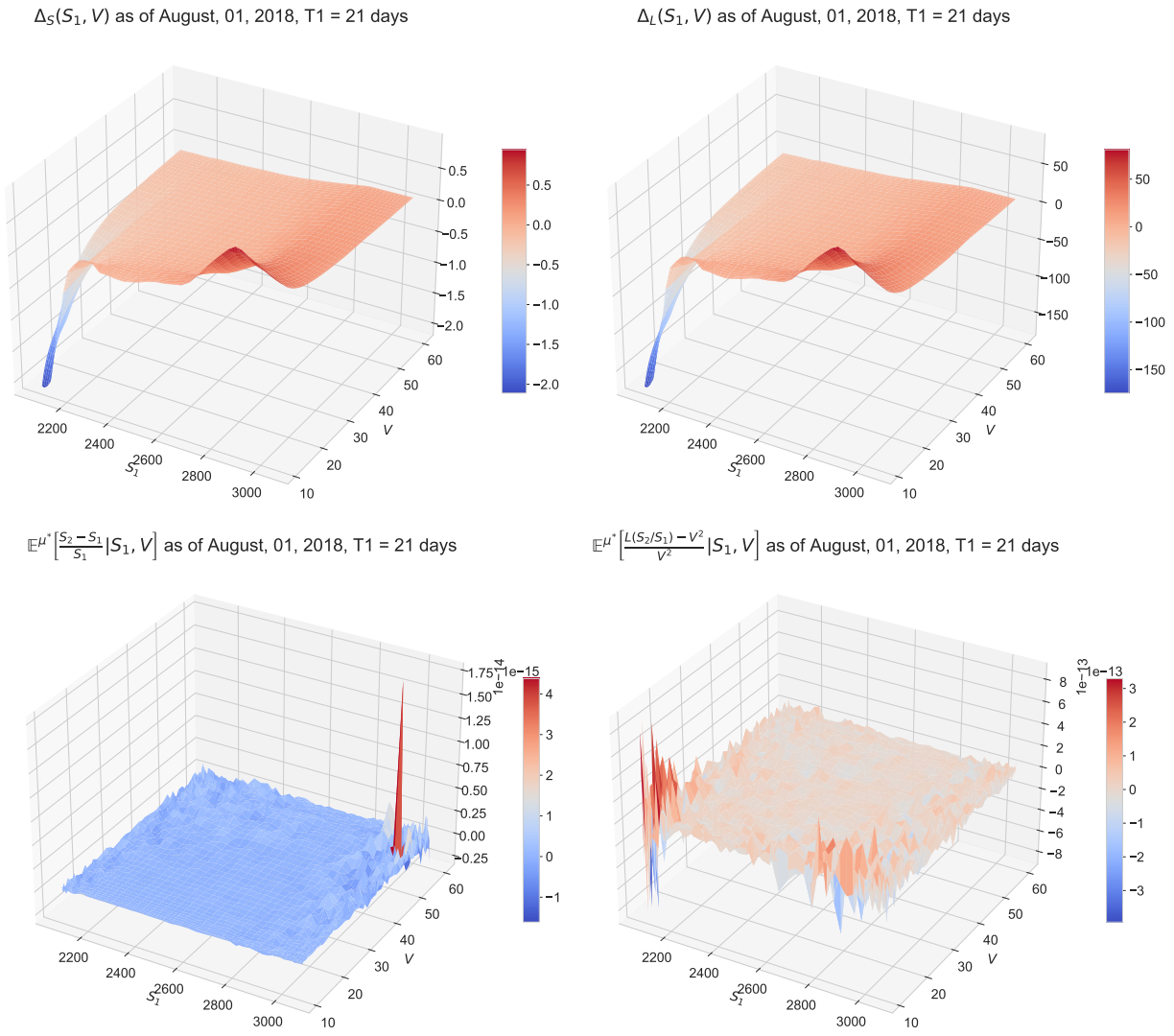

FIGURE 5.2. Functions $(s_1, v) \mapsto \Delta_S(s_1, v)$ and $(s_1, v) \mapsto \Delta_L(s_1, v)$ (upper figures) and martingale and consistency checks (bottom figures) as of August 1, 2018 for $T_1 = 21$ days.

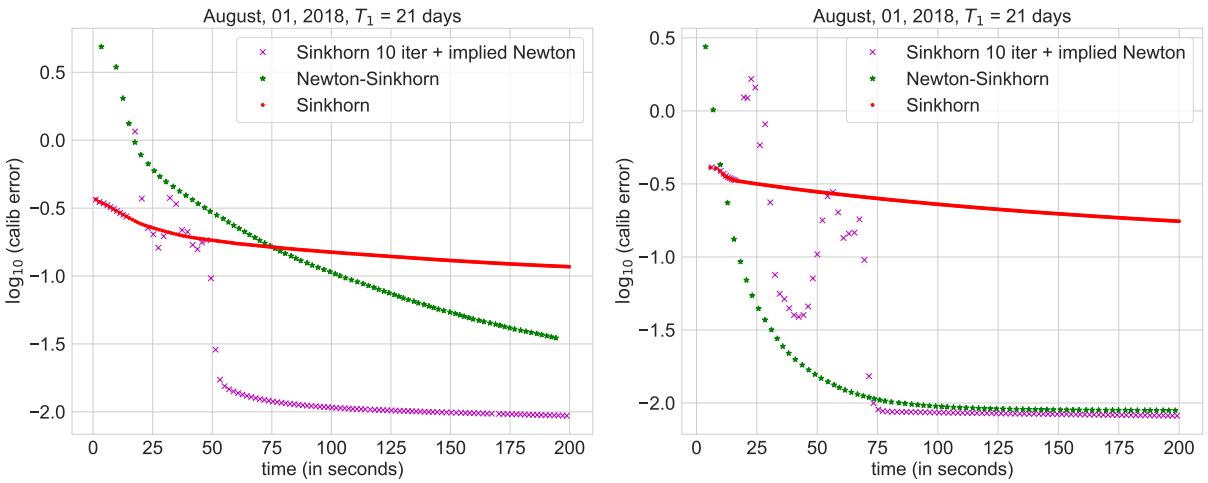

FIGURE 5.3. Comparison of the calibration errors as of August 01, 2018 for $T_1 = 21$ days with a lognormal prior (left figure) and an independent prior (right figure).

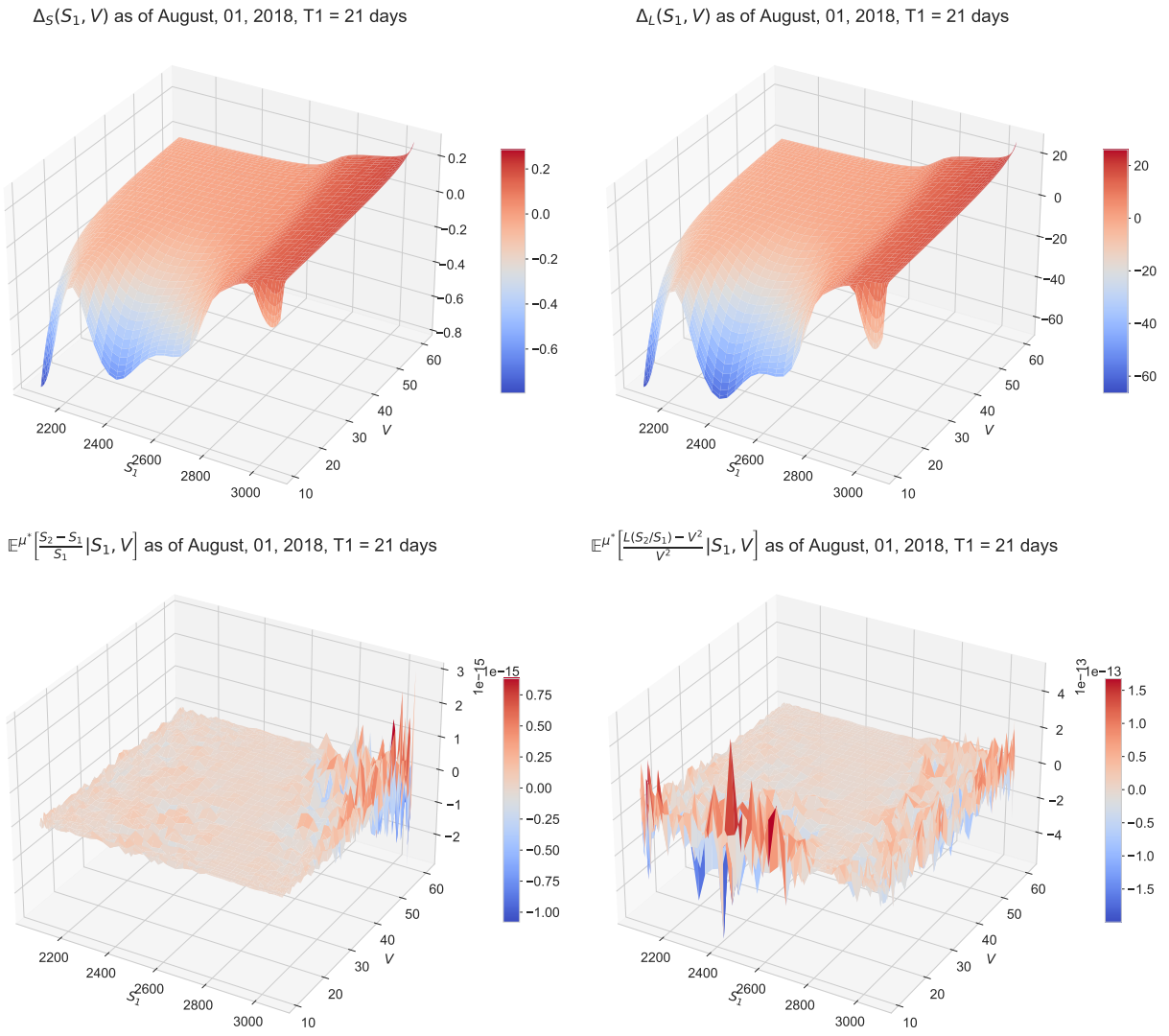

FIGURE 5.4. Functions $(s_1, v) \mapsto \Delta_S(s_1, v)$ and $(s_1, v) \mapsto \Delta_L(s_1, v)$ (upper figures) and martingale and consistency checks (bottom figures) as of August 1, 2018 for $T_1 = 21$ days when the reference measure is the product measure.