Lecture Notes

Derivatives Modeling
Modeling & Implementation Issues

Stéphane CRÉPEY

Université Paris Cité
https://www.lpsm.paris/pageperso/crepey

January 5, 2023

Contents

Chapter 0

Introduction

These lecture notes are largely based on (Crépey, 2013). II.§4.A is reproduced from Wikipedia. Modulo notation or other cosmetic changes II.§4.B and II.§5.E are Sections 2 and 3 in Fukasawa et al. (2021). VI.§1, VI.§2, VI.§3, VII.§3, and VIII respectively rely on (Crépey and Dixon, 2020), (Chataigner et al., 2020) (Abbas-Turki et al., 2022), (Chataigner et al., 2021) and (Chataigner et al., 2020)[1]. VII.§2 is (Guo et al., 2021), with minimal modifications made for pedagogical purposes. We express our warm thanks to all co-authors!

The numbering is local to the current environment, e.g. B means Part B in the current section, §2.B refers to Part B of Section 2 (other than the current one) in the current chapter, IV.§3.B to Part B of Section 3 in Chapter IV (other than the current chapter).

§1 Prerequisites

A Probability Theory

At the undergraduate level (including conditional expectation). Gaussian law of mean vector μ and covariance matrix Γ, exponential law of parameter λ and Poisson law of parameter λ respectively denoted by $\mathcal{N}(\mu,\Gamma)$, $\mathcal{E}_\lambda$ and $\mathcal{P}_\lambda$.

Lemma 1 (Gaussian conditioning) *Let X be a Gaussian vector, I and J be two complementary sets of indices of the coordinates of X, $X_I := (X_i)_{i\in I}$ and $X_J := (X_j)_{j\in J}$, with covariance matrix of $\begin{pmatrix} X_I \\ X_J \end{pmatrix}$ denoted by $\begin{pmatrix} \Gamma_I & \Gamma_{J,I}^\top \\ \Gamma_{J,I} & \Gamma_J \end{pmatrix}$ (where Γ_I and Γ_J are the respective covariance matrices of X_I and X_J). The conditional distribution of X_J given X_I is*

$$\mathcal{N}\left(\Gamma_{J,I}\Gamma_I^{-1}X_I, \Gamma_J - \Gamma_{J,I}\Gamma_I^{-1}\Gamma_{J,I}^\top\right).$$

Statistics Linear regression, principal component analysis (PCA).

Hereafter $(\Omega,\mathcal{A},\mathbb{Q})$ denotes a probability space. That is, Ω is a set of elementary events ω and $\mathcal{A}$ is a σ-field of measurable events $A\subseteq\Omega$, including Ω itself, closed under complement and countable union; $\mathbb{Q}(A)$ is the probability of an event $A\in\mathcal{A}$. The expectation, variance, and covariance with respect to $\mathbb{Q}$ are denoted by $\mathbb{E},\mathbb{V}\mathrm{ar}$ and $\mathbb{C}\mathrm{ov}$. The expectation with respect to an arbitrary probability measure, say $\mathbb{P}$, will be denoted by $\mathbb{E}^{\mathbb{P}}$ (so $\mathbb{E}=\mathbb{E}^{\mathbb{Q}}$).

[1] with respective github.com repositories mfrdixon/GP-CVA, mChataign/DupireNN, BouazzaSE/NeuralXVA, mChataign/Beyond-Surrogate-Modeling-Learning-the-Local-Volatility-Via-Shape-Constraints, and mChataign/smile-Completion.

A.1 Stochastic Processes and Analysis

Filtrations and stopping times in continuous time, Markov chains, Brownian motion and Poisson process, Itô calculus, Girsanov theorem, stopping times and (super)martingales, Markov processes, all at the level of Crépey (2013, Part I).

Hereafter a constant horizon $T \in (0, +\infty)$ is fixed, corresponding to the final maturity of a financial derivative in the financial interpretation. The probability space $(\Omega, \mathcal{A}, \mathbb{Q})$ is endowed with a filtration $\mathfrak{F} = (\mathfrak{F}_t, t \in [0,T])$, with respect to which all processes are adapted and time-t conditional expectation is meant, at each point in time $t \in [0,T]$. We assume the so called "usual conditions", according to which our filtration $\mathfrak{F}$ is right-continuous and complete. Hence, in particular, any local martingale[2] admits a càdlàg[3] modification, so that we may and do restrict ourselves to local martingales in a càdlàg version.

B Mathematical Finance

Notions of risk-free asset S^0 growing at some bounded from below, time integrable short rate process r, starting from a conventional value of 1 at time 0, and of the risk-free discount factor $\beta = e^{-\int_0^\cdot r_s ds}$ (inverse of the risk-free asset); of a primary market, underlying a financial derivative, consisting in the risk-free asset S^0 and a finite number of risky assets, denoted in vector form by S, possibly paying dividends $\mathcal{D}$, with S and $\mathcal{D}$ modeled as (càdlàg) semimartingales, hence valid stochastic integrators of any càglàd[4] integrands as explained in IX.§2.

Notions of physical probability measure $\mathbb{P}$ versus risk-neutral probability measure $\mathbb{Q}$, in the sense of the following:

Definition 1 *A risk-neutral measure on the primary market is a probability measure $\mathbb{Q} \sim \mathbb{P}$ such that*

$$d(\beta \widehat{S})_t := d(\beta S)_t + \beta_t d\mathcal{D}_t \tag{1}$$

is a $\mathbb{Q}$ local martingale[5].

Example 1 *We consider a primary market composed of the risk-free asset $S^0 = \beta^{-1}$ and of S reducing to one nonnegative asset, which may represent a stock, an index, a futures price, the value of a commodity, or any traded asset reasonably modeled in the form of a (nonnegative) jump-diffusion. The riskless interest rate r in the economy and the dividend yield q on S[6] are assumed to be bounded from below and time integrable. Setting $\kappa = r - q$ and $\alpha = e^{-\int_0^\cdot \kappa_t dt}$, we have[7]:*

$$d(\beta_t \widehat{S}_t) := d(\beta_t S_t) + \beta_t q_t S_t dt = \beta_t (dS_t - \kappa_t S_t dt) = \beta_t \alpha_t^{-1} d(\alpha_t S_t). \tag{2}$$

Notions of self-financing trading strategy π (initial wealth), ζ càglàd[8] and[9] $r\zeta^0 S^0$ time-integrable[10]. with wealth process

$$\begin{aligned} V &= \pi + \int_0^\cdot \zeta_t^0 dS_t^0 + \int_0^\cdot \zeta_t dS_t + \int_0^\cdot \zeta_t d\mathcal{D}_t, \text{ i.e.} \\ \beta V &= \pi + \int_0^\cdot \zeta_t d(\beta S)_t + \int_0^\cdot \zeta_t \beta_t d\mathcal{D}_t, \end{aligned} \tag{3}$$

[2] the notion of martingale localized by means of stopping times as reviewed in IX.§1.
[3] see IX.§1.
[4] see the introductory paragraph to Chapter IX.
[5] cf. Definition IX.3.
[6] the exact interpretation of r and q depends on the nature of the underlying S.
[7] cf. IX.(16).
[8] see the introductory paragraph to Chapter IX.
[9] an additional technical integrability requirement that can be dropped, switching to the second line below for defining a self-financing trading strategy.
[10] e.g. ζ^0 progressive and such that $\int_0^T |r_t \zeta_t^0 S_t^0| dt$ finite a.s..

admissible in the sense

$$\beta V \geq -c \tag{4}$$

for some constant c (that may depend on the strategy), resp. $\mathbb{Q}$ admissible (given a risk-neutral measure $\mathbb{Q}$) in the sense

$$\beta V \geq M \tag{5}$$

for some $\mathbb{Q}$ martingale M (that may depend on the strategy); of (resp. $\mathbb{Q}$) arbitrage opportunity, or self-financing (resp. $\mathbb{Q}$) admissible trading strategy $(\pi = 0, \zeta)$ that can gain and cannot lose, i.e. such that

$$\mathbb{P}(V_T \geq 0) = 1\,,\ \mathbb{P}(V_T > 0) > 0. \tag{6}$$

Theorem 1 *If there exists a risk-neutral measure $\mathbb{Q}$ on the primary market, then this market is free of arbitrage opportunities, and even of $\mathbb{Q}$ arbitrage opportunities.*

In particular:

Corollary 1 *In the setup of Example 1, the primary market is free of $\mathbb{Q}$ arbitrage opportunities as soon as the process $\beta\widehat{S}$ or, equivalently*[11]*, the process αS, is a local martingale under a probability measure $\mathbb{Q} \sim$ the physical one.*

B.1 Financial Derivatives

Example 2 *Forward contracts on a risky asset S correspond to linear payoff functions $(S-K)$, vanilla calls/puts to the payoff functions $\phi(S) = (S-K)^{\pm}$.*

The first part of Theorem 1 (and a partial converse to it, which also holds true) motivates us to work hereafter under a risk-neutral probability measure $\mathbb{Q}$, with related time-t conditional expectation denoted by $\mathbb{E}_t$.

Definition 2 (i) *Given a European financial derivative on S promising a $\mathbb{Q}$ integrable payoff $\phi(S_T)$ at time T, the real-valued process Π such that, for $t \in [0,T]$,*

$$\beta_t \Pi_t = \mathbb{E}_t(\beta_T \phi(S_T)), \tag{7}$$

is called the $\mathbb{Q}$ price of the related European claim;
(ii) *Given an American derivative with payoff $\phi(S_\vartheta)$ at any stopping time ϑ chosen by the holder of the claim, denoting by Θ_t the family of all the $[t,T]$ valued stopping times, the real-valued process $\widetilde{\Pi}$ such that, for $t \in [0,T]$,*

$$\beta_t \widetilde{\Pi}_t = \max_{\vartheta \in \Theta_t} \mathbb{E}_t(\beta_\vartheta \phi(S_\vartheta)), \tag{8}$$

assuming $(\phi(S_t))$ integrable under $\mathbb{Q}$ and the max achieved in (8), is called the $\mathbb{Q}$ price of the related American claim.

Hence the discounted $\mathbb{Q}$ price $\beta\Pi$ of a European option is a $\mathbb{Q}$ martingale; the discounted $\mathbb{Q}$ price $\beta\widetilde{\Pi}$ of an American option is the $\mathbb{Q}$ Snell envelope of $\beta\phi(S_\cdot)$, i.e. the smallest $\mathbb{Q}$ supermartingale $\geq \beta_\cdot \phi(S_\cdot)$[12].

[11] in view of (2), by Theorem IX.1.
[12] cf. (El Karoui, 1981).

Notions of (resp. $\mathbb{Q}$) completeness of the primary market, of (resp. $\mathbb{Q}$) hedging and (resp. $\mathbb{Q}$) replication of a European financial derivative written on the primary market, where "$\mathbb{Q}$" is in reference to the embedded class of admissible strategies (admissible versus $\mathbb{Q}$ admissible); vector space of the $\mathbb{Q}$ integrable and $\mathbb{Q}$ replicable payoffs.

If S is Markov[13]and that $r_t = r(t, S_t)$ for a measurable function of $r = r(t, S)$, then there exist measurable functions $u = u(t, S)$ and $v = v(t, S)$ such that

$$\Pi_t = u(t, S_t) \text{ and } \widetilde{\Pi}_t = v(t, S_t)\,,\ t \in [0, T]. \tag{9}$$

In $\mathcal{C}^{1,2}$ regularity cases, related Itô formulas translate, via Lemma IX.6, into partial differential equations, for which (9) provides a Feynman-Kac representation. Hence all the randomness of the price of the financial derivative is encoded in that of S. The derivative pricing problem is reduced to the computation of the (deterministic) pricing function u and v. This is the basic mechanism through which deterministic methods can be used to compute derivative prices, though these are in essence stochastic processes.

The following process is the main focus of a derivative trader or risk manager.

Definition 3 *With $\check{\Pi} = \Pi$ or $\widetilde{\Pi}$ as relevant and recalling* (1)*, the process p such that*

$$\beta p = \int_0^{\cdot} \left(-d(\beta_s \check{\Pi}_s) + \zeta_s d(\beta_s \widehat{S}_s) \right) \tag{10}$$

is the profit-and-loss (negative of the tracking error) of the bank having sold the option and using the price-and-hedge strategy $(\check{\Pi}, \zeta)$.

In view of the above:

Corollary 2 *The profit-and-loss p is a $\mathbb{Q}$ martingale in the European case and a $\mathbb{Q}$ submartingale in the American case.*

C Optimization

Gradient descents and stochastic gradient descents[14] for convex and nonconvex problems.

[13] cf. IX.(26).
[14] see IX.§5.

ANALYTICS

Chapter I

Benchmark Models

After recalling a few model-free results, we give a succint primer of four reference models: the Black-Scholes, local volatility, Heston, and Merton models.

§1 Model-Free Results

We consider the setup of Example 0.1, in the case of constant riskless interest rate r in the economy and dividend yield q on S. Consistent with no arbitrage requirements[1], we assume that the process αS is a $\mathbb{Q}$ martingale, so that

$$\mathbb{E}S_T = S_0 e^{\kappa T}. \tag{1}$$

By application of 0.(7) and 0.(8), the $\mathbb{Q}$ price process of a European vanilla option with payoff $\phi(S_T)$ at the maturity time T, e.g. call/put $(S-K)^{\pm}$ for a so called strike $K \geq 0$, assuming that $\phi(S_T)$ is $\mathbb{Q}$ integrable, is defined, for $t \in [0,T]$, by

$$\Pi_t = e^{-r(T-t)}\mathbb{E}_t\phi(S_T); \tag{2}$$

Assuming $\phi(S_\cdot)$ càdlàg and $\mathbb{E}\big(\sup_{t\in[0,T]}|\phi(S_t)|\big) < +\infty$, the corresponding American option price is defined, for $t \in [0,T]$, by[2]

$$\widetilde{\Pi}_t = \max_{\vartheta\in\Theta_t} \mathbb{E}_t\big(e^{-r(\vartheta-t)}\phi(S_\vartheta)\big), \tag{3}$$

where Θ_t denotes the family of all the $[t,T]$ valued stopping times.

A Forwards and Call-Put Parity

By application of (1)-(2), a forward contract with payoff $S_T - K$ has the $\mathbb{Q}$ price

$$F_t(T,K) = S_t e^{-q\tau} - Ke^{-r\tau} = C_t(T,K) - P_t(T,K), \tag{4}$$

where $\tau = T-t$ and $C_t(T,K)$ and $P_t(T,K)$ are the time-t $\mathbb{Q}$ prices of the European vanilla call and puts with payoffs $(S_T-K)^{\pm}$.

Remark 1 *In strict local martingale market models where αS fails to be a true $\mathbb{Q}$ martingale, $\mathbb{Q}$ prices of European vanilla calls and puts may fail to satisfy the call-put parity relationship* (4)*, as well as*

[1] cf. Theorem 0.1 and Paragraph 0.B.1.

[2] assuming the existence of a maximiser in (3).

the no-static-arbitrage bounds of Proposition 3[3] below, so that the Black-Scholes implied volatility[4] of $\mathbb{Q}$ prices may fail to be well defined, or differ between calls and puts (Jacquier and Keller-Ressel, 2018). It is to avoid these pathologies that we restrict ourselves to models where αS is a true $\mathbb{Q}$ martingale.

The $\mathbb{Q}$ price $Se^{-q\tau}-Ke^{-r\tau}$ of the forward contract is therefore the same for any probability measure $\mathbb{Q}\sim$ the physical one and for which αS is a martingale. In this sense the price of a forward contract is model-free. The T forward value of S at time t, i.e. the value F_t^T of the strike K for which $F_t(T,K)$ vanishes, is

$$F_t^T = S_t e^{\kappa\tau}. \tag{5}$$

We denote by $\widetilde{\mathbb{Q}}$ the probability measure with $\mathbb{Q}$ density $\frac{S_T}{\mathbb{E}S_T} = \frac{S_T}{S_0 e^{\kappa T}} =: \nu_T$ and by $\widetilde{\mathbb{E}}$ the $\widetilde{\mathbb{Q}}$ expectation operator. Hence the Bayes formula $\mathbb{E}\chi = \widetilde{\mathbb{E}}(\frac{\chi}{\nu_T})$ holds, for any $\mathbb{Q}$ integrable random variable χ.

Proposition 1 *The European vanilla call $\mathbb{Q}$ price at time 0, $C_0 = \mathbb{E}e^{-rT}(S_T-K)^+$, satisfies*

$$C_0 = S_0 e^{-qT}\widetilde{\mathbb{Q}}(S_T>K) - Ke^{-rT}\mathbb{Q}(S_T>K). \tag{6}$$

Proof. Decompose $(S_T-K)^+$ by $x^+ = x\mathbb{1}_{x>0}$ and use the above Bayes formula for $\chi = S_T\mathbb{1}_{S_T>K}$. ∎

Hence, as opposed to forward contracts, calls and puts, as derivatives with nonlinear payoffs more generally, have $\mathbb{Q}$ prices that effectively depend on the detailed specification of the $\mathbb{Q}$ martingale αS.

See Figure 1 and play with the python code by Clint Howard on

https://clinthoward.github.io/portfolio/2017/04/16/BlackScholesGreeks for classical payoffs that can be expressed as linear combinations of calls and puts.

Example 1 *Bull and bear spreads (top panels of Figure 1) allow investors to make directional bets on S, at a more affordable price than vanilla calls and puts. Straddle and butterfly spreads (bottom panels of Figure 1) allow investors to bet on the future volatility (low or high) of S.*

B Calls and Puts: Bounds

We let $B_t = e^{-r(T-t)}$, $A_t = e^{-q(T-t)}$, and we forget the indices 0 in $A_0, B_0, C_0, P_0, \widetilde{C}_0, \widetilde{P}_0$, to alleviate the notation. So

$$r\geq 0 \Leftrightarrow B\leq 1\,,\ r>0\Leftrightarrow B<1\,,\ q\geq 0\Leftrightarrow A\leq 1\,,\ q>0\Leftrightarrow A<1.$$

The following results are stated and proved at time 0 for notational simplicity. We write "a.s." for almost surely, i.e. "with probability one", and "poss." for possibly, i.e. "with (strictly) positive probability".

Proposition 2 *We have $\widetilde{C}\geq C$[5], $\widetilde{P}\geq P$.*

Proof. For any payoff function ϕ with $\phi(S_\cdot)$ càdlàg and $\mathbb{E}\big(\sup_{t\in[0,T]}|\phi(S_t)|\big)<+\infty$, (3) yields

$$\widetilde{\Pi}_0 = \max_{\vartheta\in\Theta_0}\mathbb{E}\big(e^{-r\vartheta}\phi(S_\vartheta)\big)\geq \mathbb{E}\big(e^{-rT}\phi(S_T)\big) = \Pi_0,$$

by (2). ∎

Proposition 3 *We have $(SA-KB)^+\leq C\leq SA, (KB-SA)^+\leq P\leq KB$.*

Proof. By $\chi\geq 0\Rightarrow\mathbb{E}\chi\geq 0$ and martingality of αS. ∎

Proposition 4 **(i)** *If $B\leq 1\leq A$, then*[6] $\widetilde{C}=C\geq(S-K)^+$.

[3] cf. (1).
[4] cf. Lemma 4 below.
[5] assuming $\mathbb{E}\big(\sup_{t\in[0,T]}S_t\big)<+\infty$.
[6] assuming $\mathbb{E}\big(\sup_{t\in[0,T]}S_t\big)<+\infty$.

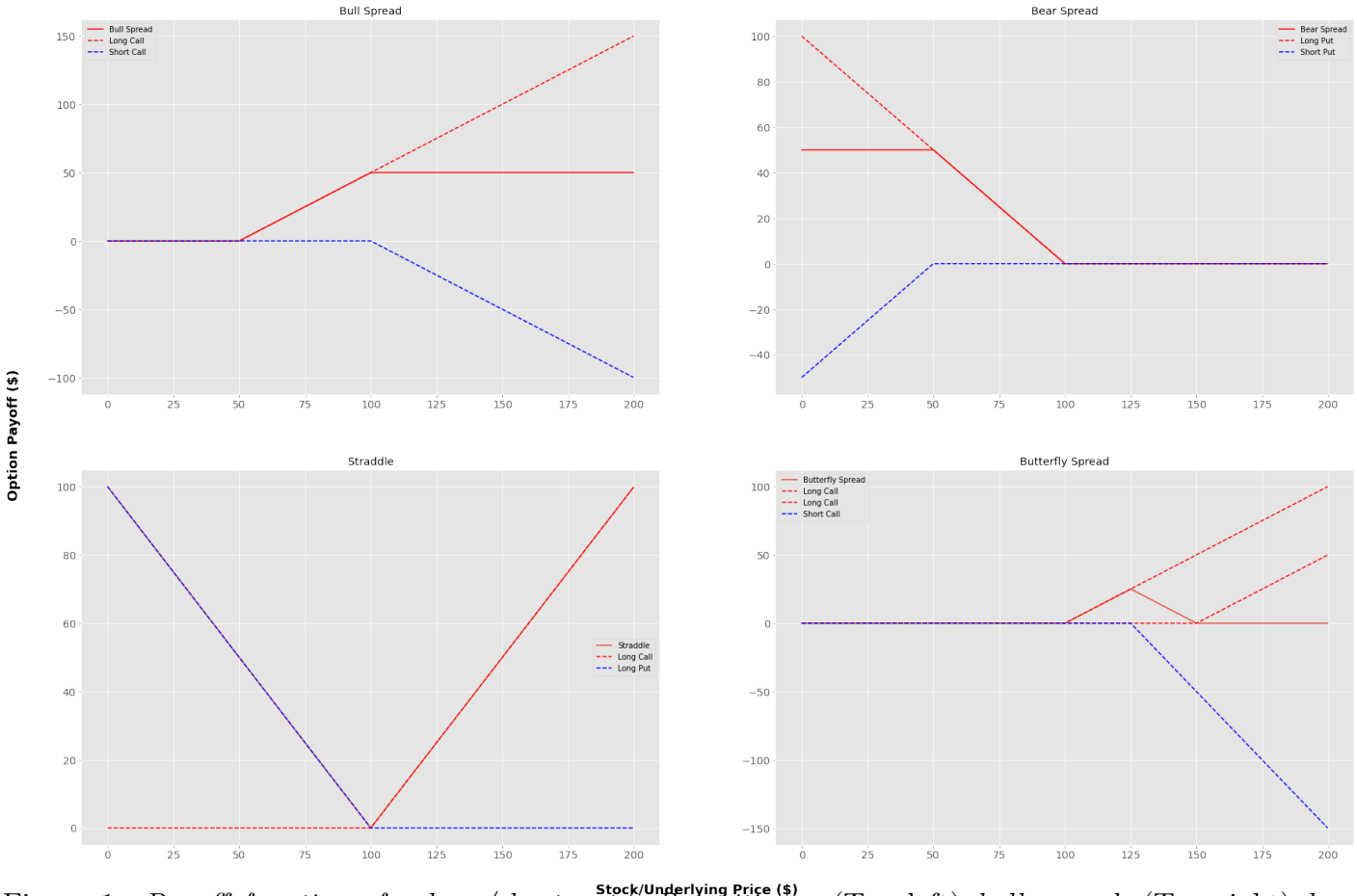


Figure 1: *Payoff functions for long/short put/call positions:* (Top left) *bull spread;* (Top right) *bear spread;* (Bottom left) *straddle;* (Bottom right) *butterfly spread.*

(ii) *If* $B \geq 1 \geq A$, *then* $\widetilde{P} = P \geq (K - S)^+$.

Proof. As both proofs are similar, we only prove (i). By Propositions 2 and 3, we have $(SA - KB)^+ \leq C \leq \widetilde{C}$. Moreover $B \leq 1 \leq A$ implies $SA - KB \geq S - K$, hence $(SA - KB)^+ \geq (S - K)^+$. Finally, for any $\vartheta \in \Theta_0$, the conditional Jensen inequality gives

$$\begin{aligned}\mathbb{E}\big(e^{-r\vartheta}(S_\vartheta - K)^+\big) &= \mathbb{E}\big((e^{-q\vartheta}e^{-\kappa\vartheta}S_\vartheta - e^{-r\vartheta}K)^+\big)\\ &= \mathbb{E}\Big(\big(e^{-q\vartheta}\mathbb{E}_\vartheta(e^{-\kappa T}S_T) - e^{-r\vartheta}K\big)^+\Big) = \mathbb{E}\big(\big(\mathbb{E}_\vartheta(e^{-q\vartheta}e^{-\kappa T}S_T - e^{-r\vartheta}K)\big)^+\big)\\ &\leq \mathbb{E}\mathbb{E}_\vartheta\big((e^{-q\vartheta}e^{-\kappa T}S_T - e^{-r\vartheta}K)^+\big) \leq \mathbb{E}(e^{-rT}S_T - e^{-rT}K)^+,\end{aligned}$$

assuming $B \leq 1 \leq A$[7]. The inequality $\widetilde{C} \leq C$ then follows from (3) and (2). ∎

Proposition 5 *Assuming positive* S *and* K*:*
(i) *If* $B \leq 1 \leq A$, *with at least one of the two inequalities strict, and* $S_T > K$ *poss., then* $C > (S-K)^+$.
(ii) *If* $B \geq 1 \geq A$, *with at least one of the two inequalities strict, then* $P > (K - S)^+$.

Proof. As both proofs are similar, we only prove (i). We have already seen in Proposition 4 and its proof that

$$(S - K)^+ \leq (SA - KB)^+ \leq C.$$

Moreover $S_T > K$ poss. implies $C > 0$. Hence $(SA - KB)^+ = 0$ implies

$$C > 0 = (SA - KB)^+ = (S - K)^+,$$

whereas $(SA - KB)^+ > 0$ implies, if $B \leq 1 \leq A$ with at least one of the two inequalities strict, that

$$C \geq (SA - KB)^+ > (S - K)^+. \blacksquare$$

[7] so that $-q\vartheta \leq -qT$ and $e^{-q\vartheta}e^{-\kappa T} \leq e^{-rT}$.

Remark 2 *The above bounds for $\mathbb{Q}$ prices (having assumed a $\mathbb{Q}$ true martingale αS) are also the classic no-static-arbitrage bounds for calls and puts, as opposed to the (potentially tighter) no-dynamic-arbitrage bounds also taking into account the admissibility issue regarding the trading strategies (Cox and Hobson, 2005; Jacquier and Keller-Ressel, 2018).*

C Calls and Puts: Shape Constraints

Proposition 6 **(i)** *C and P are respectively nonincreasing and nondecreasing in K;*
(ii) *C and P are B-Lipschitz in K;*
(iii) *C and P are convex in K;*
(iv) *If $B \leq 1 \leq A$, then C is nondecreasing in T; if $B \geq 1 \geq A$, then P is nondecreasing in T.*

Proof. **(i)** By nonnegativity of bull and bear spread payoffs[8], respectively, and $\chi \geq 0 \Rightarrow \mathbb{E}\chi \geq 0$.

(ii) By the bound $|K_1 - K_2|$ on bull and bear spread payoffs, where K_1 and K_2 are the strikes defining the spread options[9], so that the time-0 $\mathbb{Q}$ price of a bull or bear spread is $\leq B|K_1 - K_2|$.

(iii) By nonnegativity of butterly spread payoffs, for any defining strikes $K - k, K, K + k$[10], and the property $\chi \geq 0 \Rightarrow \mathbb{E}\chi \geq 0$.

(iv) for calls follows from the fact that the price at T_1 of the difference (calendar spread) between two calls of same strikes and maturities $T_2 \geq T_1$ is nonnegative if $B \leq 1 \leq A$, by the final inequality in Proposition 4(i). Hence the time-0 $\mathbb{Q}$ price of the calendar spread is also nonnegative, by an application of the tower rule. Similar argument regarding puts. ∎

D Breeden and Litzenberger Formula

We have

$$\partial_K C_0 = e^{-rT}\partial_K \mathbb{E}(S_T - K)^+ = -e^{-rT}\mathbb{E}\mathbb{1}_{\{S_T > K\}} = -e^{-rT}\mathbb{Q}(S_T > K). \tag{7}$$

Hence, whenever the model S admits a density of transition probability $\gamma_0(T, K) = \partial_K \mathbb{Q}(S_T \leq K)$ from $(0, S_0)$ to (T, K), the following Breeden and Litzenberger (1978) formula holds:

$$\partial^2_{K^2} C_0 = e^{-rT}\gamma_0(T, K) = \partial^2_{K^2} P_0, \tag{8}$$

by call/put parity (4).

D.1 Delta in Homogeneous Models

By homogeneous models, we mean models in which the European vanilla option prices are degree one homogeneous with respect to the pair (S_0, K) (for given values of the remaining parameters and risk factors in the model), which we denote by

$$C(0, \alpha S_0, T, \alpha K) = \alpha C(0, S_0, T, K),\ \alpha > 0,$$

or, equivalent to this (assuming the embedded differentiability),

$$S_0 \partial_S C(0, S_0, T, K) + K\partial_K C(0, S_0, T, K) = C(0, S_0, T, K). \tag{9}$$

In a homogeneous model, (7) yields

$$\begin{aligned} S_0 \partial_S C(0, S_0, T, K) &= C(0, S_0, T, K) - K\partial_K C(0, S_0, T, K) \\ &= C_0 + Ke^{-rT}\mathbb{Q}(S_T > K) = S_0 e^{-qT}\widetilde{\mathbb{Q}}(S_T > K), \end{aligned}$$

[8] see the top panels in Figure 1.
[9] see the top panels in Figure 1.
[10] see the bottom right panel in Figure 1.

by (6). Hence

$$\Delta_0 := \partial_S C(0, S_0, T, K) = e^{-qT}\widetilde{\mathbb{Q}}(S_T > K). \tag{10}$$

E Carr-Madan Payoff Decomposition Formula

The following Carr and Madan (2001) formula holds.

Lemma 1 *Given a twice differentiable payoff function φ, we have for every nonnegative fixed x and S:*

$$\begin{aligned}\varphi(S) = \varphi(x) + \varphi'(x)(S-x) + \int_0^x \varphi''(K)(K-S)^+ dK \\ + \int_x^{+\infty} \varphi''(K)(S-K)^+ dK.\end{aligned} \tag{11}$$

The time-t $\mathbb{Q}$ price of the European option with payoff $\varphi(S_T)$ at time T can be decomposed in terms of European vanilla calls and puts time-0 $\mathbb{Q}$ prices $C_t(T,K)$ and $P_t(T,K)$ as

$$\begin{aligned}\Pi_t = \varphi(x)e^{-r\tau} + \varphi'(x)(S_t e^{-q\tau} - xe^{-r\tau}) + \int_0^x \varphi''(K)P_t(T,K)dK \\ + \int_x^{+\infty} \varphi''(K)C_t(T,K)dK.\end{aligned} \tag{12}$$

Proof. The Taylor formula of order 2 with remainder in integral form yields[11]:

$$\varphi(S) = \varphi(x) + \varphi'(x)(S-x) + \int_{K=x}^{S} (S-K)\varphi''(K)dK.$$

Moreover, for $x, S \geq 0$,

$$\begin{aligned}&\int_{K=x}^{S} (S-K)\varphi''(K)dK \\ &= \mathbb{1}_{S\geq x}\int_{K=x}^{S} (S-K)\varphi''(K)dK + \mathbb{1}_{S\leq x}\int_{K=S}^{x} \varphi''(K)(K-S)dK \\ &= \mathbb{1}_{S\geq x}\int_{K=x}^{+\infty} (S-K)^+\varphi''(K)dK + \mathbb{1}_{S\leq x}\int_{K=0}^{x} \varphi''(K)(K-S)^+dK \\ &= \int_{K=x}^{+\infty} (S-K)^+\varphi''(K)dK + \int_{K=0}^{x} \varphi''(K)(K-S)^+dK,\end{aligned}$$

where the last identity holds because

$$\begin{aligned}K \geq x \Rightarrow \mathbb{1}_{S\geq x}(S-K)^+ = (S-K)^+ \\ K \leq x \Rightarrow \mathbb{1}_{S\leq x}(K-S)^+ = (K-S)^+.\end{aligned}$$

This yields to the payoff (which immediately implies the price) decomposition formula. ∎

E.1 Variance Swaps, Log-Contracts, and Semi-Static Replication in Stochastic Volatility Models

The following semi-static variance swap replication formula is valid in any stochastic volatility model (with $r = q = 0$, here, for notational simplicity).

[11] see https://en.wikipedia.org

Proposition 7 *Assuming, for some progressive and locally square integrable process σ,*

$$dS_t = \sigma_t S_t dW_t\,, \quad i.e.\ \frac{S_\cdot}{S_0} = e^{\int_0^\cdot \sigma_t dW_t - \frac{1}{2}\int_0^\cdot \sigma_t^2 dt}, \tag{13}$$

then it holds:

$$\xi := \frac{1}{T}\int_0^T \sigma_t^2 dt = \left(\frac{-2}{T}\right)\ln(\frac{S_T}{S_0}) + \int_0^T \frac{2dS_t}{TS_t}, \tag{14}$$

where

$$\ln(\frac{S_T}{S_0}) = \frac{S_T - S_0}{S_0} - \int_0^{S_0} \frac{1}{K^2}(K - S_T)^+ dK - \int_{S_0}^{+\infty} \frac{1}{K^2}(S_T - K)^+ dK. \tag{15}$$

Proof. The identity (14) is a reformulation of the right-hand side in (13) at time T, whereas (15) follows from (11) applied to $\varphi = \ln$ and $x = S_0$. ∎

Financial interpretation: in a stochastic volatility model, (the variable leg of) a variance swap, ξ, can be replicated by a short position in $\frac{2}{T}$ log contracts with payoff $\ln(\frac{S_T}{S_0})$ and the dynamic trading strategy with $\zeta = \frac{2}{TS}$ reflected by the integral in the right-hand side in (14); in turn, a log contract can be statically replicated by (a continuum of) calls and puts. In practice a finite number of calls and puts is used and the corresponding replication error also needs to be hedged dynamically.

F Fourier Pricing Formulas

F.1 Lewis European Vanillas Pricing Formula

Proposition 8 *The two probabilities in* (6) *can be represented in terms of the characteristic function $\Phi_T(z) = \mathbb{E}[\exp(izX_T)]$ of the log-spot $X_T = \ln(S_T)$ (assumed to exist, with $i^2 = -1$) as*

$$\begin{aligned}
\mathbb{Q}(S_T > K) &= \frac{1}{2} + \frac{1}{\pi}\int_0^\infty \mathfrak{Re}\left[\frac{e^{-izk}\Phi_T(z)}{iz}\right] dz \\
\widetilde{\mathbb{Q}}(S_T > K) &= \frac{1}{2} + \frac{1}{\pi}\int_0^\infty \mathfrak{Re}\left[\frac{e^{-izk}\Phi_T(z-i)}{iz\Phi_T(-i)}\right] dz,
\end{aligned} \tag{16}$$

with $\Phi_T(-i) = \mathbb{E}S_T = S_0 e^{\kappa T}$ in $\widetilde{\mathbb{Q}}(S_T > K)$.

Proof. By IX.(45), we have, setting $k = \ln(K)$,

$$\mathbb{Q}(S_T > K) = \mathbb{Q}\Big(X_T > k\Big) = \frac{1}{2} + \frac{1}{\pi}\int_0^\infty \mathfrak{Re}\left[\frac{e^{-izk}\Phi_T(z)}{iz}\right] dz$$

and the analogous formula holds for $\widetilde{\mathbb{Q}}(S_T > K)$ with $\Phi_T(z)$ replaced by

$$\widetilde{\Phi}_T(z) = \widetilde{\mathbb{E}}[\exp(izX_T)] = \frac{\mathbb{E}(e^{izX_T}e^{X_T})}{\mathbb{E}e^{X_T}} = \frac{\mathbb{E}(e^{i(z-i)X_T})}{\mathbb{E}e^{X_T}} = \frac{\Phi_T(z-i)}{\Phi_T(-i)}.$$ ∎

F.2 Carr and Madan European Vanillas Pricing Formula

In many models the characteristic function Φ_T can be computed explicitly. This is, for instance, the case in all affine jump-diffusion models, i.e.[12] Markov models with affine[13] generator coefficients. Knowing Φ_T, the formulas (6) and (16) enable one to compute C_0 numerically, by quadrature.

However, fast Fourier transform (FFT) algorithms cannot be used to evaluate the integrals in (16), due to the singularity of their integrands at $z = 0$. We now present an alternative formula, due to Carr and Madan (1999), that is amenable to valuation by FFT. Fast in FFT means fast in computing simultaneously the time-0 prices of a whole family of options with various strikes, as required for model calibration purposes.[14]

The idea is to compute the Fourier transform of the call price, viewed as a function $C_0(k)$ of the log-strike $k = \ln K$. The function $C_0(k)$ is not integrable, since $\lim_{k\to-\infty} C_0(k) = e^{-rT}\mathbb{E}S_T = S_0e^{-qT} > 0$. However, we obtain an integrable function by setting $C_0^\alpha(k) = e^{\alpha k}C_0(k)$ for a small enough[15] $\alpha > 0$. Letting f_T denote the density (assumed to exist) of $X_T = \ln(S_T)$, for every real z, we compute by IX.(42):

$$\begin{aligned}
\mathcal{F}C_0^\alpha(z) &= \int_{-\infty}^{\infty} e^{ikz}C_0^\alpha(k)dk = \int_{-\infty}^{\infty} e^{(\alpha+iz)k}C_0(k)dk \\
&= e^{-rT}\int_{k=-\infty}^{\infty} e^{(\alpha+iz)k}\left(\int_{x=-\infty}^{\infty}\left(e^x-e^k\right)^+ f_T(x)dx\right)dk \\
&= e^{-rT}\int_{x=-\infty}^{\infty} f_T(x)\left(\int_{k=-\infty}^{x} e^{(\alpha+iz)k}\left(e^x-e^k\right)dk\right)dx \\
&= e^{-rT}\int_{x=-\infty}^{\infty} f_T(x)\left(\frac{e^x}{\alpha+iz}\left[e^{(\alpha+iz)k}\right]_{-\infty}^{x} - \frac{1}{\alpha+iz+1}\left[e^{(\alpha+iz+1)k}\right]_{-\infty}^{x}\right)dx \\
&= e^{-rT}\int_{x=-\infty}^{\infty} f_T(x)\left(\frac{e^xe^{(\alpha+iz)x}}{\alpha+iz} - \frac{e^{(\alpha+iz+1)x}}{\alpha+iz+1}\right)dx,
\end{aligned}$$

where the last equality follows from the fact that $\lim_{c\to+\infty} e^{-(v+iz)c} = 0$, for every $v > 0$ and $z \in \mathbb{R}$. Therefore[16]

$$\begin{aligned}
\mathcal{F}C_0^\alpha(z) &= e^{-rT}\int_{-\infty}^{\infty} f_T(x)\frac{e^{(\alpha+iz+1)x}}{(\alpha+iz)(\alpha+iz+1)}dx \\
&= \frac{e^{-rT}\Phi_T(z-(\alpha+1)i)}{(\alpha+iz)(\alpha+iz+1)}.
\end{aligned} \tag{17}$$

By the inverse Fourier transform formula IX.(43) we then have, for every real k,

$$\begin{aligned}
C_0(k) = e^{-\alpha k}C_0^\alpha(k) &= \frac{e^{-\alpha k}}{2\pi}\int_{-\infty}^{\infty} e^{-ikz}\mathcal{F}C_0^\alpha(z)dz \\
&= \frac{e^{-\alpha k}}{\pi}\Re e\left[\int_0^{\infty} e^{-ikz}\mathcal{F}C_0^\alpha(z)dz\right],
\end{aligned} \tag{18}$$

where the last equality holds because the function

$$\mathbb{R} \ni k \mapsto C_0^\alpha(k)$$

is real valued, which implies that $\mathcal{F}C_0^\alpha(-z) = \overline{\mathcal{F}C_0^\alpha(z)}$[17] , for every $z \in \mathbb{R}$.

[12] see §3.D.
[13] in returns variable $x = \ln(S)$.
[14] cf. Chapter VII.
[15] so as not to compromise integrability at the other end $k \to +\infty$.
[16] assuming $\Phi_T(-(\alpha+1)i) = \mathbb{E}S_T^{\alpha+1} < +\infty$.
[17] with $\bar{\cdot}$ for complex conjugate.

Fast Fourier Transform Implementation We then approximate, by numerical integration based on Simpson's rule[18]:

$$C_0(k) \approx \frac{e^{-\alpha k}}{\pi} \Re e \left[\sum_{j=0}^{m-1} e^{-ikz_j} \mathcal{F}C_0^\alpha(z_j) w_j \right], \tag{19}$$

with m even in Simpson's rule and where, for $j = 0 \dots m-1$:

$$z_j = jh\,,\ w_j = \frac{h}{3}\big(3 + (-1)^{j+1} - \mathbb{1}_{\{j=0 \text{ or } m-1\}}\big)$$

For k of the form $k_l = \underline{k} + \frac{2\pi l}{mh}$, where $\underline{k}$ will be fixed in the end, we have:

$$kz_j = k_l z_j = \underline{k} z_j + \frac{2\pi l}{m} j.$$

Substituting this into (19) yields

$$\begin{aligned} &C_0(k_l) \approx \\ &\frac{e^{-\alpha k_l}}{\pi} \Re e \left[\sum_{j=0}^{m-1} e^{-2\pi i \frac{jl}{m}} e^{-i\underline{k} z_j} \mathcal{F}C_0^\alpha(z_j) w_j \right], \ 0 \le l \le m-1. \end{aligned} \tag{20}$$

The discrete Fourier transform $(F\varphi_l)_{0 \le l \le m-1}$ of a vector $\varphi = (\varphi_j)_{0 \le j \le m-1}$ is given by

$$F\varphi_l = \sum_{j=0}^{m-1} e^{-2\pi i \frac{jl}{m}} \varphi_j,\ 0 \le l \le m-1. \tag{21}$$

In (20), we thus recognize the discrete Fourier transform of

$$\varphi = \left(e^{-i\underline{k} z_j} \mathcal{F}C_0^\alpha(z_j) w_j \right)_{0 \le j \le m-1}.$$

Choosing $\underline{k} = \ln(S_0) + \kappa T - \frac{\pi}{h}$, so that

$$k_l = \ln(S_0) + \kappa T - \frac{\pi}{h} + \frac{2\pi}{h}\frac{l}{m}\,,\ l = 0, \dots, m-1,$$

we can thus price a call for m values of the strike K around the T forward value of the stock at time 0, $F_0^T = S_0 e^{\kappa T}$, by computing the discrete Fourier transform of φ. For m given as a power of 2, this is achieved in time $\mathrm{O}(m \ln m)$ by the recursive FFT algorithm.

G From Theory to Practice

G.1 Risk-Neutral Modeling

Throughout these notes we mostly work under a risk-neutral measure $\mathbb{Q}$, rather than under the physical measure $\mathbb{P}$. This doesn't mean that analyzing a financial market under the physical measure $\mathbb{P}$ is not important; such analysis must actually come first in the modeling process. It means simply that this task has already been done. We thus take a pricing model for granted, one that we suppose gives a realistic view of the financial market under consideration, up to an equivalent change of measure.

[18] Simpson's integration rule gives a good accuracy for a relatively small value of m.

G.2 Change of Numéraire

The above risk-neutral modeling approach can be readily extended to a martingale modeling approach with respect to an arbitrary numéraire, rather than the risk-free asset in the risk-neutral approach. This is particularly useful for dealing with interest rate derivatives[19] and foreign exchange derivatives[20].

Let thus be given a numéraire, in the form of a non-dividend-paying reference asset with $\mathbb{Q}$ price process $\widetilde{B}$ such that $\beta\widetilde{B}$ is a positive $\mathbb{Q}$ martingale. Let $\widetilde{\mathbb{Q}}$ be the pricing measure on $(\Omega, \mathcal{A})$, associated with the numéraire $\widetilde{B}$, such that

$$\frac{d\widetilde{\mathbb{Q}}}{d\mathbb{Q}}\Big|_{\mathfrak{F}_T} = \nu_T, \text{ with } \nu_t = \frac{\beta_t \widetilde{B}_t}{\widetilde{B}_0}. \tag{22}$$

The self-financing 0.(3) and (resp. $\mathbb{Q}$) admissibility conditions $\beta V \geq -c$ 0.(4) (resp. $\beta V \geq$ some $\mathbb{Q}$ martingale M (5)) can then be replaced by

$$\begin{aligned} &\widetilde{\beta}V = \pi + \int_0^{\cdot} \zeta_t^0 d(\widetilde{\beta}S^0)_t + \int_0^{\cdot} \zeta_t d(\widetilde{\beta}S)_t + \int_0^{\cdot} \zeta_t \widetilde{\beta}_t d\mathcal{D}_t \text{ and} \\ &\widetilde{\beta}V \geq -c \text{ (resp. } \widetilde{\beta}V \geq \text{some } \widetilde{\mathbb{Q}} \text{ martingale } \widetilde{M}), \end{aligned} \tag{23}$$

where $\widetilde{\beta} = 1/\widetilde{B}$. In fact, as established in Huang (1985, Proposition 4.2 and proof in Appendix II)[21], the self-financing condition (23) is invariant by change of numéraire. See Proposition II.1 below for a proof in a diffusive setup, where a suitably relaxed admissibility condition is also numéraire invariant.

By Lemma IX.7, a process $\widetilde{X}$ is a $\widetilde{\mathbb{Q}}$ martingale if and only if $\nu\widetilde{X}$ is a $\mathbb{Q}$ martingale. Considering a European option with payoff ξ at time T and $\mathbb{Q}$ price process Π, we have in particular the following $\mathbb{Q}$ martingale:

$$\beta\Pi = \nu \frac{\widetilde{B}_0 \Pi}{\widetilde{B}}. \tag{24}$$

So $\frac{\Pi}{\widetilde{B}}$ is a $\widetilde{\mathbb{Q}}$ martingale and we have, for $t \in [0, T]$,

$$\Pi_t = \widetilde{B}_t \, \widetilde{\mathbb{E}}_t(\widetilde{B}_T^{-1}\xi), \tag{25}$$

where $\widetilde{\mathbb{E}}_t$ denotes the $(\mathfrak{F}_t, \widetilde{\mathbb{Q}})$ conditional expectation.

A change-of-numéraire pricing approach can also be devised for American options.

G.3 Model Calibration

In applications we can think of $\mathbb{Q}$ as "the pricing measure chosen by the market" to price a contingent claim. For hedging purposes or in order to implement bets on specific risk factors, and also for pricing exotic or structured products, traders need to know the market pricing measure $\mathbb{Q}$.

In practice, the measure $\mathbb{Q}$ is typically estimated by calibration of a model to market data. Indeed there are two sets of constraints that the market pricing measure $\mathbb{Q}$ must satisfy. First, $\mathbb{Q}$ must satisfy structural requirements stemming from its equivalence with the physical probability measure $\mathbb{P}$. Any process must thus have the same trajectorial properties (such as continuity or lack of it) under the objective and under an equivalent pricing measure. Second, the cross-section $\Pi_t^{[\pm]}(T, K)$ of the market prices of European vanilla calls and puts quoted at any pricing time t on an underlying S must satisfy

$$\Pi_t^{[\pm]}(T, K) = \beta_t^{-1}\mathbb{E}_t\beta_T(S_T - K)^{\pm}, \ (T, K) \in \text{obs}_t, \tag{26}$$

where obs_t is the set of the most liquid options on S (European vanilla at-the-money or slightly out-of-the-money calls and puts) quoted in the market at time t.

[19] see I.II.§2 and Andersen and Piterbarg (2010); Brigo and Mercurio (2007).
[20] see Lipton (2002).
[21] see also Protter (2001, Theorems on page 184).

Constraints of type (26) are called calibration constraints. A model is said to fit the market smile, at a given time t, if it satisfies the calibration constraints (26) (cf. Chapter VII). Accounting also for synchronization and noise issues in the market data, one commonly relaxes the calibration equality constraints (26) into inequality constraints within the bid-ask spread. Quite a few classes of models can fit the smile within the bid-ask spread, provided their parameters are suitably calibrated. A further requirement can be to fit the smile dynamics priced by the market. This corresponds to additional calibration constraints associated with market prices of exotic options[22].

Finally, for being usable in practice, a pricing model needs to be constructive and implementable in real time. Concretely this leads to work with a low-dimensional "Markovian proxy" X for the "true" (if any) market factor process. However, with the emergence of machine learning based pricing and calibration procedures (Horvath, Muguruza, and Tomas, 2021), nowadays this low-dimensional requirement becomes less stringent.

G.4 Imperfect Hedging

A model calibrated to the market can be used for hedging purposes, and for dealing with more exotic products. When a bank sells a derivative, it immediately sets up a hedge composed of liquid instruments, such as the asset(s) underlying the derivative, and/or further vanilla derivatives. But for feasibility, as well as for transaction cost issues (ignored in our theoretical setup), the bank is restricted to piecewise constant hedging strategies ζ^h such that

$$\zeta^h_t = \zeta^h_{t_i} \text{ for } t_i < t \leq t_{i+1}, \tag{27}$$

where $(t_i)_{0\leq i\leq n}$ is a time-grid over $[0,T]$. In practice n may vary from one (static hedging) to the number of days or weeks between 0 and T . Since derivative payoffs are typically nonlinear, in order to get a good hedge the composition of the hedging portfolio must be updated at a high enough frequency.

In an idealized, complete market model, a continuously rebalanced hedge ζ provides a perfect hedge to an option's seller (profit-and-loss identically equal to 0 or, in the case of an American option, nonnegative and even positive in case of sub-optimal exercise by the option holder). By contrast, in the real world there are many reasons why a practical strategy ζ^h typically leads to actual profits or losses under some scenarios:

- Hedge slippage as said above, but this is only the tip of the iceberg:
- Transaction and illiquidity costs, which are ignored in our formalism, and, above all perhaps:
- Model misspecification: note that hedging ratios are typically model-dependent, even among models calibrated to the same data set.

§2 Black-Scholes and Dupire

Consistent with the risk-neutral drift condition (1), the Black and Scholes (1973); Merton (1973) model postulates the following diffusion for S, under a (risk-neutral[23]) probability measure $\mathbb{Q} \sim$ the physical one:

$$dS_t = S_t(\kappa dt + \sigma dW_t), \tag{28}$$

for a standard $\mathbb{Q}$ Brownian motion W and a constant volatility parameter σ. Or, explicitly:

$$S_t = S_0 e^{bt+\sigma W_t}, \tag{29}$$

[22] see Remark VII.2.
[23] By 0.(2).

where $b = \kappa - \frac{1}{2}\sigma^2$. As one can prove by a direct computation (or an application of the Novikov criterion) based on (29), the process αS is a $\mathbb{Q}$ martingale, as postulated in Section §1.

Equivalent to (28)-(29), we have:

$$dF_t^T = \sigma F_t^T dW_t, \tag{30}$$

where F_t^T is the T-forward time-t value of S in (5). The T-forward price F_t^T is thus a $\mathbb{Q}$ Brownian martingale with constant volatility σ. In terms of the cumulative stock price $\widehat{S}$, from 0.(2) we get

$$d(\beta_t \widehat{S}_t) = \beta_t \sigma S_t dW_t. \tag{31}$$

By the Markov property of the above process S[24], we have that

$$\mathbb{E}_t \phi(S_T) = \mathbb{E}\left(\phi(S_T) \,|\, S_t\right)$$

and therefore $\Pi_t = v(t, S_t)$ for a measurable pricing function v. Assuming that v is sufficiently regular, an application of the Itô formula yields:

$$e^{rt} d(e^{-rt} v(t, S_t)) = (\partial_t v + \mathcal{A}_S^{bs} v - rv)(t, S_t) dt + \sigma S_t \partial_S v(t, S_t) dW_t, \tag{32}$$

where $\mathcal{A}_S^{bs} = \kappa S \partial_S + \frac{\sigma^2 S^2}{2} \partial_{S^2}^2$. Since $e^{-rt} v(t, S_t) = e^{-rt} \Pi_t$ is a martingale, one can show by application of Lemma IX.6 that

$$\partial_t v + \mathcal{A}_S^{bs} v - rv(t, S_t) = 0\,, \ t \leq T.$$

Accounting for the terminal condition $\Pi_T = \phi(S_T)$, this suggests that the following Black-Scholes pricing PDE holds:

$$\begin{cases} v(T, S) = \phi(S), \ S \in (0, +\infty) \\ \partial_t v + \kappa S \partial_S v + \frac{1}{2}\sigma^2 S^2 \partial_{S^2}^2 v - rv = 0 \text{ in } [0, T) \times (0, +\infty). \end{cases} \tag{33}$$

In fact, for ϕ continuous with polynomial growth in S, the PDE (33) is known from Friedman (1983) to have a unique classical solution v in $\mathcal{C}^{1,2}\big([0, T) \times (0, +\infty)\big) \cap \mathcal{C}^0\big([0, T] \times (0, +\infty)\big)$ with polynomial growth in S (uniformly in $t \in [0, T]$).

Theorem 1 *Assuming the payoff function ϕ continuous with polynomial growth in S, the unique classical solution v with polynomial growth in S to (33) is a continuous*[25] *$\mathbb{Q}$ pricing function for the option. The hedging strategy defined, for $t \in [0, T]$, by*

$$\zeta_t^{bs} = \partial_S v(t, S_t) \tag{34}$$

units of stock S and

$$\beta_t \left(v(t, S_t) - S_t \partial_S v(t, S_t)\right) \text{ units of the riskless asset } \beta^{-1}, \tag{35}$$

is self-financing, $\mathbb{Q}$ admissible, and it replicates the option payoff $\phi(S_T)$, starting from the wealth $v(0, S_0) = \Pi_0$ at time 0.

Proof. An application of the Itô formula[26] to the solution v of (33) yields

$$\beta_T \phi(S_T) = \beta_t v(t, S_t) + \int_t^T \beta_s \partial_S v(s, S_s) \sigma S_s dW_s, \tag{36}$$

where, by (31),

$$\beta_s \partial_S v(s, S_s) \sigma S_s dW_s = \partial_S v(s, S_s) d(\beta_s \widehat{S}_s).$$

[24] see IX.§3.
[25] as can be shown: the unique continuous.
[26] on $[t, T - \epsilon]$, then taking limits on both sides of the resulting identity as $\epsilon \to 0$.

Thus

$$\beta_T\phi(S_T) = \beta_t v(t,S_t) + \int_t^T \partial_S v(s,S_s) d(\beta_s \widehat{S}_s) \tag{37}$$

and, in particular,

$$\beta_T\phi(S_T) = v(0,S_0) + \int_0^T \partial_S v(s,S_s) d(\beta_s \widehat{S}_s). \tag{38}$$

The process

$$v(0,S_0) + \int_0^{\cdot} \partial_S v(s,S_s) d(\beta_s \widehat{S}_s) = \beta_t v(t,S_t)$$

(by difference between (37) and (38)) is a $\mathbb{Q}$ local martingale, by Lemma IX.8, sandwiched between two $\mathbb{Q}$ martingales of the form $(\pm ce^{-kt}S_t^p)$ for suitable constants p,k,c, by the polynomial growth condition on v. Hence this process is itself a $\mathbb{Q}$ martingale, by application of Lemma IX.5(ii). So $\Pi_t = v(t,S_t)$ follows by taking conditional expectations in (37). Hence we have a $\mathbb{Q}$ replication strategy as stated in the theorem. ∎

Note that the above covers the special cases of European vanilla call and put options.

Remark 3 *If ϕ is bounded fom below, like with European vanilla put short or long (but call short only) positions, then so are the $\mathbb{Q}$ pricing function and the value of the replicating portfolio, hence replicability holds (not only $\mathbb{Q}$ replicability).*

Remark 4 *In the Black–Scholes model, ($\mathbb{Q}$, at least) replicability of more general, possibly path-dependent, European claims ξ, not necessarily of the form $\phi(S_T)$ but assumed $\mathbb{Q}$ square integrable, can be established based on the Brownian martingale representation theorem. This replicability of European claims in the Black-Scholes model explains why the Black-Scholes PDE (33) (hence, the Black-Scholes prices) doesn't depend on the physical drift of S, even though the latter may be responsible for fat tails or skewness in the physical returns of S.*

Remark 5 *Postulating, instead of (28), a physical model for S, namely the dynamics (28) with κ replaced by a constant μ there and W by a Brownian motion $\widehat{W}$ under $\mathbb{P}$, i.e.*

$$dS_t = S_t(\mu dt + \sigma d\widehat{W}_t),$$

then the Girsanov theorem allows showing that the probability measure $\mathbb{Q}$ defined through

$$\frac{d\mathbb{Q}}{d\mathbb{P}} = \exp\big(-\lambda\widehat{W}_T - \frac{1}{2}\lambda^2 T\big) \text{ , where } \lambda = \frac{\mu-\kappa}{\sigma},$$

is the unique risk-neutral measure[27] *on the primary market made of the stock S and the risk-free asset $e^{r\cdot}$.*

A Black-Scholes Formulas

Let $\tau = T-t$, $\mathcal{N}$ and n denote the standard Gaussian cumulative distribution and density functions, and

$$\mathrm{d}_\pm = \mathrm{d}_\pm(t,S,T,K;r,q,\sigma) = \frac{\ln(\frac{S}{K}) + \kappa\tau}{\sigma\sqrt{\tau}} \pm \frac{1}{2}\sigma\sqrt{\tau}. \tag{39}$$

The argument $(t,S,T,K;r,q,\sigma)$ will be abbreviated, when no confusion might arise, by (t,S,T,K) or (t,S), or it may sometimes even be omitted (see Table 1).

[27] namely, the one that we directly postulated under the formulation (28) of the model.

K/S	$\mathrm{d}_\pm$	$n(d_\pm)$	$\mathcal{N}(d_\pm)$
0	$+\infty$	0	1
$+\infty$	$-\infty$	0	0

σ	$\mathrm{d}_\pm$	$n(d_\pm)$	$\mathcal{N}(d_\pm)$
0	$+\infty\mathbb{1}_{\mathrm{ITMF}} - \infty\mathbb{1}_{\mathrm{OTMF}}$	$\frac{1}{\sqrt{2\pi}}\mathbb{1}_{\mathrm{ATMF}}$	$\mathbb{1}_{\mathrm{ITMF}} + \frac{1}{2}\mathbb{1}_{\mathrm{ATMF}}$
$+\infty$	$\pm\infty$	0	1/0

Table 1: *Asymptotics of* $\mathrm{d}_\pm, n(d_\pm)$, *and* $\mathcal{N}(d_\pm)$ *with respect to* K/S *and* σ. *We write* ITMF, ATMF *and* OTMF *for in-the-money forward, at-the-money forward and out-of-the-money forward for a call option, i.e. for the cases* $(S/K)e^{\kappa\tau} > 1, = 1$ *or* < 1.

Lemma 2 *For every measurable function* ϕ *such that* $\xi = \phi(S_T)$ *is* $\mathbb{Q}$ *integrable, it holds, for any fixed* $t \in [0,T]$, *that*

$$\mathbb{E}_t\phi(S_T) = \mathbb{E}\big[\phi(Se^{b\tau+\sigma\sqrt{\tau}\varepsilon})\big]|_{S=S_t},$$

with ε *standard Gaussian.*

Proof. We recall (see e.g. Lamberton and Lapeyre (1996, Proposition A.2.5)) that, if χ is measurable with respect to $\mathcal{B}$ and ξ is independent of a σ-field $\mathcal{B}$, then for every function $\varphi = \varphi(y,z)$ such that $\varphi(\chi,\xi)$ is integrable,

$$\mathbb{E}(\varphi(\chi,\xi)|\mathcal{B}) = \mathbb{E}\varphi(x,\xi)|_{x=\chi}. \tag{40}$$

Hence

$$\begin{aligned} e^{-r(T-t)}\mathbb{E}_t\phi(S_T) &= e^{-r(T-t)}\mathbb{E}\big[\phi(S_te^{b(T-t)+\sigma(W_T-W_t)})\big|\mathfrak{F}_t\big] \\ &= e^{-r\tau}\mathbb{E}\big[\phi(Se^{b\tau+\sigma\sqrt{\tau}\varepsilon})\big]|_{S=S_t} \end{aligned}$$

with ε standard Gaussian, by independence of $W_T - W_t$ with respect to $\mathfrak{F}_t$ (followed by the fact that $W_T - W_t$ equals in law $\sqrt{\tau}\varepsilon$). ∎

Lemma 3 *For* ε *standard Gaussian, we have*

$$\mathbb{E}\big[(Se^{b\tau+\sigma\sqrt{\tau}\varepsilon} - K)^+\big] = Se^{\kappa\tau}\mathcal{N}(\mathrm{d}_+(t,S)) - K\mathcal{N}(\mathrm{d}_-(t,S)).$$

Proof. Decomposing $(Se^{b\tau+\sigma\sqrt{\tau}\varepsilon} - K)^+$ via $(X-K)^+ = (X-K)\mathbb{1}_{X>K}$ yields

$$\mathbb{E}\big[(Se^{b\tau+\sigma\sqrt{\tau}\varepsilon} - K)^+\big] = \mathbb{E}(Se^{b\tau+\sigma\sqrt{\tau}\varepsilon}\mathbb{1}_{Se^{b\tau+\sigma\sqrt{\tau}\varepsilon}>K}) - K\mathbb{Q}\big(Se^{b\tau+\sigma\sqrt{\tau}\varepsilon} > K\big),$$

where

$$\begin{aligned} \mathbb{Q}\big(Se^{b\tau+\sigma\sqrt{\tau}\varepsilon} > K\big) &= \int_{y=\frac{\ln(K/(Se^{\kappa\tau}))}{\sigma\sqrt{\tau}}+\frac{\sigma\sqrt{\tau}}{2}=-\mathrm{d}_-(t,S)}^{+\infty} n(y)dy \\ &= 1 - \mathcal{N}(-\mathrm{d}_-(t,S)) = \mathcal{N}(\mathrm{d}_-(t,S)). \end{aligned}$$

Moreover,

$$\begin{aligned} &\mathbb{E}(Se^{b\tau+\sigma\sqrt{\tau}\varepsilon}\mathbb{1}_{Se^{b\tau+\sigma\sqrt{\tau}\varepsilon}>K}) \\ &= Se^{\kappa\tau}\int_{y=-\mathrm{d}_-(t,S)}^{+\infty} e^{\sigma\sqrt{\tau}y-\frac{1}{2}\sigma^2\tau}e^{-y^2/2}\frac{dy}{\sqrt{2\pi}} \\ &= Se^{\kappa\tau}\int_{y=-\mathrm{d}_-(t,S)}^{+\infty} e^{-(y-\sigma\sqrt{\tau})^2/2}\frac{dy}{\sqrt{2\pi}} \\ &= Se^{\kappa\tau}\int_{z=-\mathrm{d}_-(t,S)-\sigma\sqrt{\tau}=-\mathrm{d}_+(t,S)}^{+\infty} n(z)dz \\ &= Se^{\kappa\tau}\big(1-\mathcal{N}(-\mathrm{d}_+(t,S))\big) = Se^{-q\tau}\mathcal{N}(\mathrm{d}_+(t,S)). \end{aligned}$$ ∎

Proposition 9 *At time t with stock worth S, the Black-Scholes call price C^{bs}, delta $\Delta^{bs} = \partial_S C^{bs}$, gamma $\Gamma^{bs} = \partial^2_{S^2} C^{bs}$, theta $\Theta^{bs} = -\partial_\tau C^{bs}$, vega $\mathcal{V}^{bs} = \partial_\sigma C^{bs}$, and rho $\mathrm{P}^{bs} = \partial_r C^{bs}$ are given by:*

$$\begin{aligned}
&C^{bs}(t,S,T,K;r,q,\sigma) = Se^{-q\tau}\mathcal{N}(\mathrm{d}_+) - Ke^{-r\tau}\mathcal{N}(\mathrm{d}_-)\\
&\Delta^{bs}(t,S,T,K;r,q,\sigma) = e^{-q\tau}\mathcal{N}(\mathrm{d}_+),\\
&\Gamma^{bs}(t,S,T,K;r,q,\sigma) = e^{-q\tau}\frac{n(\mathrm{d}_+)}{S\sigma\sqrt{\tau}},\\
&\Theta^{bs} = qSe^{-q\tau}\mathcal{N}(\mathrm{d}_+) - rKe^{-r\tau}\mathcal{N}(\mathrm{d}_-) - Se^{-q\tau}n(\mathrm{d}_+)\frac{\sigma}{2\sqrt{\tau}},\\
&\mathcal{V}^{bs}(t,S,T,K;r,q,\sigma) = Se^{-q\tau}\sqrt{\tau}n(\mathrm{d}_+) = S^2\sigma\tau\Gamma^{bs}(t,S,T,K;r,q,\sigma),\\
&\mathrm{P}^{bs}(t,S,T,K;r,q,\sigma) = \tau Ke^{-r\tau}\mathcal{N}(\mathrm{d}_-).
\end{aligned} \tag{41}$$

Proof. The formula for the price follows from Lemmas 2 (applied to $\phi(S) = (S-K)^+$) and 3. The formula for the Greeks follow by differentiation of the price, using the identities

$$Se^{-q\tau}n(\mathrm{d}_+) = Ke^{-r\tau}n(\mathrm{d}_-) \text{ and } \mathrm{d}_+ - \mathrm{d}_- = \sigma\sqrt{\tau}. \tag{42}$$

For instance,

$$\begin{aligned}
\partial_\tau C^{bs} &= \partial_\tau\big(Se^{-q\tau}\mathcal{N}(\mathrm{d}_+) - Ke^{-r\tau}\mathcal{N}(\mathrm{d}_-)\big)\\
&= Se^{-q\tau}\big(n(\mathrm{d}_+)\partial_\tau\mathrm{d}_+ - q\mathcal{N}(\mathrm{d}_+)\big) - Ke^{-r\tau}\big(n(\mathrm{d}_-)\partial_\tau\mathrm{d}_- - r\mathcal{N}(\mathrm{d}_-)\big)\\
&= Se^{-q\tau}n(\mathrm{d}_+)\frac{\sigma}{2\sqrt{\tau}} - qSe^{-q\tau}\mathcal{N}(\mathrm{d}_+) + rKe^{-r\tau}\mathcal{N}(\mathrm{d}_-),
\end{aligned}$$

which yields the formula for Θ^{bs}. ∎

Sanity check that the obtained price and Greeks satisfy the Black-Scholes PDE (33):

$$\begin{aligned}
&\Theta^{bs} + \kappa S\Delta^{bs} + \frac{1}{2}\sigma^2S^2\Gamma^{bs} - rC^{bs} =\\
&\quad qSe^{-q\tau}\mathcal{N}(\mathrm{d}_+) - rKe^{-r\tau}\mathcal{N}(\mathrm{d}_-) - Se^{-q\tau}n(\mathrm{d}_+)\frac{\sigma}{2\sqrt{\tau}}\\
&\quad + \kappa Se^{-q\tau}\mathcal{N}(\mathrm{d}_+) + \frac{1}{2}\sigma^2S^2e^{-q\tau}\frac{n(\mathrm{d}_+)}{S\sigma\sqrt{\tau}} - r\big(Se^{-q\tau}\mathcal{N}(\mathrm{d}_+) - Ke^{-r\tau}\mathcal{N}(\mathrm{d}_-)\big) =\\
&\quad - Se^{-q\tau}n(\mathrm{d}_+)\frac{\sigma}{2\sqrt{\tau}} + \frac{1}{2}\sigma^2S^2e^{-q\tau}\frac{n(\mathrm{d}_+)}{S\sigma\sqrt{\tau}} = 0. \blacksquare
\end{aligned} \tag{43}$$

In the case of a put option, we have the corresponding formulas and results deduced by call-put parity (4) hence the price

$$P^{bs}_t(T,K) = Ke^{-r\tau}\mathcal{N}(-\mathrm{d}_-) - Se^{-q\tau}\mathcal{N}(-\mathrm{d}_+), \tag{44}$$

the delta $(-e^{-q\tau}\mathcal{N}(-\mathrm{d}_+))$, and the same gamma and vega as the call of same characteristics.

Remark 6 **(i)** *P^{bs} and C^{bs} only depend on t and T through their difference τ.*
(ii) *Exchanging S and K and r and q in the Black–Scholes formula for calls yields the Black–Scholes formula for puts.*

Δ^{bs} and Γ^{bs} assess the sensitivity of the call/put price to small, respectively large, movements of the stock S.

Corollary 1 *Call and put prices are convex in S and (differentiable and) increasing in σ.*

Proof. By positivity of Γ^{bs} and $\mathcal{V}^{bs}$. ∎

Remark 7 *Γ^{bs} and $\mathcal{V}^{bs}$, like $n(\mathrm{d}_+)$, are maximum in K when d_+ vanishes, i.e. for $K = Se^{\kappa\tau+\frac{1}{2}\sigma^2\tau}$; Γ^{bs} is mostly significant for close-to-the money option with small time-to-maturity; $\mathcal{V}^{bs}$ is mostly significant for close-to-the money option with large time-to-maturity.*

Using the identity

$$n'(y) = -yn(y), \tag{45}$$

further formulas can be obtained for other second order sensitivities, i.e. the call/put

$$\begin{aligned} &\text{charm} = -\partial^{bs}_{\tau}\Delta^{bs} = \partial_S\Theta^{bs} = -\partial^2_{\tau,S}\Pi^{bs} \\ &\text{volga} = \partial^2_{\sigma^2}C^{bs}\,,\ \text{vanna} = \partial^2_{S,\sigma}C^{bs}. \end{aligned} \tag{46}$$

See Figures 2, 3 and play with the original python code by Clint Howard on
https://clinthoward.github.io/portfolio/2017/04/16/BlackScholesGreeks

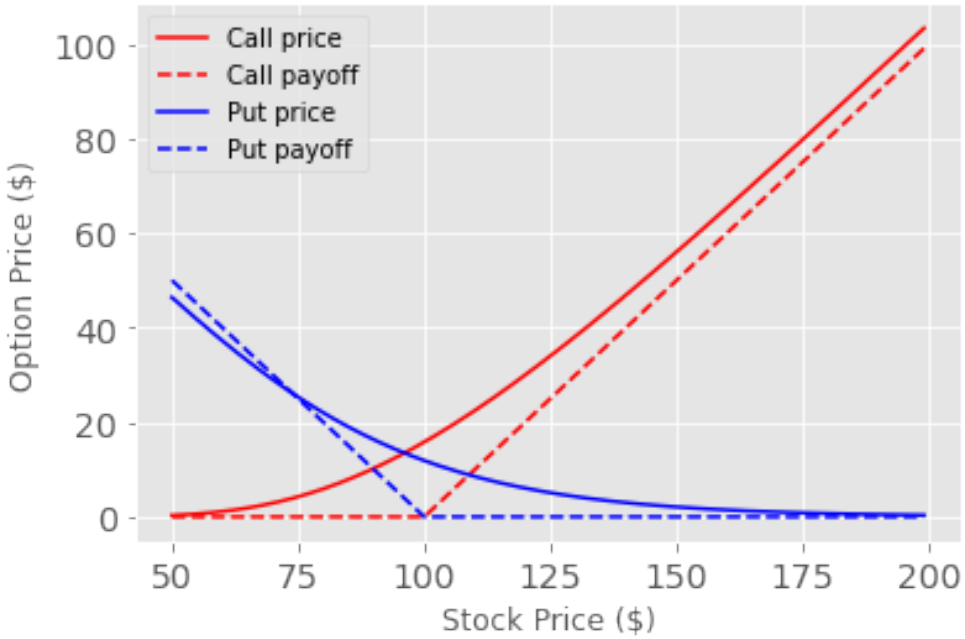


Figure 2: *Option price sensitivity to stock price.*

In view of the last identity in (41):

Corollary 2 *A portfolio of European vanilla options of the same maturity is gamma neutral if and only if it is vega neutral.*

All the above formulas and results admit straightforward extensions to the case where r, q and σ are time-integrable functions: simply replace $r\tau$, $q\tau$ and $\sigma\sqrt{\tau}$ in all the computations and results above by, respectively, $\int_t^T r(s)ds$, $\int_t^T q(s)ds$, and $\left(\int_t^T \sigma^2(s)ds\right)^{\frac{1}{2}}$.

A.1 Black Fomulas

Given the particular importance of the Black-Scholes model with zero risk-free and dividend rates, or Black model, for modeling forward prices or rates, we introduce the following additional notation, where $\sigma = \sigma(\cdot)$ is a deterministic volatility function:

$$\begin{aligned} &c^{bl}(t,F,T,K;\sigma) = F\mathcal{N}(\mathrm{d}^{bl}_+) - K\mathcal{N}(\mathrm{d}^{bl}_-), \\ &\delta^{bl}(t,F,T,K;\sigma) = \partial_F c^{bl}(t,F,T,K;\sigma) = \mathcal{N}(\mathrm{d}^{bl}_+), \end{aligned} \tag{47}$$

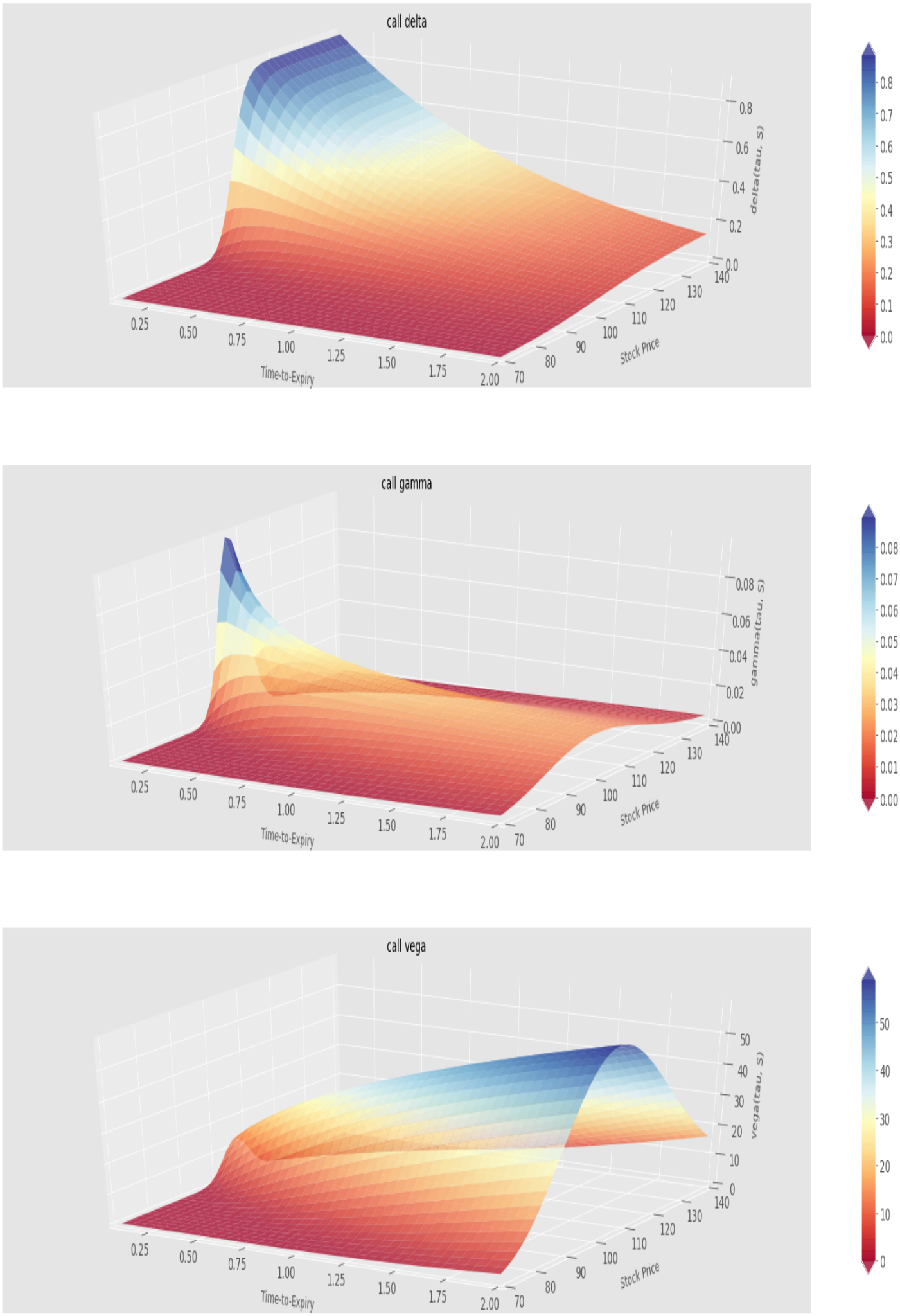


Figure 3: *Greek sensitivities to time-to-expiry and stock price.*

where

$$\mathrm{d}_{\pm}^{bl} = \frac{\ln(F/K)}{\Sigma\sqrt{\tau}} \pm \frac{1}{2}\Sigma\sqrt{\tau}, \text{ with } \Sigma^2\tau = \int_t^T \sigma^2(s)ds. \tag{48}$$

Again the argument $(t, F, T, K; \sigma)$ will be abbreviated, when no confusion can arise, by (t, F, T, K) or (t, F) or, sometimes even (as in $\mathrm{d}_{\pm}^{bl}$ above), omitted.

Equity calls with stochastic interest rates In practice interest rates are stochastic and accounting for their stochasticity is of course key for handling interest rate derivatives[28], but also equity derivatives with long maturities, such as the ones that may be embedded in convertible bonds or other structured products. The above Black formulas can be used for extending the Black-Scholes formulas to equity call options in an economy with a stochastic (bounded from below and time-integrable) risk-free short rate process r—still assuming a deterministic, time-integrable dividend yield $q(\cdot)$ on S.

This is now done in the setup of a martingale pricing model with respect to the numéraire defined by the $\mathbb{Q}$ price process B^T of a T discount bond[29]. We let $\beta_t^T = \frac{1}{B_t^T}$ and we call T-forward-neutral the measure $\mathbb{Q}^T$ associated with the numéraire B^T. Since $B_T^T = 1$, the general martingale measure pricing formula (25) applied with $\widetilde{B} = B^T$ and $\widetilde{\mathbb{Q}} = \mathbb{Q}^T$ yields, for ξ integrable under $\mathbb{Q}^T$:

$$\Pi_t = B_t^T \, \mathbb{E}_t^{\mathbb{Q}^T} \xi. \tag{49}$$

For guaranteeing no arbitrage in our model as per Corollay 0.1, whilst ensuring the $\mathbb{Q}$ integrability of αS with the motivation explained in Remark 1, we want to model the process $\alpha S = \beta S e^{\int_0^{\cdot} q(s)ds}$ as a $\mathbb{Q}$ martingale. This leads to the following formulas for the time-t $\mathbb{Q}$ price $F_t(T, \mathcal{K})$ of a $(T, \mathcal{K})$ forward contract on S (with payoff $S_T - \mathcal{K}$ at T), hence for the time-t value of the T-forward price of S:

$$F_t(T, \mathcal{K}) = S_t e^{-\int_t^T q(s)ds} - \mathcal{K} B_t^T, \tag{50}$$

$$F_t^T = \beta_t^T S_t e^{-\int_t^T q(s)ds}. \tag{51}$$

By Lemma IX.7, the above $\mathbb{Q}$ martingale condition on $\beta S e^{\int_0^{\cdot} q(s)ds}$, i.e. on $\beta S e^{-\int_{\cdot}^T q(s)ds}$, is equivalent to a $\mathbb{Q}^T$ martingale condition on the process $\beta^T S e^{-\int_{\cdot}^T q(s)ds} = F^T$, by (51). Consistent with this requirement, we postulate a Black model, i.e. a Black-Scholes model with zero rates and deterministic volatility $\sigma(\cdot)$, for F^T under $\mathbb{Q}^T$. In the case of a call option, substituting $\xi = (S_T - K)^+ = (F_T^T - K)^+$ into (49), we obtain[30]

$$\Pi_t = C_t^{bl} := B_t^T c^{bl}(t, F_t^T, T, K; \sigma). \tag{52}$$

Proposition 10 *In a primary market defined by the numéraire (T discount bond) B^T, which is used as funding asset, along with a T-forward contract on S with strike $\mathcal{K}$, a replication strategy for the European call option with payoff $\xi = (S_T - K)^+$ at T, with initial wealth C_0^{bl} at time 0, is given, for $t \in [0, T)$, by*

$$\widetilde{\zeta}_t^{bl} = \delta^{bl}(t, F_t^T)$$

forward contracts on S and

$$-K\mathcal{N}(d_-^{bl}(t, F_t^T)) + \delta^{bl}(t, F_t^T)\mathcal{K}$$

units of the T discount bond.

[28] cf. II.§2.
[29] paying €1 at T.
[30] cf. (47).

Proof. In view of (50)-(51), the B^T relative price process of the $(T,\mathcal{K})$-forward contract on S is given, for $t\in[0,T]$, by:

$$\beta_t^T F_t(T,\mathcal{K}) = \beta_t^T\left(S_t e^{-\int_t^T q(s)ds} - \mathcal{K}B_t^T\right) = F_t^T - \mathcal{K}. \tag{53}$$

By (23), the B^T relative wealth process of a self-financing strategy $\widetilde{\zeta}$ in the forward contract $F(T,\mathcal{K})$ (and the funding asset B^T) satisfies

$$d(\beta^T V)_t = \widetilde{\zeta}_t d(\beta^T F(T,\mathcal{K}))_t = \widetilde{\zeta}_t dF_t^T, \tag{54}$$

by (53).

Given the Black model postulated on F^T under $\mathbb{Q}^T$, an application of Theorem I.1 with $r=q=0$ there yields that, for $\pi = C_0^{bl}$ (hence $\beta_0^T\pi = c^{bl}(0,F_0^T,T,K;\sigma)$) and $\widetilde{\zeta}=\widetilde{\zeta}^{bl}$ forward contracts on S, we have

$$V_T = \beta_T^T V_T = (F_T^T - K)^+ = (S_T - K)^+.$$

The budget condition implies that the number of T-bonds at time t is then

$$\begin{aligned}
&\beta_t^T\left(C_t^{bl} - \widetilde{\zeta}^{bl}B_t^T(F_t^T-\mathcal{K})\right)\\
&\quad=\beta_t^T\left(B_t^T c^{bl}(t,F_t^T) - \delta^{bl}(t,F_t^T)B_t^T(F_t^T-\mathcal{K})\right)\\
&\quad=F_t^T\mathcal{N}(\mathrm{d}_+^{bl}(t,F_t^T)) - K\mathcal{N}(\mathrm{d}_-^{bl}(t,F_t^T)) - \delta^{bl}(t,F_t^T)(F_t^T-\mathcal{K})\\
&\quad=-K\mathcal{N}(d_-^{bl}(t,F_t^T)) + \delta^{bl}(t,F_t^T)\mathcal{K},
\end{aligned}$$

by (47). ∎

Proposition 11 *In a primary market defined by the T discount bond B^T, which is used as funding asset, and the stock $S = B^T F^T e^{\int_\cdot^T q(s)ds}$, a replication strategy with initial wealth C_0^{bl} for the European call option is defined, for every time $t\in[0,T)$, by*

$$\zeta_t = \zeta_t^{bl} := e^{-\int_t^T q(s)ds}\delta^{bl}(t,F_t^T)$$

units of S and $(-K\mathcal{N}(\mathrm{d}_-^{bl}(t,F_t^T)))$ units of the T discount bond.

Proof. By (23), the B^T relative wealth process of a self-financing strategy ζ in S satisfies

$$d(\beta^T V)_t = \zeta_t d(\beta^T S)_t + \zeta_t\beta_t^T q(t)S_t dt, \tag{55}$$

where

$$\begin{aligned}
d(\beta^T S)_t + \beta_t^T q(t)S_t dt &= e^{-\int_0^t q(s)ds} d(\beta^T S e^{\int_0^\cdot q(s)ds})_t\\
&= e^{-\int_0^t q(s)ds} d(\beta^T S e^{\int_0^\cdot q(s)ds})_t e^{\pm\int_0^T q(s)ds} = e^{\int_t^T q(s)ds} d(\beta^T S e^{-\int_\cdot^T q(s)ds})_t.
\end{aligned}$$

Hence, by (51),

$$d(\beta^T V)_t = \zeta_t e^{\int_t^T q(s)ds} dF_t^T. \tag{56}$$

The proof is then concluded like the one of Proposition 10. This time the number of T bonds is

$$\begin{aligned}
&\beta_t^T(C_t^{bl} - \zeta_t^{bl}S_t)\\
&\quad=\beta_t^T\left(B_t^T c^{bl}(t,F_t^T) - \delta^{bl}(t,F_t^T)e^{-\int_t^T q(s)ds}S_t\right)\\
&\quad=F_t^T\mathcal{N}(\mathrm{d}_+^{bl}(t,F_t^T)) - K\mathcal{N}(\mathrm{d}_-^{bl}(t,F_t^T)) - \delta^{bl}(t,F_t^T)e^{-\int_t^T q(s)ds}S_t\beta_t^T\\
&\quad=F_t^T\mathcal{N}(\mathrm{d}_+^{bl}(t,F_t^T)) - K\mathcal{N}(\mathrm{d}_-^{bl}(t,F_t^T)) - \delta^{bl}(t,F_t^T)F_t^T\\
&\quad=-K\mathcal{N}(\mathrm{d}_-^{bl}(t,F_t^T)).
\end{aligned}$$ ∎

As a consistency check for the above result, note that, in the case of deterministic interest rates with B_t^T given as $e^{-\int_t^T r(s)ds}$ for some function $r(t)$, so that $\frac{\beta B^T}{B_0^T} = 1$ and $\mathbb{Q}^T = \mathbb{Q}$, we have

$$F_t^T = S_t e^{\int_t^T (r(u)-q(u))du}\,,\ \mathrm{d}_\pm^{bl}(t, F_t^T, T, K; \sigma) = \mathrm{d}_\pm^{bs}(t, S_t, T, K; r, q, \sigma)$$

and we obtain

$$\begin{aligned}\zeta_t^{bl} &= e^{-\int_t^T q(s)ds}\mathcal{N}(\mathrm{d}_+^{bl}(t, F_t^T, T, K; \sigma)) = \\ &\quad e^{-\int_t^T q(s)ds}\mathcal{N}(\mathrm{d}_+^{bs}(t, S_t, T, K; r, q, \sigma)) = \zeta_t^{bs}.\end{aligned}$$

Hence the S-component of the replication strategy in the T-forward-neutral Black model with risk factor F^T and primary assets S and B^T is the same as the one of the replication strategy in the Black-Scholes risk-neutral model with risk factor S and primary assets S and S^0.

For a put option all the corresponding formulas and results can be easily deduced from the previous ones by call-put parity.

A.2 Implied Volatility

The Black-Scholes model is strongly misspecified in practice. In fact, the Black–Scholes pricing formulas are essentially used by traders for conveying information about the relative value of different options in the market. The idea is to express prices in a unit of measurement, implied volatility, that is less sensitive to the strike and maturity of an option than its money-value. Black–Scholes formulas are thus effectively used in the reverse-engineering mode for determining, given a European vanilla price observed on the market, the corresponding value of the Black–Scholes volatility consistent with that option price.

Definition 1 *Given values of r and q inferred at time t, from riskless bonds for r and from call-put parity for q, the Black-Scholes implied volatility of a European vanilla (call or put) option is the value Σ_t such that*

$$\Pi^{bs}(t, S_t, T, K; r, q, \Sigma_t) = \Pi_t^\star(T, K), \tag{57}$$

where $\Pi_t^\star(T, K)$ denotes the market price of the option at time t.

The Black–Scholes formulas are then no more than "wrong formulas into which to put a wrong number [the implied volatility of an option] to get the right result [an option market price]".

Lemma 4 (i) *The equation (57) yields a unique Σ_t provided the market price lies within the no-static-arbitrage bounds of the option, i.e.*[31] *in $((Se^{-q\tau} - Ke^{-r\tau})^+, Se^{-q\tau})$ for the call price and in $((Ke^{-r\tau} - Se^{-q\tau})^+, Ke^{-r\tau})$ for the put price.*
(ii) *Within the setup of any $\mathbb{Q}$ (true) martingale model for αS, European calls and puts of same characteristics have the same implied volatility.*

Proof. **(i)** holds because the Black-Scholes price of a vanilla option is differentiable with respect to σ and increases[32] from one arbitrage bound to the other[33] when σ increases from 0 to $+\infty$.

(ii) follows from the call-put parity (4) that holds in any $\mathbb{Q}$ martingale model for αS[34], which includes the Black–Scholes model as special case. ∎

Given the second part in Corollary 1, the equation (57) can be solved numerically by dichotomy. Building on the vega formula in (41), one can also use a Newton–Raphson zero search, i.e. iteratively solve for Σ' the linearized problem $\Pi^{bs} + \mathcal{V}^{bs}(\Sigma' - \Sigma) = \Pi^\star$, where Π^{bs} and $\mathcal{V}^{bs}$ are the Black–Scholes

[31] cf. Proposition 3 and Remark 1.
[32] cf. the positive vega formula in (41).
[33] cf. the second panel in Table 1.
[34] cf. Remark 1.

price and vega corresponding to the current Σ in the algorithm. This is typically faster than a search by dichotomy, but not for market prices $\Pi^\star$ close to the arbitrage bounds, for which the vega sensitivity of the option vanishes.

Proceeding in this way for a range of strikes K and a fixed maturity T, one commonly obtains

- a symmetrical smile on foreign exchange derivative markets,
- a negative skew on equity derivative or markets,
- a smirk on interest rate derivative markets.

See for instance the SPX options smirks on the bottom panel in Figure 4, flattening as the maturity increases.

Implied volatilities tend to be larger than realized volatilities $\widehat{\sigma}$ (statistically estimated standard deviations of stock returns on a time step h divided by $\sqrt{\mathrm{h}}$), because setting up a hedge is more costly in practice than in a theoretical Black–Scholes world with volatility $\widehat{\sigma}$, due to model risk and transaction costs that are all ignored in the Black–Scholes model.

They tend to be negatively skewed at low strikes because a (risk-neutral) Black–Scholes model underestimates the probability of downward jump (or large movements) of a stock or stock index, which traders correct by increasing the price (hence, implied volatility) of far out-of-the-money (OTM) puts. Negative skewness of implied volatilities can be related to a negative skewness, in the sense of a negative third moment, of the risk neutral distribution of stock returns ("$\mathbb{E}(\delta S)^3 < 0$"). Consistent with a jump fear interpretation, the skew is inverted, i.e. a positive implied volatility skew and $\mathbb{E}(\delta S)^3 > 0$ hold in the case of safe haven underliers, such as gold or other negative beta assets in the sense of the capital asset pricing model (CAPM, see the top panel of Figure 4).

Implied volatilities tend to be positively smiled (convex in strike) because of liquidity premia that are ignored in Black–Scholes and larger far from the money. A positive curvature of implied volatilities can be related to an excess kurtosis of the risk neutral distribution of stock returns ("$\mathbb{E}(\delta S)^4 > 3(\mathbb{E}(\delta S)^2)^2$").

The above discussion is summarized by the bottom part of Figure 5, where the upper part refers to further connections guessed in Derman, Kani, and Zou (1996); Derman (1999) and detailed with proofs in Berestycki, Busca, and Florent (2002) between market implied volatilities and the corresponding local volatilities[35].

Black (i.e. lognormal for zero interest and dividend yields) or Bachelier (normal) implied volatilities, denoted hereafter by ${}^{bl}\Sigma$ and ${}^{ba}\Sigma$, are defined similarly to the Black-Scholes implied volatility, in reference to the respective Black formula (47) and to the Bachelier formula (see e.g. Schachermayer and Teichmann (2008)).

B Local Volatility

Proposition 12 *Given $S_t = S$ and Black-Scholes parameters r, q, σ, the function*

$$(T, K) \mapsto C(T, K) = C^{bs}(t, S, T, K; r, q, \sigma) \tag{58}$$

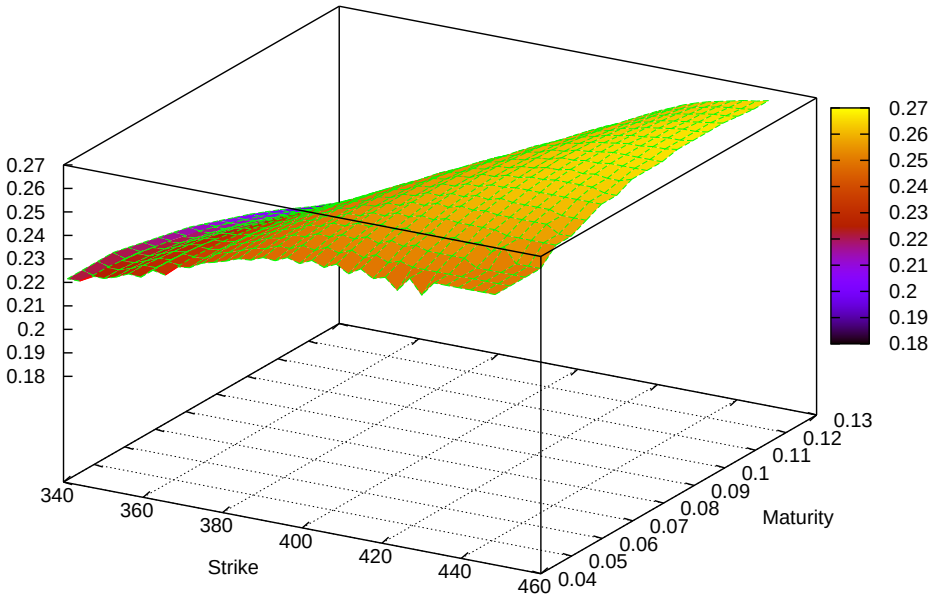
is the unique bounded classical solution to the following Dupire equation:

$$\begin{cases} C(t, K) &= (K - S)^+, \ K > 0 \\ \partial_T C + \kappa K \partial_K C - \frac{1}{2}\sigma^2 K^2 \partial^2_{K^2} C + qC &= 0, \quad T > t, \ K > 0. \end{cases} \tag{59}$$

Proof. In view of Remark 6, we have

$$\partial_t C^{bs}(t, S, T, K; r, q) = -\partial_T C^{bs}(t, S, T, K; r, q)$$

[35] cf. §2.B.



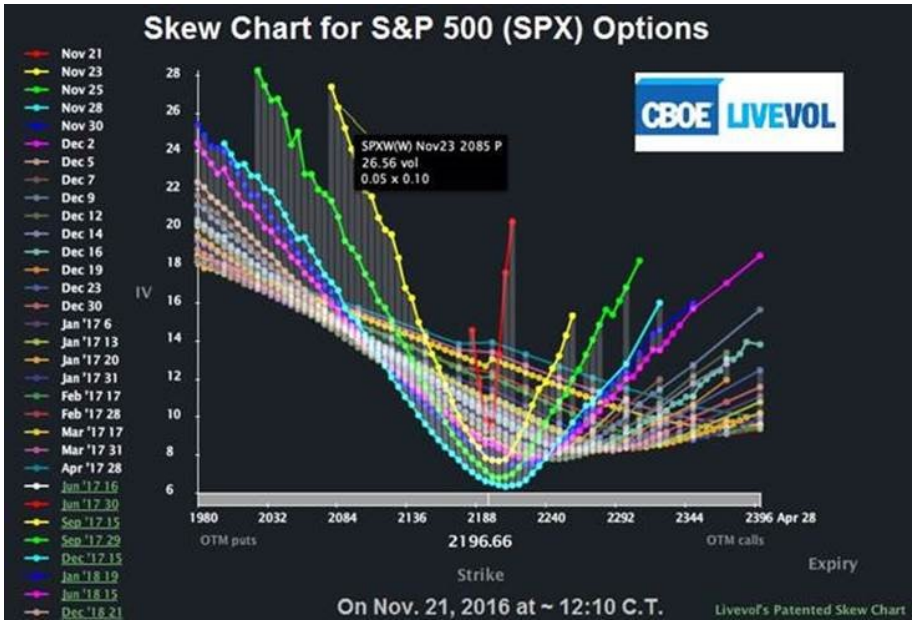


Figure 4: *(Bottom) SPX options smiles and skews; (Top) An implied volatility surface on gold futures [Source: CBOE].*

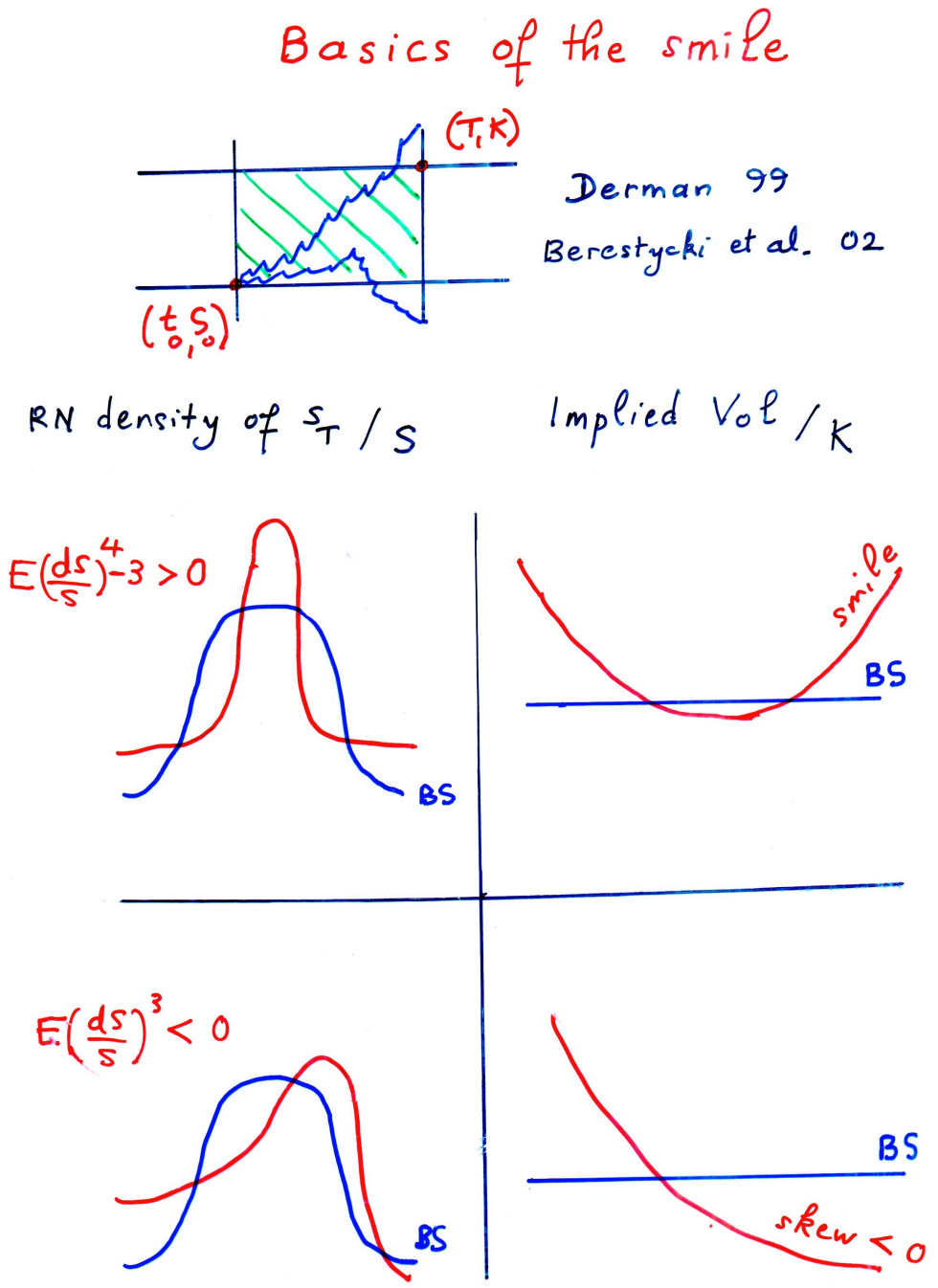


Figure 5: *Risk-neutral density return versus implied volatility patterns.*

and

$$\begin{aligned}&\big(\partial_T + (r-q)K\partial_K - \frac{1}{2}\sigma^2K^2\partial^2_{K^2} + q\big)C^{bs}(t,S,T,K;r,q)\\&\quad= \big(-\partial_t + (r-q)K\partial_K - \frac{1}{2}\sigma^2K^2\partial^2_{K^2} + q\big)P^{bs}(t,K,T,S;q,r)\\&\quad= \big(-\partial_t - (q-r)K\partial_K - \frac{1}{2}\sigma^2K^2\partial^2_{K^2} + q\big)P^{bs}(t,K,T,S;q,r) = 0,\end{aligned}$$

by the pricing equation for the put option with strike S valued at the stock level K, in the Black-Scholes model with interest rate and dividend parameters q and r. This shows the Dupire equation for calls. The fact that this equation admits a unique bounded classical solution follows from the result recalled before Theorem 1. ∎

Corollary 3 *In the Black–Scholes model, the following Dupire formula (or the analogous one for calls with C instead of P) holds:*

$$\frac{(\partial_T + \kappa K\partial_K + q)P}{K^2\partial^2_{K^2}P} = \frac{\sigma^2}{2}. \tag{60}$$

Proof. This is just a reformulation of the Dupire equation for puts, which can be established like the one for calls. ∎

More generally, in any nonarbitrable model with put prices $\mathcal{C}^{1,2}$ in (T,K), the left-hand side in (60) is nonnegative (Roper, 2010). But in general it depends on (t,S,T,K). Namely, the so called local volatility $\sigma(T,K)$, such that the left-hand side in (60) equates $\frac{\sigma(T,K)^2}{2}$, also typically depends on (t,S).

Local volatility models (Dupire, 1994a; Derman and Kani, 1994) correspond to the special case where it does actually not or, equivalently (*admitted*), under a (risk-neutral[36]) probability measure $\mathbb{Q} \sim$ the physical one:

$$dS_t = S_t(\kappa dt + \sigma(t,S_t)dW_t)\,,\ t \geq 0, \tag{61}$$

for some function $\sigma(\cdot,\cdot)$ ensuring the well posedness of the stochastic differential equation (61)[37]. In (61), the function $\sigma(\cdot,\cdot)$ that appears, valued at (t,S_t), is then also the one that appears, valued at (T,K), in the right-hand side of (60)[38].

§3 Stochastic Volatility and Jumps

As explained in §1.A.2, lognormal (or even local volatility) models are strongly misspecified. This leads us to consider various extensions of these models, adding stochastic volatility or/and jumps into the picture.

A Heston Model

The best known stochastic volatility model is the Heston (1993) model, which postulates affine dynamics for the instantaneous variance process V_t, i.e., under a (risk-neutral[39]) measure $\mathbb{Q} \sim$ the physical one: starting from $(v_0,S_0) \in \mathbb{R}^2_+$, for $t \geq 0$,

$$\begin{cases} dv_t = -\mu(v_t - \theta)dt + \eta\sqrt{v_t}dB_t \\ dS_t = S_t(\kappa dt + \sqrt{v_t}dW_t), \end{cases} \tag{62}$$

where:

[36] by application of Corollary 0.1.
[37] at least, in the weak sense, i.e. when a $\mathbb{Q}$ Brownian motion W and a process S solving (61) are sought for simultaneously.
[38] for more details, see, e.g. Bouchard and Chassagneux (2016, Section 7.2.1).
[39] by application of Corollary 0.1, to the market consisting of the stock and the risk-free asset.

- W and B are two $\mathbb{Q}$ Brownian motions with correlation ρ,
- $\mu \geq 0$ is the speed of mean-reversion of the instantaneous variance v_t,
- $\theta \geq 0$ is the long-term variance mean, i.e. $\theta = \lim_{t\to\infty} \mathbb{E}v_t$ (see below),
- η is the volatility of the volatility parameter[40].

Remark 8 *Despite the fact that the Heston SDE for v is nonLipschitz, it can be shown to have a unique strong solution*[41]*. If $\frac{2\mu\theta}{\eta^2} > 1$, the process never reaches 0, otherwise 0 is a reflecting boundary*[42]*.*

By (Abi Jaber, Larsson, and Pulido, 2019, Lemma 7.3), the Heston process αS is a $\mathbb{Q}$ martingale[43]*, in line with the general setup of Section §1.*

In view of the identities in the first two lines of II.(55)[44], $\mathbb{E}v_t$ (for all t) and $\mathbb{E}\int_0^T v_t dt$ are finite. Hence the stochastic integral that appears in the first line of (62) is a true martingale. Taking expectations in the time-integrated version of this first line then yields $\mathbb{E}v_t = v_0 + \mu(\theta t - \int_0^t \mathbb{E}v_s ds)$, i.e. an ODE $f(0) = v_0, f'(t) = \mu(\theta - f(t))$ for $f(t) = \mathbb{E}v_t$, with solution $\mathbb{E}v_t = f(t) = \theta + (v_0 - \theta)e^{-\mu t}$.

We have $d(e^{\mu t}(v_t - \theta)) = e^{\mu t}\eta\sqrt{v_t}dB_t$, i.e.

$$\begin{aligned} v_t &= \theta + e^{-\mu t}(v_0 - \theta) + \eta \int_0^t e^{-\mu(t-s)}\sqrt{v_s}dB_s \\ &= e^{-\mu t}v_0 + (1 - e^{-\mu t})\theta + \eta \int_0^t e^{-\mu(t-s)}\sqrt{v_s}dB_s. \end{aligned} \tag{63}$$

Hence the process v "forgets" its starting point v_0 and reverts to its long-term mean θ at an exponential rate μ. The parameters θ and μ determine the term structure of implied volatilities in the model, whereas v_0, ρ and η determine their level, slope and convexity. For short maturities, however, the convexity of the smile in the Heston or even more general stochastic volatility models can hardly be reconciled with highly convex market smiles: a purely diffusive stochastic volatility requires time for the model to depart from a Black-Scholes behavior.

Also note that, in view of the first line in (63), where the stochastic integral is a true martingale as explained above:

$$\frac{1}{T}\mathbb{E}\int_0^T v_t dt = \theta + (v_0 - \theta)\frac{1 - e^{-\mu T}}{\mu T},$$

hence a simple explicit formula for (the variable leg of) a variance swap in this model–at least, a limiting one that would bear on the instantaneous variance of S, instead of its realized variance in traded variance swap contracts.

From a hedging viewpoint, two (non-redundant) risky assets, e.g. S and the $\mathbb{Q}$ price process of a European option (on top of the riskless asset $S^0 = \beta^{-1}$)[45], allow one to perfectly replicate any payoff in this model[46]: cf. X.§4.C for a similar situation in a stylized stochastic volatility model with only two volatility regimes (values). So the model is incomplete in the sense of hedging with S, but the addition of a single volatility-dependent asset is enough to complete the market.

[40] the lognormal volatility of v_t is $\eta v_t^{-\frac{1}{2}}$.
[41] see for instance Jeanblanc, Yor, and Chesney (2009, Section 6.3).
[42] see IV.§7.B for related simulation issues.
[43] note that the Novikov criterion is not applicable here and see also Filipovic and Mayerhofer (2009, Section 6, case with $r = 0$ there).
[44] stated there in a slightly more general setup with time-dependent coefficient θ.
[45] such an extended market is still nonarbitrable by application of Corollary 0.1.
[46] under mild measurability and integrability restrictions, typically satisfied in applications.

B Merton Model

As already apparent in the basic jump-to-ruin model of X.§2 (see part D there), jumps are one way to generate more extreme short-term smiles than stochastic volatility models. But the jump-to-ruin model, with its single jump of S to 0, is too simplistic for pricing and hedging applications. The Merton (1976) model, instead, is obtained by adding an independent *compound Poisson* process to the Black-Scholes model. So

$$\frac{dS_t}{S_{t-}} = \kappa dt + \sigma dW_t + J_{(t)} dN_t - \lambda \bar{J} dt, \tag{64}$$

where under a (risk-neutral[47]) probability measure $\mathbb{Q} \sim$ the physical one:

- $(N_t)_{t\geq 0}$ is a Poisson process with jump intensity λ and with ordered jump times denoted by T_l,
- the $J_l \equiv J_{(T_l)}$ are i.i.d. jump sizes such that $j_1 := \ln(1+J_1)$ is $\mathcal{N}(\varrho, \nu)$-distributed, so that

$$\bar{j} := \mathbb{E} j_1 = \varrho, \ \bar{J} := \mathbb{E} J_1 = e^{\varrho + \frac{\nu}{2}} - 1,$$

- Each J_l is independent from $\mathfrak{F}_{T_l-}$[48].

By application of the standard (diffusive) Itô formula between jumps supplemented by an explicit calculation of the impact of the jumps at jump times, the model can be rewritten explicitly in terms of the log-spot $X_t = \ln(S_t)$ as[49]:

$$dX_t = a dt + \sigma dW_t + d(\sum_{l=1}^{N_t} j_l), \tag{65}$$

where $a = b - \lambda \bar{J}$, in which $b = \kappa - \frac{1}{2}\sigma^2$. Thus

$$X_t = x + at + \sigma W_t + \sum_{l=1}^{N_t} j_l, \tag{66}$$

where $x = \ln(S_0)$. Hence

$$S_t = S_0 e^{at + \sigma W_t} \prod_{l=1}^{N_t} (1 + J_l) \tag{67}$$

and

$$\alpha_t S_t = S_0 e^{\sigma W_t - \frac{1}{2}\sigma^2 t} L_t,$$

where

$$\begin{aligned} L_t &= \prod_{l=1}^{N_t} (1+J_l) e^{-\lambda \bar{J} t}, \ t \geq 0 \text{ , i.e. } L_0 = 1 \text{ and, for } t \geq 0 \\ dL_t &= L_{t-} \, d(\sum_{l=1}^{N_t} J_l - \lambda \bar{J} t) = -\lambda \bar{J} L_t dt + L_{t-} J_{(t)} dN_t, \end{aligned} \tag{68}$$

a finite variation process. We compute

$$\begin{aligned} \mathbb{E} L_t = \mathbb{E}\mathbb{E}(L_t | N_t) = e^{-\lambda \bar{J} t} \mathbb{E}((\mathbb{E}(1+J_1))^{N_t}) = \\ e^{-\lambda \bar{J} t} \mathbb{E}((1+\bar{J})^{N_t}) = e^{-\lambda \bar{J} t} e^{\lambda t((1+\bar{J})-1)} = 1, \end{aligned} \tag{69}$$

by the well known formula $\mathbb{E} e^{iuN_t} = e^{\lambda t(e^{iu}-1)}$ for the characteristic function of N_t (that follows a Poisson distribution with parameter λt). Hence, by independence between W and N, $\mathbb{E}(\alpha_t S_t) = S_0$ and αS is a $\mathbb{Q}$ martingale, in line with the setup of Section §1.

[47] by application of Corollary 0.1.
[48] see He, Wang, and Yan (1992, Eq. (3.3) page 80).
[49] cf. IX.(27) and IX.(39)–(41).

As in the Heston stochastic volatility model[50], in the simple jump-to-ruin model, the addition of one extra non-redundant risky hedging asset (on top of S) is enough to complete the model[51]. By contrast, in the Merton model, perfect replication of any payoff would require a continuum of risky assets (one extra hedging asset "per possible value of J"). In this sense the Merton model is an intrinsically incomplete model. Hence the corresponding $\mathbb{Q}$ price may not be enough, it needs to be risk-adjusted to be in line with a target hurdle rate for the bank shareholders.

C Bates Model

The Bates (1996) model is the following combination of the Heston and the Merton models:

$$\begin{cases} dv_t = -\mu(v_t - \theta)dt + \eta\sqrt{v_t}dB_t \\ \frac{dS_t}{S_{t-}} = (\kappa - \lambda\bar{J})dt + \sqrt{v_t}dW_t + J_{(t)}dN_t \end{cases} \tag{70}$$

Denoting the stock in the Heston model by S^{he}, (62) and the second line in (68) yield[52]:

$$\begin{aligned} d(S^{he}_t L_t) = S^{he}_t dL_t + L_{t-}dS^{he}_t = \\ S^{he}_t L_{t-}\left(\kappa dt + \sqrt{v_t}dW_t + d(\sum_{l=1}^{N_t} J_l - \lambda\bar{J}t)\right). \end{aligned} \tag{71}$$

Hence, by the well-posedness of the Bates SDE[53], $S^{he}L$ coincides with the stock S in the Bates model (70). By independence between S^{he} and L, this also shows that αS is integrable in the case of the Bates model, as already seen for αS^{he} and for L in (69).

D Log-Spot Characteristic Functions in the Heston, Merton, and Bates Models

The risk-neutral log-spot characteristic function

$$\Phi_T(z) = \mathbb{E}e^{izX_T} = \mathbb{E}[S_T^{iz}],$$

where $i^2 = -1$ and $X_t = \ln(S_t)$, is explicitly known in the above stochastic volatility and jump models, which all belong to the class of affine jump-diffusions (Duffie, Pan, and Singleton, 2000; Duffie, Filipović, and Schachermayer, 2003; Keller-Ressel, 2008). These denote the time-homogenous Markov models X[54] with characteristic function exponentially affine in the initial state x of X, i.e.

$$\mathbb{E}e^{izX_t} = e^{A(t,z)+B(t,z)x}, \ t \geq 0.$$

In jump-diffusion setups, differentiating e^{izX_t} by the Itô formula shows that the coefficients of the generator $\mathcal{A}_x$ of X[55] are then affine in x and that $t \mapsto A(t,z)$ and $B(t,z)$ satisfy a decoupled system of Ricatti ODEs that can efficiently be solved numerically, or even explicitly in special cases such as the ones of Propositions 13 or II.3[56] below.

Lemma 5 *In the setup of the Heston model, let, for $\tau \geq 0$ and z complex,*

$$C(\tau, z) = \mu\left[\tau y_- - \frac{1}{c}\ln(\frac{1 - ge^{-p\tau}}{1-g})\right], \ D(\tau, z) = \frac{1 - e^{-p\tau}}{1 - ge^{-p\tau}}y_-, \tag{72}$$

[50] cf. the end of Part A.
[51] cf. X.§2.F.
[52] by the semimartingale integration by parts formula, or by the integration by parts formula for continuous Itô processes between jumps, manually completed at jump times
[53] cf. Remark 8 regarding its Heston volatility component.
[54] e.g. IX.(20)-(25), for functions $b, \sigma, \delta(t,x) = b, \sigma, \delta(x)$ in (25), and IX.(39) for $\delta(t,x,y) = \delta(t,x)$.
[55] cf. (28)-(41).
[56] in a time-inhomogenous extension of the theory.

where

$$p = \sqrt{y^2 - 4wc},\ y_\pm = \frac{y \pm p}{2c},\ g = \frac{y_-}{y_+} \tag{73}$$

and

$$w = -\frac{1}{2}z(i+z),\ y = \mu - \rho\eta i z,\ c = \frac{\eta^2}{2}. \tag{74}$$

(i) *For $\Im m(z) \in [-1, 0]$, we have $\Re e(C), \Re e(D) \leq 0$.*
(ii) *Given $T > 0$ and setting $\tau = T - t$, the function*

$$\Phi(t, v, F) = F^{iz} \exp\big[C(\tau, z)\theta + D(\tau, z)v\big], \tag{75}$$

is a classical solution to

$$\begin{aligned} &\Phi(T, v, F) = F^{iz} \text{ and, for } t < T, \\ &\partial_t \Phi + \mathcal{A}_{v,F}\Phi = 0, \end{aligned} \tag{76}$$

where

$$\mathcal{A}_{v,F} = -\mu(v - \theta)\partial_v + \frac{1}{2}vF^2\partial^2_{F^2} + \frac{1}{2}\eta^2 v\partial^2_{v^2} + \rho\eta v F \partial^2_{Fv} \tag{77}$$

(the generator of the Markovian pair-process $((v_t, F_t = S_t e^{-\kappa t}))$).

Proof. For Φ of the general form (75) with C and D continuously differentiable in their time variable τ, we have, for any fixed complex number z:

$$\begin{aligned} &\partial_t \Phi/\Phi = -\theta\,\partial_\tau C - v\,\partial_\tau D, \\ &\partial_v \Phi/\Phi = D\,,\ \partial^2_{v^2}\Phi = \partial_v(D\Phi) = D^2\Phi, \\ &F\partial_F \Phi/\Phi = iz\,,\ F^2\partial^2_{F^2}\Phi/\Phi = iz(iz-1), \\ &F\partial^2_{v,F}\Phi/\Phi = F\partial_F(D\Phi)/\Phi = Diz. \end{aligned}$$

Hence (76) is equivalent to the following equations for (C, D) : for every real z, $C(0, z) = D(0, z) = 0$ and, for $\tau > 0$,

$$\begin{aligned} &-\theta\,\partial_\tau C - v\,\partial_\tau D - \mu(v - \theta)D + \\ &\quad \frac{1}{2}viz(iz-1) + \frac{1}{2}\eta^2 v D^2 + \eta\rho v i z D = 0 \end{aligned} \tag{78}$$

or, equivalently[57],

$$\begin{aligned} &\partial_\tau C = \mu D \\ &\partial_\tau D = w + \frac{1}{2}\eta^2 D^2 + \rho\eta i z D - \mu D = \\ &\qquad cD^2 - yD + w = c(D - y_+)(D - y_-), \end{aligned} \tag{79}$$

in which the second line is a Riccati equation in D. The following computations then show that $C(\tau, z)$ and $D(\tau, z)$ as per (72)–(74) (are continuously differentiable in τ, such that $C(0, z) = D(0, z) = 0$ and) indeed satisfy (79):

$$\begin{aligned} &\partial_\tau D = \frac{pe^{-p\tau}(1 - ge^{-p\tau}) - (1 - e^{-p\tau})pge^{-p\tau}}{(1 - ge^{-p\tau})^2} y_- = \frac{(1-g)pe^{-p\tau}}{(1 - ge^{-p\tau})^2} y_-, \\ &c(D - y_+)(D - y_-) = cy_-^2\Big(\frac{D}{y_-} - \frac{y_+}{y_-}\Big)\Big(\frac{D}{y_-} - 1\Big) = \\ &\quad cy_-^2\Big(\frac{1 - e^{-p\tau}}{1 - ge^{-p\tau}} - \frac{1}{g}\Big)\Big(\frac{1 - e^{-p\tau}}{1 - ge^{-p\tau}} - 1\Big) = \\ &\quad cy_-^2 \frac{\big(g(1 - e^{-p\tau}) - (1 - ge^{-p\tau})\big)(ge^{-p\tau} - e^{-p\tau})}{g(1 - ge^{-p\tau})^2} = cy_-^2 \frac{(g-1)^2 e^{-p\tau}}{g(1 - ge^{-p\tau})^2}. \end{aligned}$$

[57] cf. (74).

where in view of (73)-(74),

$$cy_-\frac{(1-g)}{g} = cy_-(\frac{1}{g}-1) = cy_-(\frac{y_+}{y_-}-1) = c(y_+-y_-) = p.$$

Likewise, for C,

$$\begin{aligned}\mu^{-1}\partial_\tau C - D &= y_- - \frac{1}{c}\frac{1-g}{1-ge^{-pT}}\frac{gpe^{-pT}}{1-g} - \frac{1-e^{-pT}}{1-ge^{-pT}}y_- \\ &= y_-\big(1-\frac{1-e^{-pT}}{1-ge^{-pT}}\big) - \frac{1}{c}\frac{gpe^{-pT}}{1-ge^{-pT}} \\ &= y_-\frac{e^{-pT}(1-g)}{1-ge^{-pT}} - \frac{1}{c}\frac{gpe^{-pT}}{1-ge^{-pT}} = 0,\end{aligned}$$

via the identity $y_-(1-g) = \frac{1}{c}gp$ that proceeds from (73)-(74). Hence C and D as per (72)–(74) satisfy (79) and the related function Φ in (75) is a classical solution to (76).

In addition, denoting $z = \delta + i\varepsilon$ with $\delta, \varepsilon \in \mathbb{R}$, hence $iz = -\varepsilon + i\delta$ and $z^2 = \delta^2 - \varepsilon^2 + i2\delta\varepsilon$, (Abi Jaber and De Carvalho, 2022, Theorem 2.3) applied to the equation in the second line of (79) implies that $\mathfrak{Re}(D) \leq 0$ holds provided $\mathfrak{Re}(w) + \frac{\mathfrak{Im}(y)^2}{4c} \leq 0$. We compute in view of (74):

$$\begin{aligned}\mathfrak{Re}(w) + \frac{\mathfrak{Im}(y)^2}{4c} &= -\frac{1}{2}\mathfrak{Re}(z(i+z)) + \frac{\mathfrak{Im}(\rho\eta iz)^2}{2\eta^2} = \\ &\frac{1}{2}\varepsilon - \frac{1}{2}(\delta^2-\varepsilon^2) + \frac{\rho^2\eta^2\delta^2}{2\eta^2} \leq \frac{1}{2}(\varepsilon+\varepsilon^2) \leq 0 \text{ if } \varepsilon \in [-1,0],\end{aligned}$$

in which case we conclude that $\mathfrak{Re}(D) \leq 0$, hence $\mathfrak{Re}(C) \leq 0$. ∎

Proposition 13 *Let $x = X_0 = \ln(S_0), T > 0$ and*

$$\Phi_T^\kappa(z) = \exp\left[iz\big(x+\kappa T\big)\right],$$
$$\Phi_T^j(z) = e^{\lambda T\left(-iz(e^{\varrho+\frac{\nu}{2}}-1)+e^{iz\varrho-z^2\frac{\nu}{2}}-1\right)}.$$

In the Black-Scholes, Merton, Heston and Bates models we have: For $\mathfrak{Im}(z) \in [-1,0]$,

$$\begin{aligned}\Phi_T^{bs}(z) &= \Phi_T^\kappa(z)\exp\left[-\frac{1}{2}z(i+z)\sigma^2T\right],\ \Phi_T^{me}(z) = \Phi_T^{bs}(z)\Phi_T^j(z) \\ \Phi_T^{he}(z) &= \Phi_T^\kappa(z)\exp\left[C(T,z)\theta + D(T,z)v_0\right],\ \Phi_T^{ba}(z) = \Phi_T^{he}(z)\Phi_T^j(z),\end{aligned} \tag{80}$$

where the functions C and D were defined in (72)–(74).

Proof. (adapted from Abi Jaber and De Carvalho (2022, Section 2.2)). In the case of the Heston model restated in terms of v_t and the "forward" $F_t = S_te^{-\kappa t}$, an application of the Itô formula shows via Lemma 5(ii) that the process $(\Phi(t, v_t, F_t))$, with Φ defined by (75), is a local martingale. Moreover (cf. (75)), we have with $\tau = T - t$:

$$\Phi(t, v_t, F_t) = F_t^{iz}\exp\left[C(\tau,z)\theta + D(\tau,z)v_t\right] = F_0^{iz}\exp(U_t),$$

where $U_t = \int_0^t iz\, d\ln F_s + C(\tau,z)\theta + D(\tau,z)v_t$. Hence, for $z = \delta + i\varepsilon$ with $\varepsilon \in [-1,0]$ so that $\mathfrak{Re}(C), \mathfrak{Re}(D) \leq 0$ holds by Lemma 5(i):

$$\begin{aligned}\mathfrak{Re}(U_t) &\leq \int_0^t \mathfrak{Re}(iz)d\ln F_s = \int_0^t(-\varepsilon)d\ln F_s \\ &= \int_0^t(-\varepsilon)\sqrt{v_s}dW_s - \frac{1}{2}\int_0^t(-\varepsilon)v_s ds \\ &\leq \int_0^t(-\varepsilon)\sqrt{v_s}dW_s - \frac{1}{2}\int_0^t(-\varepsilon)^2v_s ds =: \mathfrak{U}_t.\end{aligned}$$

Therefore

$$|\Phi(t, v_t, F_t)| = e^{\mathfrak{Re}(U_t)} \leq e^{\mathfrak{U}_t},$$

where $e^{\mathfrak{U}}$ is such that $de^{\mathfrak{U}_t} = e^{\mathfrak{U}_t}(-\varepsilon)\sqrt{v_t}dW_t$, hence a true martingale, by (Abi Jaber, Larsson, and Pulido, 2019, Lemma 7.3)[58]. Therefore the local martingale $(\Phi(t, v_t, F_t))$ is in fact a martingale, by application of Lemma IX.5(ii). In particular $\mathbb{E}\Phi(T, v_T, F_T) = \Phi(0, v_0, F_0)$, i.e. $\mathbb{E}F_T^{iz} = S_0^{iz}e^{C(T,z)\theta+D(T,z)v_0}$. As $F_T^{iz} = S_T^{iz}e^{-iz\kappa T}$, this proves the formula for Φ_T^{he} in (80).

In the case of the Bates model $S = S^{he}L$ as per (71), we have by independence between the Brownian motions driving S^{he} and the compound Poisson process $J_{(t)}dN_t$ driving L that $\Phi_T^{ba}(z) = \Phi_T^{he}(z)\mathbb{E}L_T^{iz}$. It remains to prove that $\mathbb{E}L_T^{iz} = \Phi_T^j(z)$, which follows from the first line in (68) by computing

$$\begin{aligned}\mathbb{E}(L_T^{iz}) &= \mathbb{E}\Big(\prod_{l=1}^{N_T}(1+J_l)^{iz}\Big)e^{-iz\lambda\bar{J}T} = e^{-iz\lambda\bar{J}T}\mathbb{E}\mathbb{E}\Big(\prod_{l=1}^{N_T}(1+J_l)^{iz}|N_T\Big)\\ &= e^{-iz\lambda\bar{J}T}\mathbb{E}\bigg(\Big(\mathbb{E}\big((1+J_1)^{iz}\big)\Big)^{N_T}\bigg) = e^{-iz\lambda\bar{J}T}\mathbb{E}e^{(iz\varrho-\frac{z^2\nu}{2})N_T}\\ &= e^{-iz\lambda\bar{J}T}e^{\lambda T(e^{iz\varrho-\frac{z^2\nu}{2}}-1)} = e^{\lambda T(-iz\bar{J}+e^{iz\varrho-\frac{z^2\nu}{2}}-1)},\end{aligned}$$

where the passage to the last line uses the well known formula $\mathbb{E}e^{iuN_T} = e^{\lambda T(e^{iu}-1)}$ for the characteristic function of N_T (that follows a Poisson distribution with parameter λT).

Finally, setting $\mu = \rho = 0$ yields

$$y = 0,\ y_\pm = \frac{\pm p}{\eta^2},\ p = \sqrt{-2w}\eta,\ 1-g = 2,$$

hence when $\eta \to 0+$:

$$D(T,z) = \frac{1-e^{-pT}}{1-ge^{-pT}}y_- \to \frac{pT}{1-g}y_- = \frac{-p^2T}{2\eta^2} = wT,$$

and $C(T,z)\theta + D(T,z)v_0$ reduces to $-\frac{1}{2}z(i+z)v_0T$. Thus $\Phi_T^{he}(z)$ reduces to $\Phi_T^{\kappa}(z)\exp\big[-\frac{1}{2}z(i+z)v_0T\big]$, which thus corresponds to $\Phi_T^{bs}(z)$ for $\sigma = \sqrt{v}_0$; $\Phi_T^{ba}(z)$ reduces to $\Phi_T^{bs}(z)\Phi_T^j(z)$, which thus corresponds to $\Phi_T^{me}(z)$ for $\sigma = \sqrt{v}_0$. ∎

Knowing the log-spot characteristic function Φ_T, vanilla option prices and Greeks can then be computed by the Fourier transform techniques of §1.F. See Kahl and Jäckel (2005) and Gatheral (2011, page 20) for implementation details, including numerical issues related to the multivalued complex logarithmic integrands that may be embedded in the characteristic functions Φ_T, or the use of Gaussian quadratures methods which can be efficiently used here.

[58] cf. also Filipovic and Mayerhofer (2009, Theorem 3.3 and Section 6, case with $r = 0$ there).

Chapter II

Market Models

We review market models for foreign-exchange derivatives (assessed in a generic multivariate continuous Itô processes setup), interest rate derivatives (Libor market model), credit derivatives (Gaussian copula model), and volatility derivatives (local stochastic volatility models).

§1 Multivariate Continuous Itô Processes Market Model

A Physical Model

We call continuous Itô processes market, the model with riskless asset

$$S_t^0 = e^{\int_0^t r_s ds} = \beta_t^{-1}, \tag{1}$$

for some bounded from below progressive short rate process r such that $\int_0^{\cdot} |r_t| dt < +\infty$, and with d liquidly traded risky assets[1]

$$S_t^l = S_0^l e^{\int_0^t \left(\mu_s^l - \frac{1}{2}\sum_{p=1}^d (\sigma_s^{l,p})^2\right) ds + \int_0^t \sum_{p=1}^d \sigma_s^{l,p} dB_s^p}, \ l \in 1 \dots d,$$

for a d-variate standard Brownian motion B under the physical probability measure $\mathbb{P}$ and for progressive processes μ and σ such that $\int_0^T \left(\|\mu_t\| + |||\sigma_t|||^2\right) dt < +\infty$, where $\|\cdot\|$ and $|||\cdot|||$ are norms on vectors $\mu \in \mathbb{R}^d$ and matrices $\sigma \in \mathbb{R}^{d \otimes d}$. An application of the Itô formula yields

$$dS_t^l = S_t^l \big(\mu_t^l dt + \sum_{p=1}^d \sigma_t^{l,p} dB_t^p\big), \ l \in 1 \dots d \tag{2}$$

or, in matrix-vector form,

$$dS_t = \mathrm{diag}(S)_t (\mu_t dt + \sigma_t dB_t). \tag{3}$$

Assuming the matrix σ invertible, the vector of risk premia

$$\lambda = \sigma^{-1}(\mu - r\mathbf{1})$$

is such that

$$dS_t = \mathrm{diag}(S)_t \big(r_t \mathbf{1} dt + \sigma_t (\lambda_t dt + dB_t)\big) \tag{4}$$

or, in discounted terms,

$$d(\beta S)_t = \mathrm{diag}(\beta S)_t \sigma_t (\lambda_t dt + dB_t). \tag{5}$$

[1] assuming non-dividend-paying assets for notational simplicity.

B Risk-Neutral Setup

Assuming the Novikov condition $\mathbb{E}^{\mathbb{P}} e^{\frac{1}{2}\int_0^T \|\lambda\|_t^2 dt} < +\infty$, the process

$$B + \int_0^{\cdot} \lambda_t dt = W \tag{6}$$

is a d-variate standard Brownian motion under the probability measure $\mathbb{Q}$ such that

$$\frac{d\mathbb{Q}}{d\mathbb{P}}\Big|_{\mathfrak{F}_T} = e^{-\int_0^T \lambda_t dB_t - \frac{1}{2}\int_0^T \|\lambda\|_t^2 dt}.$$

We write $\mathbb{E}^{\mathbb{Q}} = \mathbb{E}$.

The self-financing condition 0.(3) applied to the continuous Itô processes market (5) yields

$$\beta V = \pi + \int_0^{\cdot} \zeta_t \mathrm{diag}(\beta S)_t \sigma_t (\lambda_t dt + dB_t) = \pi + \int_0^{\cdot} \zeta_t \mathrm{diag}(\beta S)_t \sigma_t dW_t. \tag{7}$$

Definition 1 *A strategy is deemed $\mathbb{Q}$ progressive-admissible if ζ is progressive and such that $\int_0^T \|\zeta_t \mathrm{diag}(\beta S)_t \sigma_t\|^2$ $+\infty$ and $\beta V \geq$ some (continuous) $\mathbb{Q}$ martingale M (possibly depending on ζ).*

Lemma 1 *A progressive strategy ζ such that $\mathbb{E}\int_0^T \|\zeta_t \mathrm{diag}(\beta S)_t \sigma_t\|^2 dt < +\infty$ is $\mathbb{Q}$ progressive-admissible.*

Proof. In view of (7), the corresponding process βV is then itself a $\mathbb{Q}$ martingale M. ∎

Lemma 2 *The financial market* (1)-(3) *admits no arbitrage among $\mathbb{Q}$ progressive-admissible strategies.*

Proof. Lemma IX.8 implies that the discounted wealth process βV of a $\mathbb{Q}$ progressive-admissible strategy (π, ζ) is a $\mathbb{Q}$ local martingale on $[0, T]$. By assumption, βV is also bounded from below by a $\mathbb{Q}$ martingale. Hence Lemma IX.5(i) implies that βV is a supermartingale on $[0, T]$, so that $\mathbb{E}(\beta V)_T \leq \mathbb{E}(\beta V)_0 = 0$, assuming V starts from $\pi = 0$ at time 0. This justifies the second implication in

$$V_T \geq 0 \text{ a.s. } \Rightarrow (\beta V)_T \geq 0 \text{ a.s. } \Rightarrow (\beta V)_T = 0 \text{ a.s. } \Rightarrow V_T = 0 \text{ a.s.},$$

which shows that $(\pi = 0, \zeta)$ is not an arbitrage, by Definition 0.6. ∎

Theorem 1 *Assume $\mathfrak{F} = \mathfrak{F}^W$, the natural completed filtration of W. Then for any $\mathfrak{F}_T$ measurable random variable ξ such that $\beta_T \xi$ is $\mathbb{Q}$ square integrable, for any $t \in [0, T]$, there exists a $\mathbb{Q}$ progressive-admissible strategy ζ replicating ξ*[2]*, starting from the initial wealth at time t given as the time-t $(\mathfrak{F}, \mathbb{Q})$ price of the option*

$$\Pi_t = \mathbb{E}(e^{-\int_t^T r_s ds} \xi | \mathfrak{F}_t)$$

(e.g. starting from the initial wealth $\pi = \mathbb{E}(e^{-\int_0^T r_s ds}\xi)$ at time 0).

Proof. By the Brownian martingale representation property[3], there exists a progressive process Z such that $\mathbb{E}\int_0^T Z_s^2 ds < +\infty$ and

$$\beta_T \xi = \mathbb{E}(\beta_T \xi) + \int_0^T Z_s dW_s = \pi + \int_0^T \zeta_s \mathrm{diag}(\beta S)_s \sigma_s dW_s = \beta_T V_T \tag{8}$$

holds for the initial wealth $\pi = \mathbb{E}(\beta_T \xi)$ and the strategy $\zeta = Z\big(\mathrm{diag}(\beta S)\sigma\big)^{-1}$, which is $\mathbb{Q}$ progressive-admissible by Lemma 1. Taking the difference between (8) and its conditional expectation with respect to $\mathfrak{F}_t$, i.e. $\beta_t \Pi_t = \pi + \int_0^t Z_s dW_s$, then yields

$$\beta_T \xi = \beta_t \Pi_t + \int_t^T Z_s dW_s = \beta_t \Pi_t + \int_t^T \zeta_s \mathrm{diag}(\beta S)_s \sigma_s dW_s,$$

for any $t \leq T$. ∎

[2] i.e. such that $V_T = \xi$.

[3] for a detailed statement and proof see e.g. https://fabricebaudoin.wordpress.com/2012/09/23/lecture-23-itos-representation-theorem/.

C Change of Numéraire

Hereafter in this section, we assume σ càglàd and we restrict ourselves to $\mathbb{Q}$ admissible strategies, i.e.[4] with "ζ progressive and such that $\int_0^T \|\zeta_t \mathrm{diag}(\beta S)_t \sigma_t\|^2 dt < +\infty$" replaced by "$\zeta$ càglàd" in Definition 1. As $\mathbb{Q}$ admissible strategies are $\mathbb{Q}$ progressive-admissible[5], the market is arbitrage-free for $\mathbb{Q}$ admissible strategies, by Lemma 2.

Let there be given a continuous numéraire with inverse $\widetilde{\beta}$ and pricing measure $\widetilde{\mathbb{Q}}$[6], hence, by I.(22):

$$\frac{d\widetilde{\mathbb{Q}}}{d\mathbb{Q}}\Big|_{\mathfrak{F}_T} = \widetilde{\beta}_0 \big(\frac{\beta}{\widetilde{\beta}}\big)_T, \ \textit{i.e.} \ \frac{d\mathbb{Q}}{d\widetilde{\mathbb{Q}}}\Big|_{\mathfrak{F}_T} = \widetilde{\beta}_0^{-1} \big(\frac{\widetilde{\beta}}{\beta}\big)_T. \tag{9}$$

Definition 2 *A càglàd strategy $\eta = (\zeta^0, \zeta)$ with corresponding (càglàd) wealth process $V = \sum_{l=0}^d \eta^l S^l$ is deemed:*
(i) *$\widetilde{\beta}^{-1}$ self-financing, if*

$$d(\widetilde{\beta} V)_t = \sum_{l=0}^d \eta_t^l d(\widetilde{\beta} S^l)_t; \tag{10}$$

(ii) *$(\widetilde{\beta}^{-1}, \widetilde{\mathbb{Q}})$ admissible, if $\widetilde{\beta} V \geq$ some $\widetilde{\mathbb{Q}}$ martingale $\widetilde{M}$ (possibly depending on ζ).*

So self-financing as per 0.(3)[7] means $(\beta^{-1} =) S^0$ self-financing; $\mathbb{Q}$ admissible as per 0.(5) means $(S^0, \mathbb{Q})$ admissible.

Proposition 1 *(i) $\widetilde{\beta}^{-1}$ self-financing, respectively (ii) $(\widetilde{\beta}^{-1}, \widetilde{\mathbb{Q}})$ admissible strategies coincide with (i) self-financing, respectively (ii) $\widetilde{\mathbb{Q}}$ admissible strategies.*

Proof. **(i)** Assuming that η is self-financing as per 0.(3), the Itô formula applied to the function $f(x,y) = xy$[8] yields

$$\begin{aligned} d(\widetilde{\beta} V)_t &= \widetilde{\beta}_t dV_t + V_t d\widetilde{\beta}_t + d\langle \widetilde{\beta}, V \rangle_t \\ &= \sum_{l=0}^d \big(\widetilde{\beta}_t \eta_t^l dS_t^l + \eta_t^l S_t^l d\widetilde{\beta}_t + \eta_t^l d\langle \widetilde{\beta}, S^l \rangle_t \big) \\ &= \sum_{l=0}^d \eta_t^l d(\widetilde{\beta} S^l)_t. \end{aligned} \tag{11}$$

Hence η is $\widetilde{\beta}^{-1}$ self-financing. Conversely, if η is $\widetilde{\beta}^{-1}$ self-financing, i.e. if $V = \sum_{l=0}^d \eta^l S^l$ is such that $d(\widetilde{\beta} V)_t = \sum_{l=0}^d \eta_t^l d(\widetilde{\beta} S^l)_t$, then, by computations similar to (11),

$$\begin{aligned} d(\widetilde{\beta}^{-1}(\widetilde{\beta} V))_t &= \widetilde{\beta}_t^{-1} d(\widetilde{\beta} V)_t + (\widetilde{\beta} V)_t d\widetilde{\beta}_t^{-1} + d\langle \widetilde{\beta}^{-1}, \widetilde{\beta} V \rangle_t \\ &= \sum_{l=0}^d \big(\widetilde{\beta}_t^{-1} \eta_t^l d(\widetilde{\beta} S^l)_t + \eta_t^l \widetilde{\beta}_t S_t^l d\widetilde{\beta}_t^{-1} + \eta_t^l d\langle \widetilde{\beta}^{-1}, \widetilde{\beta} S^l \rangle_t \big) \\ &= \sum_{l=0}^d \eta_t^l d(\widetilde{\beta}^{-1}(\widetilde{\beta} S^l))_t, \end{aligned}$$

[4] cf. 0.(5).
[5] cf. Lemma IX.9.
[6] see I.§1.G.2.
[7] with $\mathcal{D} = 0$, here.
[8] or the semimartingale integration by parts formula IX.(9), in the present continuous setup where $[\cdot,\cdot] = \langle \cdot, \cdot \rangle$, also recalling the formula IX.(12) that is used to compute $d\langle \widetilde{\beta}, V \rangle_t$.

i.e. $dV_t = \sum_{l=0}^{d} \eta_t^l dS_t^l$, so η is self-financing.
(ii) If $\widetilde{\beta}V$ dominates a $\widetilde{\mathbb{Q}}$ martingale $\widetilde{M}$, then $\beta V \geq$ dominates the process $M = \frac{\beta}{\widetilde{\beta}}\widetilde{M}$, a $\mathbb{Q}$ martingale by Lemma IX.7 and (9). Symmetrically, if βV dominates a $\mathbb{Q}$ martingale M, then $\widetilde{\beta}V$ dominates the $\widetilde{\mathbb{Q}}$ martingale $M = \frac{\widetilde{\beta}}{\beta}\widetilde{M}$. ∎

Lemma 3 *Assuming dynamics*[9]

$$d\widetilde{\beta}_t^{-1} = \widetilde{\beta}_t^{-1}\big(r_t dt + \widetilde{\sigma}_t(\lambda_t dt + dB_t)\big) = \widetilde{\beta}_t^{-1}\big(r_t dt + \widetilde{\sigma}_t dW_t\big) \tag{12}$$

for same càglàd row-process $\widetilde{\sigma}$, then :
(i) *$d\widetilde{W}_t = dW_t - \widetilde{\sigma}_t^{\top} dt = (\lambda_t - \widetilde{\sigma}_t^{\top})dt + dB_t$ is a d-variate standard Brownian motion under $\widetilde{\mathbb{Q}}$;*
(ii) *for $l \in 1 \dots d$,*

$$d(\widetilde{\beta}S^l)_t = (\widetilde{\beta}S^l)_t \|\sigma^l - \widetilde{\sigma}\|_t d\mathcal{W}_t^l, \tag{13}$$

where

$$d\mathcal{W}_t^l = \Big(\mathbb{1}_{\sigma^l \neq \widetilde{\sigma}} \frac{(\sigma^l - \widetilde{\sigma})}{\|(\sigma^l - \widetilde{\sigma})^{\mathsf{T}}\|} + \mathbb{1}_{\sigma^l = \widetilde{\sigma}}(1, 0, \dots, 0)\Big)_t d\widetilde{W}_t, \tag{14}$$

is a $\widetilde{\mathbb{Q}}$ univariate standard Brownian motion.

Proof. **(i)** follows from the Girsanov theorem in view of the formulas (6), (9) and (12);
(ii) An application of the diffusive Itô formula[10] $d(\frac{1}{X})_t = \frac{-dX_t}{X_t^2} + \frac{d\langle X\rangle_t}{X_t^3}$ to (12) yields

$$\begin{aligned} d\widetilde{\beta}_t &= -\widetilde{\beta}_t^2 \cdot \widetilde{\beta}_t^{-1}\big(r_t dt + \widetilde{\sigma}_t dW_t\big) + \widetilde{\beta}_t^3 \cdot \widetilde{\beta}_t^{-2}\|\widetilde{\sigma}^{\top}\|_t^2 dt\big) \\ &= \widetilde{\beta}_t\big(- r_t dt - \widetilde{\sigma}_t dW_t + \|\widetilde{\sigma}^{\top}\|_t^2 dt\big). \end{aligned} \tag{15}$$

This in conjunction with (4) and (6) yields

$$\begin{aligned} d(\widetilde{\beta}S^l)_t &= \widetilde{\beta}_t dS_t^l + S_t^l d\widetilde{\beta}_t + d\langle\widetilde{\beta}, S^l\rangle_t \\ &= (\widetilde{\beta}S^l)_t\big((\sigma_t^l - \widetilde{\sigma}_t)dW_t + \|\widetilde{\sigma}^{\top}\|_t^2 dt - \sigma_t^l\widetilde{\sigma}_t^{\top} dt\big) \\ &= (\widetilde{\beta}S^l)_t\big((\sigma_t^l - \widetilde{\sigma}_t)dW_t - (\sigma_t^l - \widetilde{\sigma}_t)\widetilde{\sigma}_t^{\top} dt\big) \\ &= (\widetilde{\beta}S^l)_t(\sigma_t^l - \widetilde{\sigma}_t)\big(dW_t - \widetilde{\sigma}_t^{\top} dt\big) \\ &= (\widetilde{\beta}S^l)_t(\sigma_t^l - \widetilde{\sigma}_t)d\widetilde{W}_t, \end{aligned}$$

which is (13)-(14), where the fact that $\mathcal{W}^l$ is a $\widetilde{\mathbb{Q}}$ univariate standard Brownian motion follows from Lévy's characterization of Brownian motion[11]. ∎

D Application to Exchange Options

Proposition 2 *Assuming $d = 2$ and σ deterministic:*
(i) *Under the pricing measure associated to the numéraire S^2, $\frac{S^1}{S^2}$ follows a Black model as per I.§2.A.1, with volatility function $\varsigma := \|(\sigma^1 - \sigma^2)^{\mathsf{T}}\|$ (i.e. $\frac{S^1}{S^2}$ has Black-Scholes dynamics with zero interest and dividend rates and with deterministic volatility ς);*
(ii) *The payoff $(S_T^1 - KS_T^2)^+$ at time T of a European exchange option is replicable by a self-financing admissible strategy*

$$(\zeta^0, \zeta^1, \zeta^2) = \Big(0, \mathcal{N}\big(\mathrm{d}_+^{bl}(\cdot, \frac{S^1}{S^2}, T, K; \varsigma)\big), -K\mathcal{N}\big(\mathrm{d}_-^{bl}(\cdot, \frac{S^1}{S^2}, T, K; \varsigma)\big)\Big)$$

[9] cf. (4)-(6).
[10] that applies to any positive continuous Itô process.
[11] cf. Theorem IX.2.

in (S^0, S^1, S^2)*, with price process*

$$S^1\mathcal{N}\big(\mathrm{d}_+^{bl}(\cdot, \frac{S^1}{S^2}, T, K; \varsigma)\big) - KS^2\mathcal{N}\big(\mathrm{d}_-^{bl}(\cdot, \frac{S^1}{S^2}, T, K; \varsigma)\big).$$

Proof. **(i)** By application of Lemma 3;
(ii) By (10) and Proposition 1(i), the S^2 relative wealth process $\frac{V}{S^2}$ of a self-financing strategy (ζ^1, ζ^2) in (S^1, S^2) satisfies

$$d(\frac{V}{S^2})_t = \zeta_t^1 d(\frac{S^1}{S^2})_t. \tag{16}$$

Given (i), an application of Theorem I.1 (with $r = q = 0$ there) then yields that[12], for $\pi = S_0^2 c^{bl}(0, \frac{S_0^1}{S_0^2}, T, K; \varsigma)$ (hence $\frac{\pi}{S_0^2} = c^{bl}(0, \frac{S_0^1}{S_0^2}, T, K; \varsigma)$), $\zeta^1 = \delta^{bl}(\cdot, \frac{S^1}{S^2}, T, K; \varsigma)$ and ζ^2 dictated by the budget condition $V = \zeta^1 S^1 + \zeta^2 S^2$, we have $(\frac{V}{S^2})_T = ((\frac{S^1}{S^2})_T - K)^+$, i.e. $V_T = (S_T^1 - KS_T^2)^+$ (the exchange option's payoff at T), and $\frac{V}{S^2} = c^{bl}(\cdot, \frac{S^1}{S^2}, T, K; \varsigma)$. The budget condition is equivalent to $\frac{V}{S^2} = \zeta^1 \frac{S^1}{S^2} + \zeta^2$, where

$$\frac{V}{S^2} = c^{bl}(\cdot, \frac{S^1}{S^2}, T, K; \varsigma) = \frac{S^1}{S^2}\mathcal{N}\big(\mathrm{d}_+^{bl}(\cdot, \frac{S^1}{S^2}, T, K; \varsigma)\big) - K\mathcal{N}\big(\mathrm{d}_-^{bl}(\cdot, \frac{S^1}{S^2}, T, K; \varsigma)\big)$$

and $\zeta^1 = \mathcal{N}\big(\mathrm{d}_+^{bl}(\cdot, \frac{S^1}{S^2}, T, K; \varsigma)\big)$. Hence $\zeta^2 = -K\mathcal{N}\big(\mathrm{d}_-^{bl}(\cdot, \frac{S^1}{S^2}, T, K; \varsigma)\big)$. ∎

§2 Libor Market Model of Interest-Rate Derivatives

A Short Rate and HJM Models in a Nutshell

The first introduced interest rate models were stochastic models of the short rate process (r_t), notably an Ornstein-Uhlenbeck (Gaussian) process in the case of the Vasicek (1977) model, or a square-root diffusion[13] in the case of the Cox, Ingersoll, and Ross (1985) (CIR) model. Besides the lack of econometrical realism (particularly stringent in the case of the Gaussian model), one practical shortcoming of such short rate models is their difficulty in calibrating a whole term structure of T discount bond prices $B_0^T = \mathbb{E}e^{-\int_0^T r_s ds}$[14] or the equivalent yield curve $R_0^T := -\frac{1}{T}\ln(B_0^T)$, $T \geq 0$, observed or bootstrapped from market data[15] at the pricing time 0.

Remark 1 *The yield rates* R_0^T*, much like implied volatilities for option prices, are "wrong numbers to put in the wrong formula to obtain the right result", in this case the value* R_0^T *of a supposedly constant instantaneous interest rate* r *consistent with an observed* T *discount bond price* B_0^T*, i.e.* $e^{-TR_0^T} = B_0^T$*. One also defines the corresponding instantaneous forward rates* f_0^T *such that the* T *forward value of "€1 at time* $T + dT$*",*

$$\frac{B_0^{T+dT}}{B_0^T} = e^{TR_0^T - (T+dT)R_0^{T+dT}},$$

be given by $e^{-f_0^T dT}$*, i.e.* $f_0^T = (TR_0^T)' = -\partial_T \ln B_0^T$*.*

The Heath, Jarrow, and Morton (1992) (HJM) framework puts short rate models in the more general perspective of models for the evolution of the instantaneous forward rates curve $f_t^T := -\partial_T \ln B_t^T$, reparameterized via the change of variables $T = t + \tau$ as a curve $f_t(\tau), \tau \in \mathbb{R}_+$. One avatar in this class

[12] cf. I.(48).
[13] same process as the one used in the Heston model (62) to model the instantaneous variance v.
[14] see I.§2.A.1.
[15] see (Brigo and Mercurio, 2007; Andersen and Piterbarg, 2010; Henrard, 2014).

is the pioneering[16] Hull and White (1990) model, which practically reduces to a short rate (Gaussian again) model, but one that provides an automatic fit to today's yield curve as a whole, via a risk-neutral drift coefficient function of time t.

The above models yield convenient bond and bond derivatives pricing and Greeking formulas[17], typically revolving around either Gaussian analytics, or time-inhomogenous affine Laplace transform formulas such as the ones of Proposition IX.3[18].

But most real-life interest rate derivatives, such as cap/floors and swaptions, are underlied by simply compounded (simple) rates, as opposed to instantaneous short rates r in the above models. Even if it it possible to reformulate many interest rate derivative payoffs in terms of zero-coupon prices B^S_T, themselves related to short rates via the risk-neutral pricing formula $B^S_T = \mathbb{E}_T e^{-\int_T^S r_s ds}$, the so-called market models of interest rates, our focus hereafter in this section (building on the Black framework of I.§2.A.1), have emerged as a more straightforward and convenient alternative[19].

B Libor Market Model

Most interest rate derivatives, such as forward rate agreements (FRAs), interest rate swaps (IRS), cap/floors and swaptions, are written on underlying simple interest rates relative to future time periods $[S,T)$ or successions of such time periods $[t_{i-1},t_i)$ determined by tenor structures $0<t_1<\ldots<t_{n+1}$.

Example 1 *Libor, which stands for London Inter Bank Offered Rates, were daily published simple rates based on the interest rates at which a group of banks in London claim they would agree to borrow money from each other at various time horizons such as one month, three months, six months, when needed. During more than 30 years, Libor, launched in 1986 and in the process of being replaced by more transparent alternatives[20], have been the main underlyings to most vanilla interest-rate derivatives.*

In view of the above example, we use the terminology Libor to refer to simple rates underlying interest rate derivatives. More precisely, given the pricing time $t_0=0$ and a tenor structure $0\le t_1<\ldots t_{n+1}$, we consider a primary market made of the zero-coupon bonds expiring at each of the t_{i+1}, for $i=0,\ldots,n$. For any such i we write $B^i_t = B^{t_{i+1}}_t$ and we denote by L^i the forward simple rate process associated with the zero-coupon bond price processes B^{i-1} and B^i, for $i=1,\ldots,n$ (see Figure 1). As $€B^i_t = \frac{B^i_t}{B^{i-1}_t}B^{i-1}_t$ allow securing €1 at t_{i+1} or, equally well, $€\frac{B^i_t}{B^{i-1}_t}$ at t_i[21], we have for $i=1,\ldots,n$ and $t\le t_i$,

$$\frac{B^i_t}{B^{i-1}_t}\times(1+hL^i_t)=1,\ hL^i_t=\frac{B^{i-1}_t-B^i_t}{B^i_t}, \tag{17}$$

where $h=t_{i+1}-t_i$ (assumed constant over $i\ge 1$). Since $B^i_{t_{i+1}}=1$, it follows by induction[22] that, for every $1\le i\le l+1\le n+1$,

$$B^l_{t_i}=\prod_{k=i}^l \tfrac{1}{1+hL^k_{t_i}}. \tag{18}$$

Remark 2 *Interest rate derivatives are written on the spot rates $L^i_{t_i}$. The $L^i_\cdot$ are the corresponding forward rate processes.*

[16] even though strongly misspecified (Albanese, Crépey, and Iabichino, 2021, 2022).
[17] see e.g. Lamberton and Lapeyre (1996, Section 6.2 p. 158–165) for the basic Vasicek and Cox Ingersoll Ross pricing formulas.
[18] cf. also I.§3.D and the references there, as well as Filipovic (2009).
[19] for an historical perspective, see Musiela (2022).
[20] see G.3.
[21] cf. Remark 1 in the infinitesimally short limit.
[22] over $l\ge(i-1)^+$, for each fixed $i\ge 0$.

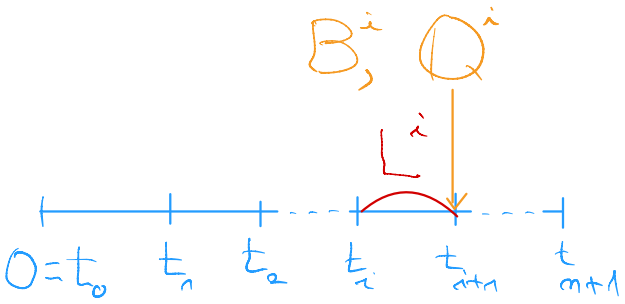


Figure 1: *Tenor structure.*

Example 2 **(i)** *A forward rate agreement (FRA) and a caplet/floorlet on the period* $[t_i, t_{i+1})$*, with strike* K*, are the financial derivatives with respective cash flows* $h(L^i_{t_i} - K)$ *and* $h(L^i_{t_i} - K)^{\pm}$ *at time* t_{i+1}*.*

(ii) *An interest rate swap and a cap/floor with tenor dates* $t_1, \ldots, t_{n+1}$ *and strike* K *are the respective collections of FRAs and caplets/floorlets with strike* K *on the successive periods* $[t_i, t_{i+1})$*, i.e. the contract with the respective cash flows* $h(L^i_{t_i} - K)$ *and* $h(L^i_{t_i} - K)^{\pm}$ *at each time* t_{i+1}*, for* $i = 1, \ldots, n$*.*

A caplet/floorlet is thus a call/put option with arrear settlement on a Libor.

Let $\mathbb{Q}^i = \mathbb{Q}^{t_{i+1}}$, $\mathbb{E}^i = \mathbb{E}^{\mathbb{Q}^i}$[23]. The quantity $B^{i-1} - B^i$ is the difference between prices of traded assets, so that, with the motivation stemming from I.§1.G.2, we want to model the ratio $hL^i = \frac{B^{i-1}-B^i}{B^i}$ as a $\mathbb{Q}^i$ martingale. Consistent with this aim, the log-normal Libor market model (LMM)[24] postulates the following $\mathbb{Q}^i$ Black dynamics for the forward Libor L^i :

$$dL^i_t = \sigma_i(t) L^i_t dW^i_t, \text{ i.e. } L^i_t = L^i_0 \exp\left(\int_0^t \sigma_i(s) dW^i_s - \tfrac{1}{2}\int_0^t \sigma_i^2(s) ds\right), \tag{19}$$

for some $\mathbb{Q}^i$-Brownian motion W^i and deterministic volatility function $\sigma_i(\cdot)$. Since the Libor process L^i is stopped at t_i, the function $\sigma_i(\cdot)$ vanishes after t_i. In particular, the time-t conditional variance of $\ln L^i_{t_{i+1}}$ is equal to

$$\frac{1}{t_{i+1} - t} \int_t^{t_i} \sigma_i^2(s) ds =: \Sigma_i^2(t). \tag{20}$$

C Caps and Floors

By the $\mathbb{Q}^i$ martingale pricing formula (49) relative to the numéraire B^i, the value F^i of the FRA of Example 2(i) is given, for time $t \le t_{i+1}$, by:

$$F^i_t = hB^i_t \mathbb{E}^i_t \left(L^i_{t_{i+1}} - K\right) = hB^i_t \left(L^i_t - K\right), \tag{21}$$

as L^i is a $\mathbb{Q}^i$ martingale (stopped at t_i). Likewise, we have in the caplet case with LMM price C^i, for $t \le t_{i+1}$[25],

$$C^i_t = hB^i_t c^{bl}(t, L^i_t, t_{i+1}, K; \Sigma_i(t)), \text{ and } \delta^i_t = \delta^{bl}(t, L^i_t, t_{i+1}, K; \Sigma_i(t)) \tag{22}$$

is the key to hedging the caplet:

[23] cf. I.§2.A.1 and Figure 1.
[24] also known as the BGM model in reference to the seminal paper by Brace, Gątarek, and Musiela (1997).
[25] cf. (47)-(48).

Proposition 3 *In a primary market defined by the numéraire B^i, used as funding asset, and the FRA of Example 2(i), an LMM replication strategy with price process C^i for the caplet is defined, for $t \in [0, t_{i+1})$, by*

$$\zeta_t = \delta_t^i$$

units of the FRA and the number of t_{i+1}-discount bonds that follows via the budget condition[26] *on the replicating portfolio.*

Proof. Denoting $\beta^i = (B^i)^{-1}$, in view of (21) the B^i relative price process of the FRA is given, for $t \leq t_{i+1}$, by:

$$\beta_t^i F_t^i = h\left(L_t^i - K\right).$$

By I.(23)[27], the wealth process V of a strategy ζ in the FRA funded in B^i satisfies

$$d(\beta^i V)_t = \zeta_t d(\beta^i F^i)_t = h\zeta_t dL_t^i, \tag{23}$$

by (21). Given the Black model postulated on L^i or, equivalently, hL^i, an application of Theorem I.1 with $r = q = 0$ there yields that, for $\pi = C_0^i$ (hence $\beta_0^i \pi = hc^{bl}(0, L_0^i, t_{i+1}, K; \sigma) = c^{bl}(0, hL_0^i, t_{i+1}, hK; \sigma)$) and $\zeta = \delta^i$, we have

$$V_{t_{i+1}} = \beta_{t_{i+1}}^i V_{t_{i+1}} = (hL_{t_{i+1}}^i - hK)^+ = h(L_{t_i}^i - K)^+,$$

which is the caplet's payoff at t_{i+1}. ∎

The analogous formulas and results for a floorlet are obtained by caplet/floorlet parity.

Caps and floors are caplet and floorlet streams relative to the tenor structure $0 < t_1 < \ldots < t_{n+1}$, hence they can be handled by additivity via the above results. Note that, due to the additive structure of the payoffs, the prices and replication strategies of caps and floors depend only on the marginal laws of the L^i at the tenor dates t_i, not on the Libor correlation.

D Adding Correlation

As opposed to caps and floors, swaptions below are sensitive to the correlation structure of L[28]. We now define a correlation structure between the L^i by expressing their joint dynamics under the so-called terminal measure $\mathbb{Q}^n$. By definition of $\mathbb{Q}^{i-1}$ and $\mathbb{Q}^i$, for $t \leq t_i$ we have[29]:

$$\nu_t^i := \frac{d\mathbb{Q}^{i-1}}{d\mathbb{Q}^i}\,|_{\mathfrak{F}_t} = \mathbb{E}_t^i \frac{d\mathbb{Q}^{i-1}}{d\mathbb{Q}^i} = \frac{B_0^i B_t^{i-1}}{B_0^{i-1} B_t^i} = \frac{B_0^i}{B_0^{i-1}}(1 + hL_t^i), \tag{24}$$

by (17). Starting from the L_0^i extracted from the B_0^i via (17), we target a dynamic model

$$dL_t^i = s_i(t) L_t^i d\mathbb{W}_t^i = \sigma_i(t) L_t^i dW_t^{i,i}, \tag{25}$$

for the row-vector volatility function $s_i(t) = (0, \ldots, 0, \sigma_i(t), 0, \ldots, 0)$ and for an n-dimensional $\mathbb{Q}^i$ Brownian motion $\mathbb{W}^i = (W^{l,i})_{1 \leq l \leq n}$ with correlation matrix $\rho = (\rho_{k,l})_{1 \leq k,l \leq n}$ (same matrix for each i). Toward this aim, we proceed by backward induction over i, starting from $\mathbb{W}^n = (W^{l,n})_{1 \leq l \leq n}$ given as an n-dimensional $\mathbb{Q}^n$ Brownian motion with the desired correlation matrix ρ and L^n modeled as $dL_t^n = \sigma_n(t) L_t^n dW_t^{n,n}$. We next define $\mathbb{W}_t^{n-1}$ as $\mathbb{W}_t^n - \rho \int_0^t \mu_u^\top du$, for some row-vector process μ. By the Girsanov theorem, for $\mathbb{W}^{n-1}$ thus defined to be a $\mathbb{Q}^{n-1}$ Brownian motion with correlation matrix ρ on $[0, t_n]$, it is sufficient that μ satisfies $d\nu_t^n = \mu_t \nu_t^n d\mathbb{W}_t^n$, where $\nu_t^n = \frac{d\mathbb{Q}^{n-1}}{d\mathbb{Q}^n}\,|_{\mathfrak{F}_t}$. Now, by (24)-(25), we have

$$d\nu_t^n = \nu_t^n \frac{hL_t^n s_n(t)}{1 + hL_t^n} d\mathbb{W}_t^n.$$

[26] self-financing condition relative to the numéraire B^i.

[27] cf. also §1.C.

[28] swaptions are also very sensitive to the term-structure of the volatility, whereas caps and floors are only sensitive to the integrated variance (Rebonato, 2005).

[29] cf. I.(22).

We thus set $\mu_t = \frac{hL_t^n s_n(t)}{1+hL_t^n}$ and

$$\begin{aligned} dL_t^{n-1} &= s_{n-1}(t)L_t^{n-1}d\mathbb{W}_t^{n-1} \\ &= s_{n-1}(t)L_t^{n-1}\left(d\mathbb{W}_t^n - \frac{hL_t^n \rho s_n^\top(t)}{1+hL_t^n}dt\right) \\ &= \sigma_{n-1}(t)L_t^{n-1}dW_t^{n-1,n} - \frac{hL_t^n\sigma_n(t)\rho_{n,n-1}}{1+hL_t^n}\sigma_{n-1}(t)L_t^{n-1}dt. \end{aligned}$$

Iterating this construction, we likewise have, for i decreasing from $n-1$ to 1, $d\mathbb{W}_t^i = d\mathbb{W}_t^{i+1} - \frac{hL_t^{i+1}\rho s_{i+1}^\top(t)}{1+hL_t^{i+1}}dt$ and, for $i \le n$,

$$\begin{aligned} dL_t^i &= s_i(t)L_t^i d\mathbb{W}_t^i \\ &= s_i(t)L_t^i\left(d\mathbb{W}_t^n - \sum_{l=i+1}^n \frac{hL_t^l \rho s_l^\top(t)}{1+hL_t^l}dt\right) \\ &= \sigma_i(t)L_t^i dW_t^{i,n} - \sum_{l=i+1}^n \frac{hL_t^l\sigma_l(t)\rho_{l,i}}{1+hL_t^l}\sigma_i(t)L_t^i dt \end{aligned} \tag{26}$$

or, in log-returns,

$$d\ln(L_t^i) = \sigma_i(t)dW_t^{i,n} - \left(\sum_{l=i+1}^n \frac{hL_t^l\sigma_l(t)\rho_{l,i}}{1+hL_t^l}\sigma_i(t) + \frac{1}{2}\sigma_i(t)^2\right)dt. \tag{27}$$

In particular, L^i is a $\mathbb{Q}^i$ martingale[30], for every $i \le n$. But, for every $i < n$, L^i has a non vanishing $\mathbb{Q}^n$ drift that depends on the $L^l, l > i$.

D.1 Correlation Structures

To set ρ, a first possibility is an historical estimate of the correlation matrix of the Libor. But a generally preferred alternative is to calibrate[31] a parametric form of ρ to market quotes of swaptions (described below). Various parameterizations of ρ are classically used in this calibration, such as $\rho_{i,l} = \exp(-\gamma|i-l|)$, or

$$\rho_{i,l} = \rho_\infty + (1-\rho_\infty)\exp\left[-|t_i - t_l|\gamma(t_i,t_l)\right], \tag{28}$$

with $\gamma(t_i,t_l) = \gamma_1 - \gamma_2\max(t_i,t_l)$, or

$$\rho_{i,l} = \exp\left[-\frac{|i-l|}{n-1}\left(-\ln\nu_\infty + \eta_1\varphi(i,l,n) + \eta_2\psi(i,l,n)\right)\right], \tag{29}$$

with

$$\begin{aligned} \varphi(i,l,n) &= \frac{i^2+l^2+il-3ni-3nl+3i+3l+2n^2-n-4}{(n-2)(n-3)} \\ \psi(i,l,n) &= \frac{i^2+l^2+il-ni-nl-3i-3l+3n+2}{(n-2)(n-3)}. \end{aligned}$$

Note, however, that the formula (28) can fail to define a correlation matrix for some values of its parameters. The formula (29), at least, is known to produce a correlation matrix provided $0 \le \eta_2 \le 3\eta_1$ and $0 \le \eta_1 + \eta_2 \le -\ln\nu_\infty$ (Brigo and Mercurio, 2007).

[30] assuming $\Sigma_i^2(t)$ finite in (20).
[31] see Chapter VII.

E Swaptions

The value at time $t \le t_1$ of the swap of Example 2(i) is given by (21). The corresponding forward swap rate S_t, i.e. the value of K for which (21) vanishes, is given, for $t \le t_1$, by

$$S_t = \frac{\sum_{i=1}^n hB_t^i L_t^i}{\sum_{i=1}^n hB_t^i} = \frac{B_t^0 - B_t^n}{\sum_{i=1}^n hB_t^i},$$

by (17). Whence the alternative swap valuation formula

$$\sum_{i=1}^n hB_t^i (L_t^i - K) = \sum_{i=1}^n hB_t^i (S_t - K). \tag{30}$$

The related swaption with maturity t_1 is an option to enter the swap at the maturity time t_1. In view of (30), the swaption's payoff is worth

$$(\textstyle\sum_{i=1}^n hB_{t_1}^i)(S_{t_1} - K)^+.$$

In the Libor market model, the swap rate process S_{t_1} is not lognormal. But it is numerically close to it, for a squared variance Σ_0^2 satisfying the Rebonato formula[32]

$$\Sigma_0^2 t_1 = \frac{1}{S_0^2} \sum_{i,l=1}^n w^i w^l L_0^i L_0^l \rho_{i,l} \int_0^{t_1} \sigma_i(t)\sigma_l(t)dt, \tag{31}$$

for weights w^l proportional to the hB_0^l. Working under the pricing measure corresponding to the numéraire $A = \sum_{i=1}^n hB^i$ then yields the following approximate time-0 price-and-delta formulas for a swaption in the LMM:

$$A_0 c^{bl}(0, S_0, t_1, K; \Sigma_0)\,,\ \delta^{bl}(0, S_0, t_1, K; \Sigma_0),$$

where Σ_0 is given by (31).

F Model Simulation

Interest rate derivatives cashflows are typically given as functions ϕ of the $L_{t_j}^i$. One thus has for the cap of Example 2(ii), by (24),

$$\begin{aligned} C_0 &= B_0^n \sum_{i=1}^n \mathbb{E}^n \left[\frac{h(L_{t_i}^i - K)^+}{B_{t_{i+1}}^n} \right] \\ &= B_0^n \mathbb{E}^n \left[\sum_{i=1}^n h(L_{t_i}^i - K)^+ \prod_{l=i+1}^n (1 + hL_{t_{i+1}}^l) \right], \end{aligned}$$

by (18). To properly discount each cash flow $h(L_{t_i}^i - K)^+$ under $\mathbb{Q}^n$, one thus needs to know the values of the $L_{t_{i+1}}^l$ with $l > i$.

For pricing and greeking by Monte Carlo in the LMM, one can simulate the $\mathbb{Q}^n$ dynamics (26) of the vector-process L by an Euler scheme[33] on a time-grid refining the tenor structure. Hull and White (2000) report that, for standard tenor lengths such as $h = 3m$ to $6m$, discretizing (27) by an Euler scheme at tenor dates is accurate enough[34]. Starting from L_0, deduced from $B_0 = (B_0^l)_{0 \le l \le n}$ by (18),

[32] see Brigo and Mercurio (2007, p. 248).

[33] see IV.§7.B.1.

[34] this at least holds provided t_0 is close enough to t_1; otherwise, of course, the discretization grid must be refined between t_0 and t_1.

we then obtain via (27), for every $i = 0, \dots, n-1$ and $l = i+1, \dots, n$:

$$L^l_{t_{i+1}} = L^l_{t_i} \exp\left[\sigma_l(t_i)\sqrt{h}(\Lambda\varepsilon^i)_l - \left(\sum_{k=l+1}^{n} \frac{hL^k_{t_i}\sigma_k(t_i)\rho_{k,l}}{1+hL^k_{t_i}}\sigma_l(t_i) + \frac{1}{2}\sigma^2_l(t_i)\right)h\right],$$

where ε^i represents a vector of n independent Gaussian random variables and the matrix Λ is a square-root of ρ, i.e. $\Lambda\Lambda^\top = \rho$. A square-root Λ of ρ can be obtained, as in item (i) below, by Cholesky decomposition of ρ[35].

Example 3 **(i)** *In the case* $n = 2 : \mathbb{W}^2 = \begin{pmatrix} W^{1,2} \\ W^{2,2} \end{pmatrix}$, *we have*

$$\frac{d\langle W^{1,2}, W^{2,2}\rangle_t}{dt} = \begin{pmatrix} 1 & \varrho \\ \varrho & 1 \end{pmatrix} = \Lambda\Lambda^\top, \tag{32}$$

where $\Lambda = \begin{pmatrix} 1 & 0 \\ \varrho & \sqrt{1-\varrho^2} \end{pmatrix}$.

(ii) *In the fully correlated case where* ε^i_l *does not depend on* l, *one standard univariate Gaussian draw* $\varepsilon^i_l = \varepsilon_i$ *per time step is enough to simulate a model trajectory. Table 1 shows one trajectory of the Libor thus simulated for* $n = 4$, $\sigma_l = 15\%$, $h = 0.5$, *and* $L^l_0 = 5\%$ *(initial term structure flat at the level* 5%*).*

t	0	t_1	t_2	t_3	t_4
$\sqrt{h}\varepsilon_i$	-0.371379	1.81768	-0.204069	0.512108	
L^1	5%	4.698%			
L^2	5%	4.699%	6.135%		
L^3	5%	4.701%	6.138%	5.918%	
L^4	5%	4.702%	6.142%	5.923%	6.36%

Table 1: Trajectory simulated in a one-factor LMM.

G Extensions

G.1 Beyond Black

Like Black-Scholes on equities, the above log-normal interest rate derivatives model is strongly misspecified. The industry standard is to use smiled extensions of this basic setup, such as the local stochastic volatility models of §4.

G.2 Multi-curve models

The interbank loan market has been very severely impacted by the global financial crisis of 2007–09 and the ensuing liquidity squeeze. In parallel to the drying up of the interbank loan market, Libor got disconnected from OIS rates (overnight or money market rates and the related derivatives) while, as more and more trades have become collateralized, their effective funding rate is the corresponding collateral repo rate, which is typically indexed on OIS. This has created a situation where the price of an interest-rate product (even the simplest flow instrument such as a FRA) involves (at least) two curves, an OIS and a Libor curve (or even distinct Libor curves corresponding to segmented Libor markets of different tenor lengths h), which entails various valuation adjustments (Mercurio, 2010; Filipović and Trolle, 2013; Crépey and Douady, 2013; Henrard, 2014)).

[35] see IV.§3.C.

G.3 The Libor transition

The situation of Libor where an underlying to financial derivatives was somehow arbitrarily fixed by a panel of key players in the market has posed major insider issues culminating in the Libor 2012 scandal. As a result, the Libor are progressively replaced by alternative benchmark rates in the form of suitable risk-free rates plus spreads (Berndt, Duffie, and Zhu, 2020; Albanese, Iabichino, and Mammola, 2020). Lyashenko and Mercurio (2019)'s related forward market model (FMM) extends the LMM by considering a stochastic evolution of (suitably modified) Libor forward rates beyond their fixing dates, i.e. of the above process L^i until t_{i+1}, for some volatility decaying to 0 on the time interval $[t_i, t_{i+1}]$ (as opposed to a volatility killed at t_i in the above). The ISDA fallback protocol, which has been set in place for ensuring a smooth transition from Libor to their replacement rates, raises various quantitative issues that have been analyzed in Henrard (2019); Piterbarg (2020a,b). See also (Andersen and Bang, 2020; Klinglera and Syrstadb, 2021).

§3 One-Factor Gaussian Copula Model of Portfolio Credit Risk

We now move to portfolio credit derivatives. The market model in this regard is still the static one-factor Gaussian copula model[36], first introduced in Li (2000).

Remark 3 *It was said that the Gaussian copula model was responsible for huge losses on CDOs in the 2007–2009 credit crisis. Of course, from theoretical point of view the shortcomings of the model are quite obvious. In particular this is a static quotation device, rather than a full-flesh dynamic model*[37]*. However, the biggest losses on securitization products in the credit crisis occurred not on synthetic CDOs that were actively risk managed by banks, with Gaussian copula deltas as explained in §5.B, but on cash CDOs, often in the form of ABS held by nonbanking institutions, which were simply not risk-managed. It is thus not the Gaussian copula model which is so much to blame here, but rather the regulation, which made it possible for banks to externalize such risks to institutions not committed to hedge them.*

One considers d reference entities (firms) with respective default (stopping) times and constant loss-given-defaults denoted by τ_l and Λ_l, for $l = 1, \dots, d$. The cumulative portfolio loss at time t is given by

$$\mathcal{L}_t = \sum_{l=1}^{d} \Lambda_l \mathbb{1}_{\{\tau_l \le t\}}. \tag{33}$$

A Credit Derivatives

A.1 Single-Name CDSs

A single-name credit default swap (CDS for short) on name l, with contractual spread S, has cumulative discounted cash flows given by, from the point of view of the seller of default protection:

$$\int_0^T \beta_t \left(S J_t^l dt + \Lambda_l dJ_t^l \right),$$

where $J^l = \mathbb{1}_{[0,\tau_l)}$ is the survival indicator process of name l. The related $\mathbb{Q}$ price[38] process, for $t \in [0, T]$, is written

$$\mathbb{E}_t \int_t^T \beta_s \left(S J_s^l ds + \Lambda_l dJ_s^l \right).$$

[36] In combination with stochastic recoveries since the 2007–2009 credit crisis.

[37] even though it is possible to dynamize this setup by the introduction of a relevant filtration, see Crépey, Bielecki, and Brigo (2014, Chapter 7) and Crépey and Song (2017).

[38] the financial meaning of which would remain to specify, in an incomplete credit market setup.

The CDS spread S is typically set such that the CDS is entered at no cost at inception so that, assuming deterministic interest rates $r_t = r(t)$[39],

$$S = \frac{\Lambda_l \mathbb{E}\int_0^T \beta_t \Lambda_l dJ_t^l}{\int_0^T \beta_t \mathbb{Q}(\tau_l > t)dt} = \frac{\Lambda_l \mathbb{E}(\beta_{\tau_l} \mathbb{1}_{\{\tau_l \le T\}})}{\int_0^T \beta_t \mathbb{Q}(\tau_l > t)dt} = \frac{\Lambda_l \int_0^T \beta_t dF_l(t)}{\int_0^T \beta_t (1 - F_l(t))dt},$$

where $F_l(t) = \mathbb{Q}(\tau_l \le t)$. So the value of either leg of the CDS on name l only depends on the law of τ_l.

A.2 CDO Tranches

A single tranche collateralized debt obligation (CDO) with attachment point a, detachment point $b > a$, maturity T and contractual spread Σ is an option with the following cumulative discounted cash flows, from the point of view of a seller of default protection:

$$\int_0^T \beta_t \left[\Sigma(b - a - L_t)dt - dL_t\right],$$

where

$$L_t = (\mathcal{L}_t - a)^+ - (\mathcal{L}_t - b)^+ = \min\left((\mathcal{L}_t - a)^+, b - a\right)$$

is the cumulative tranche loss. A CDO tranche can thus be interpreted as a bull spread[40] with strikes a and b on the porfolio loss $\mathcal{L}_t$. For instance, on the DJ iTraxx market, which is a family of CDS indices for Europe and Asia, CDO tranches were liquidly quoted until May 2009 for (a, b) in $(0\%, 3\%)$, $(3\%, 6\%)$, $(6\%, 9\%)$, $(9\%, 12\%)$ and $(12\%, 22\%)$. The tranches $(0\%, 3\%)$ and $(9\%, 12\%)$ are respectively called the equity and the senior tranche, whereas intermediate tranches are known as the mezzanine tranches. Since the 2007–09 credit crisis, only the indices are still quoted, no longer the tranches. However CDOs are still relevant in terms of risk-management as they are still present in bank portfolios.

The price process of a tranche is given, for $t \in [0, T]$, by

$$\mathbb{E}_t \int_t^T \beta_s \left[\Sigma(b - a - L_s)ds - dL_s\right]. \tag{34}$$

The tranche spread Σ is typically set such that the tranche is entered at no cost at inception, so that

$$\Sigma = \frac{\mathbb{E}\int_0^T \beta_t dL_t}{\mathbb{E}\int_0^T \beta_t (b - a - L_t)dt}.$$

Assuming deterministic interest rates $r_t = r(t)$, the values of either leg of the CDO only depends on the expected tranche losses $\mathbb{E}L_t$, $t \in [0, T]$. In fact, one has for the fees leg:

$$\mathbb{E}\int_0^T \beta_t (b - a - L_t)dt = \int_0^T \beta_t (b - a - \mathbb{E}L_t)dt. \tag{35}$$

For the protection leg, observe that $d(\beta_t L_t) = \beta_t (dL_t - r_t L_t dt)$ and $L_0 = 0$. Hence Fubini's theorem yields

$$\mathbb{E}\int_0^T \beta_t dL_t = \beta_T \mathbb{E}L_T + \int_0^T r_t \beta_t \mathbb{E}L_t dt. \tag{36}$$

B Gaussian Copula Model

Let there be given a Gaussian vector $X = (X_1, \ldots, X_d)$ with covariance matrix

$$\begin{pmatrix} 1 & \rho & \ldots & \rho \\ \rho & 1 & \ddots & \vdots \\ \vdots & \ddots & 1 & \rho \\ \rho & \ldots & \rho & 1 \end{pmatrix}. \tag{37}$$

[39] which can be extracted from the market zero-coupon curve.
[40] cf. Example I.1.

More specifically, we assume

$$X_l = \sqrt{\rho}Y + \sqrt{1-\rho}\,Y_l\,,\ l = 1,\dots,d, \tag{38}$$

for an independent standard Gaussian common factor Y and independent Gaussian random variables Y_l.

The Gaussian copula model of the defaults times $\tau_l, l = 1,\dots,d$, with respective marginal cumulative distribution functions F_l, is

$$\tau_l = F_l^{-1}(\mathcal{N}(X_l))\ \text{, i.e. } X_l = \mathcal{N}^{-1}(F_l(\tau_l))\,,\ l = 1,\dots,d, \tag{39}$$

where $\mathcal{N}$ represents the standard univariate Gaussian cumulative distribution function.

The X_l and therefore the τ_l are conditionally independent given Y in (38). For $l = 1,\dots,d$ and $t \geq 0$, let

$$x_t^l = \mathcal{N}^{-1}(F_l(t)) \tag{40}$$

and

$$p_t^{l|y} := \mathbb{Q}(\tau_l \leq t\,|\,Y = y) = \mathbb{Q}(X_l \leq x_t^l\,|\,Y = y) = \mathcal{N}\left(\frac{x_t^l - \sqrt{\rho}y}{\sqrt{1-\rho}}\right), \tag{41}$$

by (38). By conditional independence given Y, the joint cumulative distribution function F of the τ_l satisfies, for every nonnegative constants $t_l, l = 1,\dots,d$,

$$F(t_1,\dots,t_d) = \mathbb{E}\left[\mathbb{Q}\left[X_1 \leq x_{t_1}^1,\dots,X_d \leq x_{t_d}^d\,\middle|\,Y\right]\right] = \int_{-\infty}^{\infty}\prod_{l=1}^{d} p_{t_l}^{l|y} g(y)dy,$$

where g is the standard Gaussian density. In view of (33), the moment generating function $\Psi_{\mathcal{L}_t}(u) = \mathbb{E}[e^{u\mathcal{L}_t}]$ of the portfolio loss $\mathcal{L}_t$ is given, with $\Psi_{\mathcal{L}_t}^Y(u) = \mathbb{E}[e^{u\mathcal{L}_t}|Y]$, by

$$\begin{aligned}\Psi_{\mathcal{L}_t}(u) &= \mathbb{E}_Y \Psi_{\mathcal{L}_t}^Y(u) \\ &= \mathbb{E}[\prod_{l=1}^{d}\mathbb{E}[e^{u\Lambda_l \mathbb{1}_{\{\tau_l \leq t\}}}|Y]] = \int_{-\infty}^{\infty}\prod_{l=1}^{d}\left(1 - p_t^{l|y} + p_t^{l|y}e^{u\Lambda_l}\right) g(y)dy.\end{aligned} \tag{42}$$

Remark 4 *A copula function C is the joint cumulative distribution function of an $\mathbb{R}^d$ valued random vector with uniform marginals on $[0,1]$, where the latter is tantamount to*

$$C(1,\dots,1,u,1,\dots,1) = u\,,\ u \in [0,1], \tag{43}$$

for every component of C. Sklar's theorem states that for every joint cumulative distribution function $F = F(t_1,\dots,t_d)$ with marginal cumulative distribution functions $F_l = F_l(t_l)$, $l = 1\dots d$, there exists a copula function C such that, for any $t_1,\dots,t_d$,

$$F(t_1,\dots,t_d) = C\left[F_1(t_1),\dots,F_d(t_d)\right]. \tag{44}$$

The one-factor Gaussian copula model corresponds to the so-called one-factor Gaussian copula

$$C_\rho(u_1,..,u_d) := \mathcal{N}_\rho\left[\mathcal{N}^{-1}(u_1),..,\mathcal{N}^{-1}(u_d)\right], \tag{45}$$

where $\mathcal{N}_\rho$ represents the d variate Gaussian cumulative distribution function with covariance matrix (37). In fact, it is not hard to check[41] *that C_ρ is a multivariate cdf, recalling the following analytic characterization of the latter: nondecreasing and right-continuous in each of its variables, $[0,1]$ valued with respective limits 0 and 1 when its arguments jointly converge to $\pm\infty$, respectively. The additional property (43) of C_ρ follows from the fact that each component of a Gaussian vector with covariance matrix (37) is standard Gaussian. So C_ρ is indeed a copula function. Moreover, for every nonnegative $t_1,\dots,t_d$, (38)–(40) yield*

$$\begin{aligned}\mathbb{Q}(\tau_l \leq t_l\,,\ l = 1,\dots,d) &= \mathbb{Q}(X_l \leq x_{t_l}^l\,,\ l = 1,\dots,d) \\ &= \mathcal{N}_\rho\left[\mathcal{N}^{-1}(F_l(t_l))\,,\ l = 1,1\dots,d\right],\end{aligned}$$

which corresponds to the identity (44) for $C = C_\rho$ as per (45).

[41] left to the reader.

B.1 Exact CDO Pricing Schemes

We assume in this subsection that the loss-given-defaults Λ_l are commensurate or, more specifically and without loss of generality, natural numbers. With $q_t^k = \mathbb{Q}(\mathcal{L}_t = k)$ and $\Lambda = \sum_{l=1}^d \Lambda_l$ we thus have

$$\Psi_{\mathcal{L}_t}(u) = \sum_{k=0}^{\Lambda} q_t^k e^{uk}\,,\ \mathbb{E}L_t = \sum_{k=0}^{\Lambda} \big((k-a)^+ \wedge (b-a)\big) q_t^k. \tag{46}$$

The expected tranche loss $\mathbb{E}L_t$ is thus a function of the portfolio loss distribution $q_t = (q_t^k)_{0\le k\le\Lambda}$. This distribution q_t can be computed by numerical inversion of the Laplace transform $\Psi_{\mathcal{L}_t}$. Choosing $(\Lambda+1)$ as a power of 2, this inversion can be done in time $\mathrm{O}(\Lambda \ln \Lambda)$ by fast Fourier transform[42].

Alternatively, the portfolio loss distribution q_t can be computed recursively as follows. Let $q_t^{k|y}$ and $q_t^{k|y}(i)$ respectively represent $\mathbb{Q}(\mathcal{L}_t = k|y)$ and the corresponding (conditional) probability for the sub-portfolio restricted to the l first credit names of the full portfolio, so that $q_t^{k|y} = q_t^{k|y}(d)$.

Lemma 4 *We have $q_t^{k|y}(0) = \mathbb{1}_{k=0}$ and, for every $l = 1,\dots,d$ and $k = 0,\dots,\Lambda$:*

$$q_t^{k|y}(l) = p_t^{l|y} q_t^{k-\Lambda_l|y}(l-1) + (1-p_t^{l|y}) q_t^{k|y}(l-1). \tag{47}$$

Proof. Let $q_t^{k,\epsilon|y}(i)$, for $\epsilon = 0, 1$, represent the conditional probability that the aggregated loss over the i first names of the portfolio equals k and $\mathbb{1}_{\tau_i\le t} = \epsilon$. Using I_t^l as a shorthand for $\mathbb{1}_{\tau_i\le t}$, we then have

$$\begin{aligned} q_t^{k|y}(l) &= q_t^{k,I_t^l=1|y}(l) + q_t^{k,I_t^l=0|y}(l) \\ &= q_t^{k-\Lambda_l,I_t^l=1|y}(l-1) + q_t^{k,I_t^l=0|y}(l-1) \\ &= q_t^{k-\Lambda_l|y}(l-1)p_t^{l|y} + q_t^{k|y}(l-1)(1-p_t^{l|y}), \end{aligned}$$

where the first two identities are elementary and the last one follows by conditional independence with respect to Y. ∎

Once the conditional loss distribution $q_t^{k|y} = q_t^{k|y}(d)$ has been recursively computed using (47) and (41), by numerical integration we recover

$$q_t^k = \int_{-\infty}^{\infty} q_t^{k|y} g(y) dy.$$

Recursive relations analogous to (47) can also be derived for the sensitivities of the tranches with respect to the input data $F_l(t)$. Following Andersen and Sidenius (2004), we first compute $\partial_{F_l(t)}\mathbb{E}(L_t|Y)$ from

$$\big(\partial_{p_t^{l|y}}\mathbb{E}(L_t|Y)\big)\big(\partial_{F_l(t)}p_t^{l|y}\big) = \big(\partial_{p_t^{l|y}}\mathbb{E}(L_t|Y)\big)\frac{\partial_{x_t^l}p_t^{l|y}}{\partial_{x_t^l}F_l(t)}. \tag{48}$$

Now, by (41),

$$\partial_{x_t^l}p_t^{l|y} = \frac{1}{\sqrt{2\pi(1-\rho)}}\exp\Big(\frac{-(x_t^l-\sqrt{\rho}Y)^2}{2(1-\rho)}\Big)\,,\ \partial_{x_t^l}F_l(t) = \frac{1}{\sqrt{2\pi}}\exp\Big(\frac{-(x_t^l)^2}{2}\Big).$$

Moreover, by (46),

$$\partial_{p_t^{l|y}}\mathbb{E}(L_t|Y) = \sum_{k=0}^{\Lambda}\big((k-a)^+\wedge(b-a)\big)\partial_{p_t^{l|y}}q_t^{k|y}, \tag{49}$$

in which, by (47),

$$\partial_{p_t^{l|y}}q_t^{k|y} = q_t^{k-\Lambda_l|y}(\{l\}) - q_t^{k|y}(\{l\}),$$

where $q_t^{\cdot|y}(\{l\})$ denotes the conditional distribution of the aggregated loss over the portfolio deprived of name l, which can be computed recursively along the same lines as (47). We then obtain the unconditional sensitivity $\partial_{F_l(t)}\mathbb{E}L_t$ by numerical integration with respect to the Gaussian density $g(y)$, the way explained in Andersen and Sidenius (2004, Equation (7)). We finally recover the sensitivities of either leg of the CDO based on the considerations ending Paragraph A.2.

[42] FFT, see the end of I.§1.F.2.

B.2 Approximate CDO Pricing Schemes

Fast approximate schemes may also be used to compute the portfolio loss distribution q_t. Moreover, these approximate schemes do not require the assumption of commensurate losses.

We refer the reader to El Karoui and Jiao (2009); El Karoui, Jiao, and Kurtz (2008) for an efficient, easy to implement and mathematically justified approach based on Gauss-Poisson approximations of the portfolio conditional loss distributions. Related yet heuristic and harder to implement saddle-point methods (see Yang, Hurd, and Zhang (2006)) are based on the following inverse Laplace transform representation for $q_t^{x|y} = \frac{\mathbb{Q}(\mathcal{L}_t \in dx|Y=y)}{dx}$, which holds in a weak sense as explained below[43]:

$$q_t^{x|y} = \frac{1}{2\pi i} \int_{\eta - i\infty}^{\eta + i\infty} \Psi_{\mathcal{L}_t}^y(u) e^{-ux} du, \tag{50}$$

where $\Psi_{\mathcal{L}_t}^y(u) = \mathbb{E}(e^{u\mathcal{L}_t}|Y = y)$. The integration is parallel to the imaginary axis in the complex plane, with $\eta > 0$. By (50) in the weak sense, we mean that

$$\mathbb{E}\left[\varphi(\mathcal{L}_t)|Y\right] = \frac{1}{2\pi i} \int_{\eta - i\infty}^{\eta + i\infty} \left(\int_0^\infty e^{-ux} \varphi(x) dx \right) \Psi_{\mathcal{L}_t}^Y(u)\, du, \tag{51}$$

for every "regular enough" function φ, including call/put payoffs. Noting that $\partial_x \left[(ux+1)e^{-ux}\right] = -u^2 x e^{-ux}$, so $\int_0^\infty x e^{-ux}\, dx = u^{-2}$ and $\int_0^\infty e^{-ux}(x-a)^+ dx = e^{-ua} \int_a^\infty e^{-u(x-a)}(x-a) dx = e^{-ua} u^{-2}$, we obtain by application of (51) to $\varphi(\mathcal{L}_t) = (\mathcal{L}_t - a)^+$:

$$\mathbb{E}\left[(\mathcal{L}_t - a)^+|Y\right] = \frac{1}{2\pi i} \int_{\eta - i\infty}^{\eta + i\infty} \Psi_{\mathcal{L}_t}^Y(u) e^{-ua} u^{-2}\, du. \tag{52}$$

Saddle-point methods are then based on the approximation of $\Psi_{\mathcal{L}_t}^Y(u)e^{-ua}$ in (52). Saddle-point methods are then based on the approximation of $\Psi_{\mathcal{L}_t}^Y(u)e^{-ua}$ in (52) by suitable Taylor expansions around a well chosen point $u^\star$, so that the resulting integral can be computed explicitly. Depending on the expansion point $u^\star$ and the order of the expansion, we obtain a whole family of approximate pricing schemes. In the simplest case we recover the large portfolio approximation of (Vasicek, 1991).

A last possibility for computing the portfolio loss distribution q_t, or the value of a CDO tranche directly, is to proceed by Monte Carlo simulation. But simulation methods are much slower on these problems than any of the previous procedures: Gauss–Poisson or saddle-point approximations, or even, assuming commensurate Λ_l, exact fast Fourier transform or recursive schemes. Note that the integrals in all these algorithms, all of which involve the Gaussian kernel $g(y)dy$, can be computed efficiently by Gauss–Hermite quadrature.

B.3 Gaussian Copula Implied Correlation

Much like the Black-Scholes model with respect to implied volatilities, the one-factor Gaussian copula model is also used in the reverse-engineering model, for quoting CDO tranches in terms of their implied correlations. A preliminary step consists in inferring the cumulative distribution functions F_l from the respective CDS markets, based on the following pricing equations assuming deterministic interest rates[44]:

$$S_t^l(T) \int_t^T \beta_s (1 - F_l(s))\, ds - \Lambda_l \int_t^T \beta_s dF_l(s) = 0, \tag{53}$$

where $S_t^l(T)$ denotes the fair spread at time t of a CDS with maturity T on name l of the pool. For a credit name l with quoted $S_t^l(T), T \geq t$, this relation allows us to bootstrap the function F_l (for $s \geq t$) from the corresponding market CDS curve at time t. Then, given $F = (F_l(s))_{1 \leq l \leq d}$ for $s \geq t$[45]:

[43] note that the formula (50) formally corresponds by change of variable $u = iz$ to IX.(44) for $\mathcal{F}f(z) = \Phi(z) = \mathbb{E}[\exp(izX)]$ there, cf. also IX.(46). But of course the law of $\mathcal{L}_t$ is discrete, hence does not admit a density, at least not in the usual ("strong") sense.

[44] see §3.A.1.

[45] see (O'Kane and Livesey, 2004)

- the compound implied correlation of a tranche is defined as the value of the correlation $\rho_t^\dagger$ in a one-factor Gaussian copula model such that

$$\Sigma^{gc}(t,F,T,a,b;\rho_t^\dagger)=\Sigma_t^\star(T,a,b),$$

  where $\Sigma_t^\star(T,a,b)$ denotes the market tranche spread at time t and Σ^{gc} is the Gaussian copula CDO tranches pricing function;

- the base implied correlation of a tranche is defined as the value of the correlation ρ_t in a one-factor Gaussian copula model such that

$$\Sigma^{gc}(t,F,T,0,b;\rho_t)=\Sigma_t^\star(T,0,b),$$

  where $\Sigma_t^\star(T,0,b)$ denotes a synthetic market spread reconstructed from the observed market spreads of the tranches with detachment point b and below.

The base implied correlation is more stable numerically than the compound implied correlation, because the function Σ^{gc} is decreasing in its argument ρ for $a=0$, but not for $a>0$.

§4 Local Stochastic Volatility

A SABR Model

Paragraphs A.1 and A.2 below are based on https://en.wikipedia.org/wiki/SABR_volatility_model, to which we refer the reader for further developments, including how to deal with the arbitrage isssues in the SABR implied volatility formula for very low strikes or how to accomodate negative rates in a SABR setup.

Hagan, Kumar, Lesniewski, and Woodward (2002)'s SABR model[46] is a hybrid local volatility/stochastic volatility model with explicit asymptotics for the implied volatility and flexible smile dynamics (at least, for a fixed maturity T). The name stands for "stochastic alpha, beta, rho", referring to the parameters of the model. The SABR model is widely used in the industry, especially in the interest rate derivative markets.

A.1 Dynamics

The SABR model describes a single T forward price or rate, such as a forward stock price[47], a LIBOR forward rate[48] or a forward swap rate[49]. For concreteness we assume the T forward neutral setup of I.§2.A.1, but we dismiss all the indices $\cdot^T$ to alleviate the notation. The instantaneous (realized) volatility of the forward F is described by a parameter σ. SABR is a dynamic model in which both F and σ are represented by stochastic processes whose time evolution is given by the following system of stochastic differential equations:

$$dF_t=\sigma_t F_t^\beta\,dW_t,\quad d\sigma_t=\alpha\sigma_t\,dZ_t, \tag{54}$$

starting from the (assumed observed) time-0 values F_0 and σ_0. Here, W and Z are two standard Brownian motions with correlation coefficient $-1<\rho<1$ under the (T forward neutral) probability measure $\mathbb{Q}$. The constant parameters β, α satisfy the conditions $0\le\beta\le1$, $\alpha\ge0$. α is a volatility-like parameter for the volatility. ρ is the instantaneous correlation between the underlying and its volatility. The above dynamics is a stochastic version of the constant elasticity of variance (CEV) model with the

[46] see also (Rebonato, McKay, and White, 2009; Brigo and Mercurio, 2007; Antonov, Konikov, and Spector, 2019).
[47] see I.§2.A.1.
[48] see §2.B.
[49] see §2.E.

"skewness" parameter β: in fact[50], it reduces to the CEV model if $\alpha = 0$; the parameter α is often referred to as the "volvol", and its meaning is that of the lognormal volatility of the volatility parameter σ.

Remark 5 *The process F in (54) is a $\mathbb{Q}$ local martingale, so that the SABR model is non arbitrable. However, some restrictions on its parameters are necessary to ensure that F is a $\mathbb{Q}$ martingale (so that we are in line with the setup of Section I.§1). In the CEV $\alpha = 0$ case, this is the case iff $\beta \leq 1$ (Jeanblanc, Yor, and Chesney, 2009, page 366).*

A.2 Asymptotic solution

We consider a European call option on the forward F struck at K, which expires T years from now. By I.(49), the time 0 $\mathbb{Q}$ price of the option is equal to the time-0 (observed) price of the T discount bond times the expected value of the payoff $\max(F_T - K,\ 0)$ under the model (54) for the process F.

Except for the special cases of $\beta = 1$ and $\beta = 0$[51], no closed form expression for the distribution of F_T is known. The general case can be solved approximately by means of an asymptotic expansion in the parameter $\epsilon = T\alpha^2$. Under typical market conditions, this parameter is small and the approximate solution is quite accurate. Also significantly, this solution has a rather simple functional form, is very easy to implement, and lends itself well to risk management of large portfolios of options in real time.

It is convenient to express the solution in terms of the time-0 Black implied volatility[52] ${}^{bl}\Sigma_0 = {}^{bl}\Sigma_0(T,K)$ of the option. Namely, the value ${}^{bl}\Sigma_0$ of the lognormal volatility parameter in Black's model that forces it to match the SABR time-0 price is approximately given by

$$
\begin{aligned}
{}^{bl}\Sigma_0^{sabr} = \alpha\, \frac{\log(F_0/K)}{D(\zeta)} \left\{ 1 + \left[\frac{2\gamma_2 - \gamma_1^2 + 1/\left(F_{ge}\right)^2}{24} \left(\frac{\sigma_0 C(F_{ge})}{\alpha} \right)^2 + \right.\right. \\
\left.\left. \frac{\rho\gamma_1}{4} \frac{\sigma_0 C(F_{ge})}{\alpha} + \frac{2-3\rho^2}{24} \right] \epsilon \right\},
\end{aligned}
\tag{55}
$$

where $F_{ge} = \sqrt{F_0 K}$ and, for clarity, we have set $F^\beta = C\left(F\right)$. We have also set

$$
\begin{aligned}
\zeta &= \frac{\alpha}{\sigma_0} \int_K^{F_0} \frac{dx}{C(x)} = \frac{\alpha}{\sigma_0(1-\beta)} \left({F_0}^{1-\beta} - K^{1-\beta} \right), \\
\gamma_1 &= \frac{C'(F_{ge})}{C(F_{ge})} = \frac{\beta}{F_{ge}},\ \gamma_2 = \frac{C''(F_{ge})}{C(F_{ge})} = -\frac{\beta(1-\beta)}{\left(F_{ge}\right)^2},\ \text{and} \\
D(\zeta) &= \log\left(\frac{\sqrt{1-2\rho\zeta+\zeta^2} + \zeta - \rho}{1-\rho} \right).
\end{aligned}
$$

Alternatively, one can express the SABR price in terms of the Bachelier implied volatility[53] (for zero interest and dividend yields too). The corresponding time 0 implied normal volatility ${}^{ba}\Sigma_0$ can be asymptotically computed by

$$
\begin{aligned}
{}^{ba}\Sigma_0^{sabr} = \alpha\, \frac{(F_0 - K)}{D(\zeta)} \left\{ 1 + \left[\frac{2\gamma_2 - \gamma_1^2}{24} \left(\frac{\sigma_0 C(F_{ge})}{\alpha} \right)^2 + \right.\right. \\
\left.\left. \frac{\rho\gamma_1}{4} \frac{\sigma_0 C(F_{ge})}{\alpha} + \frac{2-3\rho^2}{24} \right] \epsilon \right\}.
\end{aligned}
$$

This approximation for the Bachelier implied volatility of SABR prices is generally somewhat more accurate than the approximation (55) for their Black implied volatility.

[50] see A.3 below.
[51] corresponding to the lognomal and normal cases of the Black and Bachelier model.
[52] see the last paragraph of I.§2.A.2.
[53] Gaussian rather than lognormal, see the last paragraph of I.§2.A.2.

A.3 Limiting CEV Case

In the limiting case of the CEV model with $\alpha = 0$ and $\sigma = \sigma_0$ in (54), i.e.

$$dF_t = \sigma_0 F_t^\beta \, dW_t = F_t(\sigma_0 F_t^{\beta-1})dW_t, \tag{56}$$

the asymptotics (55) need to be replaced by

$$\begin{aligned} {}^{bl}\Sigma_0^{cev} = \frac{\sigma_0}{F_{ar}^{1-\beta}} \Bigg\{ 1 + \frac{(1-\beta)(2+\beta)}{24} \left(\frac{F_0 - K}{F_{ar}}\right)^2 + \\ \frac{(1-\beta)^2}{24} \frac{\sigma_0^2 T}{F_{ar}^{2-2\beta}} + \dots \Bigg\}, \end{aligned} \tag{57}$$

where $F_{ar} = \frac{1}{2}(F_0 + K)$. Hence, at first order, ${}^{bl}\Sigma_0^{cev} = \frac{\sigma_0}{F_{ar}^{1-\beta}}$ and

$$\begin{aligned} \partial_K({}^{bl}\Sigma_0^{cev}) \approx \partial_K \frac{\sigma_0}{F_{ar}^{1-\beta}} = \partial_K \frac{\sigma_0}{(\frac{1}{2}(F_0+K))^{1-\beta}} = \\ \sigma_0 2^{1-\beta} \partial_K (F_0 + K)^{\beta-1} = \sigma_0 2^{1-\beta}(\beta-1)(F_0+K)^{\beta-2}. \end{aligned}$$

In particular, at-the-money, $({}^{bl}\Sigma_0^{cev})\big|_{K=F_0} = \sigma_0 F_0^{\beta-1}$ and

$$\begin{aligned} \partial_K({}^{bl}\Sigma_0^{cev})\big|_{K=F_0} \approx \sigma_0 2^{1-\beta}(\beta-1)2^{\beta-2}F_0^{\beta-2} = \\ \frac{1}{2}\sigma_0(\beta-1)F_0^{\beta-2} = \frac{1}{2}\partial_{F_0}(\sigma_0 F_0^{\beta-1}). \end{aligned}$$

Hence, in view of the CEV local volatility $(\sigma_0 F_t^{\beta-1})$ in (56), The ATM Black implied volatility corresponding to CEV prices coincides with the CEV local volatility at $(0, F_0)$ and the ATM Black implied volatility skew is one half of the CEV local volatility skew at $(0, F_0)$.

B A SABR/Bergomi-Type Model of Rough Volatility

This is Section 2 of Fukasawa, Horvath, and Tankov (2021), modulo notational or other minor changes made for consistency of presentation with the present notes.

The purpose of this part is to illustrate an infinite-dimensional Markov nature of rough volatility models, which will however enable us to hedge options without any "memory" of the past: While fractional Brownian motions have memory properties[54], we will see that this memory is stored in an option market.

B.1 The Model

Here we consider a 2 factor model driven by a bivariate two-sided[55] standard Brownian motion $(\widehat{W}^1, \widehat{W}^2)$ on a probability space $(\Omega, \mathfrak{F}, \mathbb{P})$, with filtration $\mathfrak{F} = (\mathfrak{F}_t)_{t\in\mathbb{R}}$ given as the augmentation of the one generated by the Brownian motion. We consider a hypothetical option market where call and put options are traded for all strike prices $K \geq 0$ and maturities $T \geq 0$. Their prices at time $t \geq 0$ are denoted by $C_t(T, K)$ and $P_t(T, K)$ respectively. We assume risk-free and dividend rates are zero for brevity. The underlying asset price process of the options is denoted by S and we suppose

$$C_t(T,K) = \mathbb{E}_t(S_T - K)^+, \quad P_t(T,K) = \mathbb{E}_t(K - S_T)^+, \quad S_t = C_t(0,T)$$

[54] long or short depending on the Hurst exponent $H >$ or $< \frac{1}{2}$.

[55] a two-sided (univariate) standard Brownian motion B can be defined as $B_t = \mathbb{1}_{t>0}B_t^1 + \mathbb{1}_{t<0}B_t^2$, where B^1 and B^2 are independent standard Brownian motions.

for all $K \geq 0$, $T \geq 0$ and $t \geq 0$, where $\mathbb{E} = \mathbb{E}^{\mathbb{Q}}$, in which $\mathbb{Q}$ is a probability measure $\sim$ the physical measure $\mathbb{P}$.

We introduce a SABR/Bergomi-type stochastic volatility model[56]

$$\begin{aligned} dS_t &= f(S_t)\sqrt{V_t^t}\left[\rho dW_t^1 + \sqrt{1-\rho^2}dW_t^2\right], \\ dV_t^u &= V_t^u g(u-t)dW_t^1, \quad t < u \end{aligned} \tag{58}$$

where (W^1, W^2) is a bivariate $(\mathfrak{F}_t)_{t\geq 0}$ Brownian motion under $\mathbb{Q}$, f and g are measurable functions on $\mathbb{R}_+$, and $\rho \in (-1, 1)$. We assume $\mathbb{E}\int_s^t g(u-r)^2 dr < +\infty$, is locally square integrable, so that

$$V_t^u = V_s^u \exp\left\{\int_s^t g(u-r)dW_r^1 - \frac{1}{2}\int_s^t g(u-r)^2 dr\,,\ 0 \leq s \leq t \leq u\right\} \tag{59}$$

and

$$\mathbb{E}_s V_t^t = V_s^t\,,\ 0 \leq s \leq t. \tag{60}$$

The case $f(S) = S$, $g(u) = \eta u^{H-1/2}$, with $H \in (0, \frac{1}{2}]$, corresponds to the rough Bergomi model of Bayer, Friz, and Gatheral (2016).

Remark 6 *We recall from https://en.wikipedia.org/wiki/Fractional_Brownian_motion that a fractional Brownian motion (fBM) is a generalization of Brownian motion. Unlike classical Brownian motion, the increments of fBm need not be independent. fBm is a continuous-time Gaussian process $B_H(t)$ on $[0, T]$ that starts at zero, has expectation zero for all $t \in [0, T]$, and has the following covariance function:*

$$\mathbb{E}[B_H(t)B_H(s)] = \frac{1}{2}(|t|^{2H} + |s|^{2H} - |t-s|^{2H}), \tag{61}$$

where H is a real number in $(0, 1)$, called the Hurst index or Hurst parameter associated with the fractional Brownian motion. The Hurst exponent describes the raggedness of the resultant motion, with a higher value leading to a smoother motion. The value of H determines what kind of process the fBm is: if $H = 1/2$ then the process is in fact a Brownian motion or Wiener process; if $H > 1/2$ then the increments of the process are positively correlated (positively autocorrelated process); if $H < 1/2$ then the increments of the process are negatively correlated (negatively autocorrelated process).

A volatility process driven by a fractional Brownian motion can be treated in the framework (58). For example, if the log (instantaneous realized) volatility is a stationary fractional Ornstein-Uhlenbeck process[57]

$$\ln\sqrt{v_t} = \frac{1}{2}\int_{-\infty}^t g(t-s)d\widehat{W}_s^1, \quad g(u) = \eta u^{H-1/2} - \eta\lambda e^{-\lambda u}\int_0^u r^{H-1/2}e^{\lambda r}dr$$

and the volatility risk premium is deterministic, then we have (58) and (59) with $f(S) = S$,

$$V_t^u = \mathbb{E}_t v_u \text{ for } t \leq u, \tag{62}$$

and a suitable family $\{V_0^t\}_{t\geq 0}$ of $\mathfrak{F}_0$ measurable random variables (recall that $\widehat{W}^1$ is a two-sided Brownian motion).

We call the curve

$$\bar{V}_t : \tau \mapsto V_t^{t+\tau} \tag{63}$$

the forward variance curve at time t.

Proposition 4 *The forward variance curve $\{\bar{V}_t\}_{t\geq 0}$ is a Markov process over the space $\mathcal{C}(\mathbb{R}^+)$ of the continuous functions on $\mathbb{R}_+$.*

[56] of deformation of the forward variance curve, automatically fitting the latter at time 0, cf. (63) below.
[57] see Barndorff-Nielsen and Basse-O'Connor (2011).

Proof. By (63) and (59), we have for $t \geq s$,

$$\bar{V}_t(\tau) = V_t^{t+\tau} = V_s^{t+\tau} \exp\left\{\int_s^t g(\tau + t - u)dW_u^1 - \frac{1}{2}\int_s^t g(\tau + t - u)^2 du\right\},$$

where $V_s^{t+\tau} = \bar{V}_s(\tau + t - s)$. Since the exponential term is independent of $\mathfrak{F}_s$, the result follows. ∎

Corollary 1 *$(S, \bar{V})$ is a Markov process with state space $[0, \infty) \times \mathcal{C}[0, \infty)$.*

Consequently, for any, possibly path-dependent, functional $\chi = \Phi(S_u, t \leq u \leq T)$, its conditional expectation $\mathbb{E}_t\chi$ in the model (58) is a function of S_t and of the curve $\bar{V}_t(\tau), \tau \geq 0$.

Now we discuss that $\bar{V}$ is an observable state, in the sense that it can be extracted from the option market at time t. By Itô's formula applied to (58), we have[58]

$$\begin{aligned} V_t^t dt &= \frac{S_t^2}{f(S_t)^2} d\langle \ln S\rangle_t, \\ \xi_{t+\tau}^t &:= \int_t^{t+\tau} V_u^u du = \int_t^{t+\tau} \frac{S_u^2}{f(S_u)^2} d\langle \ln S\rangle_u, \end{aligned} \tag{64}$$

which is the payoff leg of a weighted variance swap, with time-t $\mathbb{Q}$ price

$$\mathbb{E}_t \xi_{t+\tau}^t = \int_t^{t+\tau} (\mathbb{E}_t V_u^u) du = \int_t^{t+\tau} V_t^u du,$$

by (60). The forward variance curve $\bar{V}$ is the derivative in τ of the derivative price on the left-hand side. It is uniquely determined by call and put option prices in a model-free manner as follows. Assume $1/f$ is locally square integrable on $(0, \infty)$ and let

$$h(x) = \int_1^x \int_1^y \frac{2}{f(z)^2} dz dy \text{ , hence } h''(x) = \frac{2}{f(x)^2}.$$

Hence, by Itô's formula,

$$h(S_{t+\tau}) = h(S_t) + \int_t^{t+\tau} h'(S_u) dS_u + \int_t^{t+\tau} \frac{S_u^2}{f(S_u)^2} d\langle \ln S\rangle_u, \tag{65}$$

whereas, by the Carr-Madan payoff decomposition formula I.(11)[59],

$$\begin{aligned} h(S_{t+\tau}) &= h(S_t) + h'(S_t)(S_{t+\tau} - S_t) \\ &\quad + \int_0^{S_t} (K - S_{t+\tau})^+ h''(K) dK + \int_{S_t}^{\infty} (S_{t+\tau} - K)^+ h''(K) dK. \end{aligned} \tag{66}$$

This means a time-t $\mathbb{Q}$ price[60] of the weighted variance swap payoff leg $\xi_{t+\tau}^t$ (which is the last term in (65)) given by

$$\mathbb{E}_t \xi_{t+\tau}^t = \bar{U}_t(\tau) := 2\int_0^{S_t} P_t(t + \tau, K) \frac{dK}{f(K)^2} + 2\int_{S_t}^{\infty} C_t(t + \tau, K) \frac{dK}{f(K)^2}. \tag{67}$$

Finally we get

$$\bar{V}_t(\tau) = \partial_\tau \bar{U}_t(\tau). \tag{68}$$

[58] cf. IX.(12), with $[\cdot, \cdot] = \langle \cdot, \cdot \rangle$ in the present continuous setup.
[59] applied for $\phi = h, T = t + \tau, x = S_t$.
[60] also model-free replication price as detailed in B.2 below.

B.2 Perfect Hedging

We are considering an infinite dimensional Markov model. But we have only two factors and so, in light of the martingale representation theorem, every square integrable payoff is perfectly replicated with a dynamic portfolio of two traded assets. A natural choice of the two would be the underlying asset and the weighted variance swap with maturity T, with time-t $\mathbb{Q}$ price $\mathbb{E}_t\xi_T^t$. However, as detailed below, a synthetic asset with time-t $\mathbb{Q}$ price $\mathbb{E}_t\xi_T^0$ is more convenient than this weighted variance swap, because it is a local martingale.

Lemma 5 *We have*

$$\begin{aligned} U_t^{0,T} := \mathbb{E}_t\xi_T^0 &= \int_0^t (h'(S_0) - h'(S_u))dS_u + \\ &\quad 2\int_0^{S_0} P_t(T,K)\frac{dK}{f(K)^2} + 2\int_{S_0}^{\infty} C_t(T,K)\frac{dK}{f(K)^2} \\ dU_t^{0,T} &= Z_t^T dW_t^1, \end{aligned}$$

where

$$Z_t^T = \int_0^{T-t} \partial_\tau(\bar{U}_t^\tau)g(\tau)d\tau. \blacksquare$$

Proof. Considering the case where $t = 0$ and $\tau = T$, (65)-(66) yield

$$\begin{aligned} h(S_T) &= h(S_0) + \int_0^T h'(S_u)dS_u + \int_0^T \frac{S_u^2}{f(S_u)^2}d\langle \ln S\rangle_u, \\ h(S_T) &= h(S_0) + h'(S_0)(S_T - S_0) \\ &\quad + \int_0^{S_0} (K - S_T)^+ h''(K)dK + \int_{S_0}^{\infty} (S_T - K)^+ h''(K)dK. \end{aligned}$$

It follows that

$$\begin{aligned} \xi_T^0 = \int_0^T \frac{S_u^2}{f(S_u)^2}d\langle \ln S\rangle_u &= \int_0^T \big(h'(S_0) - h'(S_u)\big)dS_u + \\ &\int_0^{S_0} (K - S_T)^+ h''(K)dK + \int_{S_0}^{\infty} (S_T - K)^+ h''(K)dK. \end{aligned}$$

The time-t value of the corresponding semi-static replication portfolio of ξ_T^0 (replication initiated at time 0) is given by

$$\begin{aligned} \mathbb{E}_t\xi_T^0 = U_t^{0,T} &= \int_0^t (h'(S_0) - h'(S_u))dS_u + \\ &\quad 2\int_0^{S_0} P_t(T,K)\frac{dK}{f(K)^2} + 2\int_{S_0}^{\infty} C_t(T,K)\frac{dK}{f(K)^2}. \end{aligned}$$

Besides, the second lines in (64) and (58) yield

$$\begin{aligned} \mathbb{E}_t\xi_T^0 &= \mathbb{E}_t\int_0^T V_u^u du \\ &= \mathbb{E}_t\int_{u=0}^T \left\{V_0^u + \int_{s=0}^u V_s^u g(u-s)dW_s^1\right\} du \\ &= \int_0^T V_0^u du + \mathbb{E}_t\int_0^T \big(\int_{u=s}^T V_s^u g(u-s)du\big)dW_s^1 \\ &= \int_0^T V_0^u du + \int_0^t \big(\int_{u=s}^T V_s^u g(u-s)du\big)dW_s^1. \end{aligned}$$

Therefore,

$$dU_t^{0,T} = Z_t^T dW_t^1, \tag{69}$$

where

$$Z_t^T = \int_t^T V_t^u g(u-t)du = \int_0^{T-t} V_t^{t+\tau} g(\tau)d\tau$$

and

$$V_t^{t+\tau} = \bar{V}(\tau)_t = \partial_\tau(\bar{U}_t^\tau),$$

by (68). ∎

Proposition 5 *Given a constant $\Theta \in (t,T)$ for any $\mathfrak{F}_\Theta^W$ measurable and $\mathbb{Q}$ square integrable random variable χ, there exists an adapted process (ζ^S, ζ^U) such that*

$$\chi = \mathbb{E}_t\chi + \int_t^\Theta \zeta_r^S dS_r + \int_t^\Theta \zeta_r^U dU_r^{0,T}. \tag{70}$$

Proof: By the martingale representation theorem, there exists (Z^1, Z^2) such that

$$\chi = \mathbb{E}_t\chi + \int_t^\Theta Z_r^1 dW_r^1 + \int_t^\Theta Z_r^2 dW_r^2. \tag{71}$$

But (69) and the first line in (58) yield

$$dW_r^1 = \frac{1}{Z_r^T} dU_r^{0,T},$$
$$dW_r^2 = \frac{1}{\sqrt{1-\rho^2}}\left[\frac{1}{f(S_r)\sqrt{V_r^r}}dS_r - \rho dW_r^1\right].$$

Hence (71) implies that (70) holds for

$$\zeta_r^S = \frac{Z_r^2}{\sqrt{1-\rho^2}f(S_r)\sqrt{V_r^r}}, \quad \zeta_r^U = \frac{Z_r^1}{Z_r^T} - \frac{\rho}{\sqrt{1-\rho^2}}\frac{Z_r^2}{Z_r^T}. \text{ ∎}$$

§5 Benchmarking

A Implied Parameters

The Black–Scholes or the one-factor Gaussian copula pricing formulas are essentially used by traders for conveying information about the relative value of different options in the market, in the dimensionless units provided by the Black-Scholes implied volatility[61] or the Gaussian correlation[62]. These formulas are thus no more than "wrong formulas into which to put a wrong number (the implied volatility of an option or the implied correlation of a CDO tranche) to get the right result (an option market price of a CDO tranche market spread)".

We now explain how they are also commonly used for delta-hedging purposes, in the implied mode.

Remark 7 *Apart from serving as benchmark models on respective derivative markets, the Black-Scholes model and the Gaussian copula setup have of course little in common. Nevertheless, it is possible to draw some analogies between theory and practice of delta-hedging in the two models (Cousin, Crépey, and Kan, 2012).*

[61] see Section I.A.2.
[62] see Section B.3.

B Implied Delta-Hedging with the Black-Scholes Model

We consider the problem of delta-hedging a European option with maturity T and payoff ξ on an underlying S. A bank sells the option at price Π_0 at time 0 and must pay the payoff ξ at time T. The hedging strategy of the bank consists in rebalancing, at every step of a time-grid $t_i = ih$, $0 \leq i \leq n-1$, a self-financing hedge in the underlying (risky and risk-free) assets. We assume that dividends on S are paid and kept as new stock shares falling at yield q in the hedging portfolio. We are thus considering a hedging strategy of the form $\zeta_t = \zeta_{t_i} e^{q(t-t_i)}$ on $[t_i, t_{i+1})$, for $i = 0 \dots n-1$.

Remark 8 *This strategy, along with a position ζ^0 in the riskless asset $S^0 = e^{r\cdot}$ constant on each $[t_i, t_{i+1})$, and ζ^0 updated at each t_{i+1} so that $\zeta^0_{t_i} e^{rt_{i+1}} + \zeta_{t_i} e^{q(t_{i+1}-t_i)} S_{t_{i+1}} = \zeta^0_{t_{i+1}} e^{rt_{i+1}} + \zeta_{t_{i+1}} S_{t_{i+1}}$ (where the left-hand side represents the value reached by the strategy at time $t_{i+1}-$ and the right-hand side its value at t_{i+1}), defines a self-financing strategy in S^0 and S. To prove this claim, it is enough to show that a strategy with constant number c_0 of riskless assets and a number $ce^{q\cdot}$ of risky assets (with c constant) is self-financing, which follows from the following identity (cf. 0.(3)), where $V = c_0 S^0 + ce^{q\cdot} S$:*

$$d(\beta V)_t = c\, d(e^{qt}\beta S) = ce^{qt}\big(d(\beta S)_t + \beta_t S_t q dt\big).$$

Recalling 0.(2), we thus have on $[t_i, t_{i+1})$:

$$\begin{aligned} &\zeta_t d(\beta_t \widehat{S}_t) = \zeta_t\big(d(\beta_t S_t) + \beta_t q_t S_t dt\big) = \\ &\quad \zeta_{t_i} e^{-qt_i} e^{qt} d\big(d(\beta_t S_t) + \beta_t q_t S_t dt\big) = \zeta_{t_i} e^{-qt_i} d(\beta_t S_t e^{qt}). \end{aligned} \tag{72}$$

As we consider a nondividend-paying option, for which $\widehat{\Pi} = \Pi$, the discounted profit-and-loss of the trader at T is therefore[63]

$$\begin{aligned} &\beta_T p_T \\ &\quad = -\beta_T \xi + \Pi_0 + \sum_{i=1}^{n} e^{-q(i-1)h} \zeta_{(i-1)h} \left(\beta_{ih} S_{ih} e^{qih} - \beta_{(i-1)h} S_{(i-1)h} e^{q(i-1)h}\right) \\ &\quad = -\sum_{i=1}^{n} \beta_{(i-1)h} \delta\rho_i, \end{aligned} \tag{73}$$

for pnl increments $\delta\rho_i$ such that

$$\begin{aligned} \beta_{(i-1)h}\delta\rho_i = &(\beta_{ih}\Pi_{ih} - \beta_{(i-1)h}\Pi_{(i-1)h}) \\ &- e^{-q(i-1)h}\zeta_{(i-1)h}\left(e^{-\kappa ih} S_{ih} - e^{-\kappa(i-1)h} S_{(i-1)h}\right). \end{aligned} \tag{74}$$

This is valid for every discrete-time hedging scheme $(\zeta_{t_i})_{0\leq i\leq n-1}$. The Black-Scholes implied delta hedging scheme corresponds to the following specification:

$$\zeta_{t_i} = \Delta^{bs}(t_i, S_{t_i}, T, K; \Sigma_{t_i}),$$

where Σ_{t_i} denotes the Black-Scholes implied volatility of the option at time t_i.

C Implied Delta-Hedging with the Gaussian Copula Model

Our next aim is to hedge the spread risk of a CDO tranche between two successive default times of the reference entities. Note that we do not aim at hedging defaults in this approach. More precisely, our goal will be to hedge homogeneous bumps on homogeneous time intervals of the underlying CDS curves, using CDS index contracts as hedging instruments. A CDS index contract covers default risk on all names in a credit pool. CDS index contracts may be considered as kinds of averages of individual CDSs, and they can be priced essentially like the latter, using a relation of the form (53).

[63] the strategy being self-financed at grid times.

We thus rebalance, at every time step h (typically taken of the order of one week for credit derivatives), a hedging position in a primary market consisting of the risk-free asset, β^{-1}, and of q CDS index contracts with increasing maturities T_j, $j = 1, \ldots, q$, where $T_{q-1} < T \leq T_q$. Also, let $T_0 = 0$. Considering a bank that bought credit protection through a tranche and is short ζ_t^j units in the CDS index contract with maturity T_j, the discounted pnl increment[64] of the hedged position on a time interval $[t, t+h)$ is given, assuming no defaults of the underlying names on $[t, t+h)$, by:

$$\begin{aligned}\beta_t \delta\rho &= -\beta_t \delta p^\star + \sum_j \zeta_t^j \beta_t \delta p^j \\ &= (\beta_{t+h} D_{t+h} - \beta_t D_t) - \Sigma_0^* (\beta_{t+h} F_{t+h} - \beta_t F_t) - \beta_t \Sigma_0^* h \\ &\quad - \sum_j \zeta_t^j \left(\Lambda \beta_{t+h} D_{t+h}^j - \Sigma_t^j \beta_{t+h} F_{t+h}^j - \beta_t \Sigma_t^j h \right),\end{aligned} \tag{75}$$

where:

- $\delta p^\star$ (respectively δp^j) is the increment of the pnl on a unit position on the tranche (respectively on the CDS index contract with maturity T_j),
- Σ_0^* is the contractual spread of the tranche, and Σ_t^j is the spread of the CDS index contract with maturity T_j at the current time t,
- D and F, respectively D^j and F^j, denote the value processes of the default and fees legs of the tranche, respectively of the CDS index contract with maturity T_j, and
- Λ is a common and constant loss-given-default on the credit index contracts.

Note that the δp^j only depend on the value of the CDS index contracts at time $t+h$. This is due to the fact that the CDS index contracts used for hedging at time t are new CDSs, freshly emitted at t. So their value at time t is equal to zero, by definition of the fair spread Σ_t^j. The Gaussian copula implied delta hedge then consists in setting a row-vector of CDS index hedging positions $\zeta_t = (\zeta_t^j)_j$ in (75), such that

$$\Delta_t^\star = \zeta_t \Delta_t, \tag{76}$$

where $\Delta_t^\star$ (respectively Δ_t) represents the row-vector of the sensitivities of the tranche (respectively the matrix of the sensitivities of the CDS index contracts) with respect to homogeneous bumps of one basis point on the time interval $[T_{j-1}, T_j]$ of the underlying CDS curves, for $j = 1, \ldots, q$. In (76):

- the Δ_i^j are computed by assessing, using a relation analogous to (53), the impact of the j^{th} bump on the underlying CDS curves, on the default and fees legs of the i^{th} CDS index contract. Note that the Δ_i^j are not sensitive to the dependence structure (copula function) of $(\tau_1, \ldots, \tau_d)$, and that the matrix Δ is lower triangular;
- the $\Delta^{\star,j}$ are computed by assessing the impact on the default and fees legs of the tranche of the j^{th} bump on the underlying CDS curves.

More precisely, in the second item:

1. we bootstrap the marginal cumulative distribution functions F_l from the underlying CDS spread curves using (53), and we compute the base implied correlation ρ_t of the tranche;
2. we compute the associated values D and F of the default and fees legs of the tranche.

   Then, for $j = 1, \ldots, q$:
3. we recalibrate as in item i every marginal cumulative distribution function $\widetilde{F}_l$ to the corresponding CDS curve bumped by +1bp on the time interval $[T_{j-1}, T_j]$;

[64] these can then be summed up between 0 and T much like (73) in the Black-Scholes case.

4. we compute the values $\widetilde{D}$ and $\widetilde{F}$ of the default and fees legs of the tranche in the Gaussian copula model with marginal cumulative distribution functions $\widetilde{F}_l$ and with correlation parameter ρ_t;

5. we set

$$\Delta^{\star,j} = \Sigma_0^*(\widetilde{F} - F) - (\widetilde{D} - D).$$

We finally solve the triangular system (76) for the row-vector ζ_t.

D Sticky Deltas

The Black–Scholes implied delta hedge used in Paragraph B is only a partial hedge, which does no account for the volatility risk. To improve on this Black–Scholes implied delta, one can use a finite difference Black-Scholes implied delta of the form (for some small positive α)

$$(2\alpha S_t)^{-1}\times$$
$$\left(\Pi^{bs}(t,(1+\alpha)S_t,T,K;r,q,\widetilde{\Sigma}_t) - \Pi^{bs}(t,(1-\alpha)S_t,T,K;r,q,\Sigma_t)\right),$$

in which the volatility $\widetilde{\Sigma}_t$ is a suitable update of Σ_t, accounting for the correlation between stock returns and implied volatility changes. For instance, the sticky delta (or sticky moneyness) rule[65] stipulates that the implied volatility surface evolves deterministically, when parameterized in terms of the time-to-maturity τ and of the put log-forward moneyness $(\ln(\frac{K}{S_t}) - \kappa\tau)$.

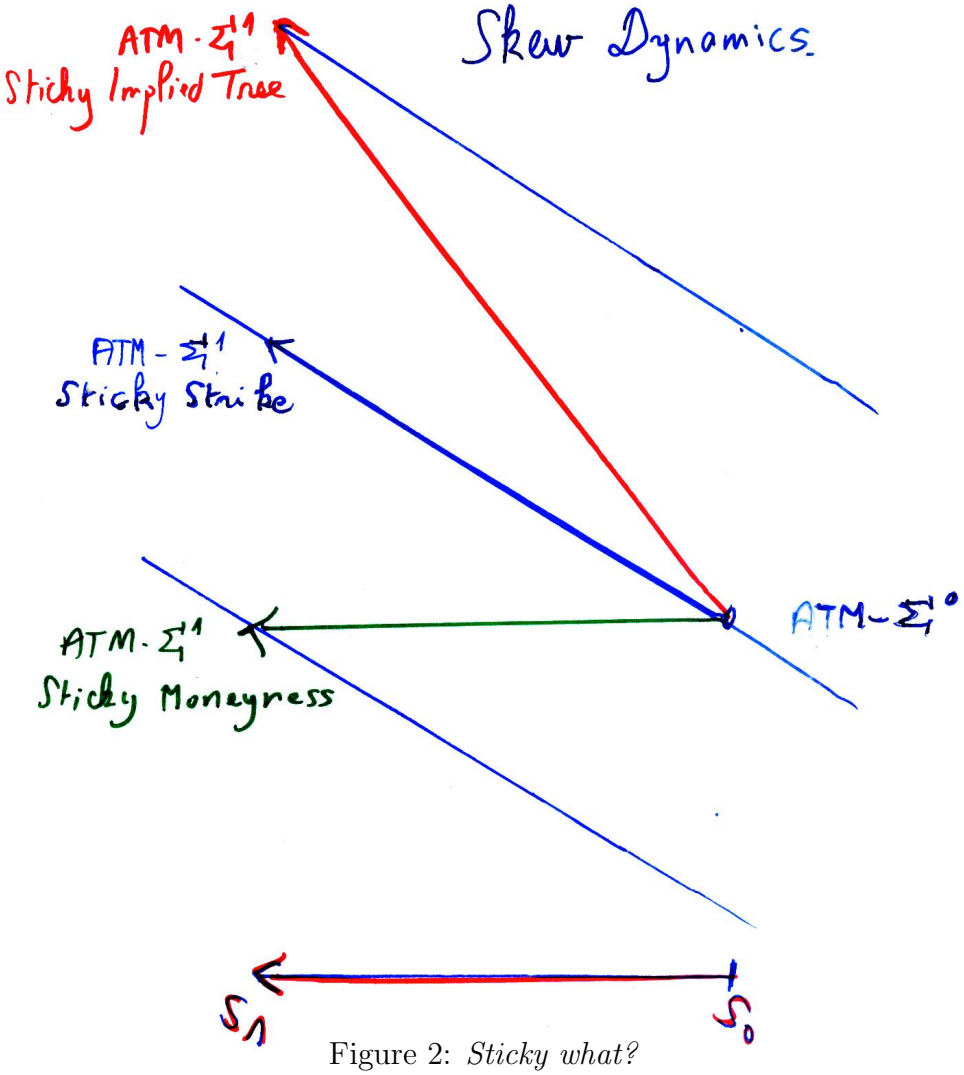


Figure 2: *Sticky what?*

[65] see Balland (2002) and Figure 2.

Likewise, to improve over the Gaussian copula implied delta of Paragraph D, we can use an updated correlation parameter $\widetilde{\rho}_t$ in item iv there in order to account for the correlation between spread moves and implied correlation changes. A sticky delta rule thus stipulates that the implied correlation smile evolves deterministically when parameterized in terms of the time to maturity τ and of a suitable notion of moneyness of a tranche.

E Hedging VIX Options: Empirical Analysis

This is Section 3 of Fukasawa, Horvath, and Tankov (2021), modulo notational or other detail changes made for self-containedness or consistency with the present notes.

In this part, we illustrate the advantages of rough volatility modeling[66] for managing a VIX option. The S&P 500 implied volatility index, VIX, is computed since 2003 as a suitable average, detailed page 10 of Chicago Board Options Exchange (2009), of all the quoted implied volatilities of maturity one month or, more precisely, as a linear interpolation, valued for $T = 30$ days, between the two averages corresponding to the two shortest quoted maturities.

We consider three models for the VIX index: the Black-Scholes model, the CIR model[67], and a rough stochastic volatility (RFSV) model where the volatility is the exponential of a fractional Brownian motion. Since we are interested in hedging short-term options, we use simplified version of the models without drift and neglect the effect of the interest rate. As a result, all models have only one parameter to be estimated.

In each model, we perform a series of backtests of dynamic hedging of a VIX option with a forward variance swap[68] with the same maturity as the option and with the duration corresponding to that of the VIX (1 month). In all tests, the hedging portfolio is readjusted daily using the closing prices of the hedging instruments. The test is performed 1000 times, starting on each working day t between Jan 10, 2012 and Apr 29, 2016. The backtest is organized as follows:

- The model parameter is estimated on the 88-day period preceding day t;
- The initial value of the hedging portfolio is initialized with the ATM VIX option price with maturity 1.5 months computed within the model, and the quantity of the hedging asset in the portfolio is initialized with the corresponding model-based hedge ratio;
- For 29 working days following day t, each day the portfolio value is readjusted following the change in the value of the hedging asset, and the hedge ratio is recomputed;
- At the end of the 29 working day period, the pnl of the hedging portfolio is recorded, and the no-hedge pnl is recorded as the difference between the option price at the beginning and at the end of the period[69].

The backtest uses synthetic forward variance curve data[70], computed from the historical prices of S&P index options, downloaded from the WRDS database. The detailed description of models and hedging procedures is given below. Table 2 presents the main results of the backtest. We see that while the Black-Scholes and CIR benchmarks appear to have similar performance, the RFSV model exhibits a much lower bias, and a RMSE which is 27% lower than the other two models. Figure 3 plots one backtest pnl evolution as function of the starting date of the backtest. The consistently low bias of the strategy based on the RFSV model is clearly visible here.

[66] cf. §4.B.
[67] cf. §2.A.
[68] cf. (63), a tradable asset as seen in (68).
[69] model initial price of the option and model-free option's payoff, respectively.
[70] a tradable asset as seen in (68).

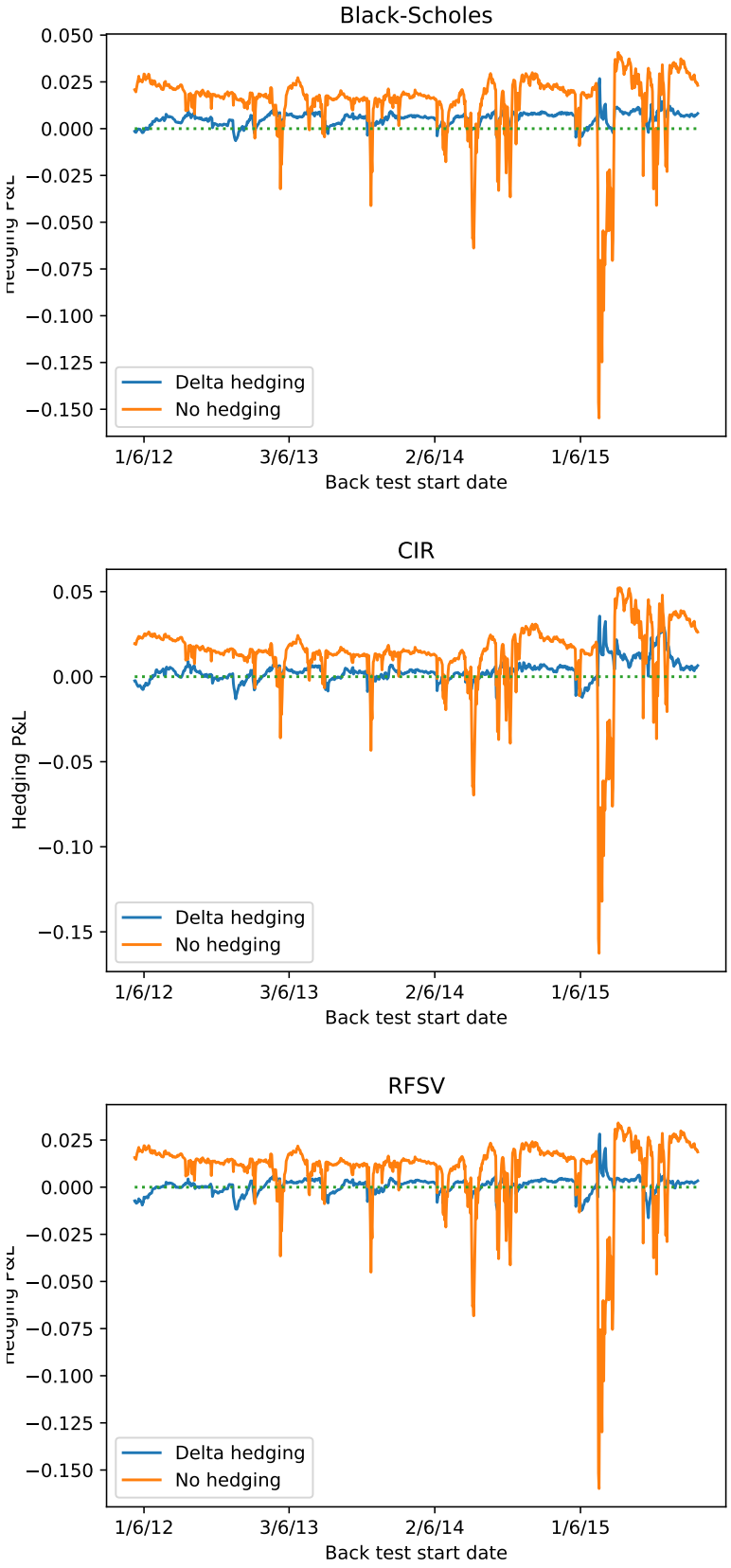


Figure 3: Backtesting pnl as function of time for the three models we study

	Black-Scholes		CIR		RFSV	
	No hedge	Hedge	No hedge	Hedge	No hedge	Hedge
Mean	0.01445	0.005336	0.01363	0.003399	0.009919	0.0006345
Std. dev.	0.01896	0.003555	0.02069	0.006506	0.01880	0.004141
RMSE	0.02384	0.006412	0.02478	0.007341	0.02125	0.004190
Red. factor		3.7176		3.7176		5.0724

Table 2: Empirical Performance of hedging strategies based on different models

E.1 Black-Scholes Model

Let VIX_t be the VIX index at time t, and let F_t^T be the T forward variance swap at time t, which refers to the same period as the VIX index, that is, $F_t^T = \mathbb{E}_t \mathrm{VIX}_T^2$[71]. Assume that the VIX index follows the log-normal dynamics $\mathrm{VIX}_t = e^{X_t}$ (with VIX_0 here conventionally set to 1), where X is an OU process $dX_t = \lambda(\theta - X_t)dt + \gamma dW_t$ under the risk-neutral probability. When close to term, the drift can be neglected (so we just set $\lambda = 0$ below) and we obtain the simple Black-Scholes dynamics

$$\begin{aligned} F_t^T &= \mathbb{E}_t e^{2X_T} = \mathbb{E}_t e^{2\gamma W_T} = e^{2\gamma W_t + 2\gamma^2 (T-t)}, \\ dF_t^T &= 2F_t^T \gamma dW_t. \end{aligned}$$

On the other hand, γ may be estimated from the volatility of VIX:

$$d\langle \mathrm{VIX} \rangle_t = \mathrm{VIX}_t^2 \gamma^2 dt.$$

We are hedging a VIX option with pay-off

$$(\mathrm{VIX}_T - K)^+ .$$

The VIX future

$$\mathrm{VIX}_t^T := \mathbb{E}_t \mathrm{VIX}_T = \mathbb{E}_t e^{\gamma W_T} = e^{\gamma W_t} e^{\frac{\gamma^2}{2}(T-t)} = \sqrt{F_t^T} e^{-\frac{\gamma^2}{2}(T-t)} \tag{77}$$

is a Black martingale with volatility γ (starting from the initial condition $e^{\frac{\gamma^2 T}{2}}$ at time 0). Neglecting the interest rate, the option price is therefore given by

$$\begin{aligned} P_t &:= \mathbb{E}_t p(t, \mathrm{VIX}_t^T) = \mathrm{VIX}_t^T \mathcal{N}(d_t^1) - K\mathcal{N}(d_t^2), \\ d_t^{1,2} &= \frac{\log \frac{\mathrm{VIX}_t^T}{K} \pm \frac{\gamma^2 (T-t)}{2}}{\gamma \sqrt{T-t}} \end{aligned}$$

or, equivalently in view of (77),

$$\begin{aligned} P_t &= \widetilde{p}(t, F_t^T) = \sqrt{F_t^T} e^{-\frac{\gamma^2}{2}(T-t)} \mathcal{N}(d_t^+) - K\mathcal{N}(d_t^-) \\ d_t^{\pm} &= \frac{\log \frac{\sqrt{F_t^T} e^{-\frac{\gamma^2}{2}(T-t)}}{K} \pm \frac{\gamma^2 (T-t)}{2}}{\gamma \sqrt{T-t}}, \end{aligned}$$

and the hedge ratio is

$$\partial_F \widetilde{p} = \partial_F \mathrm{VIX} \partial_{\mathrm{VIX}} P = \frac{\mathcal{N}(d_t^+)}{2\sqrt{F_t^T}} e^{-\frac{\gamma^2}{2}(T-t)}.$$

[71] cf. (62).

E.2 CIR Model

Assume that the VIX index follows the square root dynamics:

$$d\mathrm{VIX}_t^2 = \lambda(\theta - \mathrm{VIX}_t^2)dt + \gamma \mathrm{VIX}_t dW_t. \tag{78}$$

Since we are hedging short maturity options and cannot estimate λ and θ under the risk-neutral measure anyway, we assume that $\lambda = 0$ so that

$$d\mathrm{VIX}_t^2 = \gamma \mathrm{VIX}_t dW_t.$$

The forward variance swap is then given by

$$F_t^T = \mathbb{E}_t \mathrm{VIX}_T^2 = \mathrm{VIX}_t^2$$

and follows the dynamics

$$dF_t^T = \gamma \sqrt{F_t^T} dW_t$$

We are hedging a VIX option with pay-off

$$(\mathrm{VIX}_T - K)^+$$

with a forward variance swap. The price of the VIX option is given by

$$P(t, F_t^T) = \int_{K^2}^{\infty} (\sqrt{x} - K) p_{T-t}(F_t^T, x) dx,$$

where $p_T(v_0, x)$ is the density[72] of the CIR process at time T with the starting value v_0. The parameter γ may be estimated by observing that $\langle \mathrm{VIX} \rangle_t = \frac{\gamma^2}{4} t$, by the Itô formula applied to the square root of VIX^2 in (78).

E.3 Rough Fractional Stochastic Volatility

Assume now that the VIX index is given by

$$\mathrm{VIX}_t = C e^{X_t}, \tag{79}$$

where $C > 0$ is a constant and X is a centered Gaussian process under the risk-neutral probability. For all $s \geq 0$, let $\mathfrak{F}_s^0 := \sigma(X_r, r \leq s)$, and $\mathfrak{F}_s := \cap_{s<t} \mathfrak{F}_t^0$. The interest rate is taken to be zero. Fix a time horizon T, let $Z_t^T := \mathbb{E}_t X_T$, so that $(Z_t^T)_{t \geq 0}$ is a Gaussian[73] martingale and thus a process with independent increments[74] completely characterised by the function

$$c^T(t) := \mathbb{E}\big[(Z_t^T)^2\big].$$

The forward variance swap can be characterized as

$$\begin{aligned} F_t^T &:= \mathbb{E}_t \mathrm{VIX}_T^2 = C^2 \mathbb{E}_t e^{2X_T} = \\ & C^2 e^{2\mathbb{E}_t X_T + 2\mathbb{V}\mathrm{ar}_t X_T} = C^2 e^{2(Z_t^T + c^T(T) - c^T(t))}, \end{aligned} \tag{80}$$

since

$$\begin{aligned} \mathbb{V}\mathrm{ar}_t X_T &= \mathbb{E}_t[(X_T - Z_t^T)^2] = \mathbb{E}_t[(Z_T^T - Z_t^T)^2] = \\ & \mathbb{E}[(Z_T^T - Z_t^T)^2] = \mathbb{E}[(Z_T^T)^2] - \mathbb{E}[(Z_t^T)^2] = c^T(T) - c^T(t). \end{aligned} \tag{81}$$

[72] see Jeanblanc, Yor, and Chesney (2009, Proposition 6.3.2.1 page 358) or Lamberton and Lapeyre (1996, page 163) for the explicit CIR density formula.

[73] cf. Lemma 0.1.

[74] see e.g. Jacod and Shiryaev (2003, Theorem II.4.36).

The time-t price of a Call on the VIX is given by $P_t := \mathbb{E}_t(\mathrm{VIX}_T - K)^+$. The VIX future is the lognormal martingale

$$\begin{aligned}\mathrm{VIX}_t^T &= \mathbb{E}_t \mathrm{VIX}_T = C\mathbb{E}_t e^{X_T} = Ce^{\mathbb{E}_t X_T + \frac{1}{2}\mathbb{V}\mathrm{ar}_t X_T} \\ &= Ce^{Z_t^T + \frac{1}{2}(c^T(T) - c^T(t))} = \sqrt{F_t^T} e^{-\frac{1}{2}(c^T(T)-c^T(t))},\end{aligned} \tag{82}$$

by (81) and (80). By (79) and (81),

$$\mathbb{V}\mathrm{ar}_t \ln \mathrm{VIX}_T = \mathbb{V}\mathrm{ar}_t X_T = c^T(T) - c^T(t).$$

Hence

$$\begin{aligned}P_t &= \mathrm{VIX}_t^T \mathcal{N}(d_t^+) - K\mathcal{N}(d_t^-), \\ d_t^\pm &= \frac{\log \frac{\mathrm{VIX}_t^T}{K} \pm \frac{1}{2}(c^T(T) - c^T(t))}{\sqrt{c^T(T) - c^T(t)}}.\end{aligned}$$

Additionally assuming X continuous, applying the Itô formula and bearing in mind the martingale property of the option price, we obtain[75]:

$$dP_t = \mathcal{N}(d_t^+) d\mathrm{VIX}_t^T.$$

In view of (82), we then have in terms of the forward variance swap:

$$\begin{aligned}P_t &= \sqrt{F_t^T} e^{-\frac{1}{2}(c^T(T)-c^T(t))} \mathcal{N}(d_t^+) - K\mathcal{N}(d_t^-), \\ d_t^\pm &= \frac{\log \frac{\sqrt{F_t^T} e^{-\frac{1}{2}(c^T(T)-c^T(t))}}{K} \pm \frac{1}{2}(c^T(T) - c^T(t))}{\sqrt{c^T(T) - c^T(t)}}.\end{aligned}$$

Then, the option price dynamics takes the following form:

$$dP_t = \frac{\mathcal{N}(d_t^+) e^{-\frac{1}{2}(c^T(T)-c^T(t))}}{2\sqrt{F_t^T}} dF_t^T.$$

Assuming that

$$X_t = \sigma W_t^H,$$

where W^H is the fractional Brownian motion (starting from 0) with the Hurst parameter H, we get after some computations using the Mandelbrot-Van Ness representation

$$\begin{aligned}W_t^H = \frac{1}{\Gamma(H+1/2)} \Bigg\{ &\int_{-\infty}^0 \left[(t-s)^{H-1/2} - (-s)^{H-1/2}\right] dW_s \\ &+ \int_0^t (t-s)^{H-1/2} dW_s \Bigg\},\end{aligned}$$

where W is a standard Brownian motion and $\Gamma(z) = \int_0^\infty x^{z-1} e^{-x} dx$:

$$\begin{aligned}c^T(t) &= \frac{\sigma^2}{\Gamma^2(H+1/2)} \int_0^\infty \left[(T+s)^{H-1/2} - s^{H-1/2}\right]^2 ds \\ &\quad + \frac{\sigma^2}{\Gamma^2(H+1/2)} \int_0^t (T-s)^{2H-1} ds \\ &= f(T) - \frac{\sigma^2 (T-t)^{2H}}{2H\Gamma^2(H+1/2)}\end{aligned}$$

[75] Here we use a standard argument of the Black-Scholes model to show that $\partial_{\mathrm{VIX}} P_t = \mathcal{N}(d_t^+)$, then we argue that, since the option price is a martingale, all terms except the first order term in the underlying must cancel out (or one can also do the full computation with the Itô formula to obtain the same result).

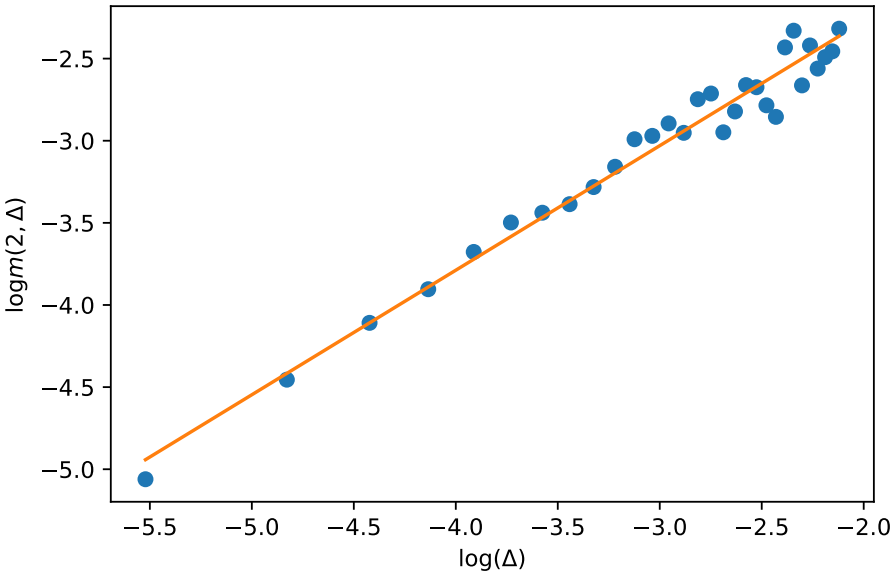


Figure 4: Estimation of the Hurst parameter

for some function $f(T)$, so that

$$c^T(T) - c^T(t) = \frac{\sigma^2 (T-t)^{2H}}{2H\Gamma^2(H+1/2)}.$$

Contrary to the previous two models, this one formally has two parameters to be estimated: σ and H. To estimate the Hurst parameter, following Gatheral, Jaisson, and Rosenbaum (2018), we define

$$m(2,\delta) = \frac{1}{\lfloor T/\delta \rfloor} \sum_{k=1}^{\lfloor T/\delta \rfloor} |\log(\mathrm{VIX}_{k\delta}) - \log(\mathrm{VIX}_{(k-1)\delta})|^2,$$

and estimate H from the half slope of $m(2,\delta)$[76] as function of δ in the log-log coordinates (see Figure 4, where δ varies from 1 to 30 days). Since this procedure requires a relatively long dataset to be precise, we perform it only once, on the VIX index time series from April 17, 2001 to April 16, 2021. This gives an estimated Hurst parameter value of 0.377, and the procedure is quite stable: when estimating on the first 10 years of the dataset, one obtains 0.380 and on the last 10 years one obtains 0.379.

These estimated values of the Hurst index are much higher than the values found by Gatheral, Jaisson, and Rosenbaum (2018) and many other authors using the daily time series of realized volatility (typically between 0.1 and 0.15). However, the VIX index is constructed from prices of one-month options on the S&P index, and using the implied volatility of one-month options as proxy for volatility, Livieri, Mouti, Pallavicini, and Rosenbaum (2018) find a value of $H = 0.32$, which is much closer to our result.

In view of the stability of the Hurst index estimation, we fix the value $H = 0.377$ for all tests, rather than estimating it before each backtest. Note that the hedging performance of the model remains very similar for $H \in [0.37, 0.39]$. This leaves us with a single parameter, σ, to estimate before each backtest, which is estimated by

$$\widehat{\sigma} = \sqrt{\frac{m(2,\delta)}{\delta^{2H}}}.$$

To further illustrate the dependence of the hedging performance on the value of the Hurst index and the importance of using a "rough volatility" specification, we performed the same test for H values

[76] which scales like δ^{2H} in the case of a fractional Brownian motion, cf. (61).

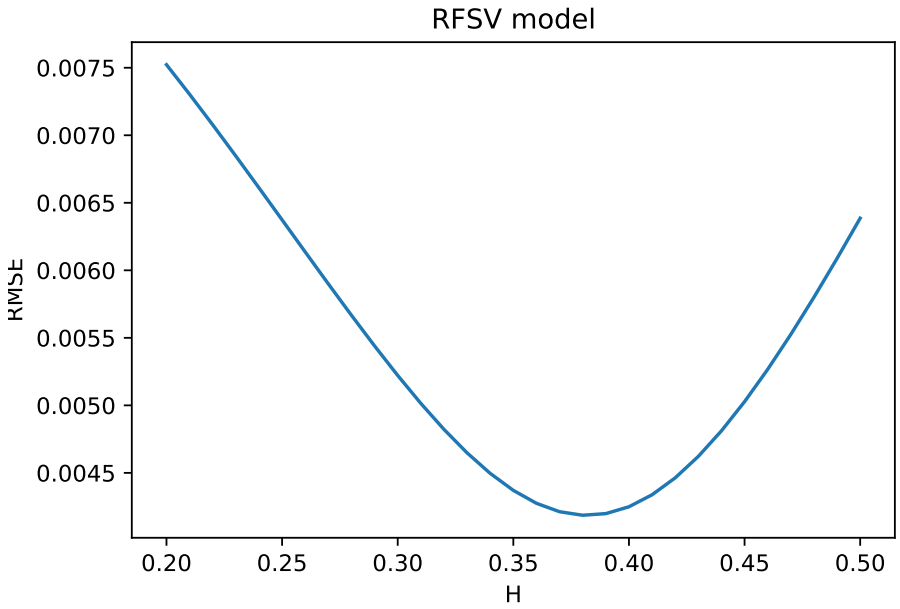


Figure 5: Dependence of the hedging RMSE on the Hurst parameter value.

ranging between 0.2 and 0.5, with step of 0.01, where the value $H = 0.5$ corresponds to the Black-Scholes benchmark. Figure 5 shows the dependence of the hedging RMSE on the value of the Hurst parameter with the minimum attained around $H = 0.38$.

NUMERICAL SCHEMES

Chapter III

Pricing and Greeking by Finite Differences

Like tree methods and as opposed to simulation methods of Chapters V and IV, finite difference methods can easily cope with early exercise features, and in low dimension they can give an accurate and robust computation of an option price and Greeks (delta, gamma, and theta at time 0). However, they are not practical for dimension greater than three or four, for then too many grid points are required for achieving satisfactory accuracy.

§1 Generic Pricing PIDE

In a Markov setup X (see around 0.(9) and IX.§3), the pricing function u of a financial derivative solves a parabolic partial integro-differential equation of the following form on the time-space domain $E = [0,T] \times \mathbb{R}^d$, where T is the maturity of the claim:

$$\left\{ \begin{array}{l} F(t,x,u(t,x),\partial_t u(t,x),\partial u(t,x),\partial^2 u(t,x),\mathcal{I}u(t,x)) = 0 \text{ on } [0,T) \times \mathcal{O} \\ u(t,x) = \psi(t,x) \text{ on } E \setminus \Big([0,T) \times \mathcal{O} \Big). \end{array} \right. \tag{1}$$

In this equation:

- $\mathcal{O}$ is an open subset of $\mathbb{R}^d$. In particular, the terminal condition at T, which is embedded in the boundary condition ψ in (1), is given by the payoff of the claim at maturity.
- ∂u and $\partial^2 u$ denote, respectively, the row-gradient and the Hessian matrix of u with respect to x.
- $\mathcal{I}u$ is the nonlocal (integral) term

$$\mathcal{I}u(t,x) = \lambda(t,x) \int_{\mathbb{R}^d} (u(t,x+\delta(t,x,y)) - u(t,x)), \tag{2}$$

  where λ, π and δ are part of the specification of X[1].

The precise definitions of the domain $\mathcal{O}$ and of the operator F depend on the specifications of the derivative at hand. In the case of a European vanilla option in space dimension one, F is a linear operator and $\mathcal{O} = (-\infty, +\infty)$ or $(0, +\infty)$. In the case of a barrier option, $\mathcal{O}$ is limited by the barriers. In the case of an American or game option, there are further obstacles in F. In any case, one deals with a monotone operator F in the sense that one has, for all $(t,x) \in [0,T] \times \mathcal{O}$, $u, v, a \in \mathbb{R}$, $p \in \mathbb{R}^d$, $A, B \in \mathbb{R}^{d \times d}$ and $I, J \in \mathbb{R}^q$:

$$\begin{array}{l} F(t,x,u,a,p,A,I) \le F(t,x,v,a,p,B,J) \text{ whenever} \\ u \le v,\ A \ge B \text{ and } I \ge J. \end{array} \tag{3}$$

Here the inequality $I \ge J$ is meant componentwise, and the inequality $A \ge B$ in the sense of the usual order on the space of the real-valued symmetric nonnegative $d \times d$-matrices, meaning that all eigenvalues of $A - B$ are nonnegative.

[1] see IX.§3.A.

A Maximum Principle

A classical solution to the deterministic pricing equation (1) is a function $u = u(t,x)$ in $\mathcal{C}([0,T]\times\mathbb{R}^d)\cap\mathcal{C}^{1,2}([0,T)\times\mathcal{O})$ satisfying (1) pointwise over $[0,T]\times\mathbb{R}^d$. A classical subsolution (respectively classical supersolution) to (1) satisfies (1) pointwise over $[0,T]\times\mathbb{R}^d$, with $=$ replaced by $\leq$ (respectively $\geq$).

Let us assume for simplicity that $\mathcal{O}$ is bounded and that the monotonicity of F is strict in its third argument u. We then have the following comparison principle:

Proposition 1 *We have $u\leq v$ for every classical subsolution u and any classical supersolution v of (1).*

Proof. Assume that $u\leq v$ doesn't hold. Then $w=u-v$ admits a positive maximum at a point $(t,x)\in[0,T)\times\mathcal{O}$. By a second-order optimality condition and, by definition of $\mathcal{I}u$, we have

$$\begin{aligned}\partial_t u(t,x)=\partial_t v(t,x),\ \mathcal{I}u(t,x)\leq\mathcal{I}v(t,x)\\ \partial u(t,x)=\partial v(t,x),\ \partial^2 u(t,x)\leq\partial^2 v(t,x).\end{aligned}$$

Hence the fact that

$$\begin{aligned}F(t,x,u(t,x),\partial_t u(t,x),\partial u(t,x),\partial^2 u(t,x),\mathcal{I}u(t,x))\leq 0\\ \leq F(t,x,v(t,x),\partial_t v(t,x),\partial v(t,x),\partial^2 v(t,x),\mathcal{I}v(t,x))\end{aligned}$$

implies, by the monotonicity of F in its arguments A and I:

$$\begin{aligned}F(t,x,u(t,x),\partial_t u(t,x),\partial u(t,x),\partial^2 u(t,x),\mathcal{I}u(t,x))\leq\\ F(t,x,v(t,x),\partial_t u(t,x),\partial u(t,x),\partial^2 u(t,x),\mathcal{I}u(t,x)).\end{aligned}$$

But $>$ should hold, since $u(t,x)>v(t,x)$, taking into account the assumed strict monotonicity of F in its argument u. ∎

Proposition 1 implies uniqueness for a classical solution to the pricing equation (1). In case F is linear with parameters (coefficients) ϱ, (Friedman, 1983) provides explicit conditions on ϱ and ψ ensuring the existence of a classical solution u to (1)[2]. However, if F is nonlinear, a classical solution doesn't generally exist and one must resort to suitable notions of weak solutions to (1).

B Weak Solutions

B.1 Viscosity Solutions

The theory of viscosity solutions defines suitable notions of weak solutions, subsolutions and supersolutions to (1) such that maximum and comparison principles hold for any nonlinear monotone operator F (Crandall et al., 1992; Fleming and Soner, 2006; Alvarez and Tourin, 1996; Barles et al., 1997; Amadori, 2003, 2007; Cont and Voltchkova, 2005; Jakobsen and Karlsen, 2006, 2005). This grants the uniqueness for a viscosity solution to (1). Existence is in turn established by various means, such as Perron's method, which is itself based on comparison principles. For practical use in the next sections, it will be enough for us to keep in mind that, under mild assumptions, the pricing equation (1) is well-posed in a suitable space of viscosity solutions.

B.2 Sobolev Solutions

As an alternative to viscosity solutions, it is possible to derive weak variational formulations of the deterministic pricing equation (1) in Sobolev functional spaces $\mathcal{H}$. In this approach, the boundary

[2] see e.g. Theorem I.1.

condition ψ is typically accounted for by a judicious choice of the space $\mathcal{H}$. Existence and uniqueness for a solution to the variational formulation of (1) is then obtained by application of a Lax–Milgram theorem[3]. In the case of the pricing equations various choices for $\mathcal{H}$ are possible (Achdou and Pironneau, 2005; Matache et al., 2004; Bally and Matoussi, 2001; Bally et al., 2002; Barles and Lesigne, 1997; Jaillet et al., 1990; Ern et al., 2004). The precise variational formulations are outside the scope of these notes. We will simply keep in mind that, under mild technical assumptions, a variational formulation of the deterministic pricing equation (1) is well-posed in a suitable Sobolev space $\mathcal{H}$. This underlies the theory of related finite element approximation schemes.

§2 Numerical Approximation

In the case of European vanilla options in simple models, the deterministic pricing equation (1) can be solved analytically. But, in general, (1) must be solved numerically. In order to approximate (1), one can either use the finite difference methods of this chapter (see also e.g. Tavella and Randall (2000); Zhu, Wu, and Chern (2004)) or resort to more general finite element methods Achdou and Pironneau (2005). Note that there is no hermetic frontier between these methods. One thus has, schematically[4]:

$$\begin{aligned} \text{Tree Methods} \subset\ & \text{Finite Differences Methods} \\ \subset\ & \text{Finite Elements Methods} \end{aligned} \tag{4}$$

A Finite Difference Methods

Finite difference methods are naturally connected with viscosity solutions of monotone equations satisfying related maximum principles. The additional complexity of using potentially more powerful finite element methods is mainly justified when the geometry of the domain makes it necessary to use an unstructured, adaptive discretization mesh. But pricing problems in finance are typically posed on rectangular domains, for which a simple finite difference grid is good enough.

A.1 Localization and Discretization

The numerical solution of the pricing equation (1) by finite differences is a four step process:

1. ***Transformations*** of the problem, such as:
   - *changes of variables* (e.g. $x = \ln(S)$),
   - *changes of unknowns* (e.g., solving the equation for $e^{-rt}u$ rather than u),
   - *changes of probability measure*[5].
2. ***Localisation*** [6] of the problem, that is:
   - replacing $\mathcal{O}$ by a *bounded domain* in (1), still denoted by $\mathcal{O}$ henceforth, and introducing a suitable boundary condition φ outside $[0,T)\times\mathcal{O}$ such that, in particular, $\varphi = \psi$ at T,
   - replacing the integration domain $\mathbb{R}^d$ in the integral term $\mathcal{I}u$, by a *bounded integration domain* $\mathcal{D}(t,x)$ such that
$$\int_{\mathcal{D}(t,x)} \pi(t,x,dy) \approx 1.$$

[3] conditions under which a bilinear form can be "inverted", see https://en.wikipedia.org/wiki/Weak_formulation and (Brézis, 2011)

[4] regarding tree methods see Chapter V.

[5] See Fournié, Lasry, Lebuchoux, and Lions (2001) for a PDE version of the Girsanov transformation.

[6] unless $\mathcal{O}$ is already bounded from the beginning

3. ***Discretizing*** the localized problem by a finite difference numerical scheme defined on a suitable mesh over $\mathcal{O}$.

4. ***Solving*** the resulting linear algebra problem numerically in the values of an approximate solution defined at mesh nodes.

To exploit the parabolic structure of the pricing equations, the time dimension is typically treated separately at step iii. The problem is then solved iteratively in time at step iv, which saves one dimension of storage cost.

A.2 Convergence Analysis

Given a mesh discretizing $[0,T]\times\mathbb{R}^d$, with time step h and space steps $k=(k_1,\dots,k_d)$, let

$$\begin{cases} F_h^k(u_h^k)=0 \text{ on } [0,T)\times\mathcal{O} \\ u_h^k=\varphi \text{ on } E\setminus\Big([0,T)\times\mathcal{O}\Big) \end{cases} \tag{5}$$

denote a fully discrete finite differences approximation scheme for (a localized version of) (1). We assume that the discretized problem (5) admits a unique solution u_h^k defined on the mesh, satisfying at every mesh node in $[0,T)\times$ or $E\setminus\Big([0,T)\times\mathcal{O}\Big)$ the related equation which appears in functional form in (5).

Under technical assumptions stated in Crandall, Ishii, and Lions (1992), the pricing equation (1) has a unique solution u in a suitable space of viscosity solutions. For simplicity we assume that u is bounded. In the purely differential case ($\lambda=0$ in X) and in case F is linear, u is in fact a classical solution to (1). The Lax equivalence theorem then states that any stable and consistent scheme F_h^k is convergent, which loosely means that $u_h^k\to u$ at mesh points as h and k go to 0, provided:

- **(consistency)** $F_h^k(u)\to 0$ at mesh points as $h,k\to 0$,
- **(stability)** u_h^k is bounded, uniformly over h,k,

where u in $F_h^k(u)$ represents the solution to (1). The essence of the Lax equivalence theorem is that any "reasonable" (consistent) scheme converges, provided it is nonexplosive (stable). We refer the reader to Morton and Mayers (1994) for the details that, in particular, involve the specification of a given norm in which u_h^k is bounded, $F_h^k(u)\to 0$ and $u_h^k\to u$ as $h,k\to 0$.

With the help of viscosity solutions, the Lax equivalence theorem can be generalized to a nonlinear monotone operator F and/or to jumps in X. For a monotone, yet still purely differential, operator F, Barles and Souganidis (1991) proved the convergence of any monotone, stable and consistent approximation scheme for (1) (assuming a viscosity solution comparison principle holds). Monotonicity of the scheme is a discrete version of the monotonicity condition (3) on F; it is satisfied by a broad family of finite difference schemes. The convergence conditions in the Barles–Souganidis theorem are thus essentially reduced to those of the Lax equivalence theorem, namely consistency and stability. Moreover, this can be extended to equations with jumps and/or to systems of equations[7].

The companion issue to convergence is convergence rate , which is the speed at which u_h^k converges to u as $h,k\to 0$. The key notion here is that of the order or the consistency of a scheme, a measure of the speed at which $F_h^k(u)\to 0$ as $h,k\to 0$. More precisely, a finite difference scheme is said to be consistent at order p in time and q in space if[8]

$$F_h^k(u)=\mathrm{O}(h^p)+\mathrm{O}(|k|^q). \tag{6}$$

The general idea is that the order of consistency (speed at which "the operator F defining u converges to the operator F_h^k defining u_h^k") determines the convergence rate of a scheme (speed at which u converges

[7] see e.g. Crépey (2013, Chapter 13).
[8] see §3.A for illustration.

to u_h^k). The order of consistency puts a bound on the convergence rate, but there can be a deterioration between order of consistency and rate of convergence in case of a lack of regularity of the domain $\mathcal{O}$ and/or of the boundary condition φ (the residual data defining u, along with F, through (1) or its localized version introduced in A.1).

B Finite Elements and Beyond

Finite element methods[9] are based on variational formulations of the pricing equation (1). They give approximate solutions defined on the whole state space E, as opposed to approximate solutions at grid points by finite difference methods. They are most naturally connected with equations expressing energy conservation principles.

They are typically heavier than finite difference methods, particularly in terms of storage cost. Indeed, a prerequisite of a finite element method is the construction of a discretization mesh, typically unstructured and adaptive, which has to be handled by the computer during the computation. This also means that finite element methods are harder to implement. One may use finite element toolboxes, but this results in less flexibility in the programming.

The additional cost of finite element methods can be justified in cases of a domain $\mathcal{O}$ with a curved boundary or depending on time t. The latter can for instance be the case for pricing a curved barrier fixed-income option, where the curved barrier arises via the exponential relation between bond rates and bond prices, or for accurately computing the exercise boundary of an American option.

With regard to the curse of dimensionality[10], an advantage of finite element methods is that they allow us to refine the approximation mesh in a more clever way than simply taking the product of univariate adaptive meshes with finite differences.

Another interesting feature of finite elements methods is the availability of powerful a priori and a posteriori error estimates theory to deal with convergence and convergence rate issues.

Practically speaking, the numerical solution of the pricing equation (1) by a finite element method, is a five steps process:

1. ***transforming*** the problem,
2. ***localizing*** it as in A.1.ii., i.e. truncating the set $\mathcal{D}$ and the integration domain in $\mathcal{I}u$, and introducing a suitable boundary condition outside the localized domain $[0,T)\times\mathcal{D}$,
3. deriving a ***weak formulation*** of the localized problem in a Sobolev space $\mathcal{H}$ and accounting for the boundary condition,
4. ***projecting*** the problem onto a finite-dimensional sub-space of finite elements $H_h^k\subset\mathcal{H}$,
5. ***solving*** the resulting high-dimensional linear algebra system in the coefficients of an approximate solution on a finite element basis.

Existence and uniqueness for a solution to the weak form of the localized problem at step iii. is typically obtained by application of a Lax–Milgram theorem. As with finite difference methods, the time dimension is, in general (but not always), treated separately. The resulting problem at step v. may thus be solved iteratively in time, at constant storage cost. To solve the linear systems arising at every time step in the algorithm, an iterative solver is required. In the context of finite element methods, where a PDE is rephrased as a variational problem, it is natural after discretization to use an optimization-based iterative solver, e.g. the conjugate gradient descent known as the "generalized minimal residual algorithm" GMRES of Saad and Schultz (1986).

[9] see e.g. Hilber, Reichmann, Schwab, and Winter (2013).
[10] see V.§4.A.

B.1 Finite Volumes

Finite volume methods can be regarded as counterparts of finite element methods in which the test-functions that are used in the variational formulation of the problem are indicator functions instead of more regular test-functions with finite elements. Finite volume methods are particularly well suited for dealing with problems with discontinuous data, such as pricing digital options.

B.2 Sparse Grid Techniques

Sparse grid methods are used to represent, integrate or interpolate high-dimensional functions (Reisinger, 2013). These methods rely on the seminal works of the Russian mathematician Smolyak, who found a clever quadrature rule to counter the curse of dimensionality[11]. The related algorithms can be tricky to implement.

For simplicity we focus on finite difference methods in the sequel.

§3 Finite Differences for Vanilla Options

We saw in I.§2 that, in the risk-neutral Black-Scholes model

$$\tfrac{dS_t}{S_t} = \kappa dt + \sigma dW_t,$$

the price process of a European vanilla option with integrable payoff $\phi(S_T)$ at T is given by $\Pi_t = v(t, S_t)$, where the pricing function v solves the following Black-Scholes PDE:

$$\begin{cases} v(T,S) = \phi(S),\ S \in (0,+\infty) \\ \partial_t v + \kappa S \partial_S v + \dfrac{1}{2}\sigma^2 S^2 \partial^2_{S^2} v - rv = 0 \text{ in } [0,T) \times (0,+\infty). \end{cases} \tag{7}$$

In log-returns $X_t = \ln(S_t)$, the option price process is given as $\Pi_t = u(t, X_t)$, where u solves the following equation:

$$\begin{cases} u(T,x) = \psi(x),\ x \in \mathbb{R} \\ \partial_t u + b\partial_x u + \dfrac{1}{2}\sigma^2 \partial^2_{x^2} u - ru = 0 \text{ in } [0,T) \times (-\infty,+\infty), \end{cases} \tag{8}$$

with $b = \kappa - \frac{\sigma^2}{2}$ and $\psi(x) = \phi(e^x)$. For "nice" terminal conditions, both (7) and (8) are well-posed in terms of classical solutions (see Part §1.A).

A Localization and Discretization in Space

Let $x = \ln(S_0)$. To solve the equation (8) numerically, we localize the space-domain to $\mathcal{D} = (x-\ell, x+\ell)$, where ℓ is chosen so that

$$\mathbb{Q}\left(|X_t - x| \le \ell,\ t \in [0,T]\right) \ge 1 - \alpha \tag{9}$$

for a "sufficiently small" $\alpha > 0$. This is achieved by setting

$$\ell = |b|T + f\sigma\sqrt{T} \tag{10}$$

for a "sufficiently high" quantile f of the Gaussian distribution.

Letting $k = \frac{2\ell}{m+1}$ and $x_j = x - \ell + jk$ for $0 \le j \le m+1$, one then approximates the differential spatial operator

$$\mathcal{A}u = \tfrac{1}{2}\sigma^2 \partial^2_{x^2} u + b\partial_x u - ru$$

by a discrete operator $\mathcal{A}^k$ acting on $\mathbb{R}^m$-valued vectors $u^k = (u^k(t,x_1), \dots, u^k(t,x_m))^\top$.

[11] see V.§4.A.

Remark 1 *For the purpose of this section, it is convenient to include the discount term $(-ru)$ in $\mathcal{A}$; elsewhere in these notes $\mathcal{A}$ refers to the generator of the factor process X[12], without the term $(-ru)$.*

A common specification is

$$\mathcal{A}^k u^k(t,x_j) = \frac{1}{2}\sigma^2 \delta^2_{x^2} u^k(t,x_j) + b\delta_x u^k(t,x_j) - r u^k(t,x_j), \tag{11}$$

with

$$\delta_x u^k(t,x_j) = \frac{1}{2k}(u^k(t,x_{j+1}) - u^k(t,x_{j-1}))$$
$$\delta^2_{x^2} u^k(t,x_j) = \frac{1}{k^2}(u^k(t,x_{j+1}) - 2u^k(t,x_j) + u^k(t,x_{j-1})),$$

where $u^k(t,x_0) = u^k(t,x-\ell)$ and $u^k(t,x_{m+1}) = u^k(t,x+\ell)$ are to be understood as notation for quantities to be defined below in terms of the $u^k(t,x_j)$, $j = 1\dots m$. By Taylor expansion, one can show that δ_x and $\delta^2_{x^2}$ are consistent approximations of order two for the spatial differential operators ∂_x and $\partial^2_{x^2}$, meaning that, for every regular test-function $\varphi(x)$, we have:

$$|\delta_x\varphi(x_j) - \partial_x\varphi(x_j)|,\ |\delta^2_{x^2}\varphi(x_j) - \partial^2_{x^2}\varphi(x_j)| \ = \ \mathrm{O}(k^2).$$

If $|\kappa|/\sigma^2$ is "large", a less accurate but more stable approximation for ∂_x is given by

$$\delta_x u^k(t,x_j) = \begin{cases} \frac{1}{k}(u^k(t,x_j) - u^k(t,x_{j-1})) & \text{if} \quad b<0 \\ \frac{1}{k}(u^k(t,x_{j+1}) - u^k(t,x_j)) & \text{if} \quad b>0. \end{cases}$$

This so-called upwind[13] discretization of $b\partial_x$ follows the characteristics of the limiting hyperbolic transport equation[14] in which $\sigma = 0$.

One then seeks an $\mathbb{R}^m$-valued time functional $u^k(t)$ satisfying the following system of ODEs: for $1 \le j \le m$,

$$\begin{cases} u^k(T,x_j) = \psi(x_j) \\ \dfrac{du^k}{dt}(t,x_j) + \mathcal{A}^k u^k(t,x_j) = 0 \text{ for } 0 \le t < T, \end{cases} \tag{12}$$

where the quantities $u^k(t,x\pm\ell)$ in (12) are to be understood as follows, given a function φ defined over the space-boundary $[0,T]\times\partial\mathcal{O}$ of the localized domain:

- in the case of a so-called Dirichlet boundary condition,

$$u^k(t,x\pm\ell) = \varphi(t,x\pm\ell);$$

- in the case of a so-called Neumann boundary condition,

$$u^k(t,x-\ell) = u^k(t,x_1) - k\partial_x\varphi(t,x-\ell)$$
$$u^k(t,x+\ell) = u^k(t,x_m) + k\partial_x\varphi(t,x+\ell).$$

Note that by letting

$$\alpha = \frac{\sigma^2}{2k^2} - \frac{b}{2k}\ ,\ \beta = -\left(\frac{\sigma^2}{k^2} + r\right)\ ,\ \gamma = \frac{\sigma^2}{2k^2} + \frac{b}{2k}, \tag{13}$$

[12] cf. IX.(28).
[13] recalling that the pricing equations are posed in backward time, with a terminal condition at time T.
[14] see e.g. Richtmyer and Morton (1967).

one has

$$\mathcal{A}^k u^k(t) = A^k u^k(t) + w^k(t), \tag{14}$$

with in the case of a Dirichlet boundary condition:

$$A^k = \begin{bmatrix} \beta & \gamma & 0 & ,\dots, & 0 & 0 \\ \alpha & \beta & \gamma & 0 & ,\dots, & 0 \\ 0 & \alpha & \beta & \gamma & ,\dots, & 0 \\ 0 & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & ,\dots, & \alpha & \beta & \gamma \\ 0 & 0 & 0 & ,\dots, & \alpha & \beta \end{bmatrix}, \; w^k(t) = \begin{bmatrix} \varphi(t, x-\ell)\alpha \\ 0 \\ \vdots \\ 0 \\ \varphi(t, x+\ell)\gamma \end{bmatrix}, \tag{15}$$

while in the case of a Neumann boundary condition:

$$A^k = \begin{bmatrix} \beta+\alpha & \gamma & 0 & ,\dots, & 0 & 0 \\ \alpha & \beta & \gamma & 0 & ,\dots, & 0 \\ 0 & \alpha & \beta & \gamma & ,\dots, & 0 \\ 0 & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & ,\dots, & \alpha & \beta & \gamma \\ 0 & 0 & 0 & ,\dots, & \alpha & \beta+\gamma \end{bmatrix}, \; w^k(t) = \begin{bmatrix} -\alpha k \partial_x \varphi(t, x-\ell) \\ 0 \\ \vdots \\ 0 \\ \gamma k \partial_x \varphi(t, x+\ell) \end{bmatrix}. \tag{16}$$

B θ-Schemes in Time

We now discuss time-discretization of the ODE system (12). We will focus on the so-called θ-schemes, which in the case of the parabolic equation (8) may be summarized as follows[15]. Given a parameter $\theta \in [0,1]$, we choose a time discretization step h such that $T = nh$ and we construct a fully discrete approximation $u_h^k(t_i, x_j) \equiv u_i^j$, where the $\mathbb{R}^m$-valued vectors $u_i = (u_i^j)^{1 \le j \le m}$ satisfy:

$$\begin{cases} u_n = \psi, \text{ and, for } i = n-1, \dots, 0, \\ h^{-1}(u_{i+1} - u_i) + \mathcal{A}^k \left(\theta u_i + (1-\theta) u_{i+1}\right) = 0. \end{cases}$$

Or equivalently in view of (14):

$$\begin{cases} u_n = \psi, \text{ and for } i = n-1, \dots, 0, \\ \left[I - h\theta A^k\right] u_i = \left[I + h(1-\theta) A^k\right] u_{i+1} + h w_i, \end{cases} \tag{17}$$

where

$$w_i = \theta w^k(t_i) + (1-\theta) w^k(t_{i+1}). \tag{18}$$

For $\theta = 0$ we get the so-called explicit scheme, for $\theta = 1$ the fully implicit scheme and for $\theta = \frac{1}{2}$ the Crank-Nicholson scheme.

Once we have computed the approximated prices u_i^j, we can recover the delta and the gamma,

$$\Delta = \partial_S v = e^{-x} \partial_x u \,, \; \Gamma = \partial_{S^2}^2 v = e^{-2x} \left(\partial_{x^2}^2 u - \partial_x u\right),$$

by their approximations

$$\Delta_i^j = e^{-x_j} \frac{u_i^{j+1} - u_i^{j-1}}{2k}$$
$$\Gamma_i^j = e^{-2x_j} \left(\frac{u_i^{j+1} - 2u_i^j + u_i^{j-1}}{k^2} - \frac{u_i^{j+1} - u_i^{j-1}}{2k} \right).$$

For m odd, $x = \ln(S_0)$ lies in the space grid. Otherwise some interpolation has to be used to recover the approximate price and Greeks at x.

[15] see, e.g., Lamberton and Lapeyre (1996); Morton and Mayers (1994).

B.1 The Explicit Scheme

Let us first discuss the explicit scheme $\theta = 0$. By definition of A^k in (15) (in the case of a Dirichlet condition), the θ-scheme (17), with $\theta = 0$, can be rewritten as: $u_n = \psi$ and for all $i = n-1, \dots, 0$, $j = 1, \dots, m$:

$$u_i^j = p_- u_{i+1}^{j-1} + p\, u_{i+1}^j + p_+ u_{i+1}^{j+1}, \tag{19}$$

in which

$$p_- = h\alpha\,,\ p = 1 + h\beta\,,\ p_+ = h\gamma$$

and $u_{i+1}(x \pm \ell) := \varphi((i+1)h, x \pm \ell)$. Using the specification of the coefficients α, β and γ in (13), one can show that this scheme is[16]:

- stable, provided $h \leq \frac{k^2}{\sigma^2 + rk^2}$ (and $\sigma^2 > |b|k$, but for $\sigma > 0$ this is always satisfied for sufficiently small k);
- consistent of order one in time and two in space.

B.2 Implicit Schemes

For $\theta > 0$, one has to solve, in (17), at every time step $i < n$, a linear equation

$$Mu_i = Nu_{i+1} + hw_i, \tag{20}$$

where w_i was defined in (18), and $M = I - h\theta A^k$ and $N = I + h(1-\theta)A^k$ are tridiagonal matrices of the form

$$\begin{pmatrix} p & p_+ & 0 & \dots & 0 & 0 \\ p_- & p & p_+ & 0 & \dots & 0 \\ 0 & p_- & p & p_+ & \dots & 0 \\ 0 & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & p_- & p & p_+ \\ 0 & 0 & 0 & \dots & p_- & p \end{pmatrix}. \tag{21}$$

Example 1 *In the case* (15) *of a Dirichlet boundary condition, one has for* M *:*

$$p_- = -\theta h\alpha, \quad p = 1 - \theta h\beta, \quad p_+ = -\theta h\gamma, \tag{22}$$

and for N *:*

$$p_- = (1-\theta)h\alpha, \quad p = 1 + (1-\theta)h\beta, \quad p_+ = (1-\theta)h\gamma,$$

(consistent with (19) *for the explicit scheme* $\theta = 0$*).*

One can show[17] that implicit schemes for the Black-Scholes equation (8) are:

- unconditionally stable for $\theta \geq \frac{1}{2}$; otherwise they are like the explicit scheme subject to a suitable stability condition;
- consistent of order one in time and two in space, except for the Crank–Nicholson scheme ($\theta = \frac{1}{2}$), which is consistent of order two in time and space.

In Figure 1 we have plotted the relative error at time 0 as a function of the spot price S_0, when pricing a European vanilla call option with the explicit, the fully implicit and the Crank–Nicholson schemes. As expected the Crank–Nicholson is more accurate, at least around the money.

[16] See §2.A.2 for the notions of stability and consistency.
[17] see Lamberton and Lapeyre (1996, Theorem 5.2.4 page 137).

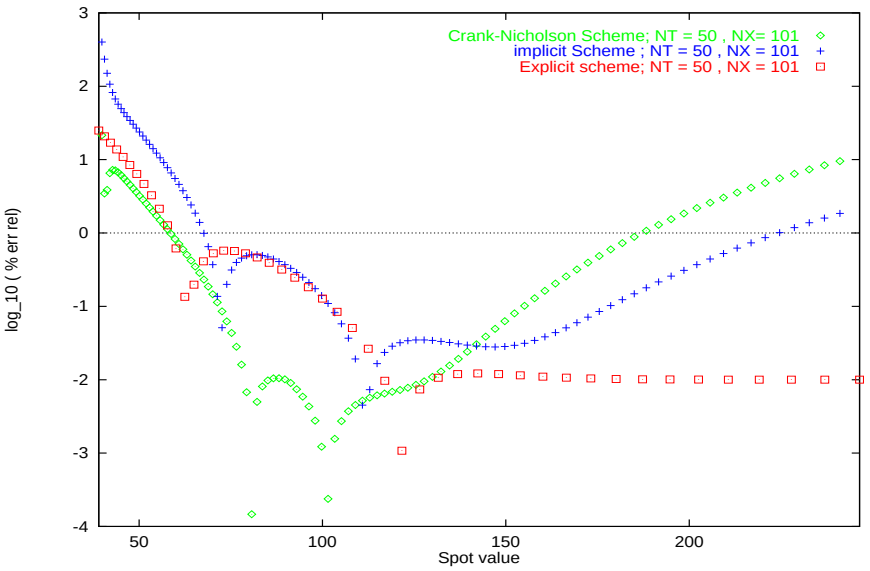


Figure 1: *Pricing of a European call option by θ-schemes, with $\theta = 0, \frac{1}{2}$ and 1 ($r = 10\%, \sigma = 20\%, T = 1y, K = 100$).*

C Solving the Linear Systems

An implicit θ-scheme (with $\theta > 0$) requires, at each time step, the solution of a linear system of the form $Mu = v$, where u and v are m-dimensional vectors and the matrix M is tridiagonal. We now describe two algorithms for solving such linear systems.

C.1 Solution by Gauss Factorization

Let $M = LU$ denote the Gauss factorization of a (regular) matrix M, where L is lower triangular and U is upper triangular, with only ones on the diagonals. The linear system $MUu = v$ can be decomposed into $Lz = v, Uu = z$. It is easy to see that if, as in our case, the matrix M is tridiagonal, then so are also L and U, so that only the upper diagonal of U and the two diagonals of L need be found. We thereby obtain the following procedure for solving $Mu = v$, known as the Thomas algorithm (Morton and Mayers, 1994; Lamberton and Lapeyre, 1996):

- $b'_m = b_m$, $z_m = v_m$ and, for j decreasing from $m-1$ to 1,

$$b'_j = b_j - c_j a_{j+1}/b'_{j+1},$$
$$z_j = v_j - c_j z_{j+1}/b'_{j+1};$$

- $u_1 = z_1/b'_1$, and, for j increasing from 2 to m,

$$u_j = (z_j - a_j u_{j-1})/b'_j$$

(assuming non zero pivots b'_j).

C.2 Iterative Solution

An alternative, which in the case of a tridiagonal system is only justified by its programming simplicity, is to use an iterative scheme. The matrix M resulting from the discretisation of a parabolic PDE is

typically diagonal-dominant, meaning that it has a "large" diagonal part D as compared with the residu $R = M - D$ (see e.g. (22)). With the so-called successive over-relaxation scheme, the linear system $Mu = v$ is rewritten as $Du = v - Ru$. Starting from an initial condition u^0, a solution is then computed as the limit of the "contracting Picard iteration" (since D is "large" and D^{-1} is therefore "small"):

$$u^{k+1} = D^{-1}(v - Ru^k); \tag{23}$$

Or, more generally, given an over-relaxation paramater $1 \leq \omega < 2$,

$$u^{k+1} = u^k + \omega(\widetilde{u}^{k+1} - u^k) \text{ where } \widetilde{u}^{k+1} = D^{-1}(v - Ru^k).$$

The detailed algorithm appears as follows: set $k = 0$ and until $|u^{k+1} - u^k|$ is less than some specified tolerance:

• ***(Jacobi iteration)*** Form an auxiliary vector $\widetilde{u}^{k+1} = (\widetilde{u}_j^{k+1})_{1\leq j\leq m}$, for $1 \leq j \leq m$, by

$$\widetilde{u}_j^{k+1} = \frac{1}{M_{jj}}(v_j - \sum_{l<j} M_{j,l}u_l^k - \sum_{l>j} M_{j,l}u_l^k). \tag{24}$$

Here a possible refinement known as Gauss–Seidel iteration consists in using $\widetilde{u}_l^{k+1}$ instead of u_l^k in the first sum.

• ***(over-relaxation)*** Let $u^{k+1} = u^k + \omega(\widetilde{u}^{k+1} - u^k)$ and set $k = k + 1$.

D Adding Jumps

We now add jumps in S, considering the Merton model

$$\frac{dS_t}{S_{t-}} = (\kappa - \lambda\bar{J})dt + \sigma dW_t + J_{(t)}dN_t \tag{25}$$

of I.§3.B. By the usual martingale and Markov arguments, the price process of the option can be represented as $\Pi_t = u(t, X_t)$, with $X_t = \ln(S_t)$, where the pricing function u solves the partial integro-differential equation

$$\begin{cases} u(T, x) = \psi(x), \ x \in \mathbb{R} \\ \partial_t u + \mathcal{C}u + \mathcal{I}u = 0 \text{ in } [0, T) \times \mathbb{R}, \end{cases} \tag{26}$$

with

$$\mathcal{C}u(t, x) = \left(\frac{1}{2}\sigma^2\partial^2_{x^2}u + a\partial_x u - ru\right)(t, x)$$
$$\mathcal{I}u(t, x) = \lambda\int_{\mathbb{R}} (u(t, x + y) - u(t, x))\, n(y)dy,$$

in which $a = b - \lambda\bar{J} = \kappa - \frac{1}{2}\sigma^2 - \lambda\bar{J}$ and where n is the $\mathcal{N}(\varrho, \nu)$-density.

D.1 Localization

Localization is essentially performed as in A, except that:

- the coefficient b is replaced by a in (10);
- the boundary $\partial\mathcal{D} = \{x - \ell\} \cup \{x + \ell\}$ is replaced by the "thick" boundary layer $\partial\mathcal{D}^j = [x - \ell - \underline{z}^-, x - \ell] \cup [x + \ell, x + \ell + \overline{z}^+]$, where $\underline{z}$ and $\overline{z}$ are such that $\int_{\underline{z}}^{\overline{z}} n(z)dz \approx 1$[18];

[18] cf. cf. step ii in §2.A.1.

- we solve the following localized problem on $\bar{\mathcal{D}}_j = \mathcal{O} \cup \partial\mathcal{D}^j$:

$$\begin{cases} u(T,x) = \psi(x),\ x \in \mathbb{R} \\ u(t,x) = \varphi(t,x),\ (t,x) \in [0,T) \times \partial\mathcal{D}^j \\ (\partial_t u + \mathcal{C}u + \mathcal{J}u + \mathcal{K})(t,x) = 0,\ (t,x) \in [0,T) \times \mathcal{O}, \end{cases} \tag{27}$$

where φ is a Dirichlet condition such that $\varphi(T,\cdot) = \psi$ and, for $(t,x) \in [0,T) \times \mathcal{O}$, with

$$\begin{array}{ll} \mathcal{J}u(t,x) & = \lambda \int_{\mathcal{D}} u(t,y) n(y-x) dy - \lambda u(t,x) \\ \mathcal{K}(t,x) & = \lambda \int_{\partial\mathcal{D}^j} \varphi(t,y) n(y-x) dy. \end{array} \tag{28}$$

D.2 Discretization

Let

$$k = \frac{2\ell}{m+1},\ \imath = \left[\frac{\underline{z}^-}{k}\right] + 1,\ \jmath = \left[\frac{\overline{z}^+}{k}\right] + 1;$$

let $x_j = x - \ell - \underline{z}^- + jk$ for $j = 1 - \imath, \ldots, m + \jmath$; we write $w_l = w(lk)$ for any integer l.

Finite Differences in Space To approximate the differential operator $\mathcal{C}$, one can use the finite difference

$$\mathcal{C}^k u(t,x_j) = [a\delta_x^\eta + \tfrac{\sigma^2}{2}\delta_{x^2}^2 - r]u(t,x_j),$$

with

$$\delta_{x^2}^2 u(t,x_j) = \tfrac{u(t,x_{j+1}) - 2u(t,x_j) + u(t,x_{j-1})}{k^2}$$

$$\delta_x^\eta u(t,x_j) = \tfrac{u(t,x_j) - u(t,x_{j-1})}{k} + \eta \tfrac{u(t,x_{j+1}) - 2u(t,x_j) + u(t,x_{j-1})}{k},$$

where, for stability reasons, η is chosen so that

$$\eta = \begin{cases} \frac{1}{2} & \text{if } ka \le \frac{\sigma^2}{2} \\ 0 & \text{if } ka > \frac{\sigma^2}{2} \text{ and } a > 0 \\ 1 & \text{if } ka > \frac{\sigma^2}{2} \text{ and } a < 0. \end{cases}$$

Jumps Approximation We approximate (writing n_{l-j} as a shorthand for $n(x_l - x_j)$)

$$\begin{array}{l} \mathcal{J}u(t,x_j) \approx \mathcal{J}^k u(t,x_j) = \lambda k \sum_{l;x_l \in \mathcal{O}} u(t,x_l) n_{l-j} - \lambda u(t,x_j) \\ \mathcal{K}(t,x_j) \approx \mathcal{K}^k(t,x_j) = \lambda k \sum_{l;x_l \in \partial\mathcal{D}^j} \varphi(t,x_l) n_{l-j}, \end{array} \tag{29}$$

in which the sums can be quickly computed for every j by discrete FFT as follows[19]. The discrete correlation f of two real sequences g and h of period m is defined by

$$f_j = \sum_{l=1}^{m} g_{j+l} h_l\,,\ 1 \le j \le m.$$

The discrete correlation theorem says that $Ff = (Fg)(\overline{Fh})$, where Ff, Fg and Fh are the discrete Fourier transforms[20] of f, g and h, and $\overline{Fh}$ denotes the complex conjugate of Fh. So, to compute f (e.g., either sum in (29)), we FFT the two data sets g and h, we multiply one Fourier transform by the complex conjugate of the other and we inverse transform the product. Formally, the result is a complex vector of length m. However, all its imaginary parts are equal to zero since the original data sets are both real.

[19] see also the end of I.§1.F.2 and D'Halluin, Forsyth, and Vetzal (2005).
[20] see I.(21).

θ-Schemes in Time Given a time grid $(t_i)_{0\le i\le n}$ with uniform time-step h, one can solve (27) by the following θ-scheme, in which $u_i^j \approx u(t_i, x_j)$, $u_i = (u_i^j)^{1\le j\le m}$ and where C^k, J^k and $\mathcal{K}_i^k$ are the matrix-vector forms of $\mathcal{C}^k$, and $\mathcal{J}^k$ and $\mathcal{K}^k(t,\cdot)$:

$$\begin{aligned} h^{-1}(u_{i+1}-u_i) + C^k[\theta_c u_i + (1-\theta_c)u_{i+1}] + J^k[\theta_j u_i + (1-\theta_j)u_{i+1}] + \\ [\theta_j \mathcal{K}_i^k + (1-\theta_j)\mathcal{K}_{i+1}^k] = 0, \end{aligned} \tag{30}$$

for some constant parameters $\theta_c,\ \theta_j \in [0,1]$. We give the developed j by j form of (30) in two specific cases. In both cases we impose a Dirichlet condition φ on $[0,T]\times\partial\mathcal{D}^j$, with $u_\cdot^l$ understood as $\varphi_\cdot^l$ for every $l \in \{1-\imath,\dots,0\}\cup\{m+1,\dots,m+\jmath\}$ in (31) and (32).

Explicit scheme $\theta_c = \theta_j = 0$: For every $i = n-1,\dots,0$, for every $j = 1,\dots,m$:

$$\begin{aligned} u_i^j = h\Big(\tfrac{\sigma^2}{2k^2} - \tfrac{a}{k}\eta\Big)u_{i+1}^{j-1} + \Big(1 - h\Big(\tfrac{\sigma^2}{k^2} + \tfrac{a}{k}(1-2\eta) + r\Big)\Big)u_{i+1}^j + h\Big(\tfrac{\sigma^2}{2k^2} + \tfrac{a}{k}(1-\eta)\Big)u_{i+1}^{j+1} \\ + h\lambda\left(k\sum_{l\in\{1-\imath,\dots,m+\jmath\}} n_{l-j}u_{i+1}^l - u_{i+1}^j\right). \end{aligned} \tag{31}$$

This scheme is computationally feasible but potentially unstable, and it only has order one of time-consistency.

Asymmetric scheme $\theta_c = \frac{1}{2}$, $\theta_j = 0$: For $i = n-1,\dots,0$, one must solve the linear system $Mu_i = v_{i+1}$, where M is the m-dimensional tridiagonal matrix

$$M = \begin{pmatrix} p & p_+ & 0 & \cdot & \cdot & 0 \\ p_- & p & p_+ & 0 & \cdot & 0 \\ 0 & p_- & p & p_+ & 0 & \cdot \\ \cdot & 0 & & & & 0 \\ \cdot & \cdot & 0 & p_- & p & p_+ \\ 0 & \cdot & \cdot & 0 & p_- & p \end{pmatrix} \text{ with } \begin{cases} p_- = -\frac{h}{2}\Big(\frac{\sigma^2}{2k^2} - \frac{a}{k}\eta\Big) \\ p = 1 + \frac{h}{2}\Big(\frac{\sigma^2}{k^2} + \frac{a}{k}(1-2\eta) + r\Big) \\ p_+ = -\frac{h}{2}\Big(\frac{\sigma^2}{2k^2} + \frac{a}{k}(1-\eta)\Big) \end{cases}$$

and where, for $j = 1,\dots,m$,

$$\begin{aligned} v_{i+1}^j = &- p_- u_{i+1}^{j-1} + (2-p)u_{i+1}^j - p_+ u_{i+1}^{j+1} + h\lambda\left(k\sum_{l\in\{1-\imath,\dots,m+\jmath\}} n_{l-j}u_{i+1}^l - u_{i+1}^j\right) \\ &+ \mathbb{1}_{j=1}p_- u_i^{j-1} + \mathbb{1}_{j=m}p_+ u_i^{j+1}. \end{aligned} \tag{32}$$

This scheme is stable and efficient but some accuracy is lost due to the asymmetric treatment of the continuous and jump parts.

E American Options

By the nonlinear Feynman-Kac formula for obstacle problems, the price of an American vanilla option in the Black-Scholes model at time 0 is given by

$$\sup_{\vartheta\in\Theta} \mathbb{E}e^{-r\vartheta}\psi(X_\vartheta) = v(0,x),$$

with $X_t = \ln(S_t)$ and $x = X_0$, where v is the unique viscosity solution to the following obstacle problem:

$$\begin{cases} v(T,\cdot) = \psi \text{ on } \mathbb{R} \\ \max\left(\partial_t v + \mathcal{A}v, \psi - v\right) = 0 \text{ on } [0,T)\times\mathbb{R}. \end{cases} \tag{33}$$

E.1 Splitting Scheme

The so-called splitting θ-scheme for the obstacle problem (33) is written in vector-form as follows, in terms of M, N and w_i as in (20): $v_n = \psi$ and, for i decreasing from $n-1$ to 0,

$$M\widetilde{v}_i = Nv_{i+1} + hw_i\,,\ v_i = \max\left(\psi, \widetilde{v}_i\right). \tag{34}$$

By the results of Barles and Souganidis (1991)[21], this scheme converges to the unique viscosity solution v of (33).

F Multi-asset Options

Alternate direction implicit (ADI) schemes (Peaceman and Rachford, 1955; Morton and Mayers, 1994) are the industry finite difference standard to cope with multivariate pricing problems. We now describe such a scheme in a bivariate Black-Scholes setting, where two underlying stock prices[22] satisfy the following stochastic differential equations:

$$\left\{\begin{array}{l} dS_t^1 = S_t^1\left(\kappa_1 dt + \sigma_{11}dW_t^1 + \sigma_{12}dW_t^2\right)\\ dS_t^2 = S_t^2\left(\kappa_2 dt + \sigma_{21}dW_t^1 + \sigma_{22}dW_t^2\right),\end{array}\right.$$

for independent Brownian motions W^1 and W^2, with

$$\left(\begin{array}{c}\kappa_1\\ \kappa_2\end{array}\right) = \left(\begin{array}{c} r-q_1\\ r-q_2\end{array}\right),\ \Sigma = \left(\begin{array}{cc}\sigma_{1,1} & \sigma_{1,2}\\ \sigma_{2,1} & \sigma_{2,2}\end{array}\right) = \left(\begin{array}{cc}\sigma_1 & 0\\ \rho\sigma_2 & \sqrt{1-\rho^2}\sigma_2.\end{array}\right)$$

In order to apply the ADI method, it is better to work directly with the independent Brownian motions W^1 and W^2. For this purpose we introduce

$$\varphi(t,x,y) = \phi(t, S_0^1 e^{b_1 t+\sigma_1 x}, S_0^2 e^{b_2 t+\sigma_2(\rho x+\sqrt{1-\rho^2}y)}),$$

with $(b_1, b_2) = (\kappa_1 - \frac{1}{2}\sigma_1^2, \kappa_2 - \frac{1}{2}\sigma_2^2)$. The payoff process $\phi(S_t^1, S_t^2)$ is thus rewritten as $\varphi(t, W_t^1, W_t^2)$.

The time-t price of a European option with payoff $\phi(S_T^1, S_T^1)$ at T is then given by

$$\mathbb{E}_t e^{-r(T-t)}\phi(S_T^1, S_T^2) = w(t, S_t^1, S_t^2),$$

or

$$\mathbb{E}e^{-r(T-t)}\varphi(T, W_T^1, W_T^2) = u(t, W_t^1, W_t^2),$$

where $w = w(t, S_1, S_2)$ satisfies a bivariate Black-Scholes equation:

$$\left\{\begin{array}{l} w(T,S_1,S_2) = \phi(T,S_1,S_2) \text{ on } (0,+\infty)^2\\ \partial_t w(t,S_1,S_2) + \kappa_1 S_1\partial_{S_1}w(t,S_1,S_2) + \kappa_2 S_2\partial_{S_2}w(t,S_1,S_2)\\ \qquad +\frac{1}{2}(\sigma_1 S_1)^2\partial^2_{(S_1)^2}w(t,S_1,S_2) + \frac{1}{2}(\sigma_2 S_2)^2\partial^2_{(S_2)^2}w(t,S_1,S_2)\\ \qquad +\rho\sigma_1\sigma_2 S_1 S_2\partial^2_{S_1,S_2}w(t,S_1,S_2) - rw(t,S_1,S_2) = 0 \quad \text{on } [0,T)\times(0,+\infty)^2.\end{array}\right. \tag{35}$$

whereas $u = u(t,x,y)$ solves the following bivariate heat equation:

$$\left\{\begin{array}{l} u(T,x,y) = \varphi(T,x,y) \text{ on } \mathbb{R}^2\\ \partial_t u(t,x,y) + \dfrac{1}{2}\partial^2_{x^2}u(t,x,y) +\\ \qquad \frac{1}{2}\partial^2_{y^2}u(t,x,y) - ru(t,x,y) = 0 \quad \text{on } [0,T)\times\mathbb{R}^2.\end{array}\right. \tag{36}$$

For the numerical solution of (36) by finite differences:

[21] see also Crépey (2013, Section 13.2.3).

[22] see IV.§6.B for explicit multi-asset option payoffs.

- ***localize*** the domain in space to a set $\mathcal{O} = (-\ell, \ell)^2$, introducing a suitable condition at the spatial boundary of $[0,T] \times \mathcal{O}$;
- introduce a time-space mesh $(t,x,y) = (ih, j_1k_1, j_2k_2)$ on $[0,T] \times \mathcal{O}$, with mesh steps h, k_1, k_2, and ***discretize*** the localized problem on the mesh by a suitable finite difference scheme, such as the one described in the next subsection.

Regarding the deltas, note that we have $w(t,S_1,S_2) = u(t,x,y)$, where

$$\begin{pmatrix} \ln S_1 \\ \ln S_2 \end{pmatrix} = \begin{pmatrix} \ln S_0^1 + b_1 T \\ \ln S_0^2 + b_2 T \end{pmatrix} + \Sigma \begin{pmatrix} x \\ y \end{pmatrix}.$$

Therefore

$$\begin{aligned} \begin{pmatrix} S_1\partial_{S_1} \\ S_2\partial_{S_2} \end{pmatrix} w(t,S_1,S_2) &= \Sigma^{-1} \begin{pmatrix} \partial_x \\ \partial_y \end{pmatrix} u(t,x,y) \\ &= \frac{1}{\sigma_1\sigma_2\sqrt{1-\rho^2}} \begin{pmatrix} \sigma_2\sqrt{1-\rho^2} & 0 \\ -\sigma_2\rho & \sigma_1 \end{pmatrix} \begin{pmatrix} \partial_x \\ \partial_y \end{pmatrix} u(t,x,y). \end{aligned}$$

The time-0 deltas $\Delta_0^1 = \partial_{S_1} w(0,S_0^1,S_0^2)$ and $\Delta_0^2 = \partial_{S_2} w(0,S_0^1,S_0^2)$ are then given in terms of u as

$$\Delta_0^1 = \tfrac{e^{-x_1}}{\sigma_1}(\partial_x u)(0,0,0)\,,\ \Delta_0^2 = \tfrac{e^{-x_2}}{\sqrt{1-\rho^2}}(\tfrac{-\rho\partial_x u}{\sigma_1} + \tfrac{\partial_y u}{\sigma_2})(0,0,0). \tag{37}$$

Let $u_i^{j_1,j_2}$ denote the solution to the discretized pricing equation. Approximate deltas are then retrieved from (37) by substituting finite differences $\delta_x u$ and $\delta_y u$ for $\partial_x u$ and $\partial_y u$, e.g.

$$(\delta_x u)_i^{j_1,j_2} = \frac{u_i^{j_1+1,j_2} - u_i^{j_1-1,j_2}}{2k_1},\ (\delta_y u)_i^{j_1,j_2} = \frac{u_i^{j_1,j_2+1} - u_i^{j_1,j_2-1}}{2k_2}.$$

F.1 The ADI Scheme

The ADI scheme consist in decomposing each time step into two parts, the first implicit in x and the second implicit in y, resulting in the following approximation scheme for (36): $u_n = \varphi$ and, for $i = n-1,\ldots,0$,

$$\begin{cases} \frac{2}{h}(u_{i+1} - u_{i+\frac12}) + \frac12\delta_{x^2}^2 u_{i+\frac12} + \frac12\delta_{y^2}^2 u_{i+1} - \frac12 r u_{i+\frac12} - \frac12 r u_{i+1} = 0 \\ \frac{2}{h}(u_{i+\frac12} - u_i) + \frac12\delta_{x^2}^2 u_{i+\frac12} + \frac12\delta_{y^2}^2 u_i - \frac12 r u_{i+\frac12} - \frac12 r u_i = 0 \end{cases}$$

or, equivalently,

$$\begin{cases} \big[(1+\frac{hr}{4})I - \frac{h}{4}\delta_{x^2}^2\big] u_{i+\frac12} = \big[(1-\frac{hr}{4})I + \frac{h}{4}\delta_{y^2}^2\big] u_{i+1} \\ \big[(1+\frac{hr}{4})I - \frac{h}{4}\delta_{y^2}^2\big] u_i = \big[(1-\frac{hr}{4})I + \frac{h}{4}\delta_{x^2}^2\big] u_{i+\frac12}, \end{cases} \tag{38}$$

in which

$$\begin{aligned} (\delta_{x^2}^2 u)_i^{j_1,j_2} &= \frac{u_i^{j_1+1,j_2} - 2u_i^{j_1,j_2} + u_i^{j_1-1,j_2}}{k_1^2} \\ (\delta_{y^2}^2 u)_i^{j_1,j_2} &= \frac{u_i^{j_1,j_2+1} - 2u_i^{j_1,j_2} + u_i^{j_1,j_2-1}}{k_2^2}. \end{aligned}$$

Each time step i takes the form

$$\begin{cases} M^{j_2} u_{i+\frac12}^{\cdot,j_2} = N^{j_2} u_{i+1}^{\cdot,j_2}\,, \text{ for every } j_2 \\ P^{j_1} u_i^{j_1,\cdot} = Q^{j_1} u_{i+\frac12}^{j_1,\cdot}\,, \text{ for every } j_1, \end{cases} \tag{39}$$

for suitable "one-dimensional tridiagonal" matrices M^{j_2}, N^{j_2}, P^{j_1}, Q^{j_1}. So each time step reduces to (m_1+m_2) implicit tridiagonal one-dimensional problems, each solvable by the Thomas algorithm of

C.1. This is in general a far better alternative than having to solve the $(m_1 m_2)$-dimensional linear system that would arise from a bivariate implicit discretization[23].

Unless simple transformations as in the above case allow elimination of the correlation from a pricing problem, additional cross-derivatives show up in the equations[24]. These can be dealt with explicitly (i.e. put on the right-hand side in (38)) and the ADI scheme is still applicable, but subject to stability conditions which become stringent in multivariate settings. See Hout and Welfert (2007) or Duffy (2006) for alternative schemes.

F.2 American Options

The time-t price of an American vanilla option is given by

$$\max_{\vartheta \in \Theta_t} \mathbb{E} e^{-r(\vartheta - t)} \varphi(\vartheta, W^1_\vartheta, W^2_\vartheta) = v(t, W^1_t, W^2_t),$$

where v is the solution to the following obstacle problem

$$\left\{ \begin{array}{l} v(T,x,y) = \varphi(T,x,y) \text{ on } \mathbb{R}^2 \\ \max\left(\partial_t v + \frac{1}{2}\partial^2_{x^2} v + \frac{1}{2}\partial^2_{y^2} v - rv, \varphi - v\right) = 0 \text{ on } [0,T) \times \mathbb{R}^2. \end{array} \right. \tag{40}$$

To solve (40):

- we ***localize*** the equation to a set $\mathcal{O} = (-\ell, \ell)^2$, introducing a suitable condition at the spatial boundary of $[0,T] \times \mathcal{O}$;
- we ***discretize and solve*** the localized problem on a grid.

A convergent finite difference approximation scheme[25] is obtained by combining the previous ADI finite difference method with the splitting method of E.1.

§4 Finite Differences for Exotic Options

In this section we deal with finite difference methods for path-dependent options on an underlying S. A unified perspective on the various situations that are studied below is possible in the setup of the functional Itô calculus of (Dupire, 2019; Cont and Fournié, 2013).

In the Black–Scholes setup to which we restrict ourselves below, explicit pricing and greeking formulas are available for many variants of the lookback and barrier options of Parts A and B. However, exotic options are smile and even smile dynamics sensitive, where the latter means that, for a given (implied or local volatility) surface reflecting the time-0 prices of vanilla options, even time-0 prices of exotic options and, a fortiori their time-0 greeks, depend (sometimes very strongly) on the joint distribution of the S and of the volatility in the future. Hence, for such products, using a pricing and hedging model embedding the correct smile and smile dynamics becomes a crucial trading issue.

A Lookback Options

The payoff of a lookback option is of the form $\phi(S_T, M_T)$, where $M_t = \sup_{0 \le s \le t} S_s$. In the Black-Scholes model, the pair (S, M) is a Markov process so that the price Π_t of a lookback option can be represented

[23] However, the sparseness of the corresponding matrix may be exploited in an iterative solution.
[24] like in the equation (35) in the original financial variables.
[25] see Villeneuve and Zanette (2002).

as $u(t, S_t, M_t)$ for a pricing function $u = u(t, S, M)$. The process $M = \max_{0 \le s \le .} S_s$ is nondecreasing continuous, hence it doesn't contribute to any bracket and therefore the[26] Itô formula yields

$$e^{rt} d\big(e^{-rt} u(t, S_t, M_t)\big) = \left(\partial_t u + \frac{1}{2}\sigma^2 S^2 \partial^2_{S^2} u + \kappa S \partial_S u - ru\right)(t, S_t, M_t)dt + \partial_M u(t, S_t, M_t)\, dM_t + \sigma S_t \partial_S u(t, S_t, M_t) dW_t.$$

From the local martingale property of the discounted price $(e^{-rt}\Pi_t)$ we deduce that, for $t < T$,

$$\begin{aligned} &\partial_t u + \frac{1}{2}\sigma^2 S^2 \partial^2_{S^2} u + \kappa S \partial_S u - ru = 0 \text{ on } \{S < M\} \\ &\partial_M u = 0 \text{ on } \{S = M\}, \end{aligned} \tag{41}$$

along with the terminal condition $u(T, S, M) = \phi(S, M)$. The pricing function u of a lookback option thus satisfies the Black-Scholes PDE on the subset $\{S \le M\}$ of a bivariate state-space (S, M). On the boundary $\{S = M\}$, u satisfies an oblique homogeneous Neumann condition $\partial_M u = 0$. The PDE (41) is bivariate (with two space dimensions S and M), but it is a simple one from a finite differences viewpoint. In fact, as the corresponding generator only acts in the S direction, this PDE can be solved at each pricing time step, going backward in a time grid, by a standard univariate finite differences θ scheme in the S direction, independently for each value of M in its space grid. As compared with the univariate case of vanilla options, the computation time, however, is one order of magnitude larger (due to added necessity of looping over the M grid).

B Barrier Options

With a barrier option the right to exercise the payoff at maturity depends on additional events such as the underlying having crossed or reached certain levels on $[0, T]$. Such options were created as a way to provide the insurance value of an option without charging as much premium.

B.1 Up-and-out barrier example.

Let us thus consider an up-and-out barrier option with trigger level H, rebate R and "vanilla component" (payoff were it not for the barrier) $\phi(S_T)$. Under certain covenants, the rebate (if triggered) of the option is delivered in arrears at the maturity time T of the option. The option is then a special case of lookback option[27]. But under the most common covenant the rebate is paid at time

$$\tau = \inf\{t \in [0, T]; S_t \ge H\},$$

in which case the effective payoff of the option is

$$\mathbb{1}_{\{\tau > T\}} \phi(S_T) + \mathbb{1}_{\{\tau \le T\}} R = \bar{\phi}(\bar{\tau}, S_{\bar{\tau}}), \tag{42}$$

paid at the random time $\bar{\tau} = \tau \wedge T$, with related price process $\Pi_t = \beta_t^{-1} \mathbb{E}_t \left[\beta_\tau \bar{\phi}(\tau, S_\tau)\right]$ (for $t \le \bar{\tau}$). In the Black-Scholes model, it follows from martingale and Markov arguments that $\Pi_t = u(t, X_t)$, with $X_t = \ln(S_t)$, where the pricing function u satisfies with $h = \ln(H)$ and $\psi(x) = \phi(e^x)$:

$$\begin{cases} u(T, x) = \psi(x) \text{ on } (-\infty, h) \\ u(t, x) = R \text{ on } [0, T] \times \{h\} \\ \partial_t u + \frac{1}{2}\sigma^2 \partial^2_{x^2} u + b \partial_x u - ru = 0 \text{ on } [0, T) \times (-\infty, h). \end{cases} \tag{43}$$

[26] semimartingale.
[27] cf. A.

B.2 Common forms of barrier options.

More generally, denoting by $v(t,x)$ the pricing function of the "vanilla component" of a barrier option (option with payoff $\phi(S_T)$ at time T), the pricing function $u = u(t,x)$ of a barrier option satisfies

$$\begin{cases} u(T,x) = \varphi(x) \text{ on } \mathcal{O} \\ u(t,x) = R(t,x) \text{ on } [0,T] \times \partial\mathcal{O} \\ \partial_t u + \frac{1}{2}\sigma^2 \partial^2_{x^2} u + b\partial_x u - ru = 0 \text{ on } [0,T) \times \mathcal{O}, \end{cases} \tag{44}$$

where, letting also $l = \ln(L)$ below, we have after localisation in space[28]:

- in the already seen case of an up-and-out barrier at the level H: $\mathcal{O} = (x-\ell, h)$, $\varphi(x) = \psi(x), R(t,h) = R$;
- in the case of a down-and-out barrier at the level L: $\mathcal{O} = (l, x+\ell)$, $\varphi(x) = \psi(x), R(t,l) = R$;
- in the case of double-out barriers at levels L and H: $\mathcal{O} = (l,h)$, $\varphi(x) = \psi(x), R(t,l) = R(t,h) = R$;
- in the case of an up-and-in barrier at the level H: $\mathcal{O} = (x-\ell, h)$, $\varphi(x) = R$, $R(t,h) = v(t,h)$;
- in the case of a down-and-in barrier at the level L: $\mathcal{O} = (l, x+\ell)$, $\varphi(x) = R$, $R(t,l) = v(t,l)$;
- in the case of double-in barriers at levels L and H: $\mathcal{O} = (l,h)$, $\varphi(x) = R$, $R(t,l) = v(t,l)$, $R(t,h) = v(t,h)$.

The localized pricing equation (44) can then be solved by finite differences θ schemes much like in §3.B.

C Asian Options

Asian options correspond to payoff processes of the form $\phi(t, S_t, I_t)$, where $I_t = \int_0^t S_s ds$. In the Black-Scholes model the pair (S_t, I_t) is Markov with generator

$$\mathcal{A}_{S,I} = \frac{1}{2}\sigma^2 S^2 \partial^2_{S^2} + \kappa S \partial_S + S\partial_I. \tag{45}$$

This generator is degenerate[29] in the I-variable, which results in PDEs "in one-and-a-half space dimension", unless it can be trimmed down to one in special cases such as the following one.

C.1 Asian Fixed Strike Put Option

This is the option with payoff

$$\xi = \left(K - \frac{I_T}{T}\right)^+ = \phi(I_T) \tag{46}$$

and related price process on $[0,T]$

$$\Pi_t = \beta_t^{-1} \mathbb{E}_t \left[\beta_T \xi\right] = u(t, S_t, I_t). \tag{47}$$

The pricing equation is written:

$$\begin{cases} u(T,S,I) = \phi(I) \\ \partial_t u + \mathcal{A}_{S,I} u = ru, \quad 0 \le t < T. \end{cases} \tag{48}$$

See Zvan, Forsyth, and Vetzal (1998) for the numerical issues related to the degeneracy of the equation in the I-variable.

[28] cf. §3.A.

[29] hyperbolic, without diffusion term (Richtmyer and Morton, 1967).

Alternatively (Rogers and Shi, 1995), one can observe that $\frac{\phi(I_T)}{S_T} = \eta_T^+$ for the process η_t such that

$$\begin{aligned}\eta_t &= \frac{1}{S_t}\left(K - \frac{I_t}{T}\right)\\ d\eta_t &= \big((-S_t^{-2})dS_t + \sigma^2 S_t^{-1}dt\big)(K - \frac{I_t}{T}) + S_t^{-1}\frac{(-S_t)}{T}dt\\ &= -\eta_t(\kappa - \sigma^2)dt - \eta_t\sigma dW_t - \frac{dt}{T},\end{aligned}$$

which is Markov with generator

$$\mathcal{A}_\eta = -\left[\frac{1}{T} + (\kappa - \sigma^2)\eta\right]\partial_\eta + \frac{1}{2}\eta^2\sigma^2\partial^2_{\eta^2}. \tag{49}$$

Introducing the measure $\widetilde{\mathbb{Q}}$ associated with the numéraire S, so that $\frac{d\widetilde{\mathbb{Q}}}{d\mathbb{Q}} = \frac{S_T}{S_0 e^{\kappa T}}$[30] and $d\widetilde{W}_t = dW_t - \sigma dt$ is a $\widetilde{\mathbb{Q}}$ Brownian motion, the process η is also a $\widetilde{\mathbb{Q}}$ Markov process and the price process (47) can be represented as

$$\Pi_t = e^{qt}S_t\widetilde{\mathbb{E}}_t\left[e^{-qT}S_T^{-1}\phi(I_T)\right] = e^{-q(T-t)}S_t\widetilde{\mathbb{E}}_t\eta_T^+ = S_t v(t,\eta_t)\,,\ t \le T \tag{50}$$

In addition, the process $\beta_t\Pi_t = e^{-rt}S_t v(t,\eta_t)$ is a $\mathbb{Q}$ martingale. But Itô calculus yields, with "$\doteq$" standing for "equality up to a $\mathbb{Q}$ local martingale":

$$\begin{aligned}e^{rt}d\left(e^{-rt}S_t v(t,\eta_t)\right) &\doteq \Big(-rS_t v(t,\eta_t) + v(t,\eta_t)\kappa S_t dt +\\ &\quad S_t\left(\partial_t v + \mathcal{A}_\eta v\right)(t,\eta_t) - \sigma^2 S_t\eta\partial_\eta v(t,\eta_t)\Big)dt\\ &= S_t\Big(\partial_t v - \left(\frac{1}{T} + \kappa\eta\right)\partial_\eta v + \frac{1}{2}\sigma^2\eta^2\partial^2_{\eta^2}v - qv\Big)dt.\end{aligned}$$

In conclusion, it comes via Lemma IX.6:

Proposition 2 *We have $\Pi_t = S_t v(t,\eta_t)$, where the pricing function $v = v(t,\eta)$ in the numéraire S solves the following one-dimensional PDE:*

$$\begin{cases} v(T,\eta) = \eta^+\\ \partial_t v - \left(\frac{1}{T} + \kappa\eta\right)\partial_\eta v + \frac{1}{2}\sigma^2\eta^2\partial^2_{\eta^2}v - qv = 0.\end{cases} \tag{51}$$

This equation can be solved numerically by finite differences. For η close to 0 the advection term is dominant in this equation (due to the first-order coefficient $\frac{1}{T}$); this poses specific numerical issues which are dealt with in Dubois and Lelievre (2005).

C.2 Hawaiian Fixed Strike Put option

This is the American counterpart of the previous option, with payoff $\phi(\vartheta, I_\vartheta)$ paid at a stopping time ϑ at the holder's convenience, where

$$\phi(t,I) = \left(K - \frac{I}{t}\right)^+. \tag{52}$$

By the nonlinear Feynman-Kac formula for obstacle problems, the related price process is written, for $t \in [0,T]$, as

$$\Pi_t = \beta_t^{-1}\max_{\vartheta\in\Theta_t}\mathbb{E}_t\left[\beta_\vartheta\phi(\vartheta, I_\vartheta)\right] = v(t,S_t,I_t), \tag{53}$$

[30] see I.§1.G.2.

where the pricing function $v = v(t, S, I)$ satisfies (in the continuous viscosity sense[31])

$$\left\{ \begin{array}{l} v(T,S,I) \;=\; \phi(T,I),\ S,I>0 \\ \max\big(\partial_t v + \mathcal{A}_{S,I} v - rv)(t,S,I)\ , \\ \qquad\qquad \phi(t,I) - v(t,S,I)\big) = 0,\ t\in(0,T),\ S,I>0, \end{array} \right. \tag{54}$$

in which $\mathcal{A}_{S,I}$ was defined in (45). The obstacle $\phi(t, I)$ is singular at time $t = 0$, so that the second line of (54) only holds for $t > 0$. The proof of the following result can be found in Crépey (2001).

Proposition 3 *We have $\Pi_t = v(t, S_t, I_t)$, $t \in (0, T]$, where v is the unique bounded viscosity solution to (54). The price of the option at time 0 is given by the radial limit*

$$\Pi_0 = \lim_{t\to 0+} v(t, S_0, S_0 t). \tag{55}$$

Figure 2 provides a numerical illustration of the radial convergence in (55). Numerically, the convergence also occurs along radii $t\alpha$ other than tS_0, yet at a slower rate than for $\alpha = S_0$. As in the European case, special numerical care is required to deal with the degeneracy of $\mathcal{A}_{S,I}$ in I (Zvan et al., 1998).

D Discretely Path Dependent Options

Discretely path-dependent options correspond to payoffs of the form $\xi = \phi(S_{t_0}, S_{t_1}, \ldots, S_{t_n})$ for a discrete set of monitoring times $(t_0 = 0, \ldots, t_n = T)$. This includes important classes of assets, such as discretely sampled Asian options (Wilmott, 1998; Wilmott et al., 1993; Demeterfi et al., 1999; Windcliff et al., 2006b), cliquet options, and volatility and variance swaps. The pricing of such products by Monte Carlo is standard in the Black-Scholes model, but it becomes tricky in a nonlinear extension of Black-Scholes known as the uncertain volatility model (UVM) of Avellaneda, Levy, and Paras (1995), where the volatility process is only known to remain in a range $[\underline{\sigma}, \bar{\sigma}]$. The PDE approach of this part, on the contrary, can easily be adapted to this setup (Wilmott, 2002; Windcliff et al., 2006a).

D.1 Cliquet Options

Let $R_i = \frac{S_{t_i} - S_{t_{i-1}}}{S_{t_{i-1}}}$ denote the return of the underlying asset on the period $[t_{i-1}, t_i]$, for $i = 1, \ldots, n$. The payoff of a cliquet option is defined by

$$\xi = \max\left(\min\left(\textstyle\sum_{i=1}^n \max(\min(R_i, C), F), \mathcal{C}\right), \mathcal{F}\right)$$

for given thresholds $F < C$ and $\mathcal{F} < \mathcal{C}$. We introduce two auxiliary processes P and Z such that, on every time interval $[t_i, t_{i+1})$,

$$P_t = S_{t_i}\ ,\ \ Z_t = \tfrac{1}{i}\textstyle\sum_{l=1}^i \max(F, \min(C, R_l)). \tag{56}$$

In the Black-Scholes model for S, the triple (S_t, P_t, Z_t) is Markov with generator[32]

$$\mathcal{A}_{S,P,Z}\,\varphi(S,P,Z) = \mathcal{A}_S^{bs}\varphi(S,P,Z) = \frac{1}{2}\sigma^2 S^2 \partial^2_{S^2}\varphi + \kappa S \partial_S \varphi.$$

Moreover, we have that

$$\xi = \max(\min(nZ_T, \mathcal{C}), \mathcal{F}) = \phi(Z_T).$$

One can then show that:

[31] see 0.§1.B.1.
[32] outside the t_is.

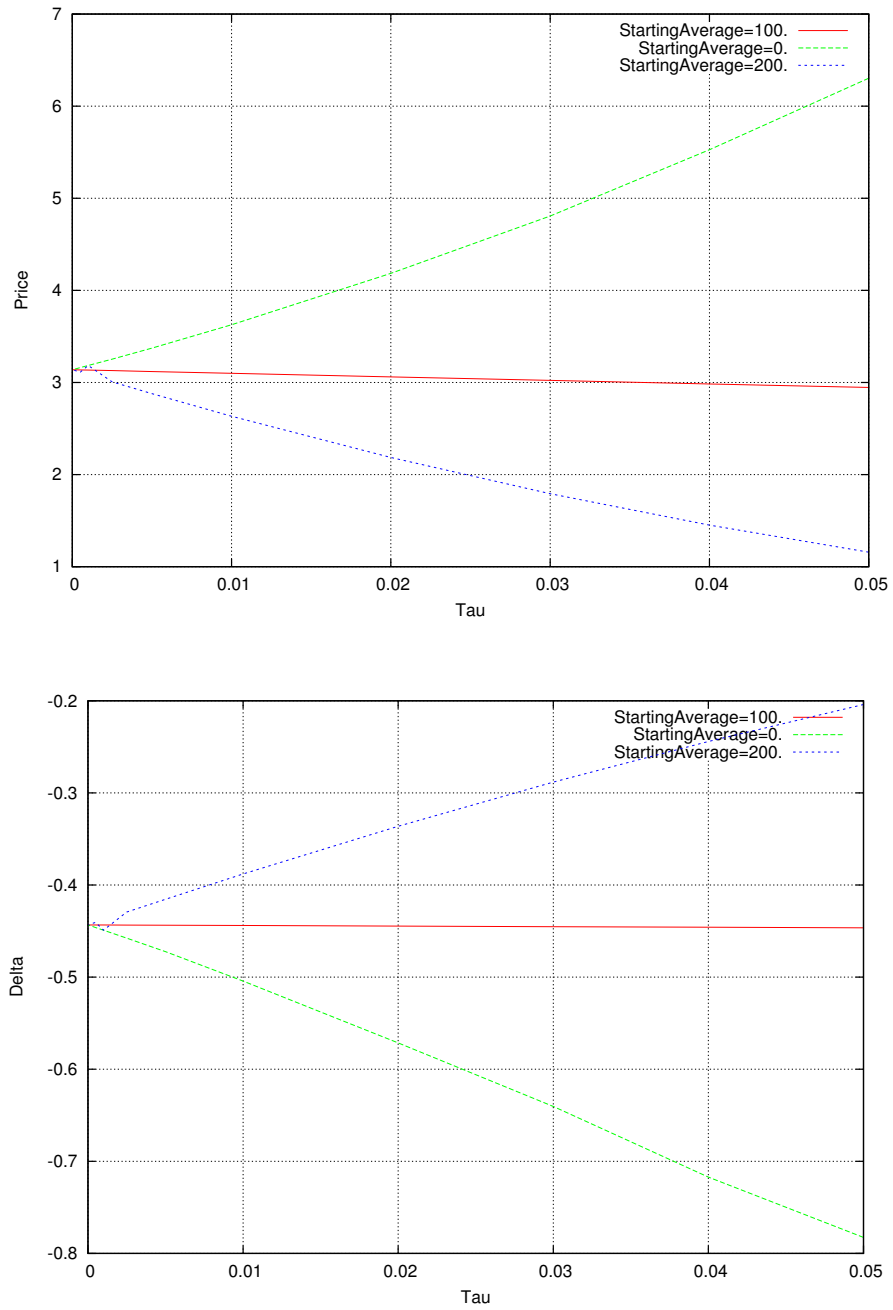


Figure 2: *Convergence of the price and delta of a Hawaiian put at* $(t, S_0, \alpha t)$ *as* $t \to 0$, *for* $\alpha = 0$, S_0 *and* $2S_0$.

Proposition 4 *The cliquet option Black-Scholes price process* $\Pi_t = e^{-r(T-t)}\mathbb{E}_t\xi$ *can be represented as* $\Pi_t = u_i(t, S_t, P_t, Z_t)$ *on every time interval* $[t_i, t_{i+1})$*, for a sequence of continuous functions* $u_i = u_i(t, S, P, Z)$ *on* $[t_i, t_{i+1}] \times (0, \infty)^3$ *such that* $(u_i)_{0\le i\le n}$ *is the unique sequence of viscosity solutions to the following PDE cascade:*

$$u_n = \phi \text{ on } \{T\} \times (0,\infty)^3 \text{ and for } i \text{ decreasing from } n-1 \text{ to } 0:$$
$$\begin{cases} u_i(t_{i+1}, S, P, Z) = u_{i+1}(t_{i+1}, S, P_+, Z_+) \text{ on } (0,\infty)^3 \\ \partial_t u_i + \frac{1}{2}\sigma^2 S^2 \partial^2_{S^2} u_i + \kappa S \partial_S u_i - r u_i = 0 \text{ on } [t_i, t_{i+1}) \times (0,+\infty)^3, \end{cases} \tag{57}$$

where in the jump condition in the first line,

$$P_+ = S\,,\ Z_+ = \tfrac{i}{(i+1)} Z + \tfrac{\rho}{(i+1)}, \tag{58}$$

in which $\rho = \max(\min(\frac{S-P}{P}, C), F)$.

The jump condition stems from the continuity[33] of the Brownian martingale $(e^{-rt}\Pi_t)$ so that, at every $t = t_{i+1}$,

$$u_i(t, S_t, P_{t_i}, Z_{t_i}) = \lim_{s\uparrow t} u_i(s, S_s, P_s, Z_s) = \lim_{s\uparrow t} \Pi_s$$
$$= \Pi_t = u_{i+1}(t, S_t, P_t, Z_t) = u_{i+1}(t, S_t, S_t, \frac{i}{(i+1)} Z_{t_i} + \frac{1}{(i+1)} \max(\min(\frac{S_t - P_{t_i}}{P_{t_i}}, C), F)).$$

To solve (57) numerically, we localize the domain in space to

$$\mathcal{O} = (\underline{S}, \overline{S}) \times (\underline{S}, \overline{S}) \times (\underline{Z}, \overline{Z}),$$

with

$$\underline{S} = 0,\ \overline{S} = S_0(1 + f\sigma\sqrt{T}),\ \underline{Z} = F,\ \overline{Z} = C,$$

for a suitable factor f, e.g. $f = 4$. We then discretize the state space. Given an adaptive P-mesh $(P^{j_1})^{1\le j_1\le m_1}$ concentrated around S_0, one can use the following (P, S)-grid, concentrated around S_0 and around the diagonal $S = P$ (Windcliff et al., 2006a):

$$(P, S)^{j_1, j_2} = (P^{j_1}, \frac{P^{j_1} P^{j_2}}{S_0})\,,\ 1 \le j_1 \le m_1,\ 1 \le j_2 \le m_2.$$

We then consider the "product" of this (P, S)-grid with a uniform grid in the Z-variable. Between monitoring times, the equation (57) can then be solved by a finite difference θ-scheme on the grid (standard Black–Scholes θ-scheme in the S-variable operating in the three dimensional state space (S, P, Z)). Some kind of interpolation is required for implementing the jump condition (first line of (57)).

D.2 Volatility and Variance Swaps

We use the same approach as for cliquet options, except that R_i and Z are now defined by $R_i = \ln(\frac{S_{t_i}}{S_{t_{i-1}}})$ and, for every $t_i \le t < t_{i+1}$,

$$Z_t = \tfrac{1}{i} \textstyle\sum_{l=1}^{i} R_l^2. \tag{59}$$

Variance and volatility swaps correspond to the payoffs $\xi = V^2 - K^2$ and $V - K$, in which $V^2 = \sum_{i=1}^{n} \ln(\frac{S_{t_i}}{S_{t_{i-1}}})^2$ is the realized variance of S and K is a constant. In both cases the payoff ξ is of the form $\phi(Z_T)$.

We proceed as for cliquet options, obtaining in this case (in Black-Scholes): $\Pi_t = u_i(t, S_t, P_t, Z_t)$ on every time interval $[t_i, t_{i+1})$, where the sequence of continuous functions $u_i = u_i(t, S, P, Z)$ on $[t_i, t_{i+1}] \times (0, +\infty)^2$ satisfies (57)-(58), but for ϕ and Z suitably redefined for volatility and variance swaps, and where ρ in the jump condition (58) is now meant as $\ln(\frac{S}{P})^2$.

[33] which is apparent from the Brownian martingale representation property.

D.3 Discretely Monitored Asian Options

Finally, we consider discretely sampled Asian options with payoffs of the form

$$\xi = \left(K - \frac{T}{n} \sum_{l=1}^{n} S_{t_l} \right)^+ .$$

Let

$$Y_t = \tfrac{1}{i} \textstyle\sum_{k=1}^{i} S_{t_i}$$

on $t_i \leq t < t_{i+1}$. The pair (S_t, Y_t) is Markov in the Black-Scholes model for S, with generator[34]

$$\mathcal{A}_{S,Y}\, \varphi(S,Y) = \mathcal{A}_S^{bs} \varphi(S,Y),$$

on every time interval (t_i, t_{i+1}). Moreover, we have that $\xi = (K - TY_T)^+ = \phi(Y_T)$. We proceed as before, obtaining in this case (in Black–Scholes):

> $\Pi_t = u_i(t, S_t, Y_t)$ on every time interval $[t_i, t_{i+1})$, for a sequence of continuous functions u_i on $[t_i, t_{i+1}] \times (0, +\infty)^2$ such that $u_n = \phi$ and, for i decreasing from $n-1$ to 0:
>
> $$\begin{cases} u_i(t_{i+1}, S, Y) = u_{i+1}(t_{i+1}, S, Y_+) \text{ on } (0,+\infty)^2 \\ \partial_t u_i + \dfrac{1}{2}\sigma^2 S^2 \partial^2_{S^2} u_i + \kappa S \partial_S u_i - r u_i = 0 \text{ on } [t_i, t_{i+1}) \times (0,+\infty)^2, \end{cases}$$
>
> with
>
> $$Y_+ = \tfrac{i}{(i+1)} Y + \tfrac{S}{(i+1)}$$
>
> in the first line.

[34] outside the t_is.

Chapter IV

Pricing and Greeking by Monte Carlo

This chapter is about Monte Carlo pricing methods. The term "Monte Carlo" for computational methods involving simulated random numbers was introduced in Metropolis and Ulam (1949). Like deterministic pricing schemes, simulation pricing schemes can be used in any Markovian (or, of course, static one-period) setup. In the case of European claims, simulation pricing schemes reduce to the well known Monte Carlo loops. For products with early exercise features, or for more general control problems, numerical schemes by simulation are available too, yet these are more sophisticated; see e.g. Sections §9 and VI.§3.

Monte Carlo methods are attractive by their genericity:

- genericity of their theoretical properties (such as the confidence interval they provide for the solution) that are insensitive to the dimension of a problem or to the irregularity of a payoff function – at least for genuine pseudo Monte Carlo methods, as opposed to the quasi Monte Carlo methods that we will also consider in this chapter,
- genericity of implementation,
- ease of parallelization.

But (pseudo) Monte Carlo methods are slow, only converging at the rate $\sigma m^{-\frac{1}{2}}$ where m is the number of simulation runs and σ is the standard deviation of the sampled payoff. To accelerate the convergence, various variance reduction techniques can be used to transform a given payoff into another one with less variance. An alternative to variance reduction is quasi Monte Carlo, which converges faster in practice than pseudo Monte Carlo (at least in low dimension). However quasi Monte Carlo estimates do not come with confidence intervals, and their performance can be strongly altered in high dimension.

§1 Principles of Monte Carlo Simulation

We want to estimate $\Theta = \mathbb{E}[\phi(X)]$, where ϕ is some function on $\mathcal{D} \subseteq \mathbb{R}^d$ and X is a $\mathcal{D}$-valued random variable. Note that Θ can be expressed in integral form as

$$\Theta = \int_{\mathcal{D}} \phi(x) d\mathbb{Q}^X(x).$$

Monte Carlo simulation is a general method for evaluating an integral as an expected value, based on the strong law of large numbers and the central limit theorem. It provides an unbiased estimate, and the error on the estimate is controlled within a confidence interval.

A Law of Large Numbers and Central Limit Theorem

For x_j i.i.d. to X with $\mathbb{E}|\phi(X)| < +\infty$, we have by the strong law of large numbers

$$\frac{1}{m}\sum_{j=1}^{m} \phi(x_j) \xrightarrow{a.s.} \Theta \text{ as } m \to \infty.$$

If, moreover, $\sigma^2 = \mathbb{V}\mathrm{ar}[\phi(X)] < +\infty$, then, by the central limit theorem, the normalized error converges in law to the Gaussian distribution:

$$\frac{\sqrt{m}}{\sigma}\left(\frac{1}{m}\sum_{j=1}^{m}\phi(x_j) - \Theta\right) \xrightarrow{\mathcal{L}} \mathcal{N}(0,1) \text{ as } m \to \infty.$$

B Standard Monte Carlo Estimator and Confidence Interval

An unbiased estimate with m trials for Θ is thus given by

$$\Theta_m = \frac{1}{m}\sum_{j=1}^{m}\phi(x_j). \tag{1}$$

The variance of this estimate is $\frac{\sigma^2}{m}$, independent of the dimension d and with unbiased estimate (squared standard error)

$$\sigma_m^2 = \frac{1}{m-1}\left[\frac{1}{m}\sum_{j=1}^{m}\phi^2(x_j) - \Theta_m^2\right]. \tag{2}$$

The speed of convergence of Θ_m to Θ is $\frac{\sigma}{\sqrt{m}}$, which is estimated by σ_m. A confidence interval I at the threshold (confidence level) $1-2\alpha$ for Θ, i.e. so that $\mathbb{Q}(\Theta \in I) = 1-2\alpha$, is given by

$$I = \left[\Theta_m - z_\alpha\sigma_m, \Theta_m + z_\alpha\sigma_m\right],$$

with $z_\alpha = \mathcal{N}^{-1}(1-\alpha)$. For instance, if the threshold is set at 95%, then $\alpha = 2.5\%$ and $z_\alpha \approx 1.96$. A natural stopping criterion consists in exiting a Monte Carlo loop when $z_\alpha\sigma_m$ becomes less than some fraction ϵ (e.g. $\epsilon = 10\text{bp} = 10^{-3}$) of Θ_m, so that one knows Θ up to a relative error equal to ϵ, at the level of confidence $1-\alpha$.

We briefly summarize some advantages and disadvantages of the Monte Carlo method.

- ***Advantages:***
  - We can implement this method very easily if we are able to simulate the random variable X.
  - The estimate is unbiased and we can build a confidence interval justified by the central limit theorem.
  - The properties of the method are not altered by irregularity of the function ϕ or dimensionality of the factor X.
- ***Disadvantage:*** Convergence is slow and therefore computing time can be very large.

§2 Simulating Uniform Numbers

To sample "random" vectors in $\mathbb{R}^d$ with specific distributions, one first draws "uniform random" vectors u_j over $[0,1]^d$. One then transforms the u_j into x_j with the desired distribution. "Uniform random vectors" u_j over $[0,1]^d$ may be obtained either:

- by sampling "i.i.d. random variables" over $[0,1]$ with a pseudo-random generator, and arranging them into sequences of length d,
- or by using a quasi-random generator (low-discrepancy sequence) in dimension d.

The use of quotation marks above indicates that simulated random numbers only "look" random and uniform (and independent, in the case of pseudo-random numbers). However, whenever a simulated sequences is generated deterministically on a computer, one can always find a statistical test for uniformity or independence that a simulated sequence will fail to pass. And, of course, the quality of a generator puts an upper bound on the quality of any simulation pricing scheme using it.

By a d-variate uniform, respectively Gaussian, random draw we henceforth mean a uniform point over $[0,1]^d$, respectively a d-variate centered and normalized Gaussian vector, with $d = 1$ by default.

A Pseudo-Random Generators

Pseudo-random generators are used to simulate independent uniform variables over $[0,1]$. L'Ecuyer L'Ecuyer (1998, 1994) formally defines a pseudo-random number generator as a quintuple $\mathcal{G} = (s, S, T, U, G)$ where S is a finite set of states, $s \in S$ is the initial state, the mapping $T : S \to S$ is the transition function, and $G : S \to U$ is the output function from S to a finite set U of outputs. Since S is finite, the sequence of states is ultimately periodic. The period is the smallest positive integer m such that $s_{j+m} = s_j$ for all sufficiently large j. The following properties are required for a good pseudo-random number generator:

1. *Large period length*: At least 2^{60}, say.
2. *Good equidistribution properties and statistical independence of successive pseudo-random draws*: The generator should pass statistical tests of uniformity and independence – general tests such as chi-square or Kolmogorov-Smirnov tests, and more specific tests such as equidistribution tests, serial tests, gap tests, partition tests.
3. *Little intrinsic structure*: Successive values produced by some generators produce undesirable lattice structures.
4. *Efficiency, fast generation, not requiring too much memory space*: Especially if we use many generators together or in parallel.
5. *Repeatability (fixing a given seed)*: Very useful for practical applications. Otherwise one can use the time of execution of the program given by the computer clock to initialize the generator.
6. *Portability*: The generator should produce the same sequence on different computers or with different compilers.
7. *Unpredictability*: one should not be able to predict the next generated value from the previous ones[1].

Note that, from the point of view of period length, built-in library generators may behave poorly. The function `rand()` in the C++ standard library thus returns a pseudo-random integer in the range 0 to `RAND_MAX`, where the value of the constant `RAND_MAX` may vary between implementations, but can only be assumed to be at least 32767.

A.1 Pseudo-Random Uniform Numbers

Linear schemes are the simplest method for constructing pseudo-random numbers. They use a linear recurrence relation to compute the next value from the previous one. The j^{th} random number is given by

$$u_j = \frac{U_j}{c} \in [0,1], \text{ where } U_j = (aU_{j-1} + b) \bmod c,$$

[1] However, this is less important in finance than for other applications such as cryptography.

for well chosen fixed integers $a > 0$, b and $c > 0$, starting from a given seed U_0. For instance, in the Fortran IMSL Library,

$$a = 397204094\,,\ b = 0 \text{ and } c = 2^{31} - 1.$$

Such generators are very simple but prone to produce a lot of regularity in sequences along with a lattice structure. Yet it is possible to combine any of these with another, using for instance the shuffling procedure of Bayes and Durham, obtaining thereby a longer period generator with better properties.

A.2 Rejection-Acceptance Method

The rejection-acceptance method allows us to draw pseudo-uniform points in an arbitrary subset of $\mathbb{R}^d$, starting from pseudo-uniform points in a larger set, based on the following

Proposition 1 *Suppose the u_j are i.i.d. uniform points over a Lebesgue set D of $\mathbb{R}^d$. Let there be given a subset Δ of D with $\frac{\lambda(\Delta)}{\lambda(D)} = \alpha \in (0,1)$, where λ represents Lebesgue measure over $\mathbb{R}^d$. Let $0' = 0$ and, for $j \geq 1$,*

$$j' = \inf\{l > (j-1)'; u_l \in \Delta\}\,.$$

Then the $v_j := u_{j'}$ are i.i.d. uniformly distributed over Δ. For every $j \geq 1$, the average acceptance time $\mathbb{E}\,(j' - (j-1)')$ equals α^{-1}.

Proof. We have $\mathbb{Q}\,(1' = l) = (1-\alpha)^{l-1}\,\alpha$, so that

$$\mathbb{E}1' = \sum\nolimits_{l\geq 1} l\mathbb{Q}\,(1' = l) = \sum\nolimits_{l\geq 1} l\,(1-\alpha)^{l-1}\,\alpha = \alpha^{-1}.$$

For any sequence of subsets Δ_j of Δ, we show by induction that, for every $m \geq 0$,

$$\mathbb{Q}\,(v_j \in \Delta_j, j = 1\ldots m) = \prod_{j=1}^{m} \frac{\lambda\,(\Delta_j)}{\lambda\,(\Delta)}.$$

For $m = 0$ this is trivially satisfied. Moreover, for $m \geq 1$ we have that

$$\begin{aligned}
&\mathbb{Q}\,(v_j \in \Delta_j, j = 1\ldots m)\\
&= \textstyle\sum_{l\geq m-1, j\geq 1} \mathbb{Q}\big(v_1 \in \Delta_1, \ldots, v_{m-1} \in \Delta_{m-1},\\
&\qquad\qquad (m-1)' = l, m' = l + j, u_{l+j} \in \Delta_m\big)\\
&= \textstyle\sum_{l\geq m-1} \mathbb{Q}\big(v_1 \in \Delta_1, \ldots, v_{m-1} \in \Delta_{m-1}, (m-1)' = l\big)\times\\
&\qquad\qquad \textstyle\sum_{j\geq 1} (1-\alpha)^{j-1}\,\frac{\lambda(\Delta_m)}{\lambda(D)}\\
&= \big(\textstyle\sum_{l\geq m-1} \mathbb{Q}\,(v_j \in \Delta_j, \ldots, v_{m-1} \in \Delta_{m-1}, (m-1)' = l)\big)\,\frac{\lambda(\Delta_m)}{\lambda(\Delta)}\\
&= \mathbb{Q}\,(v_j \in \Delta_j, j = 1\ldots m-1)\,\tfrac{\lambda(\Delta_m)}{\lambda(\Delta)}.\ \blacksquare
\end{aligned}$$

B Low-Discrepancy Sequences

Quasi-random numbers, or successive points u_j of low-discrepancy sequences (Niederreiter, 1987, 1992; Morokoff and Caflish, 1994), are not independent. But they have good equidistribution properties on $[0,1]^d$, implying good convergence properties of $\frac{1}{m}\sum_{j=1}^{m} \psi(u_j)$ to $\int_{[0,1]^d} \psi(u)du$ as $m \to \infty$.

The discrepancy can be viewed as a quantitative measure for the deviation from the uniform distribution. Let $[0,x]$ denote $\{y \in [0,1]^d,\ y \leq x\}$, where $y \leq x$ is defined componentwise. Given a $[0,1]^d$-valued sequence $u = (u_j)_{j\geq 1}$ and $x \in [0,1]^d$, let

$$D_m(u,x) = \tfrac{1}{m}\textstyle\sum_{j=1}^{m} \mathbb{1}_{[0,x]}(u_j) - \prod_{l=1}^{d} x_l.$$

Definition 1 **(i)** *A sequence u is said to be equidistributed on $[0,1]^d$ if $\lim_{m\to\infty} D_m(u,x) = 0$, for every $x \in [0,1]^d$.*
(ii) *The value $D_m(u)$ defined by*

$$D_m(u) = \sup_{x\in[0,1]^d} |D_m(u,x)|$$

is called the discrepancy of u at rank m.
(iii) *The sequence u is said to have low-discrepancy, or to be quasi-random, if $D_m(u) = \mathrm{O}(\frac{(\ln m)^d}{m})$.*

Using the law of the iterated logarithm, one can show that a pseudo-random sequence has a discrepancy $\mathrm{O}((\frac{\ln\ln m}{m})^{\frac{1}{2}})$ and is therefore not quasi-random. With respect to a lattice, quasi-random numbers have the advantage that points can be added incrementally. Low discrepancy sequences perform very well in low dimension. But, for large dimension d, the theoretical bound $\frac{(\ln m)^d}{m}$ may be meaningful only for extremely large values of m. This is essentially because, in dimension d, a lattice can only be refined by increasing the number of points by a factor 2^d. The Sobol (1976) sequence is one of the most popular quasi-random sequences in financial applications (see Figure 1). Its construction is based on number theory and can be implemented very efficiently using bitwise XOR ("exclusive or") operations.

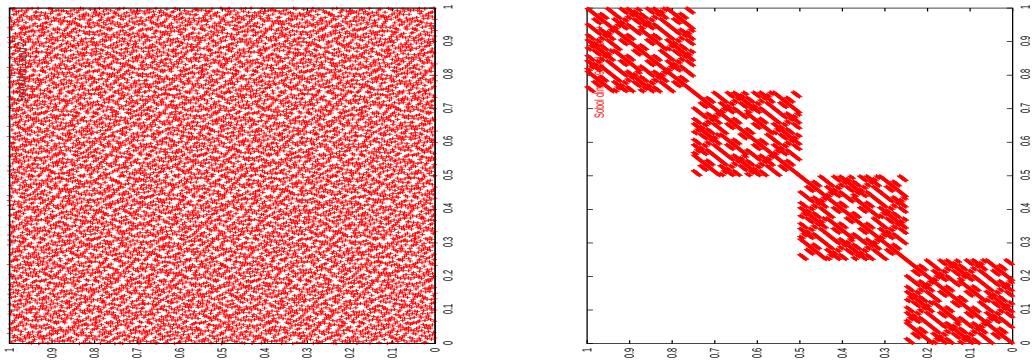


Figure 1: *Projection on the first two, respectively last two, coordinates of the first 10^4 points of the Sobol sequence in dimension $d = 160$. If a $d(\geq 2)$-dimensional sequence is uniformly distributed in $[0,1]^d$, then a two-dimensional sequence, formed by pairing two coordinates, should be uniformly distributed in the unit square. The appearance of nonuniformity in such projections is an indication of potential problems in using a high-dimensional quasi-random sequence (Morokoff and Caflish, 1994).*

§3 Simulating Non-Uniform Numbers

A Inverse Method

The inverse simulation method is based on the following elementary but fundamental:

Lemma 1 *For every real random variable X with cumulative distribution function F_X, for uniform U we have,*

$$F_X^{-1}(U) \overset{\mathcal{L}}{=} X\,,\ F_X(X) \overset{\mathcal{L}}{=} U, \tag{3}$$

where F_X^{-1} denotes the generalized inverse of the nondecreasing function F_X.

Proof. Letting $Y = F_X^{-1}(U)$ we have, for every x,

$$\mathbb{Q}\left(Y \leq x\right) = \mathbb{Q}\left(U \leq F_X(x)\right) = F_X(x),$$

as U is uniform and $F_X(x) \in [0,1]$. Thus X and Y have the same cumulative distribution function and so they have the same law. ∎

As an application of this lemma, we obtain an exponential random variable with parameter λ through $E = -\lambda^{-1}\ln(1-U)$ or, equally in law, $(-\lambda^{-1}\ln U)$, with U uniform.

A $\mathcal{P}_\lambda$-random variable P is such that $\mathbb{Q}(P=n) = e^{-\lambda}\frac{\lambda^n}{n!}$. To simulate P, we can draw a uniform number U over $[0,1]$ and set $P=\nu$ such that

$$\sum_{n=0}^{\nu} e^{-\lambda}\frac{\lambda^n}{n!} \le U < \sum_{n=0}^{\nu+1} e^{-\lambda}\frac{\lambda^n}{n!}.$$

Also, it is well known that the number of clients at time T in a queue with i.i.d. exponential inter-arrival times of parameter λ, is $\mathcal{P}_{\lambda T}$-distributed. So, to simulate P, another possibility is to draw independently $\epsilon_j = -\frac{1}{\lambda}\ln u_j$ until $\sum_{l=1}^{j}\epsilon_l > T = 1$ and to set, with the convention $\max\emptyset = 0$,

$$P = \max\{j\ge 1; \sum_{l=1}^{j}\epsilon_l < 1\} = \max\{j\ge 1; \prod_{l=1}^{j} u_l > e^{-\lambda}\}.$$

The inverse Gaussian cumulative distribution function $\mathcal{N}^{-1}$ is not known explicitly. To simulate Gaussian variables by inversion, one can use Moro's numerical approximation for $\mathcal{N}^{-1}$. However, this entails a simulation bias.

B Gaussian Pairs

Next come two exact methods for generating a standard Gaussian pair starting from uniform numbers as input data. Note that to use these methods with quasi-random numbers, we must generate the coordinates of the underlying uniform points "independently", e.g. from two different one-dimensional quasi-random sequences, or as the two coordinates of a point from a two-dimensional quasi-random sequence, but not as two successive values of a one-dimensional quasi-random sequence.

B.1 Box-Müller Method

An exact method for simulating a Gaussian pair is the Box-Müller transformation which is based on the following:

Lemma 2 *If (U,V) is bivariate uniform, then (X,Y) defined by*

$$X = \sqrt{-2\ln U}\cos(2\pi V)$$
$$Y = \sqrt{-2\ln U}\sin(2\pi V)$$

is bivariate Gaussian.

Proof. We have that $(-2\ln(U))$ and $(2\pi V)$ are $\mathcal{E}_{\frac{1}{2}}$- and $\mathcal{U}_{[0,2\pi]}$-distributed. So, for every test-function φ,

$$\begin{aligned}
\mathbb{E}\varphi(X,Y) &= \int_0^\infty\int_0^{2\pi}\varphi(\rho\cos\vartheta,\rho\sin\vartheta)\frac{1}{2}e^{\frac{-\rho^2}{2}}d(\rho^2)\frac{d\vartheta}{2\pi}\\
&= \int_0^\infty\int_0^{2\pi}\varphi(\rho\cos\vartheta,\rho\sin\vartheta)\rho d\rho e^{\frac{-\rho^2}{2}}\frac{d\vartheta}{2\pi}\\
&= \int_{\mathbb{R}}\int_{\mathbb{R}}\varphi(x,y)e^{-\frac{x^2+y^2}{2}}\frac{dxdy}{2\pi}.
\end{aligned}$$ ∎

B.2 Marsaglia Method

Lemma 3 (Change of variables formula for densities) *Let real random variables X and Y be such that $Y = f \circ X$, where f denotes a diffeomorphism between two subsets of $\mathbb{R}^d$. Let $g = f^{-1}$ and Jg denote the Jacobian matrix $(\partial_{y_j} g_i)_{1\leq i\leq d}^{1\leq j\leq d}$. Assuming X has a density p_X, then so does Y, and the density p_Y of Y at $y = f(x)$ is given by*

$$p_Y(y) = |\det(Jg)(y)|\, p_X(x) \tag{4}$$

(and $p_Y(y) = 0$ for y outside the image set of f).

Proof. This is seen by change of variables $x = g(y)$ in $\mathbb{E}\varphi(Y) = \int \varphi[f(x)]p_X(x)dx$, where φ denotes an arbitrary test-function on $\mathbb{R}^d$. ∎

Example 1 *Applying the formula (4) to $Y = S_t$ in Black-Scholes and $X = \sigma W_t = g(S_t)$, with $g(S) = \ln(\frac{S}{S_0}) - bt$ where $b = \kappa - \frac{\sigma^2}{2}$, we recover the lognormal density*

$$p_{S_t}(S) = \tfrac{1}{\sigma S\sqrt{2\pi t}} e^{-\frac{g(S)^2}{2\sigma^2 t}} = \tfrac{1}{S} p_{(\sigma W_t)}(g(S)).$$

In the Marsaglia method of simulation of a Gaussian pair that is provided by the next lemma, the uniform point (U, V) on the unit disk can be simulated by rejection-acceptance, using uniform points on the square $[0,1]^2$ as input data (with an acceptance rate of $\frac{\pi}{4}$).

Lemma 4 *If (U, V) is uniform on the unit disk D, then (X, Y) defined by*

$$X = \sqrt{\tfrac{-2\ln(\rho^2)}{\rho^2}}U$$
$$Y = \sqrt{\tfrac{-2\ln(\rho^2)}{\rho^2}}V,$$

where $\rho^2 = U^2 + V^2$, is bivariate Gaussian.

Proof. Let (ρ, θ) denote the polar coordinates of (U, V). Using the transformation

$$D \ni (U, V) \overset{f}{\longmapsto} (\rho^2, \frac{\theta}{2\pi}) \in [0,1]^2,$$

the formula (4) yields that the density of $(\rho^2, \frac{\theta}{2\pi})$ at a point (α, β) of $[0,1]^2$ is given by the density $\frac{1}{\pi}$ of (U, V) at the inverse image

$$f^{-1}(\alpha, \beta) = \big(\sqrt{\alpha}\cos(2\pi\beta), \sqrt{\alpha}\sin(2\pi\beta)\big)$$

of (α, β) under f, multiplied by the determinant of the matrix

$$Jf^{-1}(\alpha, \beta) = \begin{pmatrix} \frac{1}{2\sqrt{\alpha}}\cos(2\pi\beta) & -2\pi\sqrt{\alpha}\sin(2\pi\beta) \\ \frac{1}{2\sqrt{\alpha}}\sin(2\pi\beta) & 2\pi\sqrt{\alpha}\cos(2\pi\beta) \end{pmatrix}.$$

The determinant is π, so that the density of $(\rho^2, \frac{\theta}{2\pi})$ is equal to $\frac{\pi}{\pi} = 1$ uniformly over the square $[0,1]^2$, meaning that $(\rho^2, \frac{\theta}{2\pi}) \sim \mathcal{U}_{[0,1]^2}$. We deduce by Box-Müller that

$$\sqrt{-2\ln(\rho^2)}\begin{pmatrix} \cos\left(2\pi\frac{\theta}{2\pi}\right) \\ \sin\left(2\pi\frac{\theta}{2\pi}\right) \end{pmatrix} = \sqrt{\frac{-2\ln(\rho^2)}{\rho^2}}\begin{pmatrix} \rho\cos(\theta) \\ \rho\sin(\theta) \end{pmatrix} = \sqrt{\frac{-2\ln(\rho^2)}{\rho^2}}\begin{pmatrix} U \\ V \end{pmatrix}$$

is bivariate Gaussian. ∎

C Gaussian Vectors

A d-variate Gaussian vector X with zero mean and covariance matrix Γ can be simulated as follows:

- compute a square root of Γ, namely a matrix Σ such that $\Gamma = \Sigma\Sigma^\top$;
- generate a d-variate Gaussian vector ε.

Then $X = \Sigma\varepsilon$ is $\mathcal{N}(0,\Gamma)$-distributed. A square root $\Sigma = \Sigma^{cho}$ of Γ may be computed as the lower triangular matrix obtained by Cholesky decomposition of Γ. So, for $p = 1,\dots,d$:

$$\begin{array}{l} \Sigma_{p,p} = \sqrt{\Gamma_{p,p} - \sum_{r=1}^{p-1} \Sigma_{p,r}^2} \\ \Sigma_{q,p} = \frac{\Gamma_{p,q} - \sum_{r=1}^{p-1} \Sigma_{p,r}\Sigma_{q,r}}{\Sigma_{p,p}} \qquad \text{for} \quad q = p+1,\dots,d. \end{array} \tag{5}$$

alternatively, we can perform a spectral decomposition of Γ (as in a Principal Component Analysis), setting $\Sigma = \Sigma^{pca} = P\Lambda^{\frac{1}{2}}$, where $P\Lambda P^\top = \Gamma$ with P orthonormal and Λ diagonal. A spectral decomposition is unique up to a reordering of the PCA factors[2] along with the eigenvalues of Γ.

Example 2 *In the two-dimensional case with*

$$\Gamma = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix},$$

the Cholesky decomposition of Γ *yields*

$$\Sigma^{cho} = \begin{pmatrix} \sigma_1 & 0 \\ \rho\sigma_2 & \sqrt{1-\rho^2}\sigma_2 \end{pmatrix},$$

whereas the spectral decomposition of Γ *reads*

$$\Sigma^{pca} = \begin{pmatrix} \sigma_1\sqrt{\frac{1+\rho}{2}} & \sigma_1\sqrt{\frac{1-\rho}{2}} \\ \sigma_2\sqrt{\frac{1+\rho}{2}} & -\sigma_2\sqrt{\frac{1-\rho}{2}} \end{pmatrix}.$$

§4 Monte Carlo Acceleration Techniques

The main disadvantage of the standard Monte Carlo simulation is its convergence rate in $\sigma m^{-\frac{1}{2}}$. So, to improve the accuracy by a factor of 10, we must increase the number m of simulations by a factor of 100. Alternatively, and this is the strategy of variance reduction, we may try to rewrite Θ in terms of a new random variable with less variance than $\phi(X)$. Or, if we are ready to give up independence and the associated confidence interval, we may try to improve the convergence rate itself, which is the strategy of Quasi Monte Carlo.

A Antithetic Variables

The idea of the antithetic variables method is to introduce some "good" correlation between the terms of the estimate. To fix the ideas, we consider simulation by the inverse method, based on uniform numbers u_j on $[0,1]$ (assuming $d = 1$). In the antithetic variables method, we use each u_j twice, via $x_j = F_X^{-1}(u_j)$ and $\bar{x}_j = F_X^{-1}(1-u_j)$. These two variables have the same law, but they are not independent. An unbiased estimate Θ is provided by

$$\bar{\Theta}_m = \frac{1}{2m}\sum_{j=1}^{m} \left(\phi(x_j) + \phi(\bar{x}_j)\right).$$

[2] Eigenfactors of Γ, columns of P.

The variance of $\bar{\Theta}_m$ is

$$\bar{\sigma}_m^2 = \frac{1}{2m}\left(\sigma^2 + \mathbb{C}\mathrm{ov}(\phi(X), \phi(\bar{X}))\right),$$

with $\bar{X} = F_X^{-1}(1-U)$. The following result gives a simple condition ensuring variance reduction by this method.

Proposition 2 *If ϕ is a monotone function, then $\bar{\sigma}_m^2 \leq \frac{1}{2}\frac{\sigma^2}{m}$.*

Proof. Introducing an independent copy V of U, by the monotonicity of $\psi = \phi \circ F_X^{-1}$ we have:

$$(\psi(U) - \psi(V))\,(\psi(1-U) - \psi(1-V)) \leq 0,$$

so that

$$\mathbb{E}\left[\psi(U)\psi(1-U) + \psi(V)\psi(1-V)\right] - \mathbb{E}\left[\psi(U)\psi(1-V) + \psi(V)\psi(1-U)\right] \leq 0$$

and

$$\mathbb{C}\mathrm{ov}(\phi(X), \phi(\bar{X})) = \mathbb{C}\mathrm{ov}(\psi(U), \psi(1-U)) \leq 0. \blacksquare$$

B Control Variates

The idea of control variates is to introduce another payoff function ψ and/or factor model Y, for which we have an explicit price, and to estimate the correction term by Monte Carlo. Then we decompose

$$\Theta = \mathbb{E}[\phi(X)] = \mathbb{E}[\phi(X) - \psi(Y)] + \mathbb{E}[\psi(Y)],$$

where $\mathbb{E}[\psi(Y)]$ is known. An unbiased estimate with m trials of Θ is defined by

$$\widehat{\Theta}_m = \mathbb{E}[\psi(Y)] + \frac{1}{m}\sum_{j=1}^m \left(\phi(x_j) - \psi(y_j)\right),$$

with x_j i.i.d. for the law of X and y_j i.i.d. for the law of Y. The variance of $\widehat{\Theta}_m$ is given by

$$\begin{aligned}\widehat{\sigma}_m^2 &= \frac{1}{m}\mathbb{V}\mathrm{ar}[\phi(X) - \psi(Y)] \\ &= \frac{1}{m}\left(\sigma^2 + \mathbb{V}\mathrm{ar}[\psi(Y)] - 2\mathbb{C}\mathrm{ov}(\phi(X), \psi(Y))\right).\end{aligned}$$

Variance reduction holds if the original random variable $\phi(X)$ and the control variate $\psi(Y)$ have a sufficient positive correlation.

C Importance Sampling

The basic idea of importance sampling is to concentrate the sample distribution in the part of the space which contributes most to the payoff. To this end we introduce a changed probability measure $\widetilde{\mathbb{Q}}$, with related expectation denoted by $\widetilde{\mathbb{E}}$, such that

$$\mathrm{supp}(\mathbb{Q}) \cap \mathrm{supp}(\phi(X)) \subseteq \mathrm{supp}(\widetilde{\mathbb{Q}}) \subseteq \mathrm{supp}(\mathbb{Q}) \tag{6}$$

or, equivalently in terms of the Radon-Nikodym density of $\widetilde{\mathbb{Q}}$ with respect to $\mathbb{Q}$:

$$\frac{d\widetilde{\mathbb{Q}}}{d\mathbb{Q}} = \mu,$$

for some nonnegative random variable μ with unit mean under $\mathbb{Q}$ and such that $\frac{\phi(X)}{\mu}$ is integrable under $\widetilde{\mathbb{Q}}$, where for $\omega \notin \mathrm{supp}(\widetilde{\mathbb{Q}})$ the ratio $\frac{\phi(X)}{\mu}(\omega)$ is understood as 0 in $\widetilde{\mathbb{E}}\left[\frac{\phi(X)}{\mu}\right] = \int_\Omega \frac{\phi(X)}{\mu}(\omega)\widetilde{\mathbb{Q}}(d\omega)$. Note

that we don't restrict ourselves to equivalent measures $\widetilde{\mathbb{Q}}$, but open the door to more general measures $\widetilde{\mathbb{Q}}$ satisfying (6) in which the right-hand side inclusion expresses the property that $\widetilde{\mathbb{Q}}$ is absolutely continuous with respect to $\mathbb{Q}$, but in which the left-hand side is less restrictive than $\mathbb{Q}$ absolutely continuous with respect to $\widetilde{\mathbb{Q}}$. This is crucial, since the whole idea of importance sampling is precisely to concentrate the sample distribution in the part of the space which contributes most to the payoff, in particular by the use of measures $\widetilde{\mathbb{Q}}$ for which the left-hand side inclusion is strict in (6).

Given a measure $\widetilde{\mathbb{Q}}$ satisfying (6) or the equivalent conditions in terms of μ, it follows much like as in I.(25) (case $t = 0$ there) that

$$\Theta = \mathbb{E}[\phi(X)] = \widetilde{\mathbb{E}}\left[\frac{\phi(X)}{\mu}\right]. \tag{7}$$

The Monte Carlo estimate of Θ related to (7) is

$$\widetilde{\Theta}_m = \frac{1}{m}\sum_{j=1}^{m}\frac{\phi(\widetilde{x}_j)}{\mu_j},$$

where the $\widetilde{x}_j$ are i.i.d. for the law of X under $\widetilde{\mathbb{Q}}$.

Remark 1 *As compared with control variate an inconvenience of importance sampling is that one must simulate X under the changed measure $\widetilde{\mathbb{Q}}$. Since in practice X is typically given as the value at T of some process defined in SDE form, this concretely means that a prerequisite for using this method is to implement the change of measure in the SDE that defines X. This is done in practice by means of Girsanov transforms, which are used to rewrite the SDE in terms of fundamental martingales (Brownian motions and/or compensated jump measures) with respect to $\widetilde{\mathbb{Q}}$ (see e.g. Crépey (2013, Section 12.3.2)).*

The goal of the exercise is then to pick an admissible measure $\widetilde{\mathbb{Q}}$ minimizing the $\widetilde{\mathbb{Q}}$-variance of $\widetilde{\Theta}_m$. Since

$$\widetilde{\mathbb{E}}\widetilde{\Theta}_m = \widetilde{\mathbb{E}}\left[\frac{\phi(X)}{\mu}\right] = \Theta,$$

which doesn't depend on μ, this is equivalent to minimizing

$$\widetilde{\mathbb{E}}\widetilde{\Theta}_m^2 = \frac{1}{m}\widetilde{\mathbb{E}}\left(\frac{\phi(X)}{\mu}\right)^2.$$

The minimum variance (null in the case of a nonnegative payoff $\phi(X)$) is reached for $\mu \propto |\phi(X)|$, so (since an admissible μ must have unit mean under $\mathbb{Q}$)

$$\mu = \mu^\sharp := \frac{|\phi(X)|}{\mathbb{E}|\phi(X)|}. \tag{8}$$

But usually $\mu^\sharp$ cannot be computed explicitly. Note in particular that, for $\phi > 0$, the number Θ that we are looking for sits in the denominator of the right-hand side of (8). In practice we use

$$\mu^\flat = \frac{|\psi(Y)|}{\mathbb{E}|\psi(Y)|},$$

for approximate payoff functions ψ and/or factor model Y obeying the same rationale as for control variate in Part B. So $\phi(X)$ and $\psi(Y)$ should have as large a positive correlation as possible and $\mathbb{E}|\psi(Y)|$ should be computable explicitly.

D Efficiency Criterion

We now introduce a heuristic criterion to compare the efficiency of various simulation schemes, with or without variance reduction. This criterion takes into account not only the accuracy (variance), but also the computation time required by the simulation for each scheme. The efficiency of a method $\widetilde{M}$ with respect to a method M is defined as

$$\mathcal{E} = \lim_{m,\widetilde{m}\to\infty} \frac{\sigma_m}{\widetilde{\sigma}_{\widetilde{m}}} \sqrt{\frac{t_m}{\widetilde{t}_{\widetilde{m}}}},$$

where σ_m and t_m (respectively $\widetilde{\sigma}_{\widetilde{m}}$ and $\widetilde{t}_{\widetilde{m}}$) are the standard error and the computation times of method M based on m simulation runs (respectively $\widetilde{M}$ based on $\widetilde{m}$ simulation runs). Method $\widetilde{M}$ is considered to be more efficient than method M if $\mathcal{E} \geq 1$. For instance, $\mathcal{E} = 3$ means that for a given computation time method $\widetilde{M}$ is three times more accurate than method M, or that for a given accuracy method $\widetilde{M}$ is nine times faster than method M. Assuming computation times proportional to the sample sizes (so that $t_m = km$, where k is a factor which expresses the complexity of the algorithm for method M, and likewise $\widetilde{t}_{\widetilde{m}} = \widetilde{k}\widetilde{m}$ for method $\widetilde{M}$), then we have

$$\mathcal{E} \sim_{m,\widetilde{m}\to\infty} \frac{\sigma}{\widetilde{\sigma}} \sqrt{\frac{k}{\widetilde{k}}}.$$

Efficiency in the sense of this criterion $\mathcal{E}$ is thus asymptotically independent of the sample size.

E Quasi Monte Carlo

Quasi Monte Carlo consists in estimating

$$\Theta = \mathbb{E}[\psi(U)] = \int_{[0,1]^d} \psi(u)du$$

by $\frac{1}{m}\sum_{j=1}^m \psi(u_j)$, where u is a d-variate low-discrepancy sequence. Unlike genuine Monte Carlo, Quasi Monte Carlo doesn't provide a confidence interval. The empirical variance of the sample is not meaningful because successive terms of the sequence are not independent. This is due to the construction of low-discrepancy sequences.

However, one has the following deterministic inequality. The definition of the Hardy-Krause variation of ψ which appears in its statement is rather technical. In dimension one, its value for a regular function ψ coincides with the usual notion $\int_{[0,1]} \mid \psi'(u) \mid du$.

Theorem 1 (Koksma-Hlawka inequality) *For every sequence u of points in $\mathbb{R}^d$ and for every $m \geq 1$, one has:*

$$\left| \frac{1}{m}\sum_{j=1}^m \psi(u_j) - \int_{[0,1]^d} \psi(u)du \right| \leq V(\psi)D_m(u),$$

where $V(\psi)$ denotes the Hardy-Krause variation of ψ.

The Koksma-Hlawka inequality gives an a priori deterministic bound for the error in the approximation of $\int_{[0,1]^d} \psi(x)dx$ by $\frac{1}{m}\sum_{j=1}^m \psi(u_j)$. This error is expressed in terms of the discrepancy of the sequence u and of the variation of the function ψ. Through this inequality we understand the interest in having sequences with discrepancy D_m as small as possible. But it is often difficult to calculate or even to estimate the variation of ψ. Moreover, since for large dimensions d the asymptotic bound $\frac{(\ln m)^d}{m}$ of a low-discrepancy sequence may only be meaningful for very large values of m, and because $\frac{(\ln m)^d}{m}$ increases exponentially with d, we see that for large d the bound in the Koksma-Hlawka inequality gives no relevant information for realistic sample sizes m. Fortunately, the effective dimension of a problem,

in the sense of the number of risk factors that explain most of its variance, is often much lower than its nominal dimension d.

We benefit most from a low-discrepancy sequence by assigning the main risk factors of the problem, ordered by decreasing amount of explained variance, to the successive components of the points of a multivariate low-discrepancy sequence. Thus, even though we use a low-discrepancy sequence in the nominal dimension d of the problem, which may be high, the circumstance that the first coordinates of the quasi-random points are assigned to the main risk factors of the problem avoids many of the drawbacks generally associated with high-dimensional low-discrepancy sequences.

§5 Greeking by Monte Carlo

Price sensitivities, or Greeks, are a key issue in financial modeling. Indeed, unless exotic products are considered, derivative prices are created by supply-and-demand in the market. As we will see in Chapter VII, liquid market prices are in fact used rather than those produced by models, in a reverse-engineering mode, as calibration input data. Greeks, on the contrary, can only be computed within models. Now, in many cases, Greeks, like prices, can also be put in the form $\Theta = \mathbb{E}[\phi(X)]$, so that the Monte Carlo pricing techniques we have seen so far can also be used for greeking. This is briefly illustrated in this section in the problem of computing, by Monte Carlo, the delta of an option in the Black–Scholes model. We thus want to compute $\Delta_0 = \partial_s \mathbb{E}[\phi(S_T^s)]$, where S_T^s is the value at maturity of an underlying S with initial condition s at time 0. One obvious method consists in repricing the payoff by Monte Carlo for a perturbed initial condition in order to get a finite difference estimate for Δ_0. But such a resimulation procedure is costly and biased. In many cases, direct approaches without resimulation are possible:

- by differentiation of the payoff, provided the latter is smooth enough;
- by differentiation of the transition probability density $p_T(s, S)$ of S, provided the latter exists and is smooth enough.

In general these two approaches rely, respectively, on the theory of stochastic flows (Glasserman, 2003; Broadie and Glasserman, 1996) and on Malliavin calculus (Fournié et al., 1999, 2001). However, in the Black-Scholes model the related computations are, as we will now see, elementary.

A Differentiation of the Payoff

In case ϕ is sufficiently regular and S_T^s is differentiable with respect to s, Δ_0 may be computed, assuming commutation of the expectation and differentiation operators, by

$$\Delta_0 = \partial_s \mathbb{E}\phi(S_T^s) = \mathbb{E}\partial_s \phi(S_T^s) = \mathbb{E}\left[\phi'(S_T^s)\partial_s S_T^s\right] \tag{9}$$

(Giles and Glasserman, 2006; Glasserman, 2003). In multiplicative models such as Black-Scholes or more general homogeneous models for which $\frac{S_T}{s}$ doesn't depend on s, one has $\partial_s S_T^s = \frac{S_T^s}{s}$, so that

$$\Delta_0 = \tfrac{1}{s}\mathbb{E}[\phi'(S_T^s)S_T^s].$$

Note that ϕ here needs to be differentiable, at least outside a $\mathbb{Q}^{S_T^s}$-null set. This is, for instance, the case with a vanilla option since ϕ' (in the sense of distributions) is then defined as a step function and S_T^s has no atoms in Black–Scholes. But this is not the case for a step function ϕ, since ϕ' is then only defined as a Dirac mass at the point of discontinuity of ϕ. This method is thus applicable to the computation of the delta of a call option, but not the delta of a digital option (or gamma of a vanilla call option).

B Differentiation of the Density

Assume S admits a transition probability density $p_T(s,S)$ from $(0,s)$ to (T,S), differentiable in its first argument s (the initial condition at time 0 of the process (S_t)). Then, under mild regularity conditions,

$$\Delta_0 = \int_{S>0} \phi(S)\partial_1 p_T(s,S)\,dS = \int_{S>0} \phi(S)\frac{\partial_1 p_T(s,S)}{p_T(s,S)} p_T(s,S)\,dS,$$

so that

$$\Delta_0 = \mathbb{E}\left[\phi(S_T^s)\partial_1 \ln(p_T(s,S_T^s))\right].$$

In the Black-Scholes model, we saw in the example 1 that

$$p_T(s,S) = \frac{1}{S\sqrt{2\pi T}\sigma} e^{-\frac{\left(\ln(\frac{S}{s})-bT\right)^2}{2\sigma^2 T}}\ ,\ \partial_1 \ln(p_T(s,S_T^s)) = \frac{W_T}{s\sigma T},$$

so that

$$\Delta_0 = \mathbb{E}\left[\phi(S_T^s)\frac{W_T}{s\sigma T}\right].$$

C Finite Differences

If none of the above schemes is applicable, for a fixed bias parameter $\alpha > 0$, one can approximate Δ_0 at the order one of consistency[3] by the uncentered finite difference

$$\frac{1}{\alpha s}\left(\mathbb{E}[\phi(S_T^{(1+\alpha)s})] - \mathbb{E}[\phi(S_T^s)]\right)$$

or, at an improved order two of consistency, by the centered finite difference

$$\frac{1}{2\alpha s}\left(\mathbb{E}[\phi(S_T^{(1+\alpha)s})] - \mathbb{E}[\phi(S_T^{(1-\alpha)s})]\right). \tag{10}$$

The expectations in (10) can be estimated by Monte Carlo. In order to decrease the variance, common random numbers should be used to estimate both expectations in (10). In the Black-Scholes model, $S_T^s = s\exp(bT+\sigma W_T)$, we thereby obtain the following estimate for Δ_0:

$$\frac{1}{2\alpha s m}\sum_{j=1}^{m}\left(\phi\left((1+\alpha)se^{bT+\sigma\sqrt{T}\varepsilon_j}\right) - \phi\left((1-\alpha)se^{bT+\sigma\sqrt{T}\varepsilon_j}\right)\right), \tag{11}$$

where the ε_j are independent Gaussian draws.

Note that in the limit where α tends to 0, this method converges to differentiation of the payoff, hence for nondifferentiable payoffs numerical instabilities arise with this approach when $\alpha \to 0$.

§6 Monte Carlo Algorithms for Vanilla Options

A European Call, Put or Digital Option

Monte Carlo provides price and delta estimates with a confidence interval. The Quasi Monte Carlo method only provides price and delta estimates, without a confidence interval. The option price and delta at time 0 are

$$\Pi_0 = \mathbb{E}\left[e^{-rT}\phi(S_T)\right]\ ,\ \Delta_0 = \partial_s \mathbb{E}[e^{-rT}\phi(S_T)].$$

[3] See III.§2.A.2.

The corresponding Monte Carlo estimates are written as[4]:

$$\widehat{\Pi}_0 = \frac{1}{m} e^{-rT} \sum_{j=1}^{m} \pi^j \ , \ \widehat{\Delta}_0 = \frac{1}{m} e^{-rT} \sum_{j=1}^{m} \delta^j \text{ with } \delta^j = \partial_s \pi^j$$

$$(\widehat{\sigma}_0^{\Pi})^2 = \frac{1}{m-1} \left[\frac{1}{m} \sum_{j=1}^{m} (e^{-rT} \pi^j)^2 - \widehat{\Pi}_0^2 \right] \ , \ (\widehat{\sigma}_0^{\Delta})^2 = \frac{1}{m-1} \left[\frac{1}{m} \sum_{j=1}^{m} (e^{-rT} \delta^j)^2 - \widehat{\Delta}_0^2 \right].$$

The expressions of π^j and δ^j are detailed for each option with strike K below, assuming a Black-Scholes underlying.

• **Put**: The payoff is $(K - S_T)^+$, hence

$$\pi^j = (K - S_T^j)^+ \ , \ \delta^j = \begin{cases} -\partial_s S_T^j = -\frac{S_T^j}{s} & \text{if} \quad \pi^j \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$

• **Call**: The payoff is $(S_T - K)^+$. The call-put parity relations for the price and delta are:

$$C_0 = P_0 + s e^{-qT} - K e^{-rT} \ , \ \Delta_0^C = \Delta_0^P + e^{-qT},$$

where C/P_0 and $\Delta_0^{C/P}$ denote the call/put prices and deltas at time 0. These relations may be used for the call, in order to limit the variance, if the call is in-the-money.

• **Digital option**: The payoff is $R \mathbb{1}_{\{S_T \geq K\}}$ for some rebate R, hence

$$\pi^j = R \mathbb{1}_{\{S_T^j \geq K\}}.$$

A Monte Carlo finite difference estimate of the delta is obtained with

$$\delta^j = \frac{1}{2\alpha s} \left[\phi\left((1+\alpha) S_T^j\right) - \phi\left((1-\alpha) S_T^j\right) \right].$$

A.1 Adding Jumps

We will now add jumps in S, postulating the risk-neutral Merton model of I.§3.B:

$$S_T = s e^{aT + \sigma W_T} \prod_{l=1}^{N_T} (1 + J_l),$$

where $a = b - \lambda \bar{J}$, where N_t is a Poisson process with intensity λ and where the J_l are i.i.d. such that $\ln(1 + J_l)$ is $\mathcal{N}(\varrho, \nu)$-distributed. Essentially nothing changes except for the fact that S_T^j of A is now replaced by

$$S_T^j = s e^{aT + \sigma\sqrt{T}\varepsilon_j} \prod_{l=1}^{N_T^j} \exp(\varrho + \sqrt{\nu}\varepsilon_j^l),$$

where

- N_T^j is an independent $\mathcal{P}_{\lambda T}$-draw;
- ε_j and the ε_j^l are independent Gaussian draws.

[4] cf. (1), (2) , (11), (9).

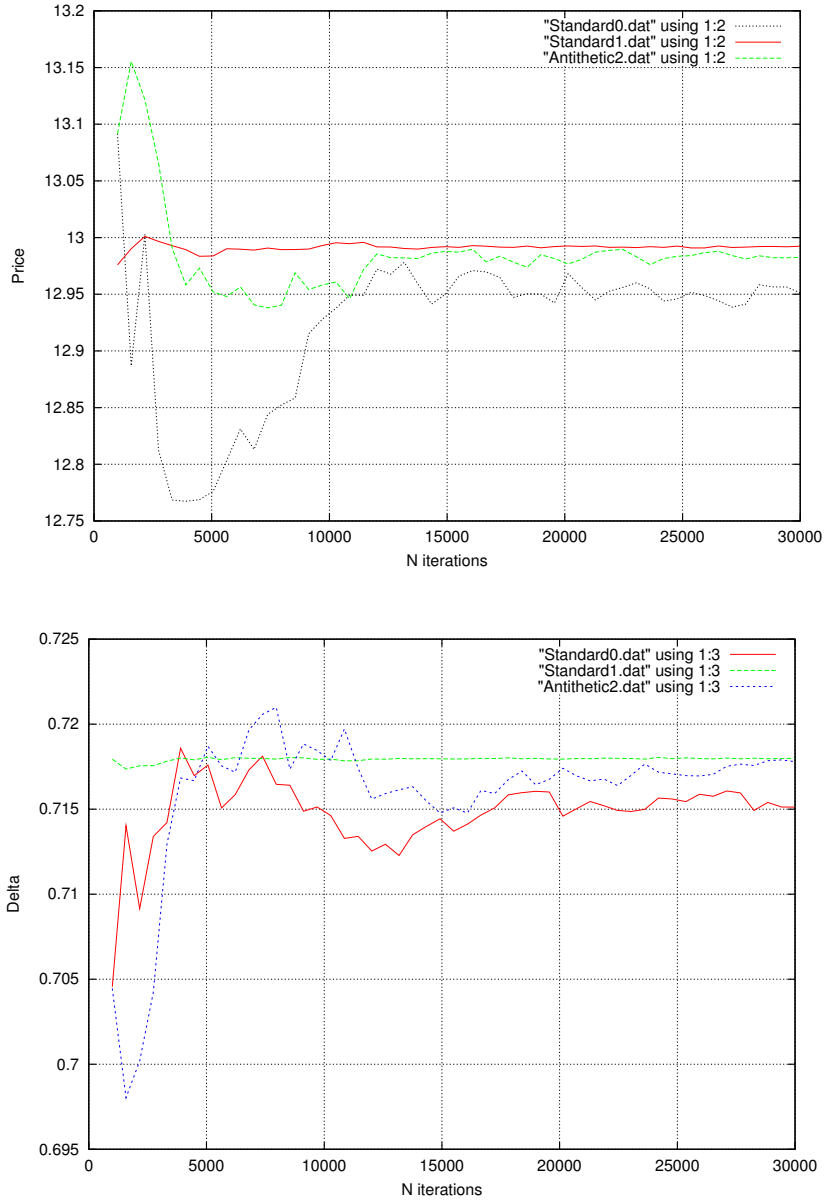


Figure 2: *European vanilla call priced in the Black-Scholes model by Monte Carlo, first using L'Ecuyer's pseudo-random numbers generator, then with antithetic variables and finally by Quasi Monte Carlo based on the one-dimensional Sobol sequence.*

B Call on Maximum, Put on Minimum, Exchange or Best of Options

Assume underlying assets evolve according to the following risk-neutral bivariate Black-Scholes model: $S_0^1 = s_1$, $S_0^2 = s_2$ and, for $t \in [0, T]$,

$$\begin{cases} \mathrm{d}S_t^1 = S_t^1(\kappa_1 dt + \sigma_1 dW_t^1) \\ \mathrm{d}S_t^2 = S_t^2(\kappa_2 dt + \sigma_2 dW_t^2), \end{cases}$$

with $\kappa_l = r - q_l$, where W^1 and W^2 are two Brownian motions with correlation ρ. So, in terms of a third Brownian motion $\widetilde{W}^2$ independent from W^1:

$$\begin{cases} S_T^1 = s_1 \exp(b_1 T + \sigma_{1,1} W_T^1) \\ S_T^2 = s_2 \exp(b_2 T + \sigma_{2,1} W_T^1 + \sigma_{2,2} \widetilde{W}_T^2), \end{cases}$$

with

$$\begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} \kappa_1 - \frac{\sigma_1^2}{2} \\ \kappa_2 - \frac{\sigma_2^2}{2} \end{pmatrix}, \quad \begin{pmatrix} \sigma_{1,1} & \sigma_{1,2} \\ \sigma_{2,1} & \sigma_{2,2} \end{pmatrix} = \begin{pmatrix} \sigma_1 & 0 \\ \rho\sigma_2 & \sqrt{1-\rho^2}\sigma_2 \end{pmatrix}.$$

Remark 2 *The latter matrix arises by Cholesky decomposition* (5) *of the covariance matrix* $h^{-1}\mathbb{C}\mathrm{ov}_t \begin{pmatrix} \sigma_1 dW_t^1 \\ \sigma_2 dW_t^2 \end{pmatrix}$

The price of an option at time 0 is

$$\Pi_0 = \mathbb{E}\left[e^{-rT}\phi(S_T^1, S_T^2)\right],$$

where ϕ denotes a payoff function. The deltas at time 0 are given by

$$\Delta_0^1 = \partial_{s_1}\mathbb{E}[e^{-rT}\phi(S_T^1, S_T^2)], \ \Delta_0^2 = \partial_{s_2}\mathbb{E}[e^{-rT}\phi(S_T^1, S_T^2)].$$

The price and delta estimates are written as

$$\widehat{\Pi}_0 = \tfrac{1}{m}e^{-rT}\textstyle\sum_{j=1}^m \pi^j, \ \widehat{\Delta}_0^l = \tfrac{1}{m}e^{-rT}\sum_{j=1}^m \partial_{s_l}\pi^j = \tfrac{1}{m}e^{-rT}\sum_{j=1}^m \delta_l^j,$$

for $l = 1, 2$. The values for π^j and δ_l^j are detailed below for each option with strike K.

• **Put on the Minimum**: The payoff is $(K - \min(S_1, S_2))^+$, so that

$$\begin{aligned} \pi^j &= \left(K - \min(S_T^{1,j}, S_T^{2,j})\right)^+ \\ \delta_1^j &= \begin{cases} -\exp(b_1 T + \sigma_{1,1} W_T^{1,j}) & \text{if} \quad \pi^j \geq 0 \quad \text{and} \quad S_T^{1,j} \leq S_T^{2,j} \\ 0 & \text{otherwise} \end{cases} \\ \delta_j^2 &= \begin{cases} -\exp(b_2 T + \sigma_{2,1} W_T^{1,j} + \sigma_{2,2} \widetilde{W}_T^{2,j}) & \text{if} \quad \pi^j \geq 0 \quad \text{and} \quad S_T^{1,j} \geq S_T^{2,j} \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

• **Call on the Maximum**: The payoff is $(\max(S_1, S_2) - K)^+$, so that

$$\begin{aligned} \pi^j &= \left(\max(S_T^{1,j}, S_T^{2,j}) - K\right)^+ \\ \delta_1^j &= \begin{cases} \exp(b_1 T + \sigma_{1,1} W_T^{1,j}) & \text{if} \quad \pi^j \geq 0 \quad \text{and} \quad S_T^{1,j} \geq S_T^{2,j} \\ 0 & \text{otherwise} \end{cases} \\ \delta_j^2 &= \begin{cases} \exp(b_2 T + \sigma_{2,1} W_T^{1,j} + \sigma_{2,2} \widetilde{W}_T^{2,j}) & \text{if} \quad \pi^j \geq 0 \quad \text{and} \quad S_T^{1,j} \leq S_T^{2,j} \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

• **Exchange Option**: The payoff is $(S_1 - KS_2)^+$, so that

$$\pi^j = \left(S_T^{1,j} - KS_T^{2,j}\right)^+$$
$$\delta_1^j = \begin{cases} \exp(b_1 T + \sigma_{1,1} W_T^{1,j}) & \text{if} \quad \pi^j \geq 0 \\ 0 & \text{otherwise} \end{cases}$$
$$\delta_j^2 = \begin{cases} -K \exp(b_2 T + \sigma_{2,1} W_T^{1,j} + \sigma_{2,2} \widetilde{W}_T^{2,j}) & \text{if} \quad \pi^j \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$

• **Best of Option**: The payoff is $[\max(S_1 - K_1, S_2 - K_2)]^+$, so that

$$\pi^j = \left[\max(S_T^{1,j} - K_1, S_T^{2,j} - K_2)\right]^+$$
$$\delta_1^j = \begin{cases} \exp(b_1 T + \sigma_{1,1} W_T^{1,j}) & \text{if} \quad \pi^j \geq 0 \quad \text{and} \quad S_T^{1,j} - K_1 \geq S_T^{2,j} - K_2 \\ 0 \end{cases}$$
$$\delta_j^2 = \begin{cases} \exp(b_2 T + \sigma_{2,1} W_T^{1,j} + \sigma_{2,2} \widetilde{W}_T^{2,j}) & \text{if} \quad \pi^j \geq 0 \quad \text{and} \quad S_T^{1,j} - K_1 \leq S_T^{2,j} - K_2 \\ 0 & \text{otherwise.} \end{cases}$$

§7 Simulation of Processes

Until now we were able to simulate S_T directly, without time-discretization of the associated SDE. However, to deal with more complex models, or even with simple models with path-dependent payoffs, we need to simulate the whole trajectory of the underlying between the pricing time 0 and the maturity T. Simulating trajectories is also necessary for testing the performances of a hedging scheme. In this case, the "path-dependent payoff" that interests us consists of the profit-and-loss at maturity of a delta-hedged option position.

A Brownian Motion

Recall that a standard Brownian motion W is a continuous process starting from 0 at time 0 with the properties that, for $0 \leq s < t$, the increment $W_t - W_s$ is independent of $\mathfrak{F}_s^W$ and is normally distributed with mean zero and variance $t - s$. By the Cholesky decomposition (5) (case $d = 3$ there) of the covariance matrix of (W_s, W_t, W_T), we can represent in law

$$\begin{cases} W_T = \sqrt{T}\varepsilon_T \\ W_t = \sqrt{t}\left(\sqrt{\frac{t}{T}}\varepsilon_T + \sqrt{1 - \frac{t}{T}}\varepsilon_t\right) \\ W_s = \sqrt{s}\left(\sqrt{\frac{s}{T}}\varepsilon_T + \rho\varepsilon_t + \sqrt{1 - \frac{s}{T} - \rho^2}\varepsilon_s\right), \end{cases}$$

for $(\varepsilon_s, \varepsilon_t, \varepsilon_T)$ standard trivariate Gaussian and for ρ in the last line defined by $s = \sqrt{st}\left(\sqrt{\frac{ts}{T^2}} + \sqrt{1 - \frac{t}{T}}\rho\right)$. Therefore, in particular,

$$\mathcal{L}(W_t \,|\, W_T) = \mathcal{N}\left(\frac{t}{T}W_T, \frac{t\,(T-t)}{T}\right) \tag{12}$$

and

$$\begin{aligned} \mathbb{C}\mathrm{ov}\left(W_s, W_t \,|\, W_T\right) = \sqrt{ts}\sqrt{1 - \frac{t}{T}}\rho = \\ \sqrt{ts}\left(\sqrt{\frac{s}{t}} - \sqrt{\frac{ts}{T^2}}\right) = s(1 - \frac{t}{T}). \end{aligned} \tag{13}$$

We now present two approaches for simulating a Brownian path on a time-grid $t_0 = 0 < t_1 < \ldots < t_n = T$. Let $h_i = t_{i+1} - t_i$ for $i = 0 \ldots n-1$, so that $h_i = h = \frac{T}{n}$ in the case of a uniform time-grid.

A.1 Forward Simulation

The forward simulation of W is defined by $W_0 = 0$ and, for $0 \leq i \leq n-1$,

$$W_{t_{i+1}} = W_{t_i} + \sqrt{h_i}\varepsilon_i$$

for independent Gaussian draws ε_i. Note that independent simulation of each W_{t_i} as $\sqrt{t_i}\varepsilon_i$ would not give the right variance for $(W_{t_{i+1}} - W_{t_i})$.

A.2 Backward Simulation

The backward simulation of W is based on the following Brownian Bridge property, which follows from (12):

$$\mathcal{L}(W_{\frac{t+s}{2}} \,|\, W_s = x, W_t = y) = \mathcal{N}(\tfrac{x+y}{2}, \tfrac{t-s}{4}) \ .$$

For this algorithm one must choose n as a power of 2. The first step is directly from 0 to T, letting $W_T = \sqrt{T}\varepsilon_T$. Intermediates steps are then filled in by taking successive subdivisions of the time intervals into halves. For every $\nu \geq 0$, assuming we have already simulated $W_{\frac{\imath T}{2^\nu}}$ for $0 \leq \imath \leq 2^\nu$, for every $0 \leq \imath \leq 2^\nu - 1$ we simulate

$$W_{\frac{(2\imath+1)T}{2^{(\nu+1)}}} = \tfrac{1}{2}(W_{\frac{\imath T}{2^\nu}} + W_{\frac{(\imath+1)T}{2^\nu}}) \ + \ \sqrt{\tfrac{T}{2^{(\nu+2)}}}\varepsilon_{\frac{(2\imath+1)T}{2^{(\nu+1)}}}$$

for independent Gaussian draws ε. This algorithm can also be adapted to a nonuniform time mesh by using the law of the Brownian Bridge at any time u between s and t:

$$\mathcal{L}\Big(W_u \,\Big|\, W_s = x, W_t = y\Big) = \mathcal{N}\left(\tfrac{t-u}{t-s}x + \tfrac{u-s}{t-s}y, \tfrac{(t-u)(u-s)}{t-s}\right).$$

Both the forward and the backward simulations of W require a vector of n independent Gaussian draws per trajectory. This vector can be computed from either n successive values of a univariate pseudo-random sequence, or from an n-variate quasi-random point. Regarding the backward simulation scheme for W, note that the values of W successively determined on each trajectory are drawn by order of decreasing variance. In case an n-variate quasi-random generator is used for generating the ε, this decreasing variance property means that the first components of every simulated quasi-random point, which are also "the more uniform ones" (recall Figure 1), are used for simulating the main directions of risk of the trajectory of W.

B Diffusions

In the case of a Brownian SDE with constant coefficients, X_t is a function of W_t. A path of X can then be simulated exactly (without time-discretisation error) as the corresponding function of a path of W. This is, for instance, the case with the risk-neutral Black-Scholes model stated in log-returns variable $X = \ln(S)$. By contrast, as soon as an SDE has non constant coefficients (and cannot simply be transformed into an SDE with constant coefficients), the only way to simulate it is by discretizing time and using the results of the constant coefficient case locally on small time intervals. However, this entails a time-discretization error.

Let us thus consider a d-dimensional diffusion

$$dX_t = b(t, X_t)dt + \sigma(t, X_t)dW_t \tag{14}$$

(jumps will be added later). As reviewed in IX.§3, this SDE admits a unique strong solution X over $[0, T]$ for coefficients Lipschitz and with linear growth in x uniformly in time. The two best-known time-discretization schemes for (14) are the Euler and the Milstein schemes.

B.1 Euler Scheme

The Euler scheme is defined by $\widehat{X}_0 = 0$ and, for every $i = 0, \dots, n-1$,

$$\widehat{X}_{t_{i+1}} = \widehat{X}_{t_i} + b(t_i, \widehat{X}_{t_i})h_i + \sigma(t_i, \widehat{X}_{t_i})(W_{t_{i+1}} - W_{t_i}).$$

Simulation is obtained with a forward algorithm by $\widehat{X}_0 = X_0$ and, for $i = 0, \dots, n-1$,

$$\widehat{X}_{t_{i+1}} = \widehat{X}_{t_i} + b(t_i, \widehat{X}_{t_i})h_i + \sigma(t_i, \widehat{X}_{t_i})\sqrt{h_i}\varepsilon_i,$$

for i.i.d. standard Gaussian draws ε_i. Let $h_i = h$ for notational simplicity.

Proposition 3 *For b, σ and ϕ "regular enough" in the sense detailed in Lapeyre, Pardoux, and Sentis (2003, page 143):*

(i) Strong convergence results *(in L_2 and trajectorial)*

$$\mathbb{E}\left(\sup_{0\le i\le n} |X_{ih} - \widehat{X}_{ih}|^2\right) = \mathrm{O}(h)$$

$$\sup_{0\le i\le n} |X_{ih} - \widehat{X}_{ih}| = \mathrm{o}(h^{\frac{1}{2}-\alpha}) \text{ almost surely, for every } \alpha > 0.$$

(ii) Convergence in law

$$|\mathbb{E}\phi(X_T) - \mathbb{E}\phi(\widehat{X}_T)| = \mathrm{O}(h).$$

Thus L^2-convergence is of order $h^{\frac{1}{2}}$, almost sure convergence is of order $h^{\frac{1}{2}}$ (essentially) and convergence in law is linear in h in regular cases. For pricing applications, the most relevant notion of convergence is that of convergence in law.

Continuous Euler scheme The continuous Euler scheme is the continuous-time approximation scheme $(\bar{X}_t)_{t\ge 0}$ defined by interpolation of the Euler scheme by a Brownian Bridge between $(t_i, \widehat{X}_{t_i})$ and $(t_{i+1}, \widehat{X}_{t_{i+1}})$, for $i = 0, \dots, n-1$. So, on $[t_i, t_{i+1}]$,

$$\bar{X}_t = \widehat{X}_{t_i} + b(t_i, \widehat{X}_{t_i})(t - t_i) + \sigma(t_i, \widehat{X}_{t_i})B^i_t,$$

where B^i is a Brownian Bridge on $[t_i, t_{i+1}]$ such that $\bar{X} = \widehat{X}$ at t_i and t_{i+1}.

B.2 Milstein Scheme

Assuming a one-dimensional and time-homogeneous diffusion, the Milstein Scheme for X appears as follows: $\widetilde{X}_0 = 0$ and, for every $i = 0, \dots, n-1$,

$$\widetilde{X}_{t_{i+1}} = \widetilde{X}_{t_i} + (b(\widetilde{X}_{t_i}) - \frac{1}{2}\sigma'(\widetilde{X}_{t_i})\sigma(\widetilde{X}_{t_i}))h_i + \sigma(\widetilde{X}_{t_i})(W_{t_{i+1}} - W_{t_i}) + \frac{1}{2}\sigma'(\widetilde{X}_{t_i})\sigma(\widetilde{X}_{t_i})(W_{t_{i+1}} - W_{t_i})^2.$$

Simulation is obtained with a forward algorithm by

$$\widetilde{X}_{t_{i+1}} = \widetilde{X}_{t_i} + (b(\widetilde{X}_{t_i}) - \frac{1}{2}\sigma'(\widetilde{X}_{t_i})\sigma(\widetilde{X}_{t_i}))h_i + \sigma(\widetilde{X}_{t_i})\sqrt{h_i}\varepsilon_i + \frac{1}{2}\sigma'(\widetilde{X}_{t_i})\sigma(\widetilde{X}_{t_i})h_i\varepsilon_i^2$$

for independent Gaussian draws ε_i. Let $h_i = h$ for notational simplicity. The following results show that convergence in law of the Milstein Scheme is again linear in h in regular cases. But the rates of L^2- and almost sure convergences are improved with respect to the Euler scheme.

Proposition 4 *For b, σ and ϕ "regular enough" in the sense detailed in Lapeyre, Pardoux, and Sentis (2003, page 144)):*

(i) Strong convergence results *(in L_2 and trajectorial)*

$$\mathbb{E}\left(\sup_{0\leq i\leq n}|X_{ih}-\widetilde{X}_{ih}|^2\right)=\mathrm{O}(h^2)$$
$$\sup_{0\leq i\leq n}|X_{ih}-\widetilde{X}_{ih}|=\mathrm{o}(h^{1-\alpha}) \text{ almost surely, for every } \alpha>0.$$

(ii) Convergence in law. *Same statement as for the Euler scheme.*

Example 3 *We consider the risk-neutral Heston model*

$$\begin{cases} dv_t=-\mu(v_t-\theta)dt+\eta\sqrt{v_t}dB_t \\ dS_t=S_t(\kappa dt+\sqrt{v_t}dW_t) \end{cases} \tag{15}$$

with $d\langle W,B\rangle=\rho dt$. The Heston SDE is not Lipschitz, but one can show[5] *existence and uniqueness for a strong solution (v,S) of (15). A commonly used time-discretization scheme for (15) consists of a Milstein scheme for v and an Euler scheme for $X=\ln(S)$, so: $\widetilde{v}_0=v_0$, $\widehat{X}_0=x=\ln(S_0)$, and, for every $i=0,\dots,n-1$,*

$$\begin{cases} \widetilde{v}_{t_{i+1}}-\widetilde{v}_{t_i}=-(\mu(\widetilde{v}_{t_i}-\theta)+\dfrac{\eta^2}{4})h_i+\eta\sqrt{\widetilde{v}_{t_i}^+h_i}\varepsilon_i+\dfrac{h_i\eta^2\varepsilon_i^2}{4} \\ \widehat{X}_{t_{i+1}}-\widehat{X}_{t_i}=(\kappa-\frac{\widetilde{v}_{t_i}}{2})h_i+\sqrt{\widetilde{v}_{t_i}^+h_i}(\rho\varepsilon_i+\sqrt{1-\rho^2}\widetilde{\varepsilon}_i), \end{cases} \tag{16}$$

for i.i.d. standard Gaussian pairs $(\varepsilon_i,\widetilde{\varepsilon}_i)$.

Note that for well-definedness of the scheme we took the positive part of $\widetilde{v}_{t_i}^+$ under the square roots in (16) (another possibility would have been to use $|\widetilde{v}_{t_i}|$), which induces a specific simulation bias. The Milstein scheme $\widetilde{v}$ has a better trajectorial convergence to v than the Euler scheme $\widehat{v}$ and so $\widetilde{v}$ is less prone to take negative values than $\widehat{v}$; hence a less biased scheme results. Time-discretisation of the Heston model, and of affine processes more generally, is a tricky issue (Broadie and Kaya, 2006; Glasserman, 2003; Gatheral, 2011).

C Adding Jumps

We now consider the $\mathbb{R}^d$-valued jump-diffusion

$$dX_t=b(t,X_t)dt+\sigma(t,X_t)dW_t+\delta(t,X_{t-},J_{(t)})dN_t, \tag{17}$$

with jump intensity $\lambda(t,X_t)$ of a point process N and distribution $\pi(t,X_{t-},dx)$ of $J_{(t)}$.

As reviewed in IX.§3, this SDE admits a unique strong solution X over $[0,T]$ for coefficients Lipschitz and with linear growth in x uniformly in time (and y, regarding $\delta(t,x,y)$).

C.1 Poisson Process

First we consider the case of a Poisson process (N_t) with constant intensity λ. To simulate a path of (N_t) over $[0,T]$, one possibility is to simulate the successive i.i.d. $\mathcal{E}_\lambda$-sojourn times of N_t until time T. Alternatively, we can first simulate $N_T\sim\mathcal{P}_{\lambda T}$ (see Part §3.A) and then use the property that, conditionally on N_T, the jump times of the process on $[0,T]$ follow the order statistics of N_T i.i.d. $\mathcal{U}_{[0,T]}$-random variables (see e.g. Cont and Tankov (2003a)).

[5] See Bouchard and Chassagneux (2016, Section 7.4 p.239) and Jeanblanc, Yor, and Chesney (2009).

C.2 Euler Scheme

To simulate (17) at the times $0 < t_1 < \ldots < t_n = T$, we set $\widehat{X}_0 = X_0$ and, for $i = 0, \ldots, n-1$:

1. we simulate
$$\check{X}_{t_{i+1}} = \widehat{X}_{t_i} + b(t_i, \widehat{X}_{t_i})h_i + \sigma(t_i, \widehat{X}_{t_i})\sqrt{h_i}\varepsilon_i,$$
2. we compute $\widehat{X}_{t_{i+1}}$ by adding to $\check{X}_{t_{i+1}}$, with probability $1 - e^{-\lambda(t_i, \widehat{X}_{t_i})h_i} \approx \lambda(t_i, \widehat{X}_{t_i})h_i$ (for small h_i), a jump term "equal to $\delta(t_i, \widehat{X}_{t_i}, x)$ with probability $\pi(t_i, \widehat{X}_{t_i}, dx)$".

Alternatively, in case λ does not depend on X, to simulate each path of X, we can first simulate the ordered jump times T_l of N, add these to the time-grid (t_i) and run the above algorithm on the resulting enlarged time-grid (still denoted by (t_i) for simplicity), with (ii) replaced by

ii'. If $t_{i+1} = T_l$ for some l, then we simulate
$$\widehat{X}_{T_l} = \check{X}_{T_l} + \delta(T_l, \check{X}_{T_l}, J_l) \text{ where } J_l \sim w(T_l, \check{X}_{T_l}, dx),$$
otherwise we set $\widehat{X}_{t_{i+1}} = \check{X}_{t_{i+1}}$.

C.3 Continuous Euler Scheme

This is the continuous-time approximation scheme defined, using interpolation with the alternative Euler scheme at the previous section, by a Brownian Bridge between $(t_i, \widehat{X}_{t_i})$ and $(t_{i+1}, \check{X}_{t_{i+1}})$, for every i.

D Monte Carlo Simulation for Processes

In the case of Monte Carlo simulation for processes, the error can be decomposed into

$$\mathbb{E}[\phi(X_T)] - \frac{1}{m}\sum_{j=1}^{m}\phi(\widehat{X}_T^j) = \Big(\mathbb{E}[\phi(X_T)] - \mathbb{E}[\phi(\widehat{X}_T)]\Big) + \left(\mathbb{E}[\phi(\widehat{X}_T)] - \frac{1}{m}\sum_{j=1}^{m}\phi(\widehat{X}_T^j)\right), \tag{18}$$

where the two terms on the right-hand side are respectively referred to as the time-discretization error (a bias term) and the Monte Carlo or simulation error (or variance term):

- for usual time-discretization schemes such as the Euler or the Milstein scheme, convergence in law is linear in h, so the time-discretization error is $\mathrm{O}(h)$;
- the (pseudo) Monte Carlo error is $\mathrm{O}(\widehat{\sigma}m^{-\frac{1}{2}})$, where $\widehat{\sigma}$ represents the standard deviation of the approximate payoff $\phi(\widehat{X}_T)$.

The overall error is $\mathrm{O}(h) + \mathrm{O}(m^{-\frac{1}{2}})$, in comparison with $\mathrm{O}(h) + \mathrm{O}(m_1^{-2})$ in the case of a typical finite difference numerical scheme with a generic number m_1 of mesh points per space dimension (see Chapter III). Taking m as m_1^d in order to balance the computation times allocated to the two methods (stochastic Monte Carlo and deterministic finite differences), we conclude that Monte Carlo is more accurate and therefore more efficient for $d > 4$, and less efficient for $d < 4$. This conclusion is consistent with the related discussion in V.§4.A. Also note that in order to balance the two terms of the error in (18), we should take m of order of n^2. This can be implemented incrementally with backward time-discretization schemes (see A.2 in the case of the Brownian motion W).

§8 Monte Carlo Methods for Exotic Options

A nice feature of Monte Carlo is that it can easily cope with path dependence. However, specific treatments must be applied in order to preserve convergence rates. A recurrent idea in this regard is use of the continuous-time Euler scheme in order to try to recover the lost information about "what happens between the points of the time-grid" of a time-discretization scheme. Let $M_t = \sup_{0\le s\le t} W_s$ denote the running supremum of the Brownian motion and let

$$W_t^\lambda = W_t + \lambda t,\ M_t^\lambda = \sup_{0\le s\le t} W_s^\lambda \tag{19}$$

for any real λ.

Lemma 5 **(i)** *The bivariate process (W, M) admits the following transition probability density between times 0 and t :*

$$p_t(w,m) = \mathbb{1}_{m\ge w^+} \frac{2(2m-w)}{\sqrt{2\pi t^3}} \exp\left[-\frac{(2m-w)^2}{2t}\right]; \tag{20}$$

(ii) *the random variable*

$$Z_t^\lambda = (2M_t^\lambda - W_t^\lambda)^2 - (W_t^\lambda)^2$$

is, conditionally on W_t^λ, $\mathcal{E}_{\frac{1}{2t}}$-distributed;
(iii) *the conditional cumulative distribution function and inverse cumulative distribution function of M_t^λ, given $W_t^\lambda = w$, are written as*

$$\begin{aligned} F_t(m\,|\,w) &= \left(1 - \exp\Big[-\frac{2}{t}m(m-w)\Big]\right),\ m \ge w^+, \\ F_t^{-1}(u\,|\,w) &= \frac{1}{2}\Big(w + \sqrt{w^2 - 2t\ln(1-u)}\Big),\ 0 \le u \le 1. \end{aligned} \tag{21}$$

Proof. **(i)** The formula (20) is obtained by crossed differentiation with respect to x and y in the following "mirror formula", which is valid for every $x \in \mathbb{R}$ and $y \ge x^+$ (Karatzas and Shreve, 1991; Revuz and Yor, 1999):

$$\mathbb{Q}(W_t \ge 2y - x) = \mathbb{Q}(W_t \le x,\ M_t \ge y).$$

(ii) By the Girsanov formula, we have, for all real x, y :

$$\begin{aligned} \mathbb{Q}(W_t^\lambda \le x) &= \mu\mathbb{Q}(W \le x)\,, \\ \mathbb{Q}(W_t^\lambda \le x,\ M_t^\lambda \le y) &= \mu\mathbb{Q}(W \le x,\ M_t \le y) \end{aligned} \tag{22}$$

for some random weight μ, depending on λ but common to both equations in (22). Therefore

$$\mathbb{Q}(M_t^\lambda \le y \,\big|\, W_t^\lambda \le x) \;=\; \mathbb{Q}(M_t \le y \,\big|\, W_t \le x).$$

This shows that the law of M_t^λ conditional on W_t^λ doesn't depend on λ. We may thus restrict our attention to the case $\lambda = 0$. Denoting $Z_t = (2M_t - W_t)^2 - (W_t)^2$ and introducing the one-to-one mapping

$$[x^+, +\infty) \ni y \mapsto z = (2y-x)^2 - x^2 \in \mathbb{R}_+,$$

we have that

$$\mathbb{Q}(Z_t \in dz\,|\,W_t = x) = \mathbb{Q}(M_t \in dy\,|\,W_t = x)$$

and therefore, by (20),

$$\mathbb{Q}(Z_t \in dz\,|\,W_t = x) = e^{\frac{-z}{2t}}\frac{dz}{2t}.$$

Part **(iii)** follows directly from (ii) by Lemma 1. ∎

A Lookback Options

We consider a lookback option with payoff $\phi(X_T, M_T)$, where X is given in the form of the following one-dimensional diffusion:

$$dX_t = b(t, X_t)dt + \sigma(t, X_t)dW_t$$

and $M_t = \sup_{0\le s\le t} X_s$. The following result, based on Lemma 5, gives a way of simulating the pair $(\bar{X}, \bar{M})$, where $\bar{M}_t = \sup_{[0,t]} \bar{X}$ and $\bar{X}$ is the continuous-time Euler scheme for X. Let $\widehat{X} = (\widehat{X}_{t_i})_{0\le i\le n}$ denote the (discrete time) Euler scheme for X and let $\widetilde{M}_i = \sup_{t_i\le t\le t_{i+1}} \bar{X}_t$ for every $i = 0, \dots, n-1$.

Proposition 5 *We have*

$$\mathcal{L}\left((\widetilde{M}_i)_{0\le i\le n-1} \,\middle|\, \widehat{X}\right) = \mathcal{L}\left((\widehat{M}_i)_{0\le i\le n-1}\right),$$

where, for every i,

$$\widehat{M}_i := \frac{1}{2}\left(\widehat{X}_{t_i} + \widehat{X}_{t_{i+1}} + \sqrt{(\widehat{X}_{t_i} - \widehat{X}_{t_{i+1}})^2 - 2\sigma(t_i, \widehat{X}_{t_i})^2 h_i \ln(1-U_i)}\right) \tag{23}$$

for independent uniforms U_i.

Proof. Letting $\lambda_i = \frac{b(t_i,\widehat{X}_{t_i})}{\sigma(t_i,\widehat{X}_{t_i})}$, in the notation of(19) we have, for $t \in [t_i, t_{i+1}]$:

$$\mathcal{L}\left(\frac{\bar{X}_t - \widehat{X}_{t_i}}{\sigma(t_i, \widehat{X}_{t_i})} \,\middle|\, \widehat{X}\right) = \mathcal{L}\left(W_t^{\lambda_i} - W_{t_i}^{\lambda_i} \,\middle|\, W_{t_{i+1}}^{\lambda_i} - W_{t_i}^{\lambda_i} = \frac{\widehat{X}_{t_{i+1}} - \widehat{X}_{t_i}}{\sigma(t_i, \widehat{X}_{t_i})}\right)$$

and (for every i and also jointly in all i, by independence between the $\widetilde{M}_i$ given $\widehat{X}$)

$$\mathcal{L}\left(\frac{\widetilde{M}_i - \widehat{X}_{t_i}}{\sigma(t_i, \widehat{X}_{t_i})} \,\middle|\, \widehat{X}\right) = \mathcal{L}\left(\sup_{t\in[t_i,t_{i+1}]} W_t^{\lambda_i} - W_{t_i}^{\lambda_i} \,\middle|\, W_{t_{i+1}}^{\lambda_i} - W_{t_i}^{\lambda_i} = \frac{\widehat{X}_{t_{i+1}} - \widehat{X}_{t_i}}{\sigma(t_i, \widehat{X}_{t_i})}\right).$$

By application of Lemma 5(ii) to the drifted Brownian motions $W_{t_i+\cdot}^{\lambda_i} - W_{t_i}^{\lambda_i}$ on $[0, h_i]$, and also using Lemma 1, the law of the $\widetilde{M}_i$ given $\widehat{X}$ is then the same as that of the

$$\widehat{X}_{t_i} + \sigma(t_i, \widehat{X}_{t_i}) F_{h_i}^{-1}\left(U_i \,\middle|\, \frac{\widehat{X}_{t_{i+1}} - \widehat{X}_{t_i}}{\sigma(t_i, \widehat{X}_{t_i})}\right) = \widehat{M}_i, \tag{24}$$

as follows from the expression of F^{-1} in (21). ∎

In summary, to simulate a pair $(\bar{X}_T, \bar{M}_T)$, which can then be substituted into a Monte Carlo loop for pricing a lookback option with payoff $\phi(X_T, M_T)$:

i. we simulate a trajectory $\widehat{X}$ of the Euler scheme for X, using n independent uniforms u_i;

ii. given this trajectory $\widehat{X}$, we simulate related $\widehat{M}_i$ by (23), using n new independent uniforms U_i.

Then we set $\bar{X}_T = \widehat{X}_T$, $\bar{M}_T = \max_i \widehat{M}_i$. If quasi-random numbers are used, we must employ a $2n$-dimensional low-discrepancy sequence in i, ii (but the use of high-dimensional low-discrepancy sequences must be considered with caution).

A.1 Black-Scholes Case

In the special case of a Black-Scholes model (Andersen and Brotherton-Ratcliffe, 1996), the Euler discretization is exact provided one works in the log-returns variable $X_t = \ln(S_t)$. In this case one can then take n equal to one in i-ii above. Letting $M_T = \max_{t\in[0,T]} S_t$, the time-0 price and delta of a lookback option with payoff ϕ are

$$\Pi_0 = \mathbb{E}\left[e^{-rT}\phi(S_T, M_T)\right] \ , \ \Delta_0 = \partial_s \mathbb{E}[e^{-rT}\phi(S_T, M_T)],$$

with related estimates

$$\widehat{\Pi}_0 = \frac{1}{m}e^{-rT}\sum_{j=1}^{m}\pi^j \ , \ \widehat{\Delta}_0 = \frac{1}{m}e^{-rT}\sum_{j=1}^{m}\partial_s\pi^j = \frac{1}{m}e^{-rT}\sum_{j=1}^{m}\delta^j.$$

• **Fixed Strike Lookback Call** The payoff is $(M_T - K)^+$, so

$$\pi^j = (M_T^j - K)^+ \ , \ \delta^j = \begin{cases} \partial_s M_T^j = \frac{M_T^j}{s} & \text{if} \quad \pi^j \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$

• **Floating Strike Lookback Put** The payoff is $(M_T - S_T)$, so

$$\pi^j = (M_T^j - S_T^j) \ , \ \delta^j = \partial_s M_T^j - \partial_s S_T^j = \frac{M_T^j - S_T^j}{s} = \frac{\pi^j}{s}.$$

Simulation of the Maximum M_T At run j:

1. S_T^j is generated as $e^{X_T^j}$, with $X_T^j = x + bT + \sigma\sqrt{T}\varepsilon_j$ for an independent Gaussian draw ε_j;
2. M_T^j is generated as $se^{\sigma F_T^{-1}(u_j \,|\, \frac{X_T^j - x}{\sigma})}$ for an independent uniform draw u_j.

B Barrier Options

With a barrier option the right to exercise the payoff at maturity depends on additional events such as the underlying having crossed or reached certain levels on $[0, T]$: see Part III.§4.B.

For instance, a barrier up-and-out option with trigger level H and rebate R corresponds to the following payoff process (considering the case of a rebate R paid at T):

$$\psi(X_T, M_T) = \phi(X_T)\mathbb{1}_{\{M_T<H\}} + R\mathbb{1}_{\{M_T\geq H\}}.$$

An approximation for the price is given by

$$e^{-rT}\mathbb{E}\psi(\bar{X}_T, \bar{M}_T),$$

with

$$\mathcal{L}(\bar{M}_T \,|\, \widehat{X}) = \mathcal{L}(\max_{0\leq i\leq n-1} \widehat{M}_i)$$

as above. We have

$$\mathbb{E}\left(\phi(\bar{X}_T)\mathbb{1}_{\{\bar{M}_T\leq H\}} \,\middle|\, \widehat{X}\right) = \phi(\widehat{X}_T)\prod_{i=0}^{n-1}\mathbb{Q}(\max_{t_i\leq t\leq t_{i+1}} \bar{X}_t \leq H \,|\, \widehat{X})$$

$$= \phi(\widehat{X}_T)\prod_{i=0}^{n-1} F_{h_i}\left(\frac{H - \widehat{X}_{t_i}}{\sigma(t_i, \widehat{X}_{t_i})} \,\middle|\, \frac{\widehat{X}_{t_{i+1}} - \widehat{X}_{t_i}}{\sigma(t_i, \widehat{X}_{t_i})}\right).$$

Likewise

$$\mathbb{E}\left(R\mathbb{1}_{\{\bar{M}_T>H\}}\,\middle|\,\widehat{X}\right)=R\left(1-\prod_{i=0}^{n-1}F_{h_i}\left(\frac{H-\widehat{X}_{t_i}}{\sigma(t_i,\widehat{X}_{t_i})}\,\middle|\,\frac{\widehat{X}_{t_{i+1}}-\widehat{X}_{t_i}}{\sigma(t_i,\widehat{X}_{t_i})}\right)\right),$$

and finally

$$\mathbb{E}\psi(\bar{X}_T,\bar{M}_T)= \\ R+\mathbb{E}\left[\left(\phi(\widehat{X}_T)-R\right)\prod_{i=0}^{n-1}F_{h_i}\left(\frac{H-\widehat{X}_{t_i}}{\sigma(t_i,\widehat{X}_{t_i})}\,\middle|\,\frac{\widehat{X}_{t_{i+1}}-\widehat{X}_{t_i}}{\sigma(t_i,\widehat{X}_{t_i})}\right)\right].$$

In this case no random draws are needed beyond those used for simulating $\widehat{X}_T$, i.e. n random draws per simulation run, where n can be taken equal to one in the special case of the Black-Scholes model.

C Asian Options

We next consider an Asian option with a payoff of the form $\phi\left(S_T,I_T\right)$, with $I_t=\int_0^T S_u\,du$. For instance, $\phi(x,y)=(\frac{I}{T}-K)^+$ in the case of a fixed strike Asian call. We assume a Black-Scholes underlying S.

A first possible approximation for I_T is the Riemann sum $\widehat{I}_T^1=\sum_{i=0}^{n-1}h_i\widehat{S}_{t_i}$, but this discretization works poorly in practice. A better discretization is given by the trapezoid rule $\widehat{I}_T^2=\sum_{i=0}^{n-1}h_i\frac{\widehat{S}_{t_i}+\widehat{S}_{t_{i+1}}}{2}$. However, one can show by Taylor expansion that this is tantamount to approximating

$$\mathbb{E}\phi\left(\bar{S}_T,\bar{I}_T\right)=\mathbb{E}\left[\mathbb{E}\left[\phi\left(\bar{S}_T,\bar{I}_T\right)\left|\widehat{S}\right.\right]\right] \tag{25}$$

by

$$\mathbb{E}\left[\phi\left(\bar{S}_T,\mathbb{E}\left[\bar{I}_T\middle|\widehat{S}\right]\right)\right]\ \text{ with }\bar{I}_t=\int_0^T\bar{S}_u\,du.$$

But this approximation involves a nonlinearity bias. An even better way is to simulate the right-hand side of (25) directly (see Lapeyre and Temam (2001)), approximating $\bar{I}_T$ conditionally on $\widehat{S}$ by

$$\sum_{i=0}^{n-1}\widehat{S}_{t_i}\int_{t_i}^{t_{i+1}}\left(1+\kappa(t-t_i)+\sigma B_t^i dt\right) \\ \approx\sum_{i=0}^{n-1}h_i\widehat{S}_{t_i}\left(1+\frac{\kappa h_i}{2}+\frac{\sigma}{h_i}\int_{t_i}^{t_{i+1}}B_t^i dt\right)=:\widehat{I}_T^3.$$

Here B^i is a Brownian Bridge between (t_i,W_{t_i}) and $(t_{i+1},W_{t_{i+1}})$, so that $\int_{t_i}^{t_{i+1}}B_t^i dt=:\varepsilon_i$ is a Gaussian random variable with (given $W_{t_i},W_{t_{i+1}}$):

$$\mathbb{E}\left(\varepsilon_i\right)=\int_{t_i}^{t_{i+1}}\left(W_{t_i}+\frac{(t-t_i)}{h_i}(W_{t_{i+1}}-W_{t_i})\right)dt=\frac{h_i}{2}(W_{t_i}+W_{t_{i+1}}) \\ \mathbb{V}\mathrm{ar}\left(\varepsilon_i\right)=2\int_{u=t_i}^{t_{i+1}}\int_{t=t_i}^{u}\mathbb{C}\mathrm{ov}(B_t^i,B_u^i)dtdu=2\int_{v=0}^{h_i}(1-\frac{v}{h_i})\frac{v^2}{2}dv=\frac{h_i^3}{12},$$

where the identity $\mathbb{C}\mathrm{ov}(B_t^i,B_u^i)=(t-t_i)(1-\frac{u-t_i}{h_i})$ resulting from (13) was used in the second line.

The previous discretization schemes can be used in conjunction with variance reduction (Kemna and Vorst, 1990; Lapeyre and Temam, 2001). The arithmetic average $A_T=\frac{I_T}{T}$ is "close" to the geometric average $J_T=\exp\left(\frac{1}{T}\int_0^T\ln(S_t)\,dt\right)$ for r and σ "small". This suggests the use of $\phi\left(S_T,TJ_T\right)$ as a control variable for the payoff $\phi\left(S_T,I_T\right)$.

Example 4 *In the case of a fixed strike Asian call, we have*

$$\phi\left(S_T, TJ_T\right) = \left(TJ_T - K\right)^+,$$

where J_T is lognormally distributed, so that $\mathbb{E}\phi\left(S_T, TJ_T\right)$ is known explicitly. Thus

$$J_T = \exp\left(\frac{1}{T}\int_0^T \ln(S_t)\,dt\right) = S_0 \exp\left(\frac{1}{T}\int_0^T (\sigma W_t + bt)\,dt\right) = S_0 \exp\left(\frac{\sigma}{T}\int_0^T W_t\,dt + \frac{bT}{2}\right),$$

with $\mathbb{V}\mathrm{ar}(\int_0^T W_t\,dt) = 2\int_0^T \int_0^u t dt du = \int_0^T u^2 du = \frac{T^3}{3}$. Therefore

$$TJ_T = \widetilde{S}_0 \exp\left(\widetilde{\sigma}\sqrt{T}\varepsilon - \frac{\widetilde{\sigma}^2 T}{2}\right),$$

with $\widetilde{\sigma} = \frac{\sigma}{\sqrt{3}}$, $\widetilde{S}_0 = TS_0 \exp\left(\frac{bT}{2} + \frac{\widetilde{\sigma}^2 T}{2}\right)$. Thus

$$\mathbb{E}(TJ_T - K)^+ = \pi^{bl}(0, \widetilde{S}_0, T, K; \widetilde{\sigma}).$$

The above Brownian Bridge techniques can be extended to arbitrary jump-diffusions.

§9 American Monte Carlo Pricing Schemes

The prices of American options are the Snell envelopes of the related payoff processes, as opposed to straight expectations (or conditional expectations, at future times) in the case of European options. The supremum in the corresponding time-0 pricing formula bears on a huge set of stopping times. Unless perhaps machine learning techniques are used[6], this set cannot be discretized numerically. As a consequence, American options cannot be priced by the standard Monte Carlo loops of previous sections. Simulation pricing schemes for American options do exist however, in the form of hybrid "forward (as in Monte Carlo loops) / backward (as deterministic finite difference and tree) schemes", in which a dynamic programming equation is run backward in time at the points of a stochastic grid simulated in a forward manner according to the underlying factor dynamics. With respect to deterministic schemes, these simulation pricing schemes present the advantage of being less severely impacted by the curse of dimensionality[7].

For pricing an American option by Monte Carlo, we can thus write a dynamic programming equation on a stochastically generated (hence nonrecombining) mesh $(X_i^j)_{0\le i\le n}^{1\le j\le m}$, so that: $\Pi_n^j = \phi(X_n^j)$ for $j = 1\dots m$ and, for $i = n-1,\dots,0$, $j = 1\dots m$:

$$\Pi_i^j = \max\left(\phi\left(X_i^j\right),\, e^{-rh}\mathbb{E}_i^j \Pi_{i+1}\right), \tag{26}$$

where $\mathbb{E}_i^j\Pi_{i+1}$ is the conditional expectation of Π_{i+1} given $X_i = X_i^j$. The issue of computation of the conditional expectations on the nonrecombining mesh $(X_i^j)_{0\le i\le n}^{1\le j\le m}$ is dealt with in B. This is only required for $i \ge 1$, since for $i = 0$ the conditional expectation reduces to an expectation, and all the Π_0^j in (26) reduce to $\widehat{\Pi}_0 := \Pi_0^0$.

As American Monte Carlo estimates are not based on the law of large numbers and the central limit theorem of Part §1.A, they do not provide a confidence interval. Of course, one can always derive a confidence interval for the estimated (if not the true) price by running the simulation loop for various (say 50) seeds of the generator and computing a standard deviation of the estimates.

[6] cf. (Becker et al., 2019).
[7] see V.§4.A.

One can also recover, from the pricing function estimated by (26), the following estimates of the exercise region and of the related optimal stopping policy (starting from time 0 for the latter):

$$\mathcal{E} = \{(i, X_i^j)\,;\, \widehat{\Pi}_i^j = \phi(X_i^j)\},$$
$$\nu^j = \inf\{i \in \mathbb{N}_n\,;\, X_i^j \in \mathcal{E}\} \wedge T. \tag{27}$$

Finally it is also possible, by simulation, to compute the option delta (Bouchard and Touzi, 2004).

Remark 3 *Alternative nonlinear simulation pricing schemes not treated in these notes are purely forward branching particle simulation schemes (Guyon and Henry-Labordère, 2012, Chapter 12).*

A Time-0 Price

The above approach, which in the language of Markov decision theory corresponds to iteration on the values, was first developed by Tsitsiklis and Van Roy (2001). We now present an approach by iteration on the policies due to Longstaff and Schwartz (2001) for pricing the option at time 0. The idea is to use the estimated optimal stopping policy ν in (27) for computing an alternative estimate of the option price at time 0 as

$$\widetilde{\Pi}_0 = \mathbb{E}e^{-\sum_0^{\nu^j - 1} rh} \phi(X_{\nu^j}^j). \tag{28}$$

The value $\widetilde{\Pi}_0$ typically underestimates the exact price Π_0. It is also possible to derive an upper bound on the true price by resorting to a dual Monte Carlo approach of Rogers (2002). Computing both estimates gives a way to end up with an interval. If this interval is too large it typically means that the basis of functions which is used for computing the conditional expectations is not well chosen (see below).

B Computing Conditional Expectations by Simulation

The nonlinear Monte Carlo pricing schemes of this section ultimately reduce to the numerical computation of conditional expectations. As we will now see, this can be done by a combination of simulation and regression tools. Let ξ and X denote real and $\mathbb{R}^d$-valued square integrable random variables. Under suitable conditions the conditional expectation $\mathbb{E}(\xi|X)$ is equal to the L^2-projection of ξ over the vector space of random variables spanned by Borel functions of X. So, using a basis $(\varphi^l)_{l\in\mathbb{N}}$ of the set of functions from $\mathbb{R}^d$ to $\mathbb{R}$,

$$\mathbb{E}(\xi|X) = \mathbb{L}(\xi|(\varphi^l(X))_{l\in\mathbb{N}}),$$

where $\mathbb{L}$ represents the L^2-projection operator. Given pairs $(X^j, \xi^j)^{1\le j\le m}$ simulated independently according to the law of (X, ξ), the conditional expectation $\mathbb{E}(\xi|X)$ may thus be simulated by linear regression of $(\xi^j)^{1\le j\le m}$ against $(\varphi^l(X^j))_{1\le l\le \rho}^{1\le j\le m}$, where the truncation order ρ is a parameter of the method. The computational cost of this regression is $O(m\rho^2)$ for forming the regression matrix, plus the time needed for solving a (typically numerically ill-conditioned) linear system of dimension ρ.

We refer the interested reader to the monograph by Györfi, Kohler, Krzyzak, and Walk (2002) for details about these simulation/regression approaches for computing a regression function

$$x \mapsto \varrho(x) = \mathbb{E}(\xi|X = x).$$

Succintly, the (truncated) regression basis may be:

- either parametric, i.e. formed of functions parameterized by a few parameters, or nonparametric, meaning in practice that it is formed of a very large set of functions, like one function per point of a discretization of the state space;
- either global, that is, formed of functions supported by the whole state space or with "large" support, or local, formed of functions with "small" support.

One typically uses either a parametric and global regression basis, such as a regression basis formed of a few monomials parameterized by their coefficients, or a nonparametric and local basis, such as a regression basis formed of the indicator functions of the cells of a grid of hypercubes partitioning the state space (method of cells). Theory tells that a global basis is preferred in the case of a "regular" regression function $\varrho(x)$, especially when a good guess is available for the shape of ϱ; this guess can then be used to define the regression basis. Otherwise a local basis is preferred, being simpler and often more robust in terms of implementation. Note that, in the simplest case of the method of cells, the "regression" doesn't involve the solution of a linear system, the "regression matrix" being diagonal in this case.

Also note that there exist alternatives to nonlinear regression for computing conditional expectations by simulation, in particular Malliavin calculus based methods (Bouchard et al., 2004; Crisan et al., 2010) and quantization methods (Bally et al., 2001; Bally and Pagès, 2003). However, the former suffer from variance issues and the latter from the curse of dimensionality (Bouchard and Warin, 2010; Glasserman, 2003).

Chapter V

Pricing and Greeking Using Trees

Tree pricing schemes are natural in finance because of their Markov chain interpretation as discrete time pricing models. From a practical point of view, trees are often rather obsolete as compared with more sophisticated finite difference or finite element technologies. However, in a number of situations, they remain an adequate and simple alternative. Moreover, from the theoretical point of view, the Markov chain interpretation underlies interesting probabilistic convergence proofs of the related (deterministic) pricing schemes.

Last but not least, trees have a unifying feature for all the pricing and greeking schemes considered so far in these notes. Indeed, there is no hermetic frontier between deterministic and stochastic pricing schemes. In essence, all these numerical schemes are based on the idea of propagating the solution, starting from a surface of the time-space domain on which it is known (typically: the maturity of a claim), along suitable (random) "characteristics" of the problem. Here "characteristics" refers to Riemann's method for solving hyperbolic first-order equations (Morton and Mayers, 2005, Chapter 4). From the point of view of control theory, all these numerical schemes can be viewed as variants of Bellman (1966)'s dynamic programming principle. Monte Carlo pricing schemes may thus be regarded as one-time-step multinomial trees, converging to a limiting jump diffusion when the number of space discretization points (tree branches) goes to infinity. The difference between a tree method in the usual sense and a Monte Carlo method is that a Monte Carlo computation mesh is stochastically generated and nonrecombining. As a converse to the left inclusion in III.(4), explicit (as obvious) but also implicit (after inversion of the involved linear systems) finite difference schemes can also be viewed as multinomial (but recombining) trees.

§4 will conclude thus chapter by a preliminary conclusion on numerical methods in finance as per Chapters III–V, before we move on to numerical methods also involving optimization (training or calibration) procedures in later chapters.

§1 Markov Chain Approximation of Jump-Diffusions

Let X^h represent a continuous-time Markov chain approximation for an $\mathbb{R}^d$-valued jump-diffusion X with infinitesimal generator $\mathcal{A}_x$ as per IX.(28)-(41) (in the compound Poisson driven jumps setup of IX.§3.A). Convergence in law of X^h to X is essentially equivalent to having, for every test-function φ,

$$\lim_{h\to 0} h^{-1}\left(\mathbb{E}_t\varphi(X^h_{t+h}) - \varphi(X^h_t)\right) = \mathcal{A}_x\varphi(x), \tag{1}$$

on every random set $\{\lim_{h\to 0} X^h_t = x\}$. See Kushner and Dupuis (1992), Ethier and Kurtz (1986) and

Jacod and Shiryaev (2003) for the related mathematical theory.

A Kushner's Theorem

Convergence in law of processes implies convergence of prices of European and American options in the approximating Markov chains to their counterparts in the limiting jump-diffusion model X. However, for establishing convergence in law, checking (1) for every test-function φ is not practical. Kushner's theorem reduces much of the burden to verification that in the limit the first two conditional moments of X^h match those of X, so that[1]

$$\begin{array}{l} \lim_{h\to 0} h^{-1}\mathbb{E}_t\left(X^h_{t+h}-X^h_t\right)=\lim_{h\to 0} h^{-1}\mathbb{E}_t\left(X_{t+h}-X_t\right) \\ \qquad =b(t,x)+\lambda(t,x)\int_{\mathbb{R}^d}\delta(t,x,y)\pi(t,x,dy) \\ \lim_{h\to 0} h^{-1}\mathbb{C}\mathrm{ov}_t\left(X^h_{t+h}-X^h_t\right)=\lim_{h\to 0} h^{-1}\mathbb{C}\mathrm{ov}_t\left(X_{t+h}-X_t\right) \\ \qquad =a(t,x)+\lambda(t,x)\int_{\mathbb{R}^d}\delta\delta^\top(t,x,y)\pi(t,x,dy) \end{array}.$$

Proposition 1 *Convergence in law of X^h to X implies that, i.e. on every random set $\{\lim_{h\to 0} X^h_t = x\}$,*

$$\lim_{h\to 0} h^{-1}\mathbb{E}_t(X^h_{t+h}-X^h_t)=b(t,x)+\lambda(t,x)\int_{\mathbb{R}^d}\delta(t,x,y)\pi(t,x,dy) \tag{2}$$

$$\lim_{h\to 0} h^{-1}\mathbb{C}\mathrm{ov}_t(X^h_{t+h}-X^h_t)=a(t,x)+\lambda(t,x)\int_{\mathbb{R}^d}\delta\delta^\top(t,x,y)\pi(t,x,dy). \tag{3}$$

Proof. For every component i and j of X, setting $\varphi=\varphi(x)=\pi^i(x):=x_i$ and $\varphi=\pi^i\pi^j$ in (1) yields respectively that, on $\{\lim_{h\to 0} X^h_t = x\}$,

$$\begin{array}{l} \lim_{h\to 0} h^{-1}\mathbb{E}_t(X^{h,i}_{t+h}-X^{h,i}_t)=\mathcal{A}_x\pi^i(x)= \\ \qquad b_i(t,x)+\lambda(t,x)\int_{\mathbb{R}^d}\delta_i(t,x,y)\pi(t,x,dy) \\ \lim_{h\to 0} h^{-1}\mathbb{E}_t\left(X^{h,i}_{t+h}X^{h,j}_{t+h}-X^{h,i}_tX^{h,j}_t\right)=\mathcal{A}_x(\pi^i\pi^j)(x). \end{array} \tag{4}$$

The first part in (4) is (2). Moreover, for fixed h, we have that

$$\begin{array}{l} \mathbb{C}\mathrm{ov}_t(X^{h,i}_{t+h}-X^{h,i}_t,X^{h,j}_{t+h}-X^{h,j}_t)+\mathbb{E}_t(X^{h,i}_{t+h}-X^{h,i}_t)\mathbb{E}_t(X^{h,j}_{t+h}-X^{h,j}_t) \\ =\mathbb{E}_t\left(X^{h,i}_{t+h}X^{h,j}_{t+h}-X^{h,i}_tX^{h,j}_t\right)-X^{h,i}_t\mathbb{E}_t\left(X^{h,j}_{t+h}-X^{h,j}_t\right) \\ \qquad -X^{h,j}_t\mathbb{E}_t\left(X^{h,i}_{t+h}-X^{h,i}_t\right). \end{array}$$

So on $\{\lim_{h\to 0} X^h_t = x\}$, $\lim_{h\to 0} h^{-1}\mathbb{E}_t(X^{h,i}_{t+h}-X^{h,i}_t)\mathbb{E}_t(X^{h,j}_{t+h}-X^{h,j}_t)=0$, by (2), and

$$\begin{array}{l} \lim_{h\to 0} h^{-1}\mathbb{C}\mathrm{ov}_t(X^{h,i}_{t+h}-X^{h,i}_t,X^{h,j}_{t+h}-X^{h,j}_t)= \\ \qquad \mathcal{A}_x(\pi^i\pi^j)(x)-x_i\mathcal{A}_x\pi^j(x)-x_j\mathcal{A}_x\pi^i(x), \end{array}$$

where, by application of IX.(29)-(30)-(41), the right-hand side coincides with

$$a_{i,j}(t,x)+\lambda(t,x)\int_{\mathbb{R}^d}\delta_i\delta_j(t,x,y)\pi(t,x,dy).$$

This yields (3). ∎

Conversely, Kushner and Dupuis (1992, Theorem 4.1 page 290)'s theorem states that the so-called local consistency conditions assure the convergence in law of X^h to X. In the diffusion case these conditions are just (2) with $\lambda=0$ there , i.e. Kushner and Dupuis (1992, Equations (4.1.3) page 71). See Kushner and Dupuis (1992, Equations (5.6.6) page 129) for the statement of the local consistency conditions in the general jump diffusion case.

In the next sections we discuss basic tree pricing schemes in the setup of the Black-Scholes model, i.e.

$$S_t=S_0e^{bt+\sigma W_t},$$

with $\kappa=r-q$ and $b=\kappa-\frac{1}{2}\sigma^2$, for a constant risk-free rate r and a constant dividend yield q on S.

[1] IX.(29)-(30)-(41).

§2 Trees for Vanilla Options

A Cox-Ross-Rubinstein Binomial Tree

The Cox, Ross, and Rubinstein (1979) tree is the following Markov chain approximation to Black-Scholes, parameterized by two positive constants $0 < \mathrm{d} < \mathrm{u}$: $S_0^h = S_0$ and, for $i = 0, \ldots, n-1$,

$$S^h_{(i+1)h} = \mathrm{u}\, S^h_{ih} \text{ (resp. } \mathrm{d}\, S^h_{ih}) \text{ with probability } p \text{ (resp. } 1-p).$$

Here $h = \frac{T}{n}$ where T is the maturity of an option with payoff function ϕ, n is the number of time steps in the tree and

$$\mathrm{u} = e^{\sigma\sqrt{h}},\ \mathrm{d} = e^{-\sigma\sqrt{h}},\ p = \tfrac{e^{\kappa h} - d}{\mathrm{u} - \mathrm{d}}. \tag{5}$$

We find it convenient to denote the time in the tree by i rather than ih. In particular, $S_i^h \equiv S_{ih}^h$, $\mathbb{E}_i$ refers to the conditional expectation with respect to the σ-field $\mathfrak{F}_i^h$ generated by $(S_0^h, \ldots, S_i^h)$ and $\mathcal{T}_i^h$, also with $\mathcal{T}_0^h = \mathcal{T}^h$, represents the set of $\mathfrak{F}^h$-stopping times ν with values in $\{i, \ldots, n\}$. The following Proposition shows that the Cox-Ross-Rubinstein tree model shares the main hedging properties of the Black-Scholes model.

Proposition 2 (i) *In the European vanilla case of a payoff $\phi(S_n^h)$, the process given for $i = 0, \ldots, n$ by*

$$\Pi_i^h := e^{-r(T-i)h}\mathbb{E}_i\phi(S_n^h) = u_i(S_i^h) \tag{6}$$

is the unique replication price process for the option, with an associated replication strategy given by

$$\delta_i^h = \frac{u_{i+1}(\mathrm{u}S_i^h) - u_{i+1}(dS_i^h)}{(\mathrm{u}-\mathrm{d})S_i^h}; \tag{7}$$

here the European pricing function u is defined by

- *$u_n(S) = \phi(S)$ for every S in the tree at time n,*
- *for $i = n-1, \ldots, 0$,*
$$u_i(S) = e^{-rh}\left[pu_{i+1}(\mathrm{u}S) + (1-p)u_{i+1}(\mathrm{d}S)\right] \tag{8}$$
for every S in the tree at time i.

(ii) *In the American vanilla case of a payoff process $(\phi(S_i^h))$, the minimal superhedging price of the option is given for $i = 0, \ldots, n$ by:*

$$\widetilde{\Pi}_i^h := \max_{\nu\in\mathcal{T}_i^h} \mathbb{E}_i\left(e^{-r(\nu-i)h}\phi(S_\nu^h)\right) = v_i(S_i^h), \tag{9}$$

with a related superhedging strategy defined as

$$\widetilde{\delta}_i^h = \frac{v_{i+1}(\mathrm{u}S_i^h) - v_{i+1}(dS_i^h)}{(\mathrm{u}-\mathrm{d})S_i^h}; \tag{10}$$

here the American pricing function v is defined by

- *$v_n(S) = \phi(S)$ for every S in the tree at time n,*
- *for $i = n-1, \ldots, 0$,*
$$v_i(S) = \max\left(\phi(S), e^{-rh}\left[pv_{i+1}(\mathrm{u}S) + (1-p)v_{i+1}(\mathrm{d}S)\right]\right) \tag{11}$$
for every S in the tree at time i.

Proof. We assume $r = q = 0$ for notational simplicity.

Case $n = 1$. First considering a European option with payoff function $\phi(S_1^h)$, let (Y_0, Z_0) denote the solution to the following equation (one time-step backward stochastic difference equation):

$$Y_0 = \phi(S_1^h) - Z_0(S_1^h - S_0), \text{ a.s.} \tag{12}$$

Note that (12) is equivalent to the following algebraic system of two equations in the two unknown numbers Y_0, Z_0 :

$$\begin{cases} Y_0 = \phi(\mathrm{u}S_0) - Z_0(u-1)S_0 \\ Y_0 = \phi(\mathrm{d}S_0) - Z_0(d-1)S_0, \end{cases}$$

which is well-posed[2], since $d < u$. By construction, the price-and-hedge (Y_0, Z_0) replicates the payoff $\phi(S_1^h)$ and we have that

$$Z_0 = \delta_0^h = \frac{\phi(\mathrm{u}S_0) - \phi(\mathrm{d}S_0)}{(\mathrm{u}-\mathrm{d})S_0}.$$

Moreover, the definition of p is such that $\mathbb{E}(S_1^h - S_0) = 0$ and therefore $Y_0 = \mathbb{E}\phi(S_1^h)$.

If the option is American, let (Y_0, Z_0, α_0) denote the solution to the following equation (one time-step reflected backward stochastic difference equation):

$$\begin{cases} Y_0 = \phi(S_1^h) + \alpha_0 - Z_0(S_1^h - S_0) \\ Y_0 \geq \phi(S_0)\,,\ \alpha_0 \geq 0\,,\ (Y_0 - \phi(S_0))\alpha_0 = 0. \end{cases} \tag{13}$$

By inspection we see, distinguishing the two cases $\phi(S_0) \leq$ and $> \mathbb{E}\phi(S_1^h)$, that the unique solution is

$$(Y_0, Z_0, \alpha_0) \;=\; \big(\max(\phi(S_0), \mathbb{E}\phi(S_1^h)),\ \delta_0^h,\ Y_0 - \mathbb{E}\phi(S_1^h)\big).$$

By construction, the price-and-hedge (Y_0, Z_0) superhedges the American payoff $\phi(S^h)$ for every holder stopping policy $\nu \in \mathcal{T}^h = \{\{0\}, \{1\}\}$ (for $n = 1$). Moreover, for every superhedging strategy $(\widetilde{Y}_0, \widetilde{Z}_0)$, we have that $\widetilde{Y}_0 \geq \phi(S_0)$ and $\widetilde{Y}_0 + \widetilde{Z}_0(S_1^h - S_0) \geq \phi(S_1^h)$ (to superhedge the payoffs respectively due in the $\nu = \{0\}$ and $\nu = \{1\}$ cases). Hence the inequality

$$\widetilde{Y}_0 \geq \max(\phi(S_0), \mathbb{E}\phi(S_1^h)) = Y_0$$

results. Therefore $Y_0 = \max(\phi(S_0), \mathbb{E}\phi(S_1^h))$ is the minimal initial wealth of an issuer superhedging strategy.

General Case. Applying the above results step-by-step backwards in the tree, we deduce that the European price process defined by $u_i(S_i^h)$, along with the hedging strategy (7) for the pricing function u^h defined by (8), replicates the option payoff $\phi(S_n^h)$ at time T. This also yields the probabilistic representation of $u_i(S_i^h)$ stated in (6).

Likewise, the American price process defined by $v_i(S_i^h)$ along with the hedge (10) for the pricing function v^h defined by (11), is a minimal superhedging strategy for the American option with payoff function ϕ. In view of (11), $(v_i(S_i^h))_{0\leq i\leq n}$ is a supermartingale dominating $(\phi(S_i^h))_{0\leq i\leq n}$, which implies the $\leq$ inequality in the probabilistic representation of $v_i(S_i^h)$ stated in (9). Moreover, let ν^i in $\mathcal{T}_i^h$ be defined as

$$\nu^i = \inf\{j \geq i;\ v_j(S_j^h) = \phi(S_j^h)\}. \tag{14}$$

Process $(v_i(S_i^h))$ is a martingale on the random time interval $\{i, \ldots, \nu^i\}$ in the sense that if $i \leq j < \nu^i$, then

$$v_j(S_j^h) = \mathbb{E}_j v_{j+1}(S_{j+1}^h),$$

[2]Unless $\frac{\phi(\mathrm{u}S_0)}{\phi(\mathrm{d}S_0)} = \frac{u-1}{d-1}$, in which case the system admits an infinity of solutions.

by the definition (11) of v and the definition (14) of ν^i. Therefore, by this martingale property of $(v_i(S_i^h))_{i\leq j<\nu^i}$, we have

$$v_i(S_i^h) = \mathbb{E}_i v_{\nu^i}(S_{\nu^i}^h) = \mathbb{E}_i \phi(S_{\nu^i}^h),$$

where the second equality holds by the definition (14) of ν^i. Therefore ν^i achieves the maximum in (9). ∎

Observe that not only these results, but also their proofs, are essentially the same as in the Black-Scholes model. One can also check that Proposition 2 holds independently of the exact definition of u and d, provided $0 < \mathrm{d} < \mathrm{u}$[3].

Cox–Ross–Rubinstein Algorithm The Cox–Ross–Rubinstein algorithm consists in a backward computation of the option price based on one of the dynamic programming equations (8) and (11) subsequent to a forward computation of the $n+1$ possible values of S_n^h. Note that, since $\mathrm{u} = 1/\mathrm{d}$, the corresponding tree is recombining in the sense that $S = \mathrm{ud}S = \mathrm{du}S$. In particular, there are only $2n+1$ possible values of the underlying in the tree between time 0 and time n.

A.1 Convergence in Law of Processes

For u and d defined by (5), the Cox-Ross-Rubinstein tree converges in various senses to the related Black-Scholes model as $h \to 0$. Convergence in law of processes thus follows, by an application of Kushner's theorem, for X^h of §1 taken as a suitable continuous-time Markov chain interpolation[4] of the Cox-Ross-Rubinstein stock process S^h (or log-returns process $\ln(S^h)$).

Convergence in law of the one-dimensional marginal $\ln(S_n^h)$ to $\ln(S_T)$ can also be established by characteristic function arguments. Assume $S_0 = 1$ for notational simplicity. Letting $\mathrm{i}^2 = -1$, for every real z, we have:

$$\begin{aligned}\mathbb{E}\left[\exp\left(\mathrm{i}z\ln S_n^h\right)\right] &= \mathbb{E}\left[\exp\left(\mathrm{i}z\ln\prod_{j=0}^{n-1}\frac{S_{j+1}^h}{S_j^h}\right)\right]\\ &= \left(\mathbb{E}\left[\exp\left(\mathrm{i}z\ln S_1^h\right)\right]\right)^n\\ &= \left(p\exp\left(\mathrm{i}z\sigma\sqrt{h}\right) + (1-p)\exp\left(-\mathrm{i}z\sigma\sqrt{h}\right)\right)^n.\end{aligned} \tag{15}$$

We compute

$$p = \frac{e^{\kappa h}-\mathrm{d}}{\mathrm{u}-\mathrm{d}} = \frac{e^{\kappa h}-e^{-\sigma\sqrt{h}}}{e^{\sigma\sqrt{h}}-e^{-\sigma\sqrt{h}}} = \frac{\kappa h+\sigma\sqrt{h}-\frac{1}{2}\sigma^2 h+\mathrm{O}(h^{\frac{3}{2}})}{2\sigma\sqrt{h}+\mathrm{O}(h^{\frac{3}{2}})} = \frac{1}{2}+\frac{b\sqrt{h}}{2\sigma}+\rho(h),$$

[3] With $e^{\kappa h}\in[\mathrm{d},\mathrm{u}]$, in order to obtain that $p = \frac{e^{\kappa h}-\mathrm{d}}{\mathrm{u}-\mathrm{d}} \in [0,1]$; otherwise p only defines a signed probability measure.
[4] See 4.3 of Kushner and Dupuis Kushner and Dupuis (1992).

where $\rho(h) = \mathrm{O}(h)$. Hence

$$p\exp\left(\mathrm{i}z\sigma\sqrt{h}\right) + (1-p)\exp\left(-\mathrm{i}z\sigma\sqrt{h}\right) \tag{16}$$

$$= \left(\frac{1}{2} + \frac{b\sqrt{h}}{2\sigma} + \rho(h)\right)\left(1 + \mathrm{i}z\sigma\sqrt{h} - \frac{1}{2}\sigma^2 z^2 h + \mathrm{o}(h)\right) \tag{17}$$

$$+ \left(\frac{1}{2} - (\frac{b\sqrt{h}}{2\sigma} + \rho(h))\right)\left(1 - \mathrm{i}z\sigma\sqrt{h} - \frac{1}{2}\sigma^2 z^2 h + \mathrm{o}(h)\right)$$

$$= 1 - \frac{1}{2}\sigma^2 z^2 h + \left(\frac{b\sqrt{h}}{2\sigma} + \rho(h)\right) 2\mathrm{i}z\sigma\sqrt{h} + \mathrm{o}(h)$$

$$= 1 + ibzh - \frac{1}{2}\sigma^2 z^2 h + \mathrm{o}(h) \tag{18}$$

$$= 1 + \left(\mathrm{i}zb - z^2\frac{\sigma^2}{2}\right)\frac{T}{n} + \mathrm{o}(\frac{1}{n}). \tag{19}$$

Therefore letting $h \to 0$ in (15) yields by Montel's theorem[5]

$$\mathbb{E}\left[\exp\left(\mathrm{i}z\ln S_n^h\right)\right] \sim \left(1 + \frac{\left(\mathrm{i}zb - z^2\sigma^2/2\right)T}{n}\right)^n$$
$$\to \exp\left(\left(\mathrm{i}zb - z^2\sigma^2/2\right)T\right) = \mathbb{E}\left[\exp\left(\mathrm{i}z\ln(S_T)\right)\right],$$

as $\ln(S_T) \sim \mathcal{N}(bT, \sigma^2 T)$.

One can even establish convergence in law of processes of (a suitable continuous-time interpolation of) the Cox-Ross-Rubinstein log-returns process $\ln(S^h)$ to its Black-Scholes counterpart $\ln(S_\cdot)$.

Time-0 Prices and Deltas Convergence in law of processes grants the convergence of prices of European and American vanilla options with integrable payoffs, e.g. call and put options. As for the deltas, in the European case, (7) yields

$$\begin{aligned} S_0\delta_0^h &= \frac{u_1(\mathrm{u}S_0) - u_1(dS_0)}{(\mathrm{u}-\mathrm{d})} \\ &= e^{-r(n-1)h}\frac{\mathbb{E}\left[\phi\left(\mathrm{u}S_0\xi^h\right)\right] - \mathbb{E}\left[\phi\left(\mathrm{d}S_0\xi^h\right)\right]}{\mathrm{u}-\mathrm{d}} \end{aligned}$$

for some random variable ξ^h. Assuming that ϕ is of class $\mathcal{C}^1$, then

$$\phi\left(\mathrm{u}S_0\xi^h\right) - \phi\left(\mathrm{d}S_0\xi^h\right) = \int_d^u S_0\xi^h\phi'\left(xS_0\xi^h\right)dx,$$

so that

$$\begin{aligned} S_0\delta_0^h &= \frac{e^{-r(n-1)h}}{(\mathrm{u}-\mathrm{d})}\int_d^u \mathbb{E}\left[S_0\xi^h\phi'\left(xS_0\xi^h\right)\right]dx \\ &= e^{-r(n-1)h}\mathbb{E}\left[S_0\xi^h\phi'\left(x^hS_0\xi^h\right)\right] \end{aligned}$$

for some $x^h \in [\mathrm{d}, \mathrm{u}]$, by the mean value property. Assuming further that the limit $y \mapsto \psi(y) = y\phi'(y)$ is Lipschitz and bounded, then

$$\left|S_0\xi^h\phi'\left(x^hS_0\xi^h\right) - \psi\left(S_0\xi^h\right)\right| = \left|\frac{1}{x^h}\psi\left(x^hS_0\xi^h\right) - \psi\left(S_0\xi^h\right)\right|$$
$$= \left|(\frac{1}{x^h} - 1)\psi\left(x^hS_0\xi^h\right) + \psi\left(x^hS_0\xi^h\right) - \psi\left(S_0\xi^h\right)\right|$$
$$\leq c\left(\left|\frac{1}{x^h} - 1\right| + S_0\xi^h\left|x^h - 1\right|\right),$$

[5] $(1 + \frac{y}{n})^n$ converges to e^y locally uniformly.

for some constant c. Hence

$$\lim_{h\to 0}\mathbb{E}\left[S_0\xi^h\phi'\left(x^hS_0\xi^h\right)\right]=\lim_{h\to 0}\mathbb{E}\left[\psi\left(S_0\xi^h\right)\right]. \tag{20}$$

Moreover, by convergence in law of S_n^h to S_T, we have

$$\lim_{h\to 0}\mathbb{E}\left[\psi\left(S_0\xi^h\right)\right]=\mathbb{E}\left[\psi\left(S_T\right)\right]=\mathbb{E}\left[S_T\phi'\left(S_T\right)\right]=e^{rT}S_0\Delta_0^{bs},$$

where Δ_0^{bs} denotes the Black-Scholes delta of the option at time 0, because

$$\begin{aligned}e^{rT}\Delta_0^{bs}&=e^{rT}\partial_{S_0}\Pi_0^{bs}=\partial_{S_0}\mathbb{E}\phi(S_T)=\partial_{S_0}\mathbb{E}\phi(S_0e^{bT+\sigma W_T})\\&=\mathbb{E}\left[\phi'\left(S_0e^{bT+\sigma W_T}\right)e^{bT+\sigma W_T}\right]=\mathbb{E}\left[\phi'\left(S_T\right)\frac{S_T}{S_0}\right].\end{aligned}$$

This shows that, for bounded and regular payoff functions, the Cox–Ross–Rubinstein delta δ_0^h of a European option converges towards the Black-Scholes delta as $h\to 0$. This result can be extended to a vanilla put payoff by density, and then to a vanilla call payoff by call–put parity.

Convergence of deltas also holds for American options, although this can't be proved by elementary computations as in the European case.

B Other Binomial Trees

To achieve convergence in law, many other choices of u, d and p can be done. The limiting law of S_n^h only depends on $pe^{i\lambda\ln(\mathrm{u})}+(1-p)\,e^{i\lambda\ln(\mathrm{d})}$ through its Taylor expansion up to $\mathrm{o}(h)$. Thus u, d or/and p can be altered as long as the first order terms of the development are not modified. Also note that a binomial tree is recombining as soon as u and d remain constant within the tree.

B.1 Random Walk Scheme

A natural idea consists in approximating the Brownian motion W in Black-Scholes by a random walk approximation, leading to

$$\mathrm{u}=e^{bh+\sigma\sqrt{h}},\ \mathrm{d}=e^{bh-\sigma\sqrt{h}},\ p=\frac{1}{2}.$$

B.2 Matching Three Moments Scheme

Here the idea is to match the first three conditional moments of the approximating Markov chain with those of the Black-Scholes model. We thereby obtain the following equations in u, d, p :

$$\begin{aligned}p\mathrm{u}+(1-p)\,\mathrm{d}&=e^{\kappa h}\\p\mathrm{u}^2+(1-p)\,d^2-e^{2\kappa h}&=e^{2\kappa h}\left(e^{\sigma^2h}-1\right)\\p\mathrm{u}^3+(1-p)\,d^3&=e^{3\kappa h}e^{3\sigma^2h},\end{aligned}$$

so that

$$\begin{aligned}\mathrm{u}&=\frac{e^{\kappa h}\rho}{2}\left[1+\rho+\sqrt{\rho^2+2\rho-3}\right]\\\mathrm{d}&=\frac{e^{\kappa h}\rho}{2}\left[1+\rho-\sqrt{\rho^2+2\rho-3}\right]\\p&=\frac{e^{\kappa h}-\mathrm{d}}{\mathrm{u}-\mathrm{d}},\end{aligned}$$

with $\rho=e^{\sigma^2h}$.

C Kamrad–Ritchken Trinomial Tree

The Kamrad–Ritchken tree Kamrad and Ritchken (1991) is a trinomial tree with $2n+1$ possible values of the underlying S throughout the option life. It consists of a symmetric 3-point approximation with space step k to $X = \ln(S)$, with up and down probabilities p_+ and p_- such that

$$k\,(p_+ - p_-) = bh$$
$$k^2\,(p_+ + p_-) = \sigma^2 h$$

and with forward probability $p = 1 - p_+ - p_-$. Equivalently, in terms of the so-called stretch parameter λ defined by $k = \lambda\sigma\sqrt{h}$,

$$p_- = \frac{1}{2\lambda^2} - \frac{b\sqrt{h}}{2\lambda\sigma}\,,\ p\ = 1 - \frac{1}{\lambda^2}\,,\ p_+ = \frac{1}{2\lambda^2} + \frac{b\sqrt{h}}{2\lambda\sigma}.$$

The stretch parameter λ is a free parameter of the geometry of the tree. It must satisfy $\lambda \geq 1$ to ensure "nonnegativity of the probabilities". The value $\lambda = 1.22474$, which corresponds to $p = \frac{1}{3}$, is reported to be a good choice for pricing an at-the-money call or put.

Note that the Kamrad–Ritchken tree essentially coincides with the explicit finite difference scheme of III.§3.B.1, except for a slightly different treatment of the discount factor (the ru term in the Black–Scholes equation).

D Multinomial Trees

In a generic multinomial tree with geometry and dynamics defined by "up" factors u_j and corresponding probabilities p_j (possibly parameterized by and then implicitly dependent on a timestep h), the algorithms for pricing European and American options are the backward schemes defined by :

- $u_n(S) = v_n(S) = \phi(S)$ for every S in the tree at time n,
- for $i = n-1, \ldots, 0$ for every S in the tree at time i:

$$u_i(S) = e^{-rh} \sum p_j u_{i+1}(\mathrm{u}_j S)$$
$$v_i(S) = \max\left(\phi\,(S)\,,\ e^{-rh} \sum p_j v_{i+1}(\mathrm{u}_j S)\right).$$

The equation ensuring convergence in law of the characteristic functions, and therefore of the one-dimensional marginals of S, is written[6]: for every real z,

$$\sum p_j \exp\left(\mathrm{i}z \ln \mathrm{u}_j\right) = 1 + \left[\mathrm{i}zb - z^2\tfrac{\sigma^2}{2}\right] h + \mathrm{o}(h),$$

i.e. with $X = \ln(S)$

$$\Phi^h(z) := \mathbb{E} \exp\left(\mathrm{i}z(X_1^h - X_0^h)\right) = 1 + \left[\mathrm{i}zb - z^2\frac{\sigma^2}{2}\right] h + \mathrm{o}(h). \tag{21}$$

Hence (assuming $\partial_z \mathrm{o}(h)$ and $\partial^2_{z^2}\mathrm{o}(h)$, for $\mathrm{o}(h)$ in (21), are also negligible with respect to h when $h \to 0$)

$$\mathbb{E}(X_1^h - X_0^h) = -\mathrm{i}\partial_z \Phi^h(0) = bh + \mathrm{o}(h)$$
$$\mathbb{E}\left[(X_1^h - X_0^h)^2\right] = -\,\partial^2_{z^2}\Phi^h(0) = \sigma^2 h + \mathrm{o}(h),$$

So the Kushner local consistency conditions are satisfied and convergence in law also holds at the level of processes.

Note that, in order to get a recombining tree, $\mathrm{u}_{j+1}/\mathrm{u}_j$ must not depend on j. Otherwise the complexity of the scheme is combinatorially exploding with n.

[6] cf. (15)-(19).

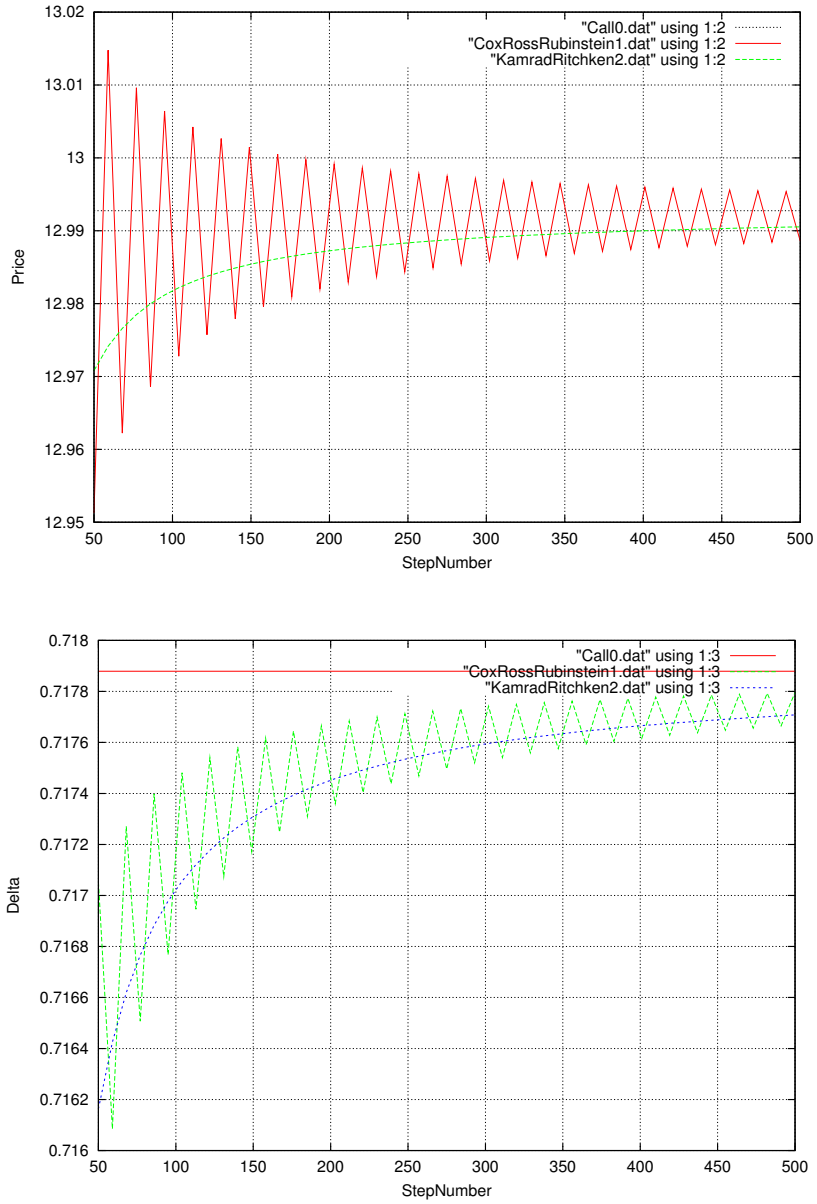


Figure 1: *European vanilla call in the Black-Scholes model, priced in a Cox–Ross–Rubinstein binomial tree and in a Kamrad–Ritchken trinomial tree.*

§3 Trees for Exotic Options

A Barrier Options

We now consider a down-and-out call option with a constant rebate R at the barrier level L (see IV.§8.B). The first idea for pricing this option within a Cox–Ross–Rubinstein tree is to apply the usual backward induction scheme, with an option price at or above the barrier constrained to R. It is possible to show that the resulting price converges to the right Black-Scholes limit. However, the convergence is slower than for vanilla options. To understand why, let us denote by l the tree node index such that

$$S_0 \mathrm{d}^l \geq L > S_0 \mathrm{d}^{l+1}.$$

For fixed n, the algorithm yields the same result for every value of the barrier between $S_0\mathrm{d}^l$ and $S_0\mathrm{d}^{l+1}$. Therefore the convergence cannot be faster than

$$\partial_L u^{bs} \left(\mathrm{d}^l - \mathrm{d}^{l+1}\right) = \mathrm{O}(h^{\frac{1}{2}}),$$

where u^{bs} denotes the Black-Scholes pricing function of the barrier option. For comparison, the convergence rate in case of a European vanilla option is $\mathrm{O}(h)$[7].

An alternative method is to determine the stretch parameter λ of a trinomial Kamrad–Ritchken tree (see §2.C) so that the barrier is hit exactly. Recall that λ must be greater than one in order to ensure stability of the scheme. One thus sets $\lambda = \frac{1}{m}\frac{\ln\left(\frac{S_0}{L}\right)}{\sigma\sqrt{h}}$, with $m = \left\lfloor \frac{\ln\left(\frac{S_0}{L}\right)}{\sigma\sqrt{h}} \right\rfloor$. For this choice of λ convergence is reported to be as fast as for vanilla options.

B Bermudan Options

Bermudan options can only be exercised at specific times. Let us consider the case of an option putable on $[T_1, T]$ for some fixed T_1 in $(0, T)$. A convergent Cox–Ross–Rubinstein algorithm for pricing this option consists of the "American" backward induction formula (11) between step $n-1$ and n_1, followed by the "European" backward induction formula (8) before n_1, where $(n_1 - 1)\,h < T_1 \leq n_1 h$. But this algorithm is very crude since it gives the same price, n being fixed, for every value of T_1 between $(n_1 - 1)\,h$ and $n_1 h$.

A better algorithm consists of two Kamrad–Ritchken trees pasted together at time T_1, i.e.:

- a first tree with stretch parameter λ_1 and number of time steps n_1 between times 0 and T_1, and
- another tree with stretch parameter λ_2 and number of time steps n_2 between times T_1 and T.

In order to get a recombining tree, we impose the following pasting condition at T_1 :

$$\lambda_1 \sqrt{\frac{T_1}{n_1}} = \lambda_2 \sqrt{\frac{T - T_1}{n_2}}.$$

For instance, we can first fix $\lambda_1 \geq 1$ (e.g., $\lambda_1 = 1.2274$) and n_1, and then set

$$n_2 = \left[\frac{n_1\,(T - T_1)}{T_1}\right] + 1.$$

Thus $n_2 T_1 \geq n_1\,(T - T_1)$ and $\lambda_2^2 = \lambda_1^2 \frac{T_1}{n_1 (T - T_1)}\, n_2 \geq \lambda_1^2 \geq 1$.

[7] see III.§3.B.1.

C Cox-Ross-Rubinstein Tree for Lookback Options

Until now we have only considered univariate trees with complexity $O(n^2)$. For the applications that remain, one needs to consider bivariate trees of complexity $O(n^3)$.

We first consider a lookback option with payoff $\phi(S_T, M_T)$, where $M_t = \sup_{0\le s\le t} S_s$. We can price this option in a bivariate tree $(S_i^h, M_i^h)_{0\le i\le n}$, where M^h corresponds to the running maximum of a Cox–Ross–Rubinstein stock S^h. The related dynamic programming equation is written as follows:

- $u_n(S, M) = \phi(S, M)$ for every S in the Cox-Ross-Rubinstein tree at time n and M in the related lattice of all possible values of M_n^h,
- for $i = n-1, \dots, 0$, for every S in the Cox-Ross-Rubinstein tree at time i and M in the related lattice of all possible values of M_i^h we have

$$u_i(S, M) = e^{-rh}\left[p u_{i+1}(\mathrm{u}S, \max(\mathrm{u}S, M)) + (1-p) u_{i+1}(\mathrm{d}S, M)\right], \tag{22}$$

having used that $\mathrm{d}S_{i+1}^h \le M_i^h$ since $\mathrm{d} = e^{-\sigma\sqrt{h}} < 1$.

D Kamrad–Ritchken Tree for Options on Two Assets

We now assume a bivariate Black-Scholes model. So, for $l = 1, 2$,

$$\mathrm{d}S_t^l = \kappa_l S_t^l dt + \sigma_l S_t^l dW_t^l,$$

where $\kappa_l = r - q_l$ and (W^1, W^2) is a bivariate Brownian motion with correlation ρ. Or, in log-returns for $l = 1, 2$,

$$dX_t^l = b_l dt + \sigma_l dW_t^l$$

with $b_l = \kappa_l - \frac{\sigma_l^2}{2}$. To approximate X we can use the bivariate Markov chain (X_i^h) such that $X_0^h = X_0$. Then for every $i = n-1, \dots, 0$ and for $l = 1, 2$,

$$X_{i+1}^{h,l} = X_i^{h,l} + \left(\kappa_l - \frac{\sigma_l^2}{2}\right) h + \sigma_l \sqrt{h}\varepsilon_i^l,$$

for i.i.d. $(\varepsilon_i^1, \varepsilon_i^2)$ such that

$$\mathbb{Q}\left(\varepsilon_0^1 = 1, \varepsilon_0^2 = 1\right) = \mathbb{Q}\left(\varepsilon_0^1 = -1, \varepsilon_0^2 = -1\right) = \tfrac{1+\rho}{4}$$
$$\mathbb{Q}\left(\varepsilon_0^1 = 1, \varepsilon_0^2 = -1\right) = \mathbb{Q}\left(\varepsilon_0^1 = -1, \varepsilon_0^2 = 1\right) = \tfrac{1-\rho}{4}.$$

We can easily check that

$$\mathbb{E}\varepsilon_0^l = 0\,,\ \mathbb{V}\mathrm{ar}\varepsilon_0^l = 1 \text{ and } \mathbb{C}\mathrm{ov}(\varepsilon_0^1, \varepsilon_0^2) = \mathbb{E}(\varepsilon_0^1\varepsilon_0^2) = 2\left(\frac{1+\rho}{4} - \frac{1-\rho}{4}\right) = \rho$$

and therefore

$$\lim_{h\to 0} h^{-1}\mathbb{E}_i\left(X_{i+1}^h - X_i^h\right) = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$
$$\lim_{h\to 0} h^{-1}\mathbb{C}\mathrm{ov}_i\left(X_{i+1}^h - X_i^h\right) = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}.$$

Thus Kushner's local consistency conditions are satisfied and (a suitable continuous-time Markov chain interpolation[8] of) X^h converges in law to X.

[8] see Section 4.3 of Kushner and Dupuis (1992).

§4 Numerical Solutions: Synthesis and Perspectives

For simple products in simple models, analytic exact or approximation formulas are available. These are very useful at the stage of model calibration (Chapter VII), which involves intensive pricing of vanilla options. However, as far as pricing exotics or dealing with less standard models is concerned, the pricing equations have to be solved numerically. To a large extent these notes deal with the numerical solution of the pricing equations by stochastic simulation methods (Chapter IV) or, provided the dimension of the model is not too large, by deterministic schemes like trees (Chapter V) or finite differences (Chapter III). In recent years, machine learning techniques have also become useful in this regard (Chapter VI). But the scope of machine learning in finance is of course broader than being instrumental in providing numerical solutions to challenging derivative procing problems. It also (actually, primarily) includes metamodeling from real data, even if the non-stationarity of financial data then raises particular challenges (Chapter VIII).

In Chapters III–V we presented reference numerical schemes in simple models, most often in Black-Scholes, although numerical solutions are typically not obligatory in this case since many explicit formulas are available. The explicit formulas are useful to assess the accuracy of the numerical schemes. The numerical schemes themselves are generic so that it is rather straightforward to extend them to arbitrary jump-diffusions.

A Accuracy versus Computational Cost

A typical benchmark of numerical accuracy in computational finance may vary from an order of $10^{-2}\%$ ("one basis point" or bp) to 1% in relative terms, depending on the application. As for computation times, the benchmark also varies greatly with the application, but as far as "real time" option pricing is concerned, "instantaneous" pricing is the target, and more than a few seconds is prohibitive. Now, a solution within a 1bp normalized error by a finite differences ADI method, which is an industry standard today as far as deterministic methods in space dimension 1 to 3 are concerned[9], requires about 300 grid points per space dimension. This yields a computation time in $\mathrm{O}(300^d)$, i.e. a computation time ranging on present day computers from a few milliseconds for $d = 1$ to a few seconds for $d = 3$. In practice, this limits the range of applicability of deterministic pricing schemes to problems in space dimension ≤ 3, unless sophisticated sparse grid or grid refinement techniques are used. This is an effect of Bellman's curse of dimensionality, referring to the fact that the computational cost of numerical integration grows exponentially with space dimension d as m_1^d, where m_1 is the number of discretization points in a generic dimension of the state space.

Table 1 provides a crude comparison of the computational costs of typical Monte Carlo and deterministic schemes (Monte Carlo algorithm with time discretisation of the underlying factor process, cf. IV.§7.D, versus an ADI PDE method). A rough conclusion[10] is that deterministic methods are more efficient in space dimension ≤ 3, but they are often harder to implement. In dimension > 3 Monte Carlo methods are preferred when available. This leads to the dictionary of Table 2 regarding the type

	CPU Time	**Accuracy**	**Memory Cost**
MC	$\mathrm{O}(nm)$	$\mathrm{O}(n^{-1} + m^{-\frac{1}{2}})$	$\mathrm{O}(1)$
PDE	$\mathrm{O}(nm)$	$\mathrm{O}(n^{-1} + m^{-\frac{2}{d}})$	$\mathrm{O}(m)$

Table 1: *Compared computational costs of a Monte Carlo pricing scheme (m simulation runs) versus a deterministic pricing scheme (m_1 mesh points per space dimension, i.e. $m = m_1^d$ space mesh points), in space dimension d; n is the number of points of a time-grid which is used in both schemes.*

of numerical method to be used, depending on the space dimension and on the nature of a pricing

[9] see III.§3.F.1.

[10] we do not detail the constants involved in these computational cost estimates.

problem. In the upper left cell of the Table, the choice between PDE and Monte Carlo should be dictated by the relative interest of performance as compared with implementation and maintainance costs, generally higher with deterministic schemes. In the lower right cell, "Nonlinear MC" refers to nonlinear simulation schemes, which can be used for solving nonlinear control problems. The latter include, in particular, American or more general game option (such as convertible bond) pricing problems, or pricing in an uncertain volatility model in which a stochastic volatility is only known to remain in a positive interval $[\underline{\sigma}, \bar{\sigma}]$ (Avellaneda et al., 1995)—which also relates to the notion of second-order BSDEs (Soner et al., 2012). Nonlinear MC schemes include nonlinear simulation/regression schemes such as those of Sections IV.§9 and VI.§3. They also include purely forward particle simulation schemes (not covered in these notes) based on the theory of branching Markov processes (Henry-Labordère, 2012; Guyon and Henry-Labordère, 2012; Henry-Labordere et al., 2014, 2019).

	European Problem	**American or Control Problems**
$d \leq 3$	PDE or MC	PDE
$d > 3$	MC	Nonlinear MC

Table 2: *Deterministic versus stochastic pricing schemes: which one is preferred?*

A.1 Markovian Dimension versus Martingale Order of Multiplicity

The space dimension d above refers to the Markovian dimension of a pricing problem, in the sense of the dimension of the state space of a related factor process, which may differ from the nominal dimension of a pricing model. For instance, when pricing an Asian option in the Black-Scholes model[11], the nominal dimension of the model is one, but the Markovian dimension of the pricing problem is two[12]. Also, in the case of models with jumps such as portfolio credit risk models, the notion of dimension d of the state space of the factor process is not always so well defined. In the case of a Markov chain model, it seems reasonable to define d as the logarithm of the dimension of the matrix generator.

A competing notion for the dimension of a pricing problem is its multiplicity μ, which is the minimal size of a family of martingales with the representation property[13] or, in financial terms, the minimal size of a family of replicating assets. In rough terms, this corresponds to the number of independent sources of noise in the model. It is often the case that $\mu = d$. However, this need not be so. We thus have:

- $\mu = 1 < 2 = d$ in the problem of pricing an Asian option in the Black-Scholes model[14].
- $d = 1 < 2 = \mu$ in the problem of pricing a vanilla option in the jump-to-ruin model[15].
- $d = n$ and $\mu = 2^n$ in the general continuous-time Markov chain bottom-up model of portfolio credit risk with n obligorsof Bielecki, Crépey, and Jeanblanc (2010), but μ reduces to n if simultaneous defaults are excluded.

The multiplicity μ[16] of a model is the relevant notion of dimension as far as a Monte Carlo pricing scheme is concerned. Thus observe that the complexity of a Monte Carlo method is linear in μ[17], whereas that of a deterministic method is exponential in d. This gives another perspective on the fact that simulation or simulation/regression (and the related machine learning) methods are better suited than deterministic ones for facing the curse of dimensionality.

[11] see III.§4.C.
[12] unless specific degenerate cases are considered, as in III.§4.C.1.
[13] see e.g. Davis and Varaiya (1974).
[14] see III.§4C.
[15] see X.§2.
[16] multiplied by the number of dates in the time-grid, in case of a time-discretised problem.
[17] there is a linear dependence in μ hidden in the $O(nm)$ in the upper left cell of Table 1.

OPTIMIZATION SCHEMES

Chapter VI

Machine Learning Techniques for Pricing and Greeking

The surge of machine learning techniques in finance is explained by the encounter between the much increased feasibility of machine learning in terms of computing power (with, in particular, the advent of GPU and now TPU computing), and an evolution of the derivatives management paradigm, following the 2008–09 global financial crisis, from a replication framework to an optimization framework for capital and collateral, going along with a growing trend towards automated trading (through platforms as soon as possible).

The numerical finance toolbox is traditionally based on three pillars that essentially correspond to the developements of previous chapters: approximate formulas, deterministic numerical schemes, simulation methods. The first include for example the quadrature schemes based on the Fourier analysis available in the affine jump diffusion models, or the formulas resulting from asymptotic analysis in the SABR model (see II.§2.G.1). The second are related to difference or finite element techniques for the PDEs of finance models, sometimes revisited in a tree formalism defined with reference to Markov chains resulting from time-discretized models. The third are based on the probabilistic representation, of the Feynman-Kac type, of the solutions of these PDEs, or the probabilistic formulation of these PDEs as backward stochastic differential equations (BSDEs, cf. V.(12)-(13) in an elementary one-period setup and see Crépey (2013)).

In this chapter we consider pricing and greeking from the point of view of statistical learning on simulated data in finance. Machine learning is then conceived, not as a way of modeling from the data (since the latter are simulated within a predefined model), but as a fourth term from the above toolbox. Actually, this is again a well-established tradition, rooted in simulation/regression schemes for the pricing of Bermudan options à la Tsitsiklis and Van Roy (2001) and Longstaff and Schwartz (2001), then extended to BSDEs (Gobet et al., 2005). Nevertheless, we can speak of a technological breakthrough in this domain: cf., to mention only two emblematic papers of this evolution, E, Han, and Jentzen (2017) or Buehler, Gonon, Teichmann, and Wood (2019).

Our purpose in putting machine learning pricing and greeking (also calibration in Chapter VII and statistical in Chapter VIII) techniques in perspective with the traditional quantitative finance methodologies is also to warn the reader about certain limitations that the mirage of "a universal solver" should not blur, including:

- only partial mathematical convergence results,
- care, time and effort required for setting up the databases,
- tedious and sometimes instable fine-tuning needed for 'scaling' the different dimensions of the problem and setting the hyperparameters,
- often long training times,

- performance sometimes hard to assess and potentially poor, dependent on the quality of the data in the first place,
- necessity to retrain the model every time the underlying reality that one tries to learn has changed.

Machine learning in finance is here to stay. It is in line with in the general spirit of quantitative finance of extracting as much information as possible from the market (and beyond, with the advent of alternative data). But it can only work as a complement and under the control of the traditional modeling techniques.

We will consider two well established, universal approximation machine learning frameworks, Gaussian processes (Rasmussen and Williams, 2006) and neural nets (Goodfellow et al., 2016), applied to respective interpolation and regression tasks, based on simulated data as explained above. Machine learning metamodels, i.e. interpolation paradigms in a sense (as opposed to the financial models as per Chapter I), are typically trained to the data by gradient descent, possibly mini-batched in the form of stochastic gradient descent, with ADAM as a popular algorithm. A crucial machine learning concen is overfitting, i.e. the risk that a good data fit be only a consequence of overparameterization, detrimental to generalization. To keep this risk under control, the data are split between a training set, which is used for driving the optimization, and a testing set, used for assessing the quality of the fit, e.g. in terms of root-mean square error (RMSE).

Given any rectangular computational training domain of interest, we tacitly rescale all inputs so that the domain becomes a unit hypercube Ω. This rescaling avoids any one independent variable dominating over another during any fitting of the market prices.

§1 Pricing and Greeking With Gaussian Processes

This section introduces Gaussian process regression for fast evaluation of financial derivatives and their sensitivities. Once the involved kernels have been learned, there is no need to use expensive derivative pricing or greeking functions. The kernels permit a closed form approximation for the sensitivities. Efficient hyper-parameter optimization procedures are available. Moreover, the advantage is not just computational: The risk estimation approach is Bayesian (Murphy, 2012)—the uncertainty in a point estimate which the model hasn't seen in the training data is quantified based on a distribution put on the metamodel parameters.

A Introduction

Statistical inference involves learning a function $Y = f(X)$ of the data, $(X, Y) := \{(x_i, y_i) \mid i = 1, \dots, n\}$. The idea of Gaussian processes (GPs) is to, without parameterizing[1] $f(X)$, place a probabilistic prior directly on the space of functions. The GP is hence a Bayesian nonparametric model that generalizes the Gaussian distributions from finite dimensional vector spaces to infinite dimensional function spaces. GPs are an example of a more general class of supervised machine learning techniques referred to as 'kernel learning', which model the covariance matrix from a set of parametrized kernels over the input. GPs extend and put in a Bayesian framework spline or kernel interpolators, as well as Tikhonov regularization (cf. VII.VII and VII.VII.A.1 and see Gupta and Reisinger (2014, Section 3.3), Rasmussen and Williams (2006) and Alvarez, Rosasco, and Lawrence (2012)). Neal (1996) also observed that certain neural networks with one hidden layer converge to a Gaussian process in the limit of an infinite number of hidden units.

Via "the kernel trick", based on Mercer's theorem (which is the infinite-dimensional analogue of the diagonalization of a symmetric matrix), GP regressions can be seen as infinite dimensional regressions against a series of latent factors.

[1] This is in contrast to nonlinear regressions commonly used in finance, which attempt to parameterize a non-linear function with a set of weights.

Compared with simpler alternatives such as splines or kernel smoothers, GP regressions offer a metamodeling framework with a probabilistic Bayesian interpretation and a quantification of the associated numerical uncertainty. Marginal likelihood maximization yields a convenient way of setting the hyperparameters. GPs can cope with noisy data but they are also interpolating in the noise-free limit. As opposed to Chebyshev interpolation (Gaß et al., 2017), which uses a deterministic node location imposed by the scheme (in conjunction with suitable interpolation weights), GPs can use an arbitrary, possibly unstructured (e.g. stochastically simulated) grid of observations.

Compared with richer alternatives such as deep neural networks (DNN), GPs typically require much less data to train. They also inherently provide "differential regularization" without the need to adopt cumbersome cross-validation techniques to tune regularization hyper-parameters, as in neural nets (NNs). Also, despite recent Bayesian deep learning developments meant to enable deep learning in small data domains, NNs are still difficult to cast in a Bayesian framework. However, unlike NNs, a kernel view does not give any hidden representations, failing to identify the useful features for solving a particular problem. More elaborate choice of priors can be used to address this issue.

Some of the numerical examples of this section are illustrated with Python code excerpts demonstrating the key features of the approach. These and additional examples are provided in the Github repository `https://github.com/mfrdixon/GP-CVA`. The examples can be run using the command `ipython notebook` (once the required packages have been loaded).

A Gaussian process (GP) regression setup involves both the randomness of financial risk factors and the Bayesian uncertainty relative to GP estimation. Hereafter we denote by E (respectively var or cov) a GP point (respectively variance or covariance) estimate.

B Gaussian Process Regressions

This part is a primer on Gaussian processes inference, written in the Bayesian statistics style. See Rasmussen and Williams (2006), MacKay (1998), and Murphy (2012, Chapter 15) for more background and detail.

We say that a random real-valued function $f : \mathbb{R}^d \to \mathbb{R}$ is drawn from a GP with a mean function μ and a covariance function, called kernel, c, i.e. $f \sim \mathcal{GP}(\mu, c)$, if for any input points $x_1, x_2, \ldots, x_n$ in $\mathbb{R}^p$, the corresponding vector of function values is Gaussian:

$$[f(x_1), f(x_2), \ldots, f(x_n)] \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{C}_{X,X}),$$

for some mean vector $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n)$, such that $\mu_i = \mu(x_i)$, and covariance matrix $\mathbf{C}_{X,X}$ that satisfies $(\mathbf{C}_{X,X})_{ij} = c(x_i, x_j)$. Unless specified otherwise, we follow the convention in the literature of assuming $\boldsymbol{\mu} = \mathbf{0}$. This choice is not a real limitation in practice since it only regards the prior and it does not prevent the mean of the predictor from being nonzero.

Kernels c can be any symmetric positive semidefinite function, which is the infinite-dimensional analogue of the notion of a symmetric positive semidefinite (i.e. covariance) matrix, i.e. such that

$$\sum_{i,j=1}^{n} c(x_i, x_j)\xi_i\xi_j \geq 0, \text{ for any points } x_k \in \mathbb{R}^p \text{ and reals } \xi_k.$$

Radial basis functions (RBF) are kernels that only depend on $||\mathbf{x} - \mathbf{x}'||$, such as the squared exponential (SE) kernel

$$c(\mathbf{x}, \mathbf{x}') = \exp\{-\frac{1}{2\ell^2}||\mathbf{x} - \mathbf{x}'||^2\}, \tag{1}$$

where the length-scale parameter ℓ can be interpreted as "how far you need to move in input space for the function values to become uncorrelated", or the Matern (MA) kernel

$$c(\mathbf{x}, \mathbf{x}') = \frac{2^{1-p}}{\Gamma(p)}\left(\sqrt{2p}\frac{||\mathbf{x} - \mathbf{x}'||}{\ell}\right)^p K_p\left(\sqrt{2p}\frac{||\mathbf{x} - \mathbf{x}'||}{\ell}\right) \tag{2}$$

(which converges to (1) in the limit where p goes to infinity), where ℓ and p are non-negative parameters, Γ is the Euler Gamma function, and K_p is the modified Bessel function of the second kind. One advantage of GPs over more naive interpolation methods is their expressability. One can combine the kernels by convolution (Melkumyan and Ramos, 2011).

GPs can be seen as distributions over the reproducing kernel Hilbert space (RKHS) of functions which is uniquely defined by the kernel function, c (Scholkopf and Smola, 2001). GPs with RBF kernels are known to be universal approximators to within an arbitrarily small epsilon band of any continuous function on any compact subset of the input space (Micchelli et al., 2006).

GPs also provide "differential regularity". GPs are RKHSs defined in terms of differential operators, with the Hilbert norm of the latent function having the effect of penalizing the gradients. Regularity of the GP interpolation is thus controllable through the choice of the kernel and smoothing parameters (Rasmussen and Williams, 2006, Section 6.2).

As compared with neural nets to be adressed in later sections, one limitation of the kernel is that, in principle, it does not reveal any hidden representations — failing to identify the useful features for solving a particular problem. However, the issue of feature discovery can be addressed by GPs through imposing "spike-and-slab" mixture priors on the covariance parameters (Savitsky et al., 2011).

We assume additive Gaussian i.i.d. noise, $y \mid \mathbf{x} \sim \mathcal{N}(f(\mathbf{x}), \varsigma^2)$, and a GP prior on $f(\mathbf{x})$, given training inputs $x_i \in X$ and training targets $y_i \in Y$. By the Gaussian conditioning Lemma 0.1, the predictive distribution of the GP evaluated at arbitrary test points X_* is:

$$(f_* \mid X, Y, X_*) \sim \mathcal{N}\big(E[f_*|X, Y, X_*], \mathrm{var}[f_*|X, Y, X_*]\big). \tag{3}$$

where, the moments of the posterior over X_* are

$$\begin{aligned} E[f_*|X, Y, X_*] &= \mu_{X_*} + \mathbf{C}_{X_*,X}[\mathbf{C}_{X,X} + \varsigma^2 I]^{-1} Y, \\ \mathrm{var}[f_*|X, Y, X_*] &= \mathbf{C}_{X_*,X_*} - \mathbf{C}_{X_*,X}[\mathbf{C}_{X,X} + \varsigma^2 I]^{-1} \mathbf{C}_{X,X_*}. \end{aligned} \tag{4}$$

Here, $\mathbf{C}_{X_*,X}$, $\mathbf{C}_{X,X_*}$, $\mathbf{C}_{X,X}$, and $\mathbf{C}_{X_*,X_*}$ are matrices that consist of the kernel, $c : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, evaluated at the corresponding points, X and X_*, and μ_{X_*} is the mean function evaluated on the test inputs X_*.

In the context of derivative pricing applications, X may correspond to a set of risk factor grid nodes, Y to the corresponding model prices (valued by analytical formulas or any, possibly approximate, classical numerical finance pricing schemes as per the previous chapters), $E[f_*|X, Y, x_*]$ to the GP regressed prices corresponding to the new value $x_* \in X_*$ of the risk factors, and $\mathrm{var}[f_*|X, Y, x_*]$ to the corresponding interpolation uncertainty. Note that the latter is only equal to 0 if $x_* \in X$ and one is in the noise-free case where ς has been set to 0.

We emphasize that, in a least square Monte-Carlo regression approach a la Longstaff and Schwartz (2001) (see IV.§9), we train function approximators, usually as linear combinations of fixed basis functions (but not only, see §3), on simulated future cash flows. By contrast, in this section, GPs are trained on values (not cash flows), on a (structured or not, deterministically or stochastically generated) grid, like a sophisticated interpolator.

C Hyper-parameter Tuning

GPs are fit to the data by optimizing *the evidence*-the marginal probability of the data given the model with respect to the learned kernel hyperparameters.

Up to a constant, the evidence has the form (see e.g. Murphy (2012, Section 15.2.4, p. 523)):

$$\log \gamma(Y \mid X, \lambda) = -\left[Y^\top (\mathbf{C}_{X,X} + \varsigma^2 I)^{-1} Y + \log\det(\mathbf{C}_{X,X} + \varsigma^2 I)\right], \tag{5}$$

where the method hyperparameters, ϱ say, include ς in (5) and parameters of c (e.g. $\varrho = [\ell, \varsigma]$, assuming an SE kernel as per (1) or an MA kernel for some exogenously fixed value of ν in (2)).

The first and second term in the brackets in (5) can be interpreted as a *model fit* and a *complexity penalty* term (see Rasmussen and Williams (2006, Section 5.4.1)). Maximizing the evidence with respect to the method hyperparameters, i.e. computing $\varrho^* = \underset{\varrho}{\text{Argmax}} \log \gamma(Y \mid X, \varrho)$, results in an automatic Occam's razor controlling the trade-off between the regression fit and the regularity of the interpolator (see Alvarez, Rosasco, and Lawrence (2012, Section 2.3) and Rasmussen and Ghahramani (2001)). In practice, the negative evidence is minimized by stochastic gradient descent (SGD). The gradient of the evidence is given analytically by

$$\partial_\varrho \log \gamma(Y \mid X, \varrho) = \text{tr}\left(\alpha\alpha^\top - (\mathbf{C}_{X,X} + \varsigma^2 I)^{-1}\right) \partial_\varrho (\mathbf{C}_{X,X} + \varsigma^2 I)^{-1}, \tag{6}$$

where

$$\alpha := (\mathbf{C}_{X,X} + \varsigma^2 I)^{-1} Y\,,\ \partial_\varsigma (\mathbf{C}_{X,X} + \varsigma^2 I)^{-1} = -2\varsigma (\mathbf{C}_{X,X} + \varsigma^2 I)^{-2} \tag{7}$$

and, in the case of the SE or MA kernels,

$$\partial_\ell (\mathbf{C}_{X,X} + \varsigma^2 I)^{-1} \quad = \quad -(\mathbf{C}_{X,X} + \varsigma^2 I)^{-2} \partial_\ell \mathbf{C}_{X,X} \tag{8}$$

(with, in the SE case, $\partial_\ell c(\mathbf{x}, \mathbf{x}') = \ell^{-3} ||\mathbf{x} - \mathbf{x}'||^2 c(\mathbf{x}, \mathbf{x}')$).

D Computational Properties

Training time, required for maximizing (5) numerically, scales poorly with the number of observations n. This stems from the need to solve linear systems and compute log determinants involving an $n \times n$ symmetric positive definite covariance matrix $\mathbf{C}_{X,X} + \varsigma^2 I = \mathbf{C}$. This task is commonly performed by computing the Cholesky decomposition of $\mathbf{C}$ incurring $\mathcal{O}(n^3)$ complexity. Prediction, however, is faster and can be performed in $\mathcal{O}(n^2)$ with a matrix-vector multiplication for each test point, and hence the primary motivation for using GPs is real-time risk estimation performance.

If uniform grids are used, we have $n = \prod_{k=1}^d n_k$, where n_k are the number of grid points per variable. However, mesh-free GPs can be used as described in G.1.

In terms of storage cost, although each kernel matrix $\mathbf{C}_{X,X}$ is $n \times n$, we only store the n-vector α in (7), which brings reduced memory requirements.

D.1 Massively scalable Gaussian processes

Massively scalable Gaussian processes (MSGP) are a recent significant extension of the basic kernel interpolation framework described above. The core idea of the framework, which is detailed in Gardner, Pleiss, Wu, Weinberger, and Wilson (2018), is to improve scalability by combining GPs with 'inducing point methods'. Using structured kernel interpolation (SKI), a small set of ν inducing points are carefully selected from the original training points. Under certain choices of the kernel, such as RBFs, a Kronecker and Toeplitz structure of the covariance matrix can be exploited by fast Fourier transform (FFT). Finally, output over the original input points is interpolated from the output at the inducing points. The interpolation complexity scales linearly with dimensionality d of the input data by expressing the kernel interpolation as a product of 1D kernels. Overall, SKI gives $\mathcal{O}(dn + 279 \ln \nu)$ training complexity and $\mathcal{O}(1)$ prediction time per test point, using the Lanczos variance estimates of Pleiss, Gardner, Weinberger, and Wilson (2018). In these notes, we primarily use the basic interpolation approach for simplicity.

D.2 Online learning

If the option pricing model is recalibrated intra-day, then the corresponding GP model should be re-trained. Online learning techniques permit performing this incrementally (Pillonetto et al., 2010). To enable online learning, the training data should be augmented with the constant model parameters. Each time the parameters are updated, a new observation (x', y') is generated from the option model

prices under the new parameterization. The posterior at test point x_* is then updated with the new training point following

$$\gamma(f_*|X,Y,x',y',x_*) = \frac{\gamma(x',y'|f_*)\gamma(f_*|X,Y,x_*)}{\int_Z \gamma(x',y'|z)\gamma(z|X,Y,x_*)dz}, \tag{9}$$

where the previous posterior $\gamma(f_*|X,Y,x_*)$ becomes the prior in the update and $f_* \in Z \subset \mathbb{R}$. Hence the GP learns over time as model parameters (which are an input to the GP) are updated through recalibration of the pricing model.

E Pricing Application

We consider an European call and a put option struck on the same underlying, with strike $K = 100$. We assume that the underlying follows Heston dynamics (in risk-neutral form, cf. I.(62)):

$$\begin{aligned} dv_t &= -\lambda(v_t - \theta)dt + \eta\sqrt{v_t}dB_t, \\ \frac{dS_t}{S_t} &= rdt + \sqrt{v_t}dW_t \end{aligned} \tag{10}$$

where the notation is detailed in Table 1. We use a Fourier Cosine method by Fang and Oosterlee (2008) to generate the European Heston option price training and testing data for the GP. We will also use this method to benchmark the GP Greeks obtained by differentiating the kernel function.

Table 1 also lists the values of the Heston parameters and terms of the European call and put option contract used in our numerical experiments. Additionally, the data is generated using an Euler time stepper for (10) using 100 time steps over a two year horizon.

Parameter description	Symbol	Value
Initial stock price	S_0	100
Initial variance	v_0	0.1
Mean reversion rate	λ	0.1
Mean reversion level	θ	0.15
Vol. of Vol.	η	0.1
Risk free rate	r	0.01
Strike	K	100
Maturity	T	2.0
Correlation W, B	ρ	-0.9

Table 1: This table shows the values of the parameters for the Heston dynamics and terms of the European call and put option contracts.

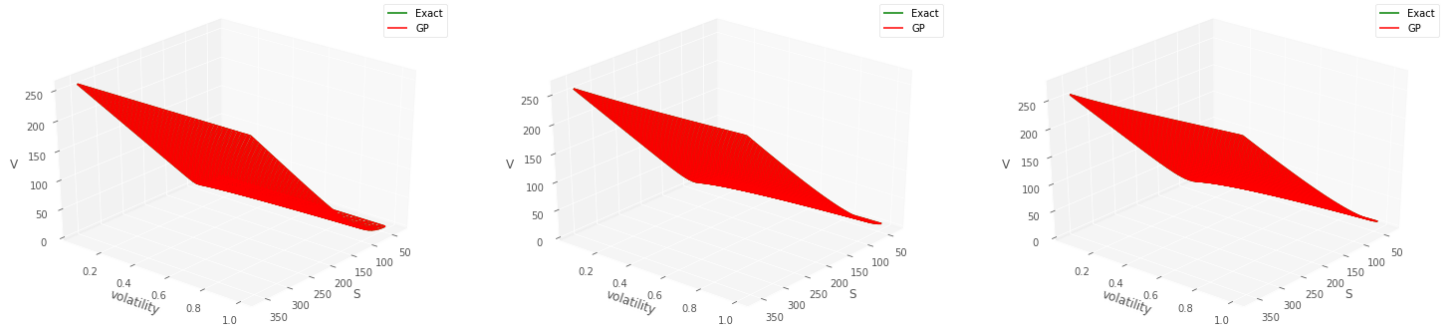
For each t_i in a grid of dates, we fit a GP to the Heston pricing function from Heston prices computed by Fourier formulas for gridded values of S and $\sqrt{v}$ (keeping time to maturity fixed). We emphasize that the Heston dynamics (10) are not used in the simulation mode in this procedure.

Listing 1 details how the GP and data are prepared to predict prices over the two dimensional grid, for a fixed time to maturity and strike. Figures 1 and 2 (top) show the comparison between the gridded semi-analytic and GP call and put price surfaces at various time to maturities, together with the GP estimate. Within each column in the figures, the same GP model has been simultaneously fitted to both the call and put prices over a 30×30 grid $\Omega^h \subset \Omega := [0,1] \times [0,1]$ of stock prices and volatilities[2], for a given time to maturity. The bottom panel of the figure shows the error surfaces between the GP and semi-analytic estimates. The scaling to the unit domain is not essential. However, we observed superior numerical stability when scaling.

[2]Note that the plot uses the original coordinates and not the re-scaled co-ordinates.

Across each column, corresponding to different time to maturities, a different GP model has been fitted. The GP is then evaluated out-of-sample over a 40×40 grid $\Omega^{h'} \subset \Omega$, so that many of the test samples are new to the model. This is repeated over the various dates t_i. The option model versus GP model are observed to produce very similar values.

(a) Price: $T - t = 1.8$ (b) Price: $T - t = 1.0$ (c) Price: $T - t = 0.2$



(a) Error: $T - t = 1.8$ (b) Error: $T - t = 1.0$ (c) Error: $T - t = 0.2$

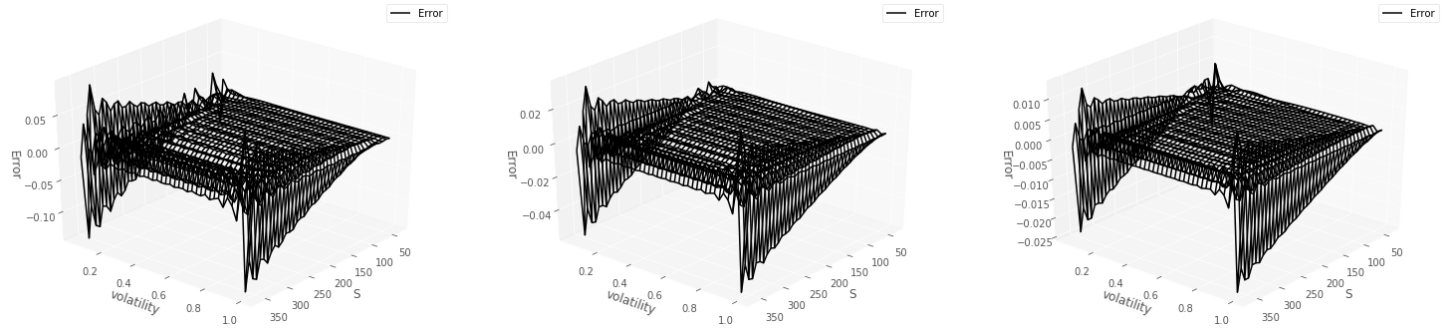
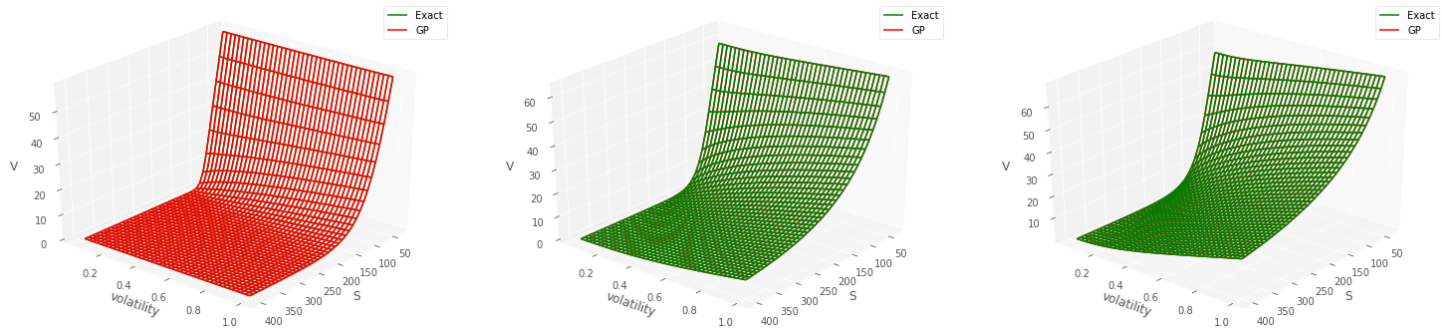


Figure 1: *This figure compares the gridded Heston GP and semi-analytic ('exact') model call prices (top) and error (bottom) surfaces at various time to maturities. The GP estimate is observed to be practically identical (on average it's slightly above the semi-analytic solution).*

(a) Price: $T - t = 1.8$ (b) Price: $T - t = 1.0$ (c) Price: $T - t = 0.2$



(a) Error: $T - t = 1.8$ (b) Error: $T - t = 1.0$ (c) Error: $T - t = 0.2$

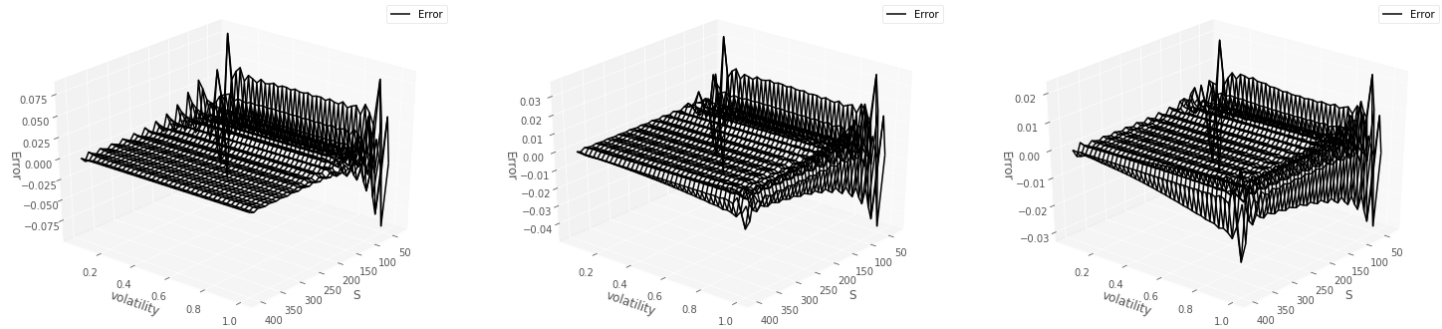


Figure 2: *Similar as Figure 1 for the put (instead of call) option.*

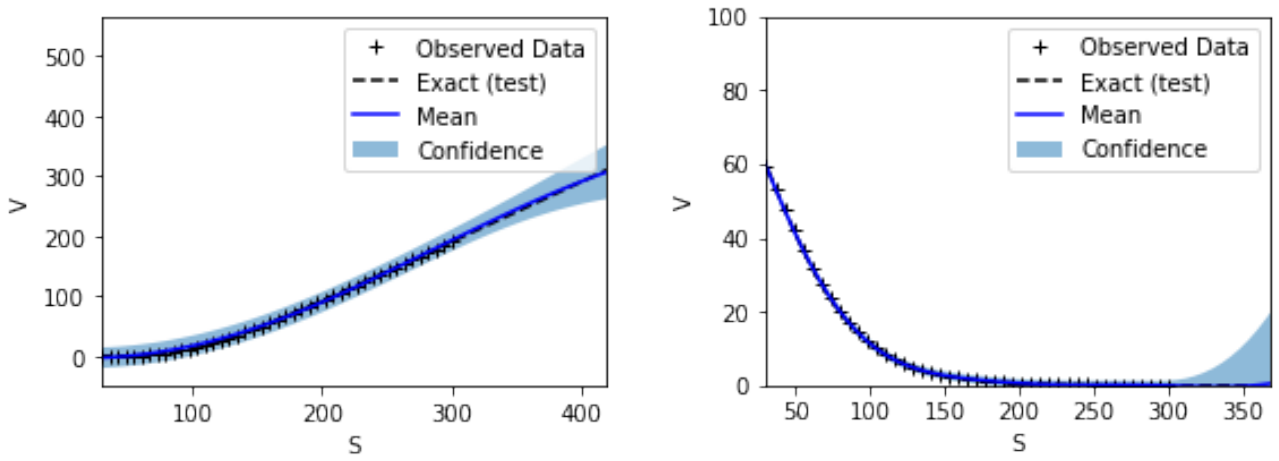
```
import PyHeston

S= 100
v0 = 0.1

lmbda = 0.1
meanV = 0.15
eta = 0.1
r = 0.01
K = 100
T = 2.0
rho =-0.9
step_size = 0.4 #used internally by Heston pricer


lb = 1
ub = 400
portfolio = {}
portfolio['call']={}
portfolio['put']={}

training_number = 30
testing_number = 40


x1_train = np.array(np.linspace(0.0,1.0, training_number), dtype='float32').reshape(
    training_number, 1)
x2_train = np.array(np.linspace(0.05,1.0, training_number), dtype='float32').reshape(
    training_number, 1)


X1_train, X2_train = np.meshgrid(x1_train, x2_train)
x_train = np.zeros(len(X1_train.flatten())*2).reshape(len(X2_train.flatten()), 2)
x_train[:,0] = X1_train.flatten()
x_train[:,1] = X2_train.flatten()

x1_test = np.array(np.linspace(0.0,1.0, testing_number), dtype='float32').reshape(
    testing_number, 1)
x2_test = np.array(np.linspace(0.05,1.0, testing_number), dtype='float32').reshape(
    testing_number, 1)

X1_test, X2_test = np.meshgrid(x1_test, x2_test)

x_test = np.zeros(len(X1_test.flatten())*2).reshape(len(X2_test.flatten()), 2)
x_test[:,0] = X1_test.flatten()
x_test[:,1] = X2_test.flatten()

portfolio['call']['price']= lambda x,y,z: PyHeston.HestonCall(lb+(ub-lb)*x, y, K, z, r,
    lmbda, meanV, varsigma, rho, step_size)
portfolio['put']['price']= lambda x,y,z: PyHeston.HestonPut(lb+(ub-lb)*x, y, K, z, r,
    lmbda, meanV, varsigma, rho, step_sizez)

for key in portfolio.keys():
    portfolio[key]['GPs'] = trainGPs(x_train, portfolio[key]['price'], timegrid)
    portfolio[key]['y_tests'], portfolio[key]['preds'], portfolio[key]['varsigmas'] =
    predictGPs(x_test, portfolio[key]['price'], portfolio[key]['GPs'], timegrid)
```

Listing 1: *This Python 3.0 code excerpt illustrates how to use a GP to fit to option prices under a Heston model. x_1 and x_2 are gridded underlying stock values and volatilities respectively. Note that the listing provides the salient details only and the reader should refer to Example-6-GP-Heston.ipynb in Github for the full implementation.*

E.1 Extrapolation

One instance where kernel combination is useful in derivative modeling is for extrapolation—the appropriate mixture or combination of kernels can be chosen so that the GP is able to predict outside the domain of the training set. Noting that the payoff is linear when a call or put option is respectively deeply in-the money, we can configure a GP as a combination of a linear kernel and, say, a SE kernel. The linear kernel is included to ensure that prediction outside the domain preserves the linear property, whereas the SE kernel captures non-linearity.

Figure 3 shows the results of using this combination of kernels to extrapolate the prices of a call struck at 110 and a put struck at 90, with time to maturity of the options fixed to two years. The linear asymptote of the payoff function is preserved by the GP prediction and the uncertainty increases as the test point is further from the training set.



(a) call price (b) put price

Figure 3: *This figure assesses the GP option price prediction in the setup of a Black-Scholes model. The GP with a Linear and SE kernel is trained on* $n = 50$ X, Y *pairs, where* $X \in \Omega^h \subset (0, 300]$ *is the gridded underlying of the option prices and* Y *is a vector of call or put prices. These training points are shown by the black '+' symbols. The exact result using the Black-Scholes pricing formula is given by the black line. The predicted mean (blue solid line) and variance of the posterior are estimated from Equation* (4) *over* $m = 100$ *gridded test points,* $X_* \in \Omega^h_* \subset [300, 400]$*, for the (left) call option struck at 110 and (right) put option struck at 90. The shaded envelope represents the 95% uncertainty band of the GP about the mean of the posterior.*

The above examples are trained on (semi)-analytic Black-Scholes and Heston prices, so the quality of the approximator can be assessed.

In a realistic application where approximators are trained on more general products in more general models, numerical methods as per the previous chapters would be required to find the values on the knot points.

F Greeking Application

The GP provides analytic derivatives with respect to the input variables

$$\partial_{X_*} E[f_*|X, Y, X_*] = \partial_{X_*}\mu_{X_*} + (\partial_{X_*}\mathbf{C}_{X_*,X})\alpha \tag{11}$$

where $\partial_{X_*}\mathbf{C}_{X_*,X} = \frac{1}{\ell^2}(X - X_*)\mathbf{C}_{X_*,X}$ and we recall from after (6) that $\alpha = [\mathbf{C}_{X,X} + \varsigma^2 I]^{-1}Y$ (and in the numerical experiments we set $\boldsymbol{\mu} = 0$). Second order sensitivities are obtained by differentiating once more with respect to X_*.

Note that α is already calculated at (pricing) training time by Cholesky matrix factorization of $[\mathbf{C}_{X,X} + \varsigma^2 I]$ with $\mathcal{O}(n^3)$ complexity, so there is no significant computational overhead from greeking.

Once the GP has learned the derivative prices, Equation (11) is used to evaluate the first order greeks with respect to the input variables over the test set. Example source code illustrating the implementation of this calculation is given in Listing 2.

Figure 4 shows (left) the GP estimate of a call option's delta $\Delta := \partial_S C$ and (right) the error between the Black–Scholes delta Δ^{bs} (cf. I.(41)) and the GP estimate. We emphasize that the GP model is trained on underlying and option pricing data and not using the option's delta. The GP delta is observed to closely track the Black–Scholes formula for the delta. Figure 5 shows (left) the GP estimate of a call option's vega $\mathcal{V} := \partial_\sigma C$, having trained on the implied volatility, and Black–Scholes option model prices and not using the option's vega. The right hand panel shows the error between the Black–Scholes vega $\mathcal{V}^{bs}$ (cf. I.(41)) and the GP estimate. The GP vega is observed to closely track the Black–Scholes formula for the vega.

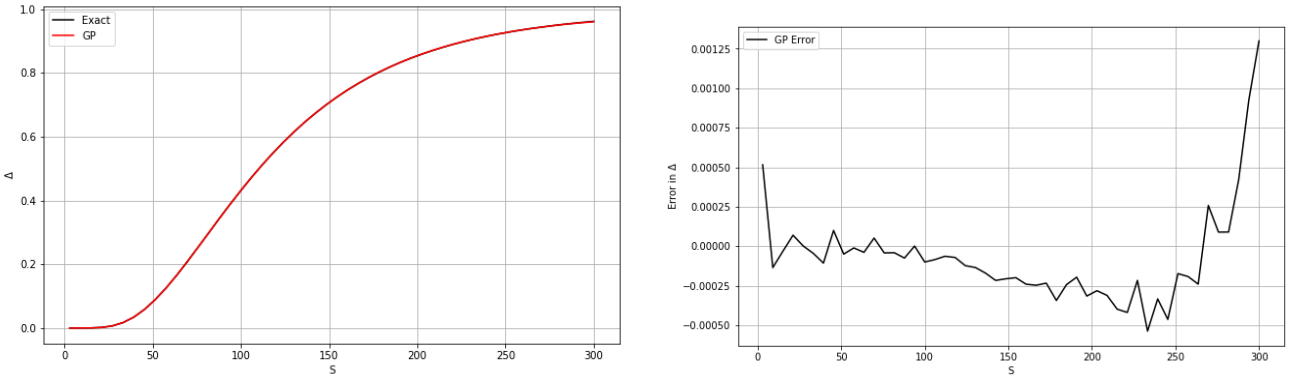


Figure 4: *This figure shows (left) the comparison of the GP estimate of the call option's delta* $\Delta := \partial_S C$ *and the BS delta formula. (Right) The error between the BS delta and the GP estimate.*

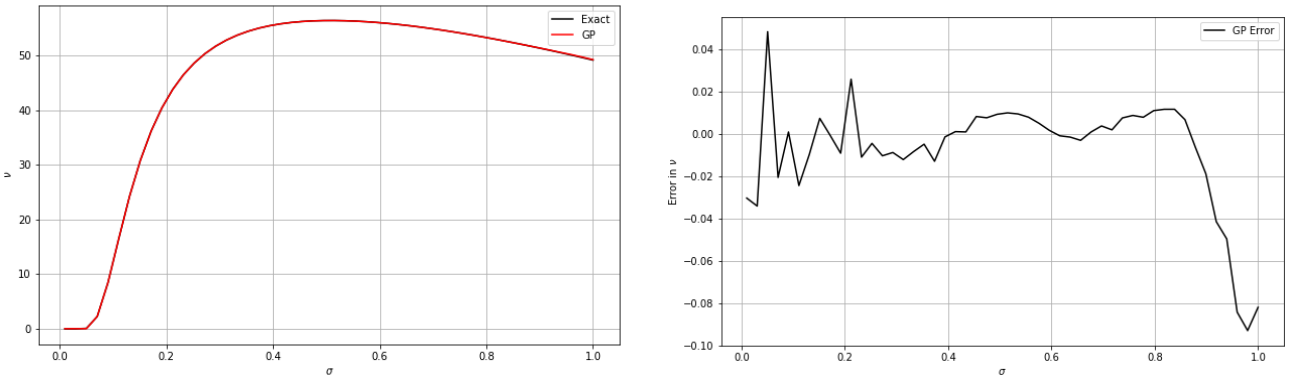


Figure 5: *This figure shows* (left) *the comparison of the GP estimate of the call option's vega* $\mathcal{V} := \partial_\sigma C$ *and the BS vega formula.* (right) *The error between the BS vega and the GP estimate.*

```python
import scipy as sp
import numpy as np
from BlackScholes import *
from sklearn import gaussian_process
from sklearn.gaussian_process.kernels import ConstantKernel, RBF


# set BS model parameters
r = 0.0002  # risk-free rate
S= 100      # Underlying spot
KC = 130    # Call strike
KP = 70     # Put strike
varsigma = 0.4 # implied volatility
T = 2.0     # Time to maturity
lb = 0.001  # lower bound on domain
```

```
ub = 300     # upper bound on domain
varsigma_n = 1e-8 # additive noise in GP

call = lambda x,y: bsformula(1, lb+(ub-lb)*x, KC, r, T, y, 0)[0]
put = lambda x,y: bsformula(-1, lb+(ub-lb)*x, KP, r, T, y, 0)[0]

training_number = 100
testing_number = 50

x_train = np.array(np.linspace(0.01,1.2, training_number), dtype='float32').reshape(
    training_number, 1)
x_test = np.array(np.linspace(0.01,1.0, testing_number), dtype='float32').reshape(
    testing_number, 1)

y_train = []

for idx in range(len(x_train)):
    y_train.append(call(x_train[idx], varsigma))
y_train = np.array(y_train)

sk_kernel = RBF(length_scale=1.0, length_scale_bounds=(0.01, 10000.0))
gp = gaussian_process.GaussianProcessRegressor(kernel=sk_kernel, n_restarts_optimizer
    =20)
gp.fit(x_train,y_train)
y_pred, varsigma_hat = gp.predict(x_test, return_std=True)

l = gp.kernel_.length_scale
rbf= gaussian_process.kernels.RBF(length_scale=l)

Kernel= rbf(x_train, x_train)
\maco_y = Kernel + np.eye(training_number) * varsigma_n
L = sp.linalg.cho_factor(\maco_y)
alpha_p = sp.linalg.cho_solve(np.transpose(L), y_train)

k_s = rbf(x_test, x_train)

k_s_prime = np.zeros([len(x_test), len(x_train)])
for i in range(len(x_test)):
    for j in range(len(x_train)):
        k_s_prime[i,j]=(1.0/l**2)*(x_train[j]-x_test[i])*k_s[i,j]
# Calculate the gradient of the mean using Equation in greeking \Section\ref{ss:greek}.
f_prime = np.dot(k_s_prime, alpha_p)/(ub-lb)

# show error between BS delta and GP delta
delta = lambda x,y: bsformula(1, lb+(ub-lb)*x, KC, r, T, y, 0)[1]
delta(x_test, varsigma)-f_prime
```

Listing 2: *This Python 3.0 code excerpt, using scikit-learn, illustrates how to calculate the Greeks of an option by differentiating the GP price model. x are gridded underlying stock values, so that f_prime is the estimate of the delta. If x were gridded volatilities, then f_prime would be the estimate of the vega. The listing provides the salient details only and the reader should refer to Example-2-GP-BS-Derivatives.ipynb in Github for the full implementation.*

G Extensions

G.1 Mesh-Free GPs

The above numerical examples have trained and tested GPs on uniform grids. This approach suffers from a stringent curse of dimensionality[3] issue, as the number of training points grows exponentially

[3] see V.§4.A.

with the dimensionality of the data (cf. D). However, use of uniform grids is by no means necessary. We show here how GPs can show favorable approximation properties with a relatively small number of simulated reference points (cf. Gramacy and Apley (2015)).

Figure 6 shows predicted Heston call prices using (left) 50 and (right) 100 simulated training points, indicated by "+"s, drawn from a uniform random distribution. The Heston call option is struck at $K = 100$ with a maturity of $T = 2$ years. Figure 7 (left) shows the convergence of the GP MSE of

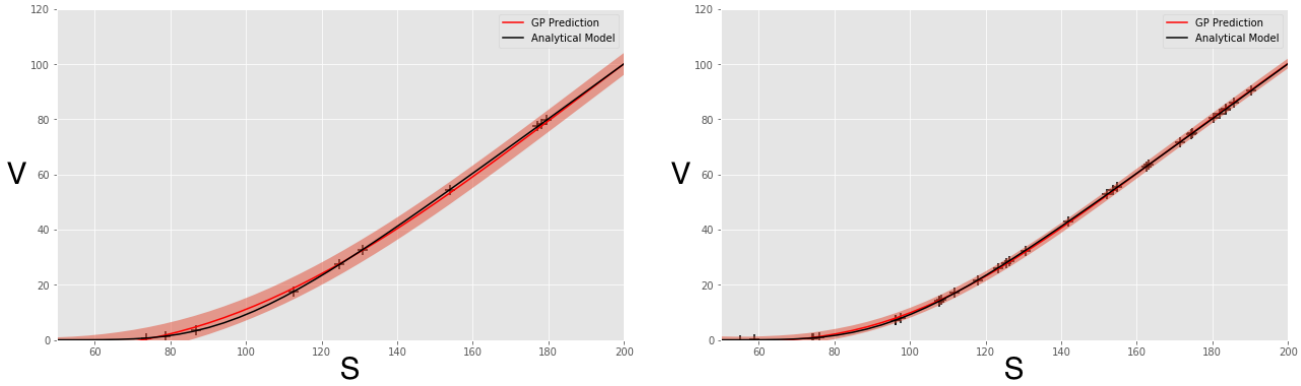


Figure 6: *GP predicted Heston Call prices and 95% uncertainty bands using (left) 50 and (right) 100 simulated training points, indicated by '+'s, drawn from a uniform random distribution.*

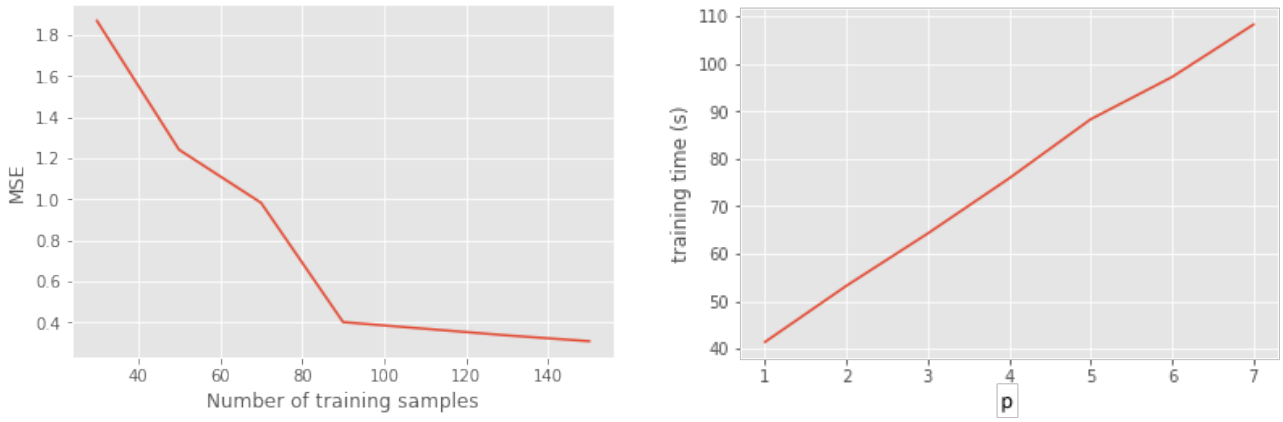


Figure 7: *(Left) The convergence of the GP MSE of the prediction is shown based on the number of simulated Heston training points.* (Right) *Fixing the number of simulated points to 100, but increasing the dimensionality d of each observation point (including more and more Heston parameters), the figure shows the wall-clock time for training a GP with SKI.*

the prediction, based on the number of Heston simulated training points.

G.2 Massively Scalable GPs

Fixing the number of simulated points to 100, but increasing the input space dimensionality, d, of each observation point (i.e. including more and more Heston parameters), Figure 7 (right) shows the wall-clock time for training a GP with SKI (see D.1). Note that the number of SGD iterations has been fixed to 1000. All performances are based on a 2.2 GHz Intel Core i7 laptop

Figure 8 shows the increase of MSGP training time and prediction time against the number of training points n from a Black Scholes model. Fixing the number of inducing points to $\nu = 30$ (see D.1), we increase the number of observations, n, in the $d = 1$ dimensional training set.

Setting the number of SGD iterations to 1000, we observe an approximate 1.4 increase in training time for a 10x increase in the training sample. We observe an approximate 2x increase in prediction time

for a 10x increase in the training sample. The reason that the prediction time grows with n (instead of being constant, cf. D) is due to memory latency in our implementation—each point prediction involves loading a new test point into memory. Fast caching approaches can be used to reduce this memory latency, but are beyond the scope of this research.

Note that training and testing times could be improved with CUDA programming on a GPU, but are not evaluated here.

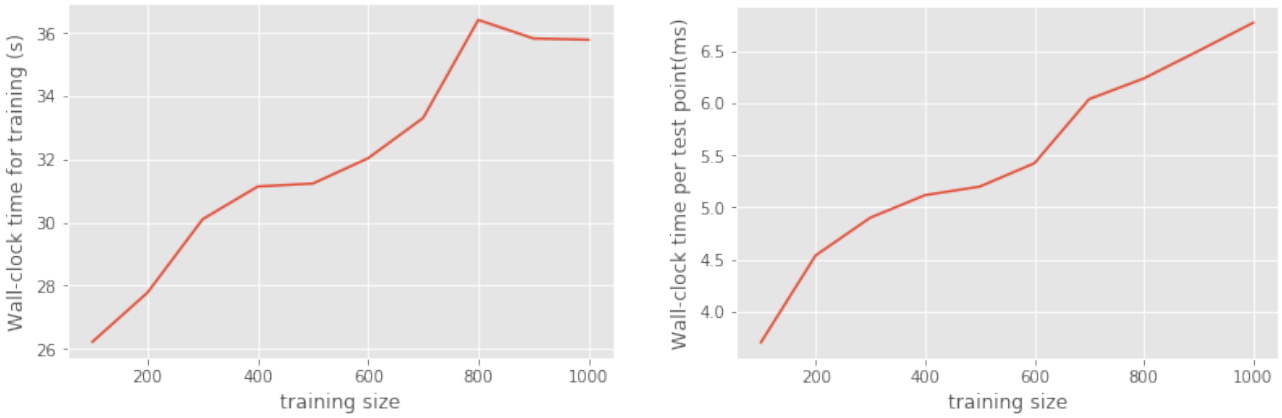


Figure 8: *(Left) The elapsed wall-clock time is shown for training against the number of training points generated by a Black-Scholes model.* (Right) *The elapsed wall-clock time for prediction of a single point is shown against the number of testing points. The reason that the prediction time increases (whereas the theory reviewed in D.1 says it should be constant) is due to memory latency in our implementation—each point prediction involves loading a new test point into memory.*

§2 Non-Arbitrable Neural Net Interpolation

There have been recent surges of literature about the learning of derivative pricing functions by machine learning surrogate models, i.e. Gaussian processes as above or, among possible alternatives for the same acceleration purpose, neural nets, which are respectively surveyed in Crépey and Dixon (2020, Section 1) and Ruf and Wang (2020). There has, however, been relatively little coverage of no-arbitrage constraints when interpolating prices. In this section we demonstrate the modification of the loss function or a feedforward neural network architecture to exclude (hard constraints approach) or penalize (soft constraints approach) arbitrages in an interpolation of a full surface of European vanilla put options.

A Problem Statement

We consider European vanilla option prices on a stock or index S under the assumption that a deterministic short interest rate term structure $r(t)$ has been bootstrapped from the zero coupon curve, and that a term structure of deterministic continuous-dividend-yields $q(t)$ on S has then been extracted from the prices of the forward contracts on S. For simplicity we assume r and q constant in the notation below.

Without restriction given the call-put parity relationship, we only consider put option prices hereafter. We denote by $P_*(T,K)$ the market price of the put option with maturity T and strike K on S, observed for a finite number of pairs (T,K) at a given day, conventionally taken as $t=0$.

Our goal is to construct, by neural net interpolation, an arbitrage-free and continuous put price surface $P:\mathbb{R}_+\times\mathbb{R}_+\to\mathbb{R}_+$ interpolating P_* up to some error term. in terms of the reduced prices $p(T,k):=e^{qT}P(T,K)$, where $k=Ke^{-(r-q)T}$, the no calendar spread arbitrage and no butterfly arbitrage conditions are written as (assuming the embedded regularity of the reduced prices):

$$\partial_T p(T,k)\geq 0\,,\ \partial^2_{k^2}p(T,k)\geq 0, \tag{12}$$

Accordingly, we are targeting a reduced put price surface satisfying (12). More broadly, see (Roper, 2010, Theorem 2.1) for the detailed statement of the static non-arbitrage relationships conditions on European vanilla call (easily transposable to put) option prices, also including, in particular, an initial condition at $T = 0$ given by the option payoffs. This initial payoff condition will be incorporated as well to our learning schemes, in a way described in D.

In both networks considered below, the derivatives that appear in (12) are available analytically via the neural network automatic differentiation capability. Hard or soft constraints can then be used to enforce the shape properties (12), exactly in the case of hard constraints and approximately (via regularization) in the case of soft constraints.

B Shape Constrained Neural Networks

We consider parameterized maps $p = p_{\mathbf{W},\mathbf{b}}$

$$(T, k) \ni \mathbb{R}^2_+ \xrightarrow{p} p_{\mathbf{W},\mathbf{b}}(T, k) \in \mathbb{R}_+, \tag{13}$$

given as deep neural networks with two hidden layers. As detailed in Goodfellow et al. (2016), these take the form of a composition of simpler functions:

$$p_{\mathbf{W},\mathbf{b}}(x) = f^{(3)}_{W^{(3)},b^{(3)}} \circ f^{(2)}_{W^{(2)},b^{(2)}} \circ f^{(1)}_{W^{(1)},b^{(1)}}(x), \tag{14}$$

where

$$\mathbf{W} = (W^{(1)}, W^{(2)}, W^{(3)}) \text{ and } \mathbf{b} = (b^{(1)}, b^{(2)}, b^{(3)})$$

are weight matrices and bias vectors, and the $f^{(l)} := \varsigma^{(l)}(W^{(l)}x + b^{(l)})$ are semi-affine, for nondecreasing (typically nonlinear) activation functions $\varsigma^{(l)}$ applied to their (vector-valued) argument componentwise. Any weight matrix $W^{(\ell)} \in \mathbb{R}^{m\times n}$ can be expressed as an n column $W^{(\ell)} = [\mathbf{w}^{(\ell)}_1, \ldots, \mathbf{w}^{(\ell)}_n]$ of m-vectors, for successively chained pairs (tn, m) of dimensions varying with $l = 1, 2, 3$, starting from $n = 2$, the number of inputs in (13), for $l = 1$, and ending up with $m = 1$, the number of outputs, for $l = 3$. The two hidden layers correspond to $f^{(1)}$ and $f^{(2)}$ in (14), were x is the input and $f^{(3)}$ yields the output.

B.1 Hard Constraints Approach

In the hard constraints case, our network is sparsely connected in the sense that, with $x = (T, k)$ as above,

$$f^{(1)}_{W^{(1)},b^{(1)}}(x) = \big(f^{(1,T)}_{W^{(1,T)},b^{(1,T)}}(T), f^{(1,k)}_{W^{(1,k)},b^{(1,k)}}(k)\big),$$

where $W^{(1,T)}, b^{(1,T)}$ and $W^{(1,k)}, b^{(1,k)}$ correspond to parameters of sub-graphs (see Figure 9) for each input T and k, and

$$f^{(1,T)}(T) := \varsigma^{(1,T)}(W^{(1,T)}T + b^{(1,T)}), \; f^{(1,k)}(k) := \varsigma^{(1,k)}(W^{(1,k)}k + b^{(1,k)}).$$

To impose the shape constraints relevant for put options, it is then enough to restrict ourselves to nonnegative weights, and to convex (and nondecreasing) activation functions, namely

$$\text{softplus}(x) := \ln(1 + e^x),$$

except for $\varsigma^{(1,T)}$, which will be taken as an S-shaped sigmoid $(1 + e^{-x})^{-1}$. Imposing non-negative constraints on weights can be achieved in back-propagation using projection functions applied to each weight after each gradient update.

Hence, the network is convex and nondecreasing in k, as a composition (restricted to the k variable) of convex and nondecreasing functions of k. In T, the network is nondecreasing, but not necessarily

convex, because the activation function for the maturity subnetwork hidden layer is not required to be convex - in fact, we choose a sigmoid function.

Figure 9 illustrates the shape preserving feed forward architecture with two hidden layers containing 10 hidden nodes. For avoidance of doubt, the figure is not representative of the number of hidden neurons used in our experiments. However, the connectivity is representative. The first input variable, T, is only connected to the first 5 hidden nodes and the second input variable, k, is only connected to the last 5 hidden nodes. Effectively, two sub-networks have been created where no information from the input layer crosses the sub-networks until the second hidden layer. In other words, each sub-network is a function of only one input variable. This property is the key to imposing different hard shape constraints w.r.t. each input variable.

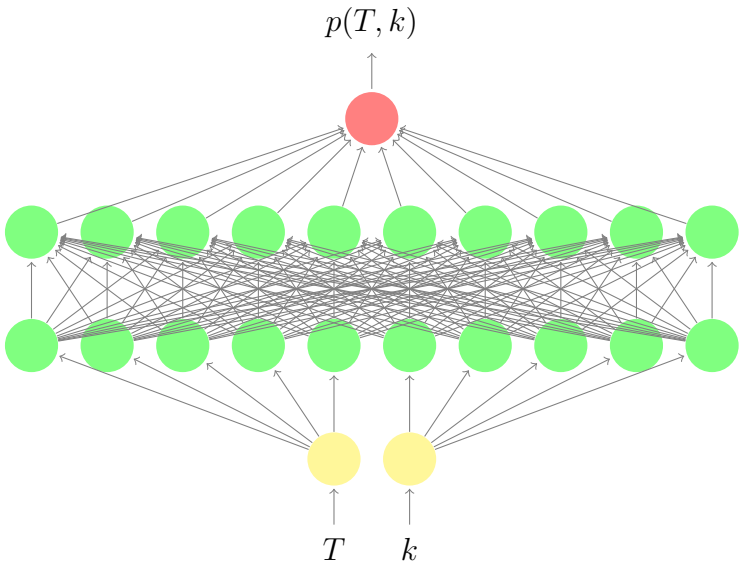


Figure 9: *A shape preserving (sparse) feed forward architecture with one hidden layer containing 10 hidden nodes. The first input variable, T, is only connected to the 5 left most hidden nodes and the second input variable, k, is only connected to the 5 right most hidden nodes.*

B.2 Soft Constraints Approach

However, sparsening the network (i.e. splitting) diminishes the expressiveness of the network, i.e. increases the approximation error. Hence, in what follows, we also consider the so called soft constraints approach using a fully connected network, where the static no arbitrage conditions (12) are favored by penalization, as opposed to imposed to hold exactly in the previous hard constraint approach.

Note that only the "hard constraints" approach theoretically guarantees the absence of arbitrage among predicted put prices. returns a positive function. While soft constraints reduce the risk of static arbitrage in the sense of mismatch between model and market prices, they do not however fully prevent arbitrages in the sense of violations of the shape conditions (12) in the predicted price surface, especially far from the grid nodes of the training set. In particular, the penalties only control the corresponding derivatives at the training points. Compliance with the no-arbitrage constraints on the majority of the points in the test set is due only to the regularity of these derivatives.

C Training Methodology

In general, to fit our fully connected or sparse networks to the available option market prices at a given time, we solve a loss minimization problem of the following form (with $\lambda = 0$ in the non-penalized cases), using observations $\{x_i = (T_i, k_i), p^\star(x_i)\}_{i=1}^n$ of n maturity-strike pairs and the corresponding market put

prices:

$$\underset{\mathbf{W},\mathbf{b}}{\text{Argmin}} \ \frac{1}{n}\sum_{i=1}^{n}\Big(|p^\star(x_i)-p(x_i)|+\lambda^\intercal\phi(x_i)\Big). \tag{15}$$

Here $p=p_{\mathbf{W},\mathbf{b}}$ and $\phi=\phi_{\mathbf{W},\mathbf{b}}$ is a regularization penalty vector

$$\phi:=[(\partial_T p)^-,(\partial^2_{k^2}p)^-].$$

The choice to measure the error $p^\star-p$ under the L_1 norm, rather than L_2 norm, in (15) is motivated by a need to avoid allocating too much weight to the deepest in-the-money options. Note that Ackerer et al. (2019) consider a combination of L_1 and L_2 norms. In a separate experiment, not reported here, we additionally investigated using the market convention of vega weighted option prices, albeit to no effect beyond simply using L_1 regularization.

As typical with neural networks, the loss function is non-convex, possessing many local minima and it is generally difficult to find a global minimum. The penalty vector favors the shape conditions (12). Of course, as soon as penalizations are effectively used (i.e. for $\lambda\neq 0$ in the soft constraints approach), a further difficulty, typically involving grid search, is the need to determine suitable values of the corresponding "Lagrange multipliers"

$$\lambda=(\lambda_1,\lambda_2)\in\mathbb{R}^2_+, \tag{16}$$

ensuring the right balance between fit to the market prices and the targeted constraints.

D Experimental Design

Our training sets are prepared using daily datasets of DAX index European vanilla options of different available strikes and maturities, listed on the 7^{th}, 8^{th} (by default below), and 9^{th}, August 2001. The corresponding values of the underlying are $S=5752.51, 5614.51$ and 5512.28. The associated interest rate and dividend yield curves are constructed from zero-coupon and forward curves, themselves obtained from quotations of standard fixed income linear instruments and from call/put parity applied to the option market prices. Each training set is composed of about 200 option market prices plus the put payoffs for all strikes present in the training grid. For each day of data (see e.g. Figures 10-11), a test set of about 350 points is generated by computing, thanks to a trinomial tree, option prices for a regular grid of strikes and maturities, in the local volatility model calibrated to the corresponding training set by the benchmark Tikhonov calibration method of Crépey (2002) (see VII.VII and VII.VII.A.1, and VII.§3.C).

Each network has two hidden layers, each with 200 neurons per hidden layer. Note that Dugas et al. (2009) only uses one hidden layer. Two was found important in practice in our case. All networks are fitted with an ADAM optimizer. In order to achieve the convergence of the training procedure toward a local minimum of the loss criterion, the learning rate is divided by 10 whenever no improvement in the error on the training set is observed during 100 consecutive epochs. The total number of epochs is limited to $10,000$ because of the limited number of market prices. Thus we opt for a batch learning with numerous epochs.

Moreover, we will assess numerically four different combinations of network architectures and optimization criteria, i.e.

- sparse (i.e. split) network and hard constraints, so $\lambda_1=\lambda_2=0$ in (15)-(16),
- sparse network but soft constraints, i.e. ignoring the non-negative weight restriction in B.1, but using $\lambda_1,\lambda_2>0$ in (15)-(16),
- dense network and soft constraints, i.e. for $\lambda_1,\lambda_2>0$ in (15)-(16),
- dense network and no shape constraints, i.e. $\lambda_1=\lambda_2=0$ in (15)-(16).

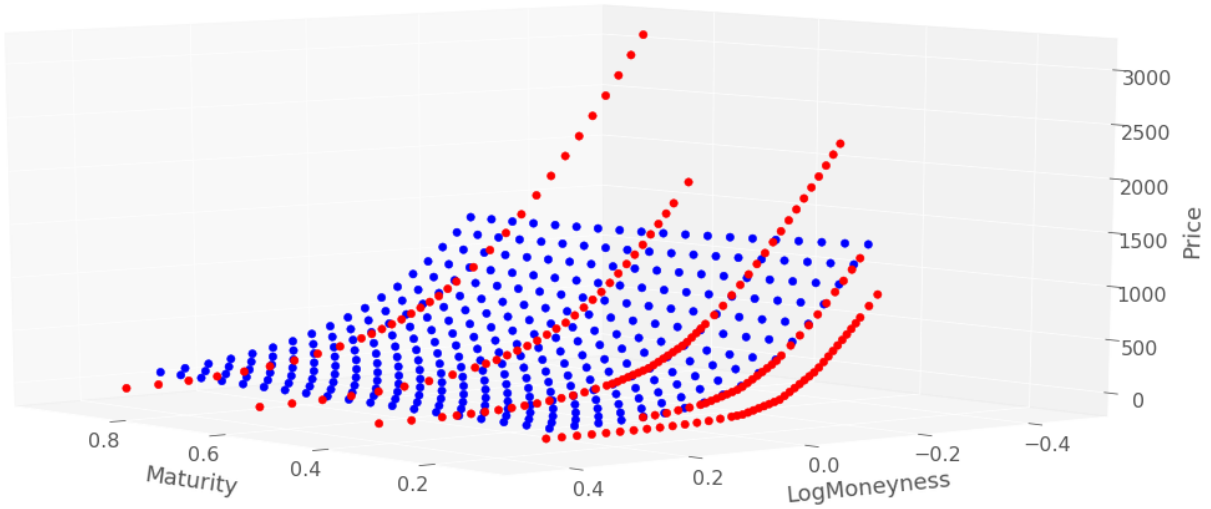


Figure 10: *DAX put prices from training grid (red points) and testing grid (blue points), 8 Aug 2001.*

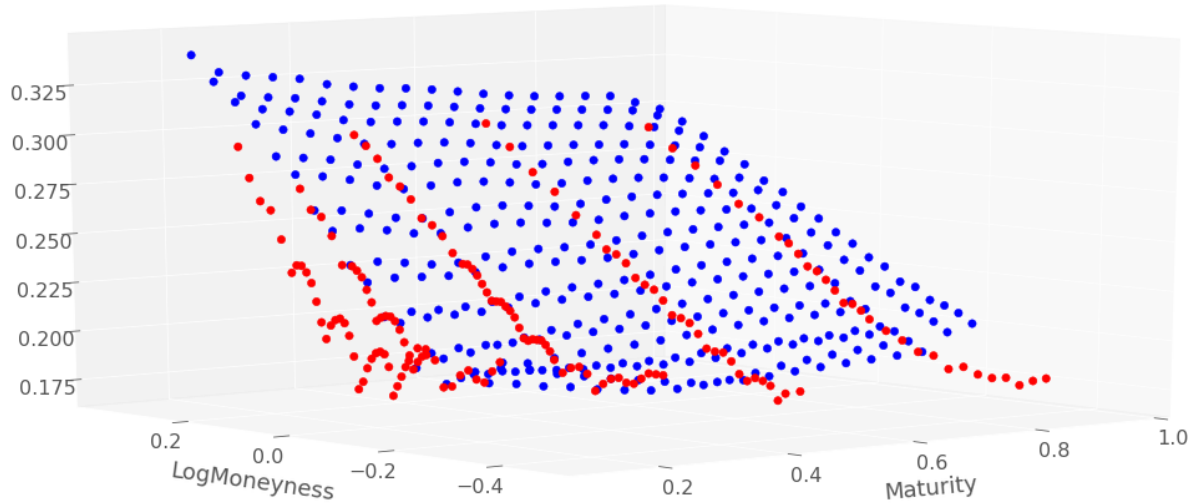


Figure 11: *Same as Figure 10 in implied volatility scale.*

In each case the error between the prices of the calibrated model and the market data are evaluated on both the training and an out-of-sample test set. Unless reported otherwise, all numerical results shown below correspond to test sets.

All our numerical experiments were run under google colab with 13 Gos of Ram and a dual core CPU of 2.2GHz.

E Numerical Results

Table 2 shows the pricing RMSEs for four different combinations of architecture and optimization criteria. For the sparse network with hard constraints, we have $\lambda = 0$. For the sparse and dense networks with soft constraints (i.e. penalization), we set $\lambda = [1.0 \times 10^5, 1.0 \times 10^3]$.

The sparse network with hard constraints is observed to exhibit significant pricing error, which suggests that this approach is too limited in practice to approach market prices. This conclusion is consistent with Ackerer et al. (2019), who choose a soft-constraints approach in the implied volatility approximation (in contrast to our approach which approximates prices).

	Sparse network		Dense network	
	Hard constraints	Soft constraints	Soft constraints	No constraints
Training dataset	28.13	6.87	2.28	2.56
Testing dataset	28.91	4.09	3.53	3.77
Indicative training times	200s	400s	200s	120s

Table 2: *Pricing RMSE (absolute pricing errors) and training times.*

Figure 12 compares the percentage errors in implied volatilities using the sparse network with hard constraints and the dense network with soft constraints approaches, corresponding to the columns 1 and 3 of Table 2. Relative errors with hard constraints exceed 10% on most the training grid oppositely to dense network with soft constraints. This confirms that the error levels of the hard constraints approach are too high to imagine a practical use of this approach: the corresponding model would be immediately arbitrable in relation to the market. Those of the soft constraint approach are much more acceptable, with high errors confined to short maturities or far from the money, i.e. in the region where prices provide little information on volatility.

Table 3 shows the fraction of points in the neural network price surface which violate the static arbitrage conditions. The table compares the same four methods listed in Table 1 applied to training and testing sets. We recall that, in theory, only the sparse network with hard constraints guarantees zero arbitrages. However, we observe that the inclusion of soft constraints reduces the number of arbitrage constraints on the training set when compared with no constraints. The trend is less pronounced for the test set. But in the absence of hard constraints, the effect of adding soft constraints is always preferable than excluding them entirely.

	Sparse network		Dense network	
	Hard constraints	Soft constraints	Soft constraints	No constraints
Training dataset	0	1/254	0	63/254
Testing dataset	0	2/360	0	44/360

Table 3: *The fraction of static arbitrage violations.*

E.1 Further Diagnostic Results

Figure 13 shows the convergence of the loss function against the number of epochs using either hard constraints or soft constraints. The spikes trigger decays of the learning rates so that the training

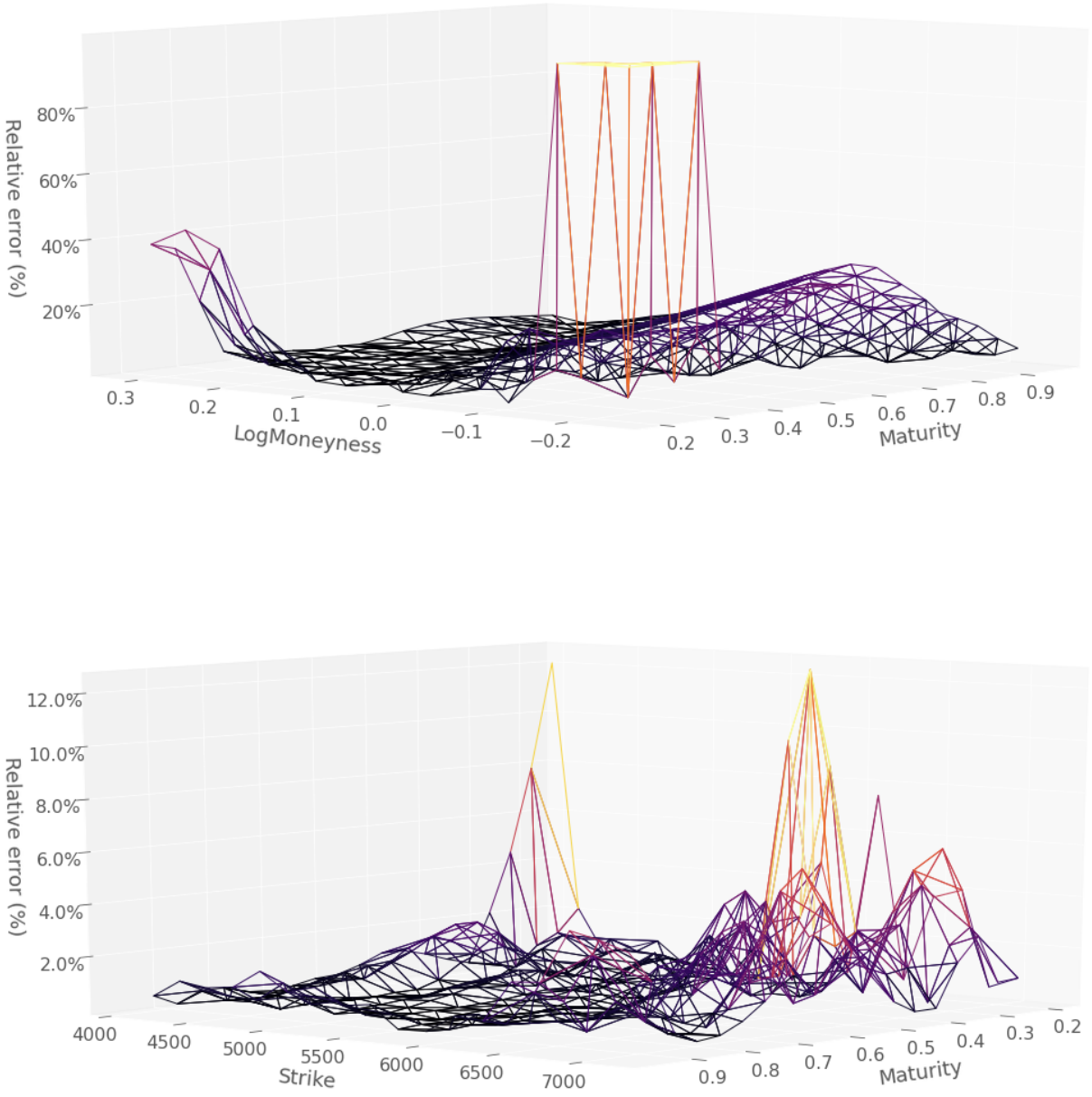


Figure 12: *Percentage relative error in the implied volatilities using (top) hard constraints (bottom) dense networks with soft constraints.*

procedure can converge toward a local minimum of the loss criterion (cf. D). We observe that the loss function converges to a much smaller value using a dense network with soft constraints and that either approach converge in at most 2000 epochs.

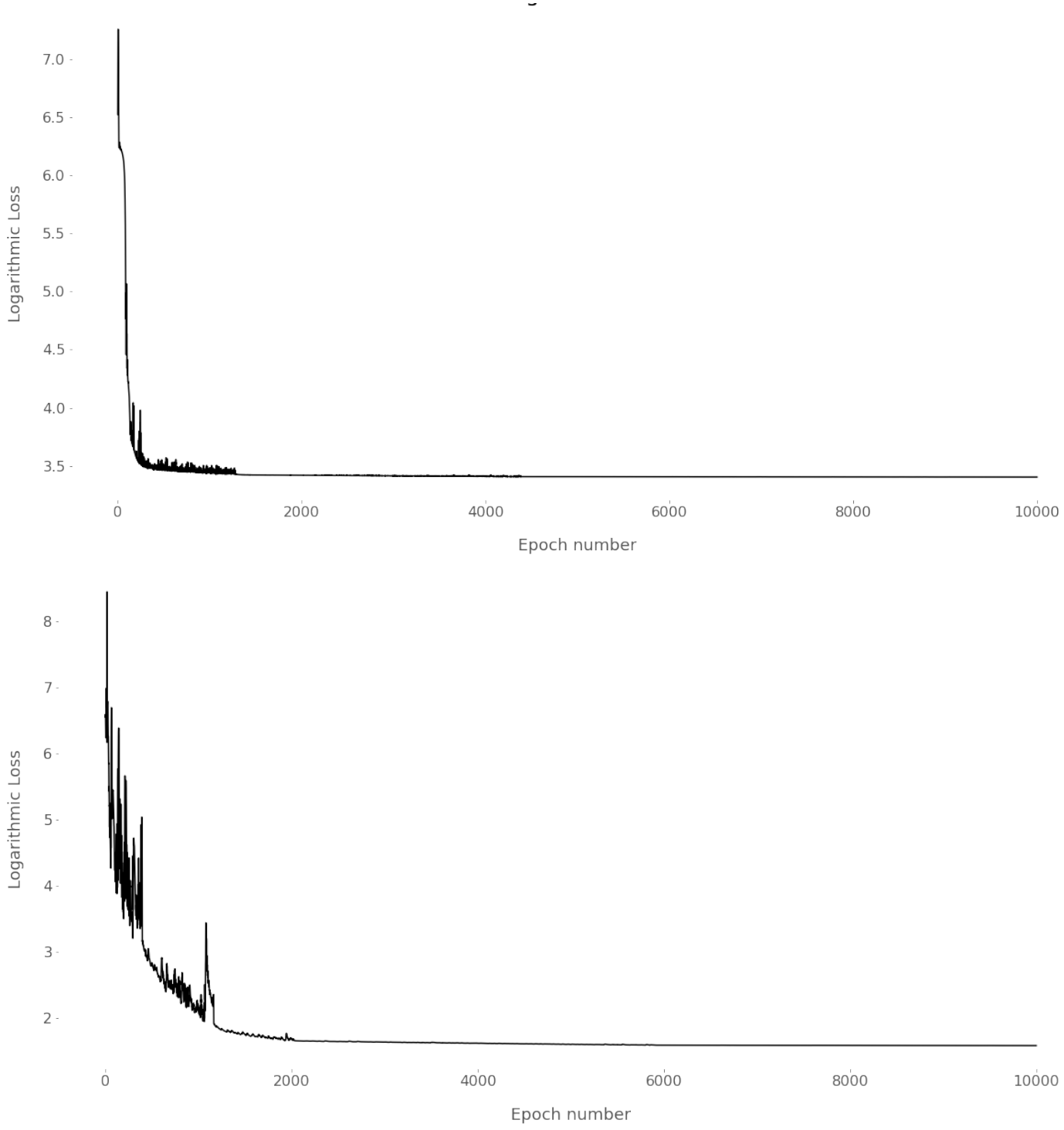


Figure 13: *Logarithmic RMSE through epochs (top) hard constraints (bottom) dense networks with soft constraints.*

Table 4 provides some further insight into the effect of architectural parameters, although it is not intended to be an exhaustive study. Here, only the number of units in the hidden layers is varied, while keeping all other parameters except the learning rate fixed, to study the effect on error in the price and implied volatility surfaces. The price RMSE for the testing set primarily provides justification for the choice of 200 hidden units per layer: the RMSE is 3.55. We further observe the effect of reduced pricing error on the implied volatility surface: 0.0036 is the lowest RMSE of the implied volatility test surface across all parameter values.

# Hidden Units	Surface	RMSE	
		Training	Testing
50	Price	3.01	3.60
	Impl. Vol.	0.0173	0.0046
100	Price	3.14	3.66
	Impl. Vol.	0.0304	0.0049
200	Price	2.73	3.55
	Impl. Vol.	0.0181	0.0036
300	Price	2.84	3.88
	Impl. Vol.	0.0180	0.0050
400	Price	2.88	3.56
	Impl. Vol.	0.0660	0.0798

Table 4: *Sensitivity of the errors to the number of hidden units. Note that these results are generated using the dense network with soft constraint.*

Table 5 shows the pricing RMSEs resulting from the application of different stochastic gradient descent algorithms under the soft constraints approach with dense network. ADAM (our choice everywhere else in this section, cf. the next-to-last column in Table 1) and RMSProp (root mean square propagation, another well known SGD procedure) exhibit a comparable performance. A Nesterov accelerated gradient procedure, with momentum parameter set to 0.9 as standard, obtains much less favorable results. As opposed to ADAM and RMSProp, Nesterov accelerated momentum does not reduce the learning rate during the optimization.

	Train RMSE	Test RMSE
ADAM	2.48	3.36
Nesterov accelerated gradient	5.67	6.92
RMSProp	2.76	3.66

Table 5: Pricing RMSEs corresponding to different stochastic gradient descents (soft constraints approach with dense network).

§3 Neural Net Regression

We consider an acceleration technique for the computation by simulation and neural net regression of conditional expectations of functionals of processes (X, Y), where an exogenous component Y (Markov by itself) is time-consuming to simulate, while the endogenous component X (jointly Markov with Y) is quick to simulate (given Y), but responsible for most of the variance of the simulated payoff. Financial applications in equity world include option pricing in rough volatility models, with rough volatility Y of a stock X. Here we rather consider highly dimensional XVA example with market risk factors Y and bank client defaults related processes X. The idea is then to over-simulate X with respect to Y. We propose a way to optimize the proposed hierarchical simulation scheme. The resulting algorithm is implemented on a Graphics Processing Unit (GPU) combining Python/CUDA and learning with PyTorch for its proximity to the CUDA programming model. We explain the various optimizations used for our implementation on GPU. A CVA benchmarking case study of the method with a reference nested Monte Carlo approach shows that the over-simulation layer is key to the success of the overall deep learning approach.

Remark 1 *Although our CVA case study only covers quadratic risk minimization (for benchmarking purposes), the approach and the proofs of this paper are valid for more general loss functions and apply to the learning of any elicitable statistics. In particular, via the Rockafellar and Uryasev (2000) representation of value-at-risk and expected shortfall of a given loss (random variable) in terms of "far*

out-of-the-money call options" on that loss, our hierarchical simulation approach is also relevant for learning value-at-risk and expected shortfall in hybrid mark-to-market and default simulation setups. Such an approach is even particularly relevant in these cases, where the fact that X is responsible for most of the variance of the payoff is then intrinsic to the far out-of-the-money feature of the corresponding "option".

A Neural Regression Setup

The state spaces of X and Y are taken as $\mathbb{R}^p$ and $\mathbb{R}^q$, for some positive integers p and q. In the (default risk) case of a Markov chain like component X, referred to hereafter as the Markov chain X case (but with transition intensities modulated by Y), we assume, without loss of generality in this case, that X evolves on the vertices $\{0,1\}^p$ of the unit cube in $\mathbb{R}^p$. We take the problem after discretisation of time (if the latter was continuous in the first place), for a time step set to one year for ease of notation.

We then consider $(X_i)_{0\leq i\leq n}$ and $(Y_i)_{0\leq i\leq n}$ as discrete-time processes on the time grid. Our goal is to estimate, for every i, conditional expectations of the form

$$\Pi_i = \mathbb{E}[\xi_{i,n}\,|X_i, Y_i], \tag{17}$$

where

$$\xi_{i,n} = f_i(X_i, \ldots, X_n, Y_i, \ldots, Y_n). \tag{18}$$

Here f_i is a measurable real function such that $\xi_{i,n}$ is a square-integrable random variable.

Conditional expectations such as (17) can be estimated via linear regression using a finite sample. This is ubiquitous in quantitative finance since the Bermudan Monte Carlo papers of Tsitsiklis and Van Roy (2001) and Longstaff and Schwartz (2001). In order to estimate the conditional expectation in (17), one draws i.i.d. samples $\{(X_i^\iota, Y_i^\iota, \xi_{i,n}^\iota)\}_{\iota\in\mathcal{I}}$ of $(X_i, Y_i, \xi_{i,n})$ where $\mathcal{I}$ is a finite set of indices. Then, given a feature map $\phi : \mathbb{R}^p \times \mathbb{R}^q \to \mathbb{R}^m$ (for some positive integer m), one linearly regresses $\{\xi_{i,n}^\iota\}_{\iota\in\mathcal{I}}$ against $\{\phi(X_i^\iota, Y_i^\iota)\}_{\iota\in\mathcal{I}}$, solving for

$$\widehat{w}_i \in \operatorname*{Argmin}_{w_i\in\mathbb{R}^m} \sum_{\iota\in\mathcal{I}} (\xi_{i,n}^\iota - w_i^\top \phi(X_i^\iota, Y_i^\iota))^2. \tag{19}$$

One then uses $\widehat{w}_i^\top \phi(X_i, Y_i)$ as an approximation for Π_i.

The above procedure is justified by the characterization, in the square integrable case, of conditional expectations as orthogonal projections, i.e.

$$\mathbb{E}[\xi_{i,n}\,|X_i, Y_i] = \varphi_i^\star(X_i, Y_i) \quad \text{a.s.},$$

where, denoting by $\mathcal{B}(E)$ the set of Borel measurable real functions on a metric space E,

$$\varphi_i^\star \in \operatorname*{Argmin}_{\varphi_i\in\mathcal{B}(\mathbb{R}^p\times\mathbb{R}^q)} \mathbb{E}[(\xi_{i,n} - \varphi_i(X_i, Y_i))^2]. \tag{20}$$

One recovers the linear regression formulation (19) by approximating the expectation by an empirical mean and restricting the search space to the functions of the form $\mathbb{R}^p \times \mathbb{R}^q \ni (x, y) \mapsto w_i^\top \phi(x, y)$, where $w_i \in \mathbb{R}^m$.

A.1 Neural Net Parameterization

Linear regression by means of a priori, explicit factors has a reasonable chance of success when $\varphi_i^\star$ is simple enough and the feature mapping ϕ can be judiciously chosen, usually from expert knowledge. This is however not always the case, e.g. when considering portfolio-wide XVA metrics, which exhibit non-trivial dependencies on the many risk factors being regressed against. It is then impossible to manually devise a satisfactory feature mapping ϕ.

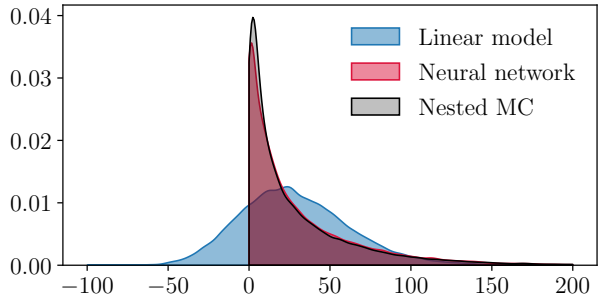


Figure 14: Density plot of the CVA of a vanilla call, at mid-life of the option.

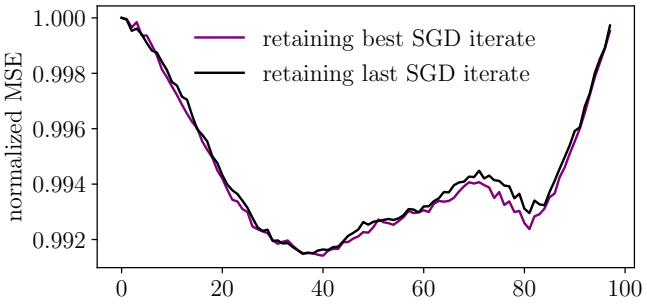


Figure 15: Out-of-sample MSEs against labels $\xi_{i,n}$ at different time-steps divided by the variance of the labels.

Figure 14 shows how a linear regression with the raw risk factors as features fails for even a simple portfolio comprised of a call option, while the neural net estimator almost matches with the nested Monte Carlo estimator (see Sections B and D for more numerical details).

In the Markov chain X case, we face the additional peculiarity of a hybrid regression setting, in view of the discrete and continuous natures of the X and Y model components.

Neural networks (Goodfellow et al., 2016) propose an alternative way to parameterize and learn the feature map. Let $\mathcal{NN}_{p+q,h,u,\varsigma}$ denote the set of functions of the form

$$\mathbb{R}^{p+q} \ni z \mapsto \zeta(z; W^{[h+1]}, \dots, W^{[1]}, b^{[h+1]}, \dots, b^{[1]}) = \zeta^{[h+1]}(z; W, b)$$

where $W^{[h+1]} \in \mathbb{R}^{1\times u}, \dots, W^{[\ell]} \in \mathbb{R}^{u\times u}, \dots, W^{[1]} \in \mathbb{R}^{u\times(p+q)}$ are the weight matrices, $b^{[h+1]} \in \mathbb{R}, \dots, b^{[\ell]} \in \mathbb{R}^u, \dots, b^{[1]} \in \mathbb{R}^u$ are the bias offsets, W and b are the respective concatenations of the $W^{[\ell]}$ and of the $b^{[\ell]}$, ς is an element-wise scalar nonlinearity and, for every $z \in \mathbb{R}^{p+q}$,

$$\begin{aligned}
&\zeta^{[0]}(z; W, b) = z\\
&\zeta^{[\ell]}(z; W, b) = \varsigma(W^{[\ell]}\zeta^{[\ell-1]}(z; W, b) + b^{[\ell]}), \quad \ell = 1, \dots, h\\
&\zeta^{[h+1]}(z; W, b) = W^{[h+1]}\zeta^{(h)}(z; W, b) + b^{[h+1]}.
\end{aligned}$$

The function $z \mapsto \zeta^{[h+1]}(z; W, b)$ can be seen as a nonlinear feature mapping from $\mathbb{R}^{p+q}$ to $\mathbb{R}^u$, parameterized by $W^{[h+1]}, \dots, W^{[1]}, b^{[h+1]}, \dots, b^{[1]}$ (for a given activation function ς). On top of the set $\mathcal{NN}_{p+q,h,u,\varsigma}$ of real-valued neural networks taking inputs from $\mathbb{R}^{p+q}$, with h hidden layers, u units per hidden layer, and ς as the activation function, we also define

$$\mathcal{NN}^{+}_{p+q,h,u,\varsigma} = \{\mathbb{R}^{p+q} \ni z \mapsto (f(z))^+ + \mu, f \in \mathcal{NN}_{p+q,h,u,\varsigma}, \mu \in \mathbb{R}\}. \tag{21}$$

This specification ensures positivity of the output when the additive constant μ is nonnegative and is useful for learning positive (e.g. XVA) functions. The additive constant μ is introduced in order to improve the fit of the first moment and hence reduce the bias.

In what follows, we identify $\mathbb{R}^{p+q}$ with $\mathbb{R}^p \times \mathbb{R}^q$ and write $\phi(z)$ or $\phi(x, y)$ interchangeably, where z is the concatenation of x and y, for every function ϕ defined over $\mathbb{R}^{p+q}$ or $\mathbb{R}^p \times \mathbb{R}^q$, and for every $x \in \mathbb{R}^p$ and $y \in \mathbb{R}^q$.

A.2 Training Algorithm

Learning the conditional expectation (17) in a positive neural net search space consists in applying the same empirical risk minimization (19) approximation as in linear regression, using this time $\mathcal{NN}^{+}_{p+q,h,u,\varsigma}$ as the search space, i.e. solving for

$$\widehat{\varphi}_i \in \operatorname*{Argmin}_{\varphi_i \in \mathcal{NN}^{+}_{p+q,h,u,\varsigma}} \sum_{\iota\in\mathcal{I}} (\xi^{\iota}_{i,n} - \varphi_i(X^{\iota}_i, Y^{\iota}_i))^2. \tag{22}$$

This is achieved by using an iterative gradient-based optimization algorithm, which we will assume to be mini-batch stochastic gradient descent.

In the context of learning a positive output (e.g. an XVA), the addition of a ReLU activation $(\cdot)^+$ at the output layer in (21) can jeopardize the learning as the gradient may vanish at a certain SGD iteration and the parameters are then frozen irrespective of the number of subsequent iterations. Thus, for more stability of the learning procedure, we first perform the first half of SGD steps on the network without the ReLU at the output layer. Then, still without the ReLU, we fine-tune the weights of the output layer by optimizing with respect to those weights only (freezing the weights of the hidden layers), which can be done in closed form in the case of quadratic risk minimization.

Remark 2 *This step isn't achievable in closed-form in the case of, for example, quantile regression*[4]*. However, even in this case, the optimization problem is still convex and as such easier to solve numerically.*

Finally, we restore the ReLU at the output layer and we finish the last half of the SGD iterations.

We also chose to retain the best set of parameters among those explored during the SGD iterations. Figure 15 shows the corresponding improvement in generalization when applied in the context of the case study of Sections B and D.

The ensuing learning scheme is detailed in Algorithm 0. Note that we presented vanilla SGD iterations only for the sake of simplicity. In practice, accelerated SGD methods like Adam (Kingma and Ba (2014)) are used instead.

A.3 Backward Learning

In the setup of the path-wise pricing problem (17), at each pricing time i, a separate learning problem is solved by Algorithm 0. Since the algorithm returns for each problem a local minimum, it is possible to end up with an approximation of the pricing function $\mathbb{E}[\xi_{i,n}\,|X_i = x, Y_i = y]$ (cf. (17)) with noisy paths (i.e. with respect to time i) if the local minima are not close to each other, even for fixed x and y. Yet, for two consecutive time-steps i and $i+1$, the learning problems are similar. One possible refinement is, after having learned Π_{i+1}, to initialize the parameters of the network at time i with the parameters of the network trained at time $i+1$. This not only smoothes the results across regression times, but also accelerates convergence.

We obtain an algorithm which starts the learnings at time step n and, proceeding backward in time until time step 1, reuses each time the previous solution as an initialization for the next learning. The ensuing backward learning scheme is detailed in Algorithm 1. This process of reusing knowledge from a different but related learning task can be seen as a form of transfer learning (Pan and Yang, 2009; Bozinovski, 2020).

Remark 3 *A variation on the above would be forward learning. We favor the backward learning scheme because it is the only one that is amenable to more general backward stochastic differential equations, such as the equations for the FVA and the KVA in Crépey, Sabbagh, and Song (2020). In addition, in these XVA applications, the labels/features corresponding to times i close to the final maturity n of the portfolio have a lower/higher variance. Hence the training task corresponding to a higher i is easier.*

A.4 Separable Case

Next we present a fine-tuning which is applicable when $\xi_{i,n} = \sum_{j=1}^{p} \xi_{i,n}^{(j)}$, where, for every $1 \le j \le p$, denoting by $x^{(j)}$ the j^{th} component of $x \in \mathbb{R}^p$, one has (cf. (18))

$$\xi_{i,n}^{(j)} = f_i^{(j)}(X_i^{(j)}, \dots, X_n^{(j)}, Y_i, \dots, Y_n)$$

[4] cf. Remark 1.

Algorithm 0: Baseline learning scheme for training at a given time-step i

name : BaseAlg

input : $\{(X_i^\iota, Y_i^\iota, \xi_{i,n}^\iota), \iota \in \mathcal{I}\}$, a partition $\mathcal{B}$ of $\mathcal{I}$, a number of epochs $E \in \mathbb{N}^\star$, a learning rate $\eta > 0$, initial values for the network parameters W, b and μ

output: Trained parameters W_{best}, b_{best} and μ_{best}

define $\mathcal{L}(W, b, \mu, \text{batch}, \text{pos}) = \begin{cases} \frac{1}{|\text{batch}|} \sum_{\iota \in \text{batch}} (\zeta^{[h+1]}(X_i^\iota, Y_i^\iota; W, b) + \mu - \xi_{i,n}^\iota)^2 & \text{if pos} = 0 \\ \frac{1}{|\text{batch}|} \sum_{\iota \in \text{batch}} ((\zeta^{[h+1]}(X_i^\iota, Y_i^\iota; W, b))^+ + \mu - \xi_{i,n}^\iota)^2 & \text{if pos} = 1 \end{cases}$

$\mathcal{L}_{\text{best}} \leftarrow \infty$, pos $\leftarrow 0$

for *epoch* $= 1, \ldots, E$ **do** `// loop over epochs`

  for *batch* $\in \mathcal{B}$ **do** `// loop over batches`

    for $\ell = 1, \ldots, h+1$ **do**

      $W^{[\ell]} \leftarrow W^{[\ell]} - \eta \nabla_{W^{[\ell]}} \mathcal{L}(W, b, \mu, \text{batch}, \text{pos})$

      $b^{[\ell]} \leftarrow b^{[\ell]} - \eta \nabla_{b^{[\ell]}} \mathcal{L}(W, b, \mu, \text{batch}, \text{pos})$

    end

    $\mu \leftarrow \mu - \eta \partial_\mu \mathcal{L}(W, b, \mu, \text{batch}, \text{pos})$

  end

  if *epoch* $= \left\lfloor \frac{E}{2} \right\rfloor$ **then** `// tune weights of last layer`

    $(W^{[h+1]}, b^{[h+1]}) \leftarrow \underset{\widetilde{W}^{[h+1]}, \widetilde{b}^{[h+1]}}{\text{Argmin}} \mathcal{L}(\{W^{[0]}, \ldots, W^{[h]}, \widetilde{W}^{[h+1]}\}, \{b^{[0]}, \ldots, b^{[h]}, \widetilde{b}^{[h+1]}\}, \mu, obs, 0)$

    pos $\leftarrow 1$

  end

  if $\mathcal{L}(W, b, \mu, \mathcal{I}, 1) < \mathcal{L}_{best}$ **then** `// keep track of best parameters`

    $\mathcal{L}_{\text{best}} \leftarrow \mathcal{L}(W, b, \mu, obs, 1)$

    $W_{\text{best}} \leftarrow W$

    $b_{\text{best}} \leftarrow b$

    $\mu_{\text{best}} \leftarrow \mu$

  end

end

Algorithm 1: Backward learning scheme

input : $\{(X_i^\iota, Y_i^\iota, \xi_{i,n}^\iota), \iota \in \mathcal{I}, 1 \le i \le n\}$, a partition $\mathcal{B}$ of $\mathcal{I}$, a number of epochs $E \in \mathbb{N}^\star$, a learning rate $\eta > 0$

output: $\widehat{\varphi}_1, \ldots, \widehat{\varphi}_n$

initialize parameters W_{n+1}, b_{n+1} and μ_{n+1} of the network at terminal time-step n

for $i = n \ldots 1$ **do**

  $W_i, b_i, \mu_i \leftarrow \text{BaseAlg}(\{(X_i^\iota, Y_i^\iota, \xi_{i,n}^\iota), \iota \in \mathcal{I}\}, \mathcal{B}, E, \eta, W_{i+1}, b_{i+1}, \mu_{i+1})$

  $\widehat{\varphi}_i \leftarrow \{x \mapsto \zeta^{[h+1]}(x, y; W_i, b_i) + \mu_i\}$

end

for some real function $f_i^{(j)}$ such that $\xi_{i,n}^{(j)}$ is square-integrable. Then

$$\mathbb{E}[\xi_{i,n}^{(j)}|X_i, Y_i] = \mathbb{E}[\xi_{i,n}^{(j)}|X_i^{(j)}, Y_i] = \Pi_i^{(j)},$$

which can be learned separately for each coordinate j.

In the Markov chain case with state space $\{0,1\}^p$ of X, we can write

$$\Pi_i^{(j)} = \mathbb{E}[\xi_{i,n}^{(j)}|\{X_i^{(j)} = 1\}, Y_i]X_i^{(j)} + \mathbb{E}[\xi_{i,n}^{(j)}|\{X_i^{(j)} = 0\}, Y_i](1 - X_i^{(j)}). \tag{23}$$

Thus, for every i we have two sub-learning problems, respectively conditional on $\{X_i^{(j)} = 1\}$ and $\{X_i^{(j)} = 0\}$, and the feature $X_i^{(j)}$ is no longer needed in the regressions. Algorithms 0 and 1 can be easily adapted to this setting by averaging over the respective samples where $X_i^{(j)} = 1$ and 0 (instead of averaging over the whole dataset as before). A requirement is to have enough samples for both events, but this can be facilitated by the approach presented in Section C.

Separability comes in handy when learning for example a CVA or an MVA for each counterparty of a bank (whether default indicator based as in (28) or default intensity based as in (30)). However, it is not applicable to FVA computations and KVA computations, which can only be addressed at the level of the overall portfolio of the bank (Albanese et al., 2021).

A.5 Python/CUDA Optimized Implementation Using GPU

Contrary to most use-cases of machine learning where the final product is the trained model and thus execution time is only critical during inference, in the case of learning from simulated data in pricing applications, the training process itself is part of the final product. Hence particular care is needed when writing the training procedures.

We implemented Algorithm 1 using Python programming with the CUDA API (Application Programming Interface). Because the considered problem involves high variances and thus requires a sufficiently large sample size, both training and inference are not easy to achieve in a reasonable execution time. First, we need to leverage the manycore parallel architecture of GPUs that involves streaming multiprocessors, which are used for the simulation, learning and inference phases. All phases are intertwined and performed inline. Hence we need to carefully optimize each part of the algorithm.

On the simulation side, due to their intrinsically parallel nature, Monte Carlo simulations easily lend themselves to parallelization on GPUs. Nevertheless, various optimizations are needed to have achieve a reasonable solution executed within a few seconds (cf. Figure 20 in Section D). We chose to use Python and the CUDA kernels are compiled *just-in-time* using the module numba, which allows to dynamically generate CUDA kernels at run-time.

Regarding learning, we opted for PyTorch for its proximity to the CUDA programming model and its *just-in-time* compiler allowing for static computation graphs and automatic fusion, whenever appropriate, of the kernels associated with the PyTorch operations used by the model.

We used most of the optimization techniques already presented in Abbas-Turki, Diallo, and Crépey (2018), except those related to regressions since these are replaced here by neural networks. We also introduced several additional optimizations, the most important one being to judiciously manage the CPU and GPU memories. A naive solution would involve the CPU/GPU virtual unified memory (NVIDIA Corporation, 2020) and let the compiler choose. However, this usually results in sub-optimal memory accesses. Our choice rather targets an efficient use of the GPU memory space, a reduction of CPU/GPU transfer and an optimized transfer when needed. These optimizations and implementation choices are developed in the accompanying Github repository[5].

[5]https://github.com/BouazzaSE/NeuralXVA, see the coverpage of the paper.

B CVA Case Study

We now introduce a CVA case study, to be pursued in Section D. In this context the reference probability measure represents a risk-neutal measure chosen by the market, to which the model is calibrated in mark-to-market terms.

B.1 Market and Credit Model

We consider a bank trading derivative contracts in different economies e with various clients c. The currency corresponding to the economy labeled by 0 is taken as the reference currency. Let there be given the short rate process $r^{\langle e\rangle}$ in each economy e, as well as the exchange rate process $\chi^{\langle e\rangle}$ from the currency of each economy $e \neq 0$ to the reference currency. Each client c of the bank has a stochastic default intensity process $\gamma^{\langle c\rangle}$ and a default-time $\tau^{\langle c\rangle}$. For notational convenience we also define $\chi^{\langle 0\rangle} = 1$ and we denote by $\gamma^{\langle 0\rangle}$ the default intensity of the bank itself.

Continuous-Time Limit For every economy e, the short-rate $r^{\langle e\rangle}$ and the exchange rate $\chi^{\langle e\rangle}$ against the reference currency respectively follow Vasicek and log-normal dynamics

$$\begin{aligned} dr_t^{\langle e\rangle} &= a^{\langle e\rangle}(b^{\langle e\rangle} - r_t^{\langle e\rangle})dt + \sigma^{r,\langle e\rangle} d\widetilde{B}_t^{r,\langle e\rangle} \\ d\log \chi_t^{\langle e\rangle} &= (r_t^{\langle 0\rangle} - r_t^{\langle e\rangle} - \frac{1}{2}|\sigma^{\chi,\langle e\rangle}|^2)dt + \sigma^{\chi,\langle e\rangle} dB_t^{\chi,\langle e\rangle}. \end{aligned}$$

For both the bank ("$c = 0$") and every counterparty c ($\neq 0$), the process $\gamma^{\langle c\rangle}$ (funding spread for $c = 0$ and default intensity for $c \geq 1$) follows CIR dynamics

$$d\gamma_t^{\langle c\rangle} = \alpha^{\langle c\rangle}(\delta^{\langle c\rangle} - \gamma_t^{\langle c\rangle})dt + \nu^{\langle c\rangle}\sqrt{\gamma_t^{\langle c\rangle}} dB_t^{\gamma,\langle c\rangle}.$$

In the above, for every e, $\widetilde{B}^{r,\langle e\rangle}$ is a $\mathbb{Q}^{\langle e\rangle}$-brownian motion and, for every client c and economy e, $B^{\chi,\langle e\rangle}$ and $B^{\gamma,\langle c\rangle}$ are $\mathbb{Q}^{\langle 0\rangle}$-brownian motions. Here $\mathbb{Q}^{\langle e\rangle}$ is the risk-neutral measure corresponding to the numeraire $(\exp(\int_0^\cdot r_s^{\langle e\rangle} ds))_t$, and $a^{\langle\cdot\rangle}$, $b^{\langle\cdot\rangle}$, $\sigma^{\cdot,\langle\cdot\rangle}$, $\alpha^{\langle\cdot\rangle}$, $\delta^{\langle\cdot\rangle}$, $\nu^{\langle\cdot\rangle}$ are model parameters calibrated using liquid market instruments.

By the fundamental theorem of asset pricing, for any asset Z priced in a foreign currency $e \geq 1$, $\exp(-\int_0^\cdot r_s^{\langle e\rangle} ds)Z$ and $\exp(-\int_0^\cdot r_s^{\langle 0\rangle} ds)\chi^{\langle e\rangle} Z$ are martingales with respect to $\mathbb{Q}^{\langle e\rangle}$ and $\mathbb{Q}^{\langle 0\rangle}$ respectively. In particular,

$$\mathbb{E}^{\mathbb{Q}^{\langle e\rangle}}[\exp(-\int_0^t r_s^{\langle e\rangle} ds)Z_t] = Z_0 = \frac{1}{\chi_0^{\langle e\rangle}}\mathbb{E}^{\mathbb{Q}^{\langle 0\rangle}}[\exp(-\int_0^t r_s^{\langle 0\rangle} ds)\chi_t^{\langle e\rangle} Z_t].$$

Thus,

$$\begin{aligned} (\frac{d\mathbb{Q}^{\langle e\rangle}}{d\mathbb{Q}^{\langle 0\rangle}})_t &= \exp(\int_0^t (r_s^{\langle e\rangle} - r_s^{\langle 0\rangle})ds)\frac{\chi_t^{\langle e\rangle}}{\chi_0^{\langle e\rangle}} \\ &= \exp(-\frac{1}{2}\big(\sigma^{\chi,\langle e\rangle}\big)^2 t + \sigma^{\chi,\langle e\rangle} B_t^{\chi,\langle e\rangle}). \end{aligned}$$

Hence, by Girsanov's theorem, if we define $B_t^{r,\langle e\rangle}$ such that:

$$d\widetilde{B}_t^{r,\langle e\rangle} = dB_t^{r,\langle e\rangle} - \sigma^{\chi,\langle e\rangle} d\langle B^{r,\langle e\rangle}, B^{\chi,\langle e\rangle}\rangle_t,$$

then $B_t^{r,\langle e\rangle}$ is a $\mathbb{Q}^{\langle 0\rangle}$-brownian motion. In particular, assuming $d\langle B^{r,\langle e\rangle}, B^{\chi,\langle e\rangle}\rangle_t = \rho^{\langle e\rangle} dt$, we get the following $\mathbb{Q}^{\langle 0\rangle}$ dynamics for the short-rate of economy e:

$$dr_t^{\langle e\rangle} = (a^{\langle e\rangle}(b^{\langle e\rangle} - r_t^{\langle e\rangle}) - \rho^{\langle e\rangle}\sigma^{\chi,\langle e\rangle})dt + \sigma^{r,\langle e\rangle} dB_t^{r,\langle e\rangle}.$$

For every counterparty c, the default time $\tau^{\langle c\rangle}$ can be modeled as a the stopping time $\inf\{t > 0; \int_0^t \gamma_s^{\langle c\rangle} ds \geq \epsilon^{\langle c\rangle}\}$, where $\epsilon^{\langle c\rangle}$ is a standard exponential. That is, for every $t \geq 0$,

$$\mathbb{1}_{\{\tau^{\langle c\rangle} \leq t\}} = 1 \Leftrightarrow \tau^{\langle c\rangle} \leq t \Leftrightarrow \int_0^t \gamma_s^{\langle c\rangle} ds \geq \epsilon^{\langle c\rangle}. \tag{24}$$

We then define our model (X, Y) as the collection X of all the default indicator processes and Y of all the short-rate, FX, and credit spread processes r, χ and γ (except for the formal $\chi^{\langle 0\rangle} = 1$), endowed with the filtration generated by the innovation in the model, i.e. the collection of all the Gaussian and exponential variables involved at the increasing time steps $i \in 1 \dots n$. Note that both Y (by itself) and (X, Y) (jointly) are Markov processes with respect to this filtration.

For the instruments, we assume a book comprised of interest rate swaps which we assume to be priced at par. For a given swap instrument, we denote the set of its reset dates by $\mathcal{R}$ and by t_- and t_+ the reset dates respectively immediately preceding and following t. We assume that successive reset dates are regularly spaced by δ, that the swap is spot starting (ie $0 \in \mathcal{R}$) and that the swap is paying fixed (δs, where s is the swap rate) and receiving floating ($\frac{1}{\mathrm{ZC}_{t_-}(t)} - 1$ where $\mathrm{ZC}_t(t')$ is the price of a zero-coupon bond[6] at time t with maturity t') at each reset date $t \in \mathcal{R} \setminus \{0\}$. Denoting by SP_t the price of the swap at time t in units of the underlying currency[7], we have for all $t \leq \bar{t} := \max \mathcal{R}$:

$$\mathrm{SP}_t = \begin{cases} \dfrac{\mathrm{ZC}_t(t_+)}{\mathrm{ZC}_{t_-}(t_+)} - \mathrm{ZC}_t(\bar{t}) - \delta s \displaystyle\sum_{t' \in \mathcal{R}, t' > t} \mathrm{ZC}_t(t') & \text{if } t \notin \mathcal{R} \setminus \{0\} \\ \dfrac{1}{\mathrm{ZC}_{t_-}(t)} - \mathrm{ZC}_t(\bar{t}) - \delta s (1 + \displaystyle\sum_{t' \in \mathcal{R}, t' > t} \mathrm{ZC}_t(t')) & \text{if } t \in \mathcal{R} \setminus \{0\} \\ 1 - \mathrm{ZC}_0(\bar{t}) - \delta s \displaystyle\sum_{t' \in \mathcal{R} \setminus \{0\}} \mathrm{ZC}_0(t') & \text{if } t = 0 \end{cases}$$

Notice that there is a small non-Markovianity induced by the dependence on the previous reset date, which can be solved by including the short rates of that date among the risk factors.

Discrete Version We then consider an Euler time-discretization of the above setup. We use the same notation for the continuous-time processes and their discrete-time approximations (with time-step equal to 1 year to alleviate the notation). So, all the ε that appear below denoting independent standard Gaussian draws: at each time step j, for each economy e (with, in particular, $\sigma^{\chi}_{\langle 0\rangle} = 0$),

$$r_{j+1}^{\langle e\rangle} - r_j^{\langle e\rangle} = \left(a_{\langle e\rangle}\left(b_{\langle e\rangle} - r_j^{\langle e\rangle}\right) - \rho_{(e)}\sigma^r_{\langle e\rangle}\right) + \sigma^r_{\langle e\rangle}\varepsilon_j^{\langle e\rangle},$$

$$\log \frac{\chi_{j+1}^{\langle e\rangle}}{\chi_j^{\langle e\rangle}} = \left(r_j^{\langle 0\rangle} - r_j^{\langle e\rangle} - \frac{1}{2}\right) + \sigma^{\chi}_{\langle e\rangle}\widetilde{\varepsilon}_j^{\langle e\rangle}$$

and, for each client c,

$$\gamma_{j+1}^{\langle c\rangle} = \left(\gamma_j^{\langle c\rangle} + \alpha_{\langle c\rangle}\left(\delta_{\langle c\rangle} - \gamma_j^{\langle c\rangle}\right) + \nu_{\langle c\rangle}\sqrt{\gamma_j^{\langle c\rangle}}\widetilde{\varepsilon}_j^{\langle c\rangle}\right)^+.$$

The $a_{\langle c\rangle}, b_{\langle e\rangle}, \sigma^r_{\langle e\rangle}, \sigma^{\chi}_{\langle e\rangle}$, and the $\alpha_{\langle c\rangle}, \delta_{\langle c\rangle}, \nu_{\langle c\rangle}$, where c ranges over clients and e over economies, are parameters to be calibrated on market quotes of liquid instruments (except for $\sigma^{\chi}_{\langle 0\rangle} = 0$).

The client default times are such that, for each client c,

$$\tau^{(c)} \leq i \Leftrightarrow \sum_{j+1 \leq i} \frac{(\gamma_{j+1}^{\langle c\rangle}) + (\gamma_j^{\langle c\rangle})}{2} \geq \epsilon^{(c)}, \tag{25}$$

where the $\epsilon^{(c)}$ denote independent standard exponentials.

[6] Note that the price of a zero-coupon bond has a closed-form under our affine short-rate model.

[7] The swap prices are then to be multiplied by the cross-currency exchange rate processes to have all prices in the same reference currency.

Remark 4 *In practice, the Euler-Marayama discretizations above are stepping through a refined simulation time grid. This same grid is also used when integrating numerically some of the above diffusions, e.g. the default intensities in* (25)*, or for defining risk-neutral discount factors* β_i *associated with the reference currency via* $(-\ln \beta_i)$ *given as a numerical integral*[8] *of* $r^{\langle 0\rangle}$ *on* $[0, i]$ *using the fine time grid. Pricing and checking for default events, instead, is only done at the pricing time steps. Hence, although we step through the fine time grid in our discretized diffusions, we only need to store the values of the processes at the pricing time steps.*

B.2 Learning the CVA

We denote by $\mathrm{MtM}_i^{\langle c\rangle}$ the mark-to-market at time i, from the point of view of the bank and in units of the reference currency, of all the contracts with the client c. By mark-to-market we mean trade additive counterparty-risk-free valuation, i.e. the risk-neutral conditional expectation of the future contractually promised cash flows, expressed in units of the reference currency and discounted at the risk-free rate $r^{\langle 0\rangle}$. We restrict ourselves to interest-rate derivatives for which mark-to-market valuation at i is a function of Y_i, by the nature of the cash-flows and the Markov property of Y. The CVA of the bank then corresponds to the risk-neutral conditional expectation of its future risk-free discounted client default losses. Namely, the CVA of the bank at the time step i is given by[9]

$$\mathrm{CVA}_i = \sum_c \mathrm{CVA}_i^{\langle c\rangle} \mathbb{1}_{\{i<\tau^{\langle c\rangle}\}}, \tag{26}$$

for a (pre-default) CVA of the client c such that

$$\mathrm{CVA}_i^{\langle c\rangle} = \mathbb{E}[\sum_{j=i}^{n} \beta_i^{-1}\beta_{j+1}(\mathrm{MtM}_{j+1}^{\langle c\rangle})^+ \mathbb{1}_{j<\tau^{\langle c\rangle}\le j+1} \Big| X_i, Y_i]. \tag{27}$$

Hence $\mathrm{CVA}_i^{\langle c\rangle} = \mathbb{1}_{\{i<\tau^{\langle c\rangle}\}}\varphi_i^{\langle c\rangle}(Y_i)$, where (cf. (20) and (23))

$$\varphi_i^{\langle c\rangle} \in \operatorname*{Argmin}_{\varphi\in\mathcal{B}(\mathbb{R}^q)} \mathbb{E}[(\sum_{j=i}^{n-1} \beta_i^{-1}\beta_{j+1}(\mathrm{MtM}_{j+1}^{\langle c\rangle})^+ \mathbb{1}_{j<\tau^{\langle c\rangle}\le j+1} - \varphi(Y_i))^2 \Big| i<\tau^{(c)}]. \tag{28}$$

We also mention the following intensity-based formula for the CVA of the client c (cf. Albanese et al. (2021, Eq. (60))):

$$\widetilde{\mathrm{CVA}}_i^{\langle c\rangle} = \mathbb{E}\Big[\sum_{j=i}^{n-1} \beta_i^{-1}\beta_j(\mathrm{MtM}_j^{\langle c\rangle})^+\gamma_j^{\langle c\rangle} \exp\big(-\sum_{s=i}^{j-1}\gamma_s^{\langle c\rangle}\big)\Big| Y_i\Big] \mathbb{1}_{\{i<\tau^{(c)}\}}, \tag{29}$$

which converges to the same continuous-time limit as $\mathrm{CVA}_i^{\langle c\rangle}$ when the time discretisation step[10] goes to zero. Hence $\widetilde{\mathrm{CVA}}_i^{\langle c\rangle} = \mathbb{1}_{\{i<\tau^{\langle c\rangle}\}}\widetilde{\varphi}_i^{\langle c\rangle}(Y_i)$, where (cf. (20) and (23))

$$\widetilde{\varphi}_i^{\langle c\rangle} \in \operatorname*{Argmin}_{\varphi\in\mathcal{B}(\mathbb{R}^q)} \mathbb{E}\Big[(\sum_{j=i}^{n-1} \beta_i^{-1}\beta_{j+1}(\mathrm{MtM}_i^{\langle c\rangle})^+\gamma_j^{\langle c\rangle} \exp\big(-\sum_{s=i}^{j-1}\gamma_s^{\langle c\rangle}\big) - \varphi(Y_i))^2 \Big| i<\tau^{\langle c\rangle}\Big]. \tag{30}$$

We reiterate that Algorithm 1 with hierarchical simulation of (X, Y) is generically applicable to all the XVA metrics. The focus on the CVA in our case study is for benchmarking purposes only. Besides, were it for the CVA only, the regression learning scheme with minimal variance would obviously be the

[8] Although it is also possible to jointly simulate exactly $r^{\langle 0\rangle}$ and its integral without the need for numerical integration, see for example Glasserman (2003).

[9] Assuming that the netting set for a given client is the whole set of transactions with this client.

[10] Conventionally set to one in this paper.

one based on (29), where a CVA is computed separately for each client based on its default intensity. At the other extreme of the spectrum, equivalently to (26)-(27), one can rewrite the CVA of the bank using a single expectation conditional on the default states of all clients, as

$$\mathrm{CVA}_i = \mathbb{E}\Big[\sum_c \sum_{j=i}^{n} \beta_i^{-1}\beta_{j+1}(\mathrm{MtM}_{j+1}^{\langle c\rangle})^+ \mathbb{1}_{j<\tau^{\langle c\rangle}\le j+1}\Big| X_i, Y_i\Big]. \tag{31}$$

Hence $\mathrm{CVA}_i = \varphi_i^\star(X_i, Y_i)$, where

$$\varphi_i^\star \in \underset{\varphi\in\mathcal{B}(\mathbb{R}^p\times\mathbb{R}^q)}{\mathrm{Argmin}}\ \mathbb{E}\big[(\sum_c \sum_{j=i}^{n} \beta_i^{-1}\beta_{j+1}(\mathrm{MtM}_{j+1}^{\langle c\rangle})^+ \mathbb{1}_{j<\tau^{\langle c\rangle}\le j+1} - \varphi(X_i, Y_i))^2\big]. \tag{32}$$

On top of the regression schemes (30) and (32) associated with the formulations (29) and (31), another computational alternative in each case is nested Monte Carlo as detailed in Abbas-Turki, Diallo, and Crépey (2018). This variety of approaches will be useful for benchmarking purposes.

B.3 Preliminary Results Using IID Data

In the following experiments, we assume that a bank is trading derivatives in 10 economies with 8 clients. Implementing the discretized mark-to-market and credit model, we get a total of 10 interest rates, 9 cross-currency rates, 8 default intensities. This yields 27 diffusive mark-to-market risk factors and 8 default indicator processes. For time-stepping, we use $n = 100$ pricing time steps and 25 simulation sub-steps per pricing time step (see Remark 4). We consider a portfolio of 500 interest rate swaps with random characteristics (notional, currency and counterparty), the MtMs are thus analytic. All swaps are priced at par at inception. For all the runs of the simulations in this section, whether they be for training or testing, we use $\mu = 16384$ paths for the market risk factors.

We implemented the learning procedure of Algorithm 1 in PyTorch with custom CUDA kernels for label generation during the backward iterations. Moreover we implemented an optimized CUDA benchmark involving nested simulations, using the intensity-based formulation (29) for the inner CVA computations. For the nested Monte Carlo, we used 128 inner paths. The nested Monte Carlo CVA is only computed at few pricing times due to the heavy calculation.

The comparison between the two panels of Figure 16 reveals a difficulty with the neural net learning approach of Algorithm 1 applied to the defaults-based formulation (31) on the basis of i.i.d. simulated data. In this case, represented by the left panel in Figure 16, the network only learns a rather crude and noisy approximation of the CVA conditional to each training time: it is only on the mean that the learned CVA agrees with the nested Monte Carlo estimator; on the tails it largely fails. As visible from the right panel, the CVA learned using the intensity-based formulation, instead, yields satisfactory results on a wide range of quantiles of the targeted distribution.

C Hierarchical Simulation and its Analysis

The above difficulty of learning from defaults based on (31) holds despite of an optimized training scheme. As should always be first scrutiny with machine learning, the problem in fact comes from the data, i.e. from the simulation part in our case. Specifically, a large variance of the estimated population loss function jeopardizes the learning approach, which we address in what follows by a suitable hierarchical simulation approach.

C.1 Variance Contributions using Automatic Relevance Determination

In this part we show how to hierarchize the variance impact of explanatory variables using automatic relevance determination (ARD). As detailed in Rasmussen and Williams (2006, Sections 5.1, 5.4.3, 6.6,

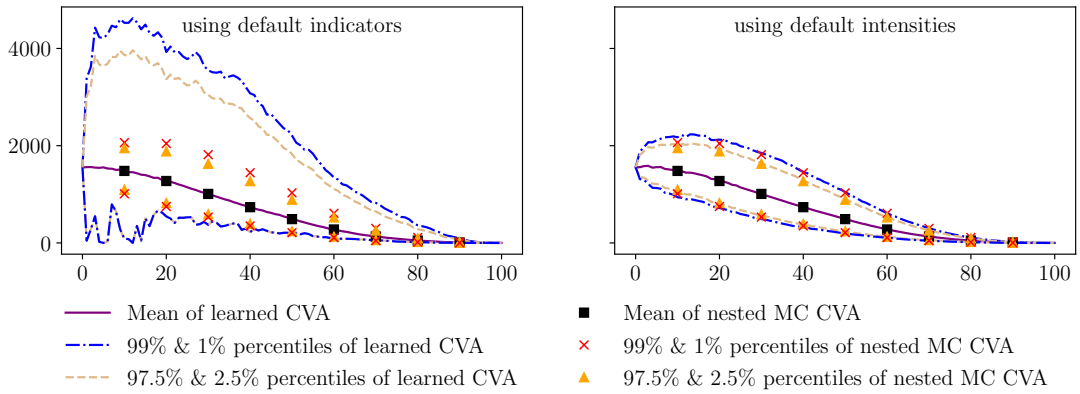


Figure 16: CVA learned using default indicators vs. using default intensities (X axis pricing times, Y axis CVA levels). Statistics computed using out-of-sample paths.

and 8.3.7), ARD is a Bayesian procedure for feature selection and consists in estimating the relevance of the features by maximizing a marginal likelihood. In our setup, we apply a Gaussian process regression based ARD to quantify empirically the impact of the variances of X and Y on that of ξ.

Toward this aim, we treat the vector of the diffusion parameters, denoted by ν, as a latent variable endowed with some instrumental distribution. Given ν, we sample time-averages of the variances of $X_1, \ldots, X_n$, $Y_1, \ldots, Y_n$ and $\xi_{1,n}, \ldots, \xi_{n,n}$,

$$
\begin{aligned}
V(X|\nu) =& \frac{1}{n+1} \sum_{i=0}^{n} \widehat{\mathbb{E}}^{\nu}[(X_i - \widehat{\mathbb{E}}^{\nu}[X_i])^2] \\
V(Y|\nu) =& \frac{1}{n+1} \sum_{i=0}^{n} \widehat{\mathbb{E}}^{\nu}[(Y_i - \widehat{\mathbb{E}}^{\nu}[Y_i])^2] \\
V(\xi|\nu) =& \frac{1}{n+1} \sum_{i=0}^{n} \widehat{\mathbb{E}}^{\nu}[(\xi_{i,n} - \widehat{\mathbb{E}}^{\nu}[\xi_{i,n}])^2]
\end{aligned}
$$

meant componentwise in the vector cases of X and Y, where $\widehat{\mathbb{E}}^{\nu}$ is an empirical average over paths sampled for a given realization ν of the diffusion parameters. Then, based on a finite sample of ν and on the corresponding realizations of the triple $(V(X|\nu), V(Y|\nu), V(\xi|\nu))$, we perform a Gaussian process regression (Rasmussen and Williams, 2006) of $V(\xi|\nu)$ against $V(X|\nu)$ and $V(Y|\nu)$. In this procedure we use an anisotropic kernel $\exp(-\sum_{j=1}^{p} \frac{(v_j - v'_j)^2}{2\lambda_{x,j}^2} - \sum_{j=1}^{q} \frac{(w_j - w'_j)^2}{2\lambda_{y,j}^2})$, where the hyperparameters $\lambda_{x,1}, \ldots, \lambda_{x,p}$ and $\lambda_{y,1}, \ldots, \lambda_{y,q}$ are characteristic length-scales for the corresponding components of $V(X|\nu)$ and $V(Y|\nu)$. Maximizing the marginal likelihood on the dataset allows to recover those length-scales and these can then be interpreted as relevance estimates for our input variables. The higher the inverse length-scale gets, the more the corresponding variable influences the output (payoff variance, in our case).

The above procedure is then itself randomized, i.e. run multiple times on the restricted datasets corresponding to different sub-samplings of ν. This provides a distribution of the fitted hyper-parameters λ in the above, while being also less prone to over-fitting and local minima issues. A similar analysis was used in Bergstra and Bengio (2012) to study the relevance of different neural network hyper-parameters with respect to the validation loss.

Figure 17 reveals the dominance of the impact of the variance of X on that of ξ in the context of our CVA case study, here for a single client of the bank and relying on the CVA representation (28), where Y is the vector of mark-to-market risk factor processes and X is the default indicator process of the client.

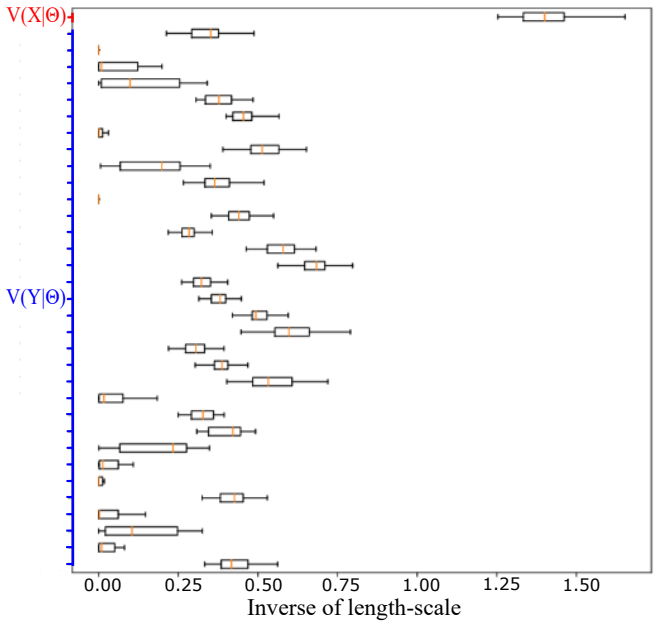


Figure 17: Single client CVA (28): Box-plot of the inverse length-scales obtained by randomized Gaussian process regressions of the conditional variances of the cash flows ξ against the conditional variances of the risk factors X and Y, where conditional here is in reference to the parameters of the model treated as a random vector with a postulated distribution.

C.2 Learning on Hierarchically Simulated Paths

If X contributes more to the variance of ξ than Y, then, in order to improve the efficiency of the associated simulation/regression scheme, an idea is to simulate more realizations of X than Y, even if this means giving up the independence of the simulation setup. More precisely, we simulate μ i.i.d paths $Y^1, \dots, Y^\mu$ of Y and, for every $k \in \{1, \dots, \mu\}$ and $i \in 1\{1, \dots n\}$, we simulate ν i.i.d realizations $X_i^{k,1}, \dots, X_i^{k,\mu}$ of X_i conditional on Y^k. For every i, this yields to a sample $(X_i^{k,l}, Y_i^k, \xi_{i,n}^{k,l})$, $k \in \{1, \dots, \mu\}, l \in \{1, \dots, \nu\}$ of $(X_i, Y_i, \xi_{i,n})$ of size $\mu\nu$, where, within each block k, independence between the $X^{k,l}$ only holds conditionally on Y^k.

Algorithm 1 is then run on the resulting hierarchically simulated dataset by taking $\mathcal{I} = \{1, \dots, \mu\} \times \{1, \dots, \nu\}$, with by convention $Y^{k,l} = Y^k$ for all l. For implementation efficiency reasons pertaining to memory contiguity, the set of indices of the $\imath$-th batch, with $1 \leq \imath \leq |\mathcal{B}|$, is chosen to be $\{(k,l) \in \{1, \dots, \mu\} \times \{1, \dots, \nu\} : (\imath - 1)\frac{|\mathcal{I}|}{|\mathcal{B}|} \leq (l-1)\nu + (k-1) + 1 < \imath\frac{|\mathcal{I}|}{|\mathcal{B}|}\}$.

Hierarchical simulation in the above sense can be thought of as a form of data augmentation procedure (see e.g. Shorten and Khoshgoftaar (2019)), but in a simulation setup where one knows how to generate the data perfectly, hence no discriminator is required. In this framework, the main question is then to which extent one should augment the data, i.e. the choice of the hierarchical simulation parameters μ and ν, which is the focus of the sequel of this section.

Remark 5 *Hierarchical simulation is different in nature from importance sampling that favors particular events, e.g., in a credit risk setup, default vs. survival (see e.g. Carmona and Crépey (2010)). In an XVA setup, for instance, some metrics, like the CVA, need default events for being properly estimated, whereas others, like the FVA, require survival events. Hence what one needs is richness regarding both default and survival events, which is precisely what hierarchical simulation provides.*

C.3 Choosing the Hierarchical Simulation Factor

Assume that simulating Y_i costs P times more than simulating X_i given a path $\{Y_j\}_{j \leq i}$ in terms of computation time. The hierarchical simulation factor ν can be chosen so as to minimize the variance ($\mathbb{V}\text{ar}$) of the loss $\frac{1}{\mu\nu}\sum_{k=1}^{\mu}\sum_{l=1}^{\nu} g_i(\theta, X_i^{k,l}, \dots, X_n^{k,l}, Y_i^k, \dots, Y_n^k)$ with respect to ν, under a budget constraint $\mu(\nu + P) = B$, where g_i is the point-wise loss of our learning task at time-step i, e.g. $g_i(\theta, X_i, \dots, X_n, Y_i, \dots, Y_n) = (\xi_{i,n} - \varphi_\theta(X_i, Y_i))^2$ in our CVA case study, and φ_θ is the neural net (element of $\mathcal{NN}^+_{p+q,h,u,\varsigma}$) with parameters θ.

For ease of notation in this and the next part, we write $g_i(\theta, X^{k,l}, Y^k)$ and $g_i(\theta, X, Y)$ instead of $g_i(\theta, X_i^{k,l}, \ldots, X_n^{k,l}, Y_i^k, \ldots, Y_n^k)$ and $g_i(\theta, X_i, \ldots, X_n, Y_i, \ldots, Y_n)$ (it is then implied that X and Y play formally the role of vectors containing their path from time-step i up to n).

Proposition 1 *The hierarchical simulation factor that minimizes the variance of the loss* $\frac{1}{\mu\nu}\sum_{k=1}^{\mu}\sum_{l=1}^{\nu} g_i(\theta, X^{k,l}, Y^k)$ *with respect to* ν*, subject to the budget constraint* $\mu(\nu + P) = B$*, is*

$$\nu_i^\theta = \sqrt{\frac{Q_i^\theta P}{R_i^\theta}}, \tag{33}$$

where

$$R_i^\theta = \mathbb{C}\mathrm{ov}\big(g_i(\theta, X^{1,1}, Y^1), g_i(\theta, X^{1,2}, Y^1)\big) = \mathbb{V}\mathrm{ar}\big(\mathbb{E}(g_i(\theta, X^{1,1}, Y^1)|Y^1)\big)$$
$$Q_i^\theta = \mathbb{E}\big(\mathbb{V}\mathrm{ar}(g_i(\theta, X^{1,1}, Y^1)|Y^1)\big) = \mathbb{V}\mathrm{ar}(g_i(\theta, X^{1,1}, Y^1)) - R_i^\theta.$$

Proof. After rearranging terms, one can show that

$$\mathbb{V}\mathrm{ar}(\frac{1}{\mu\nu}\sum_{k=1}^{\mu}\sum_{l=1}^{\nu} g_i(\theta, X^{k,l}, Y^k)) = \frac{R_i^\theta}{B}(\frac{1}{\nu}(\nu - \sqrt{\frac{Q_i^\theta P}{R_i^\theta}})^2 + (\sqrt{\frac{Q_i^\theta}{R_i^\theta}} + \sqrt{P})^2),$$

where

$$Q_i^\theta = \mathbb{E}[(g_i(\theta, X^{1,1}, Y^1))^2] - \mathbb{E}[g_i(\theta, X^{1,1}, Y^1)g_i(\theta, X^{1,2}, Y^1)]$$
$$R_i^\theta = \mathbb{E}[g_i(\theta, X^{1,1}, Y^1)g(\theta, X^{1,2}, Y^1)] - (\mathbb{E}[g_i(\theta, X^{1,1}, Y^1)])^2. \blacksquare$$

The quotient

$$\frac{Q_i^\theta}{R_i^\theta} = \frac{\mathbb{V}\mathrm{ar}\big(\mathbb{E}(g_i(\theta, X^{1,1}, Y^1)|Y^1)\big)}{\mathbb{E}\big(\mathbb{V}\mathrm{ar}(g_i(\theta, X^{1,1}, Y^1)|Y^1)\big)}$$

in (33) measures the relative contributions of X and Y to the variance of the loss estimator (note that $Q_i^\theta + R_i^\theta = \mathbb{V}\mathrm{ar}(g_i(\theta, X^{1,1}, Y^1))$, by the total variance formula). To estimate the values of Q_i^θ and of R_i^θ therein, one only needs to simulate $(X^{1,1}, X^{1,2}, Y^1)$, i.e., with respect to the bare simulation of (X, Y), one extra simulation of X conditional on each realization of Y.

As a fixed value of ν has to be chosen throughout all the simulation and training task, for the above result to be of practical use, ν_i^θ has to be reasonably stable with respect to both pricing time steps i and SGD iterations (the transfer learning scheme of Section A.3 is advantageous in this respect in that it stabilizes the learning). If so, it leads to the following:

Heuristic 1 *Choose for* N *the average of the values* N_i^θ *obtained during the SGD iterations and the time steps. Make for* M *the corresponding choice deduced from the budget constraint, i.e.* $M = \frac{B}{N+P}$.

Note that N depends only on P, and M on P and B. If P is not analytically known, it can be deduced from simulation times of experiments corresponding to the same M but different N. Namely, let B and B' the budgets corresponding to configurations (M, N) and (M, N'). We have

$$\frac{B}{B'} = \frac{P + N}{P + N'}. \tag{34}$$

One can deduce P by identifying the ratio in (34) to that of the execution times of (M, N) and (M, N'). For doing so, it is preferable to choose M large enough to avoid time measurement noise that may be due to caching or parallelization of the simulations.

D CVA Case Study Continued

In order to improve the learning (32) of the defaults-based CVA (31) (cf. Section B.3), we apply to it the hierarchical simulation technique. Let $(r^1, \chi^1, \gamma^1), \ldots, (r^\mu, \chi^\mu, \gamma^\mu)$, be i.i.d sample paths of the triple of processes (r, χ, γ). Let $\{\epsilon^{k,l}, 1 \leq k \leq \mu, 1 \leq l \leq \nu\}$ be i.i.d samples of ϵ, the vector defined by the right-hand side in (25) when c ranges over clients. Then we can define $\mu\nu$ samples of the vector of the default indicator processes of the clients at every pricing time i based on (25). Figure 18 illustrates the ensuing simulation scheme for the default indicator of a generic client of the bank, with sampled default times $\tau^{k,l}$.

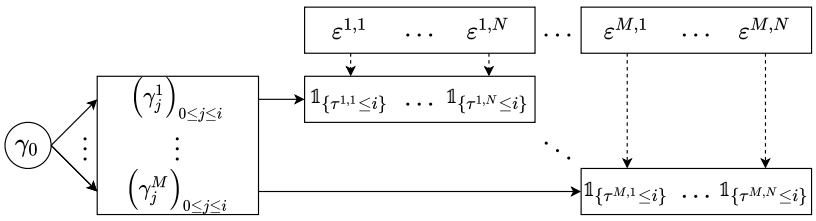


Figure 18: Default simulation scheme

We then learn the CVA process at different time steps for the whole portfolio at once based on (31), trying different combinations of the number of market paths μ and of the hierarchical simulation factor ν. Figures 19 and 20 and show the relative RMSE of the trained neural network against the nested Monte Carlo benchmark[11], the simulation and training times on the GPU and the host RAM usage, as functions of the number of diffusion paths M and of the hierarchical simulation factor N. We already see some configurations $(\frac{1}{2}M, N)$ being better than $(M, \frac{1}{2}N)$, as they achieve a similar accuracy with less memory footprint. For example, $(32768, 1024)$ is better than $(65536, 512)$, given that the former is 30% faster to simulate and price, while also occupying 23% less CPU memory. For the execution times in Figure 20, the runs were done on a server with an Intel Xeon Gold 6248 CPU and 4 Nvidia Tesla V100 GPUs (out of which we used only one). For performance comparison reasons, we use for all (M, N) configurations the same number of epochs $E = 8$ and number of batches $|\mathcal{B}| = 32$, which yields a total of 256 stochastic gradient descent steps during any training task.

The dominance of the impact of the variance of X on that of ξ has been demonstrated in Figure 17. Figure 21 shows the $\sqrt{\frac{Q_i^\theta}{R_i^\theta}}$ (cf. (33)) obtained in the base case $\nu = 1$. With respect to the discussion introducing Heuristic 1, one can note that these are quite stable, of the order of a few tens, with respect to both pricing time steps i and SGD iterations. To obtain from the $\sqrt{\frac{Q_i^\theta}{R_i^\theta}}$ the ν_i^θ in (33), one needs to multiply them by $\sqrt{P}$ (e.g. if a market simulation is 100 times slower than an ensuing default simulation, then the factors displayed in Figure 21 must be multiplied by 10). Solving the equation (34) for P on the basis of the columns $M = 65536$ in Figures 19-20 yields $P \approx 497$. So the numbers in Figure 21 need to be multiplied by $\sqrt{497} \approx 22.3$ to get the optimal ν as per Heuristic 1. In view of this, we expect an optimal hierarchical simulation factor ν of the order of a few hundreds.

More results for $\nu = 1, 32, 64, 128, 512$ are shown in Figure 22, which are to be compared to the right plot in Figure 16 obtained when learning the CVA relying on the intensity-based formula (29). In line with the above expectations, one needs $\nu = 512$ in order to have a close enough match between the 1, 2.5, 97.5 and 99-th percentiles of the CVA learned from defaults and those of the nested Monte Carlo estimator (or of the intensity-based CVA learner represented by the right panel in Figure 16). These results show that hierarchical simulation is essential to a defaults-based CVA learner.

Even after writing an optimized GPU implementation for the nested Monte Carlo estimator, it takes at least 32 minutes on the same hardware as above to compute that estimator for $M = 16384$ and $\sqrt{M} = 128$ inner paths[12], compared to approximately 8 minutes in the case of the learning approach

[11] RMSE restricted to the realizations where the benchmark is non-zero.

[12] However, when doing the error computations and in all plots, we used 1024 inner paths to get benchmark CVAs that

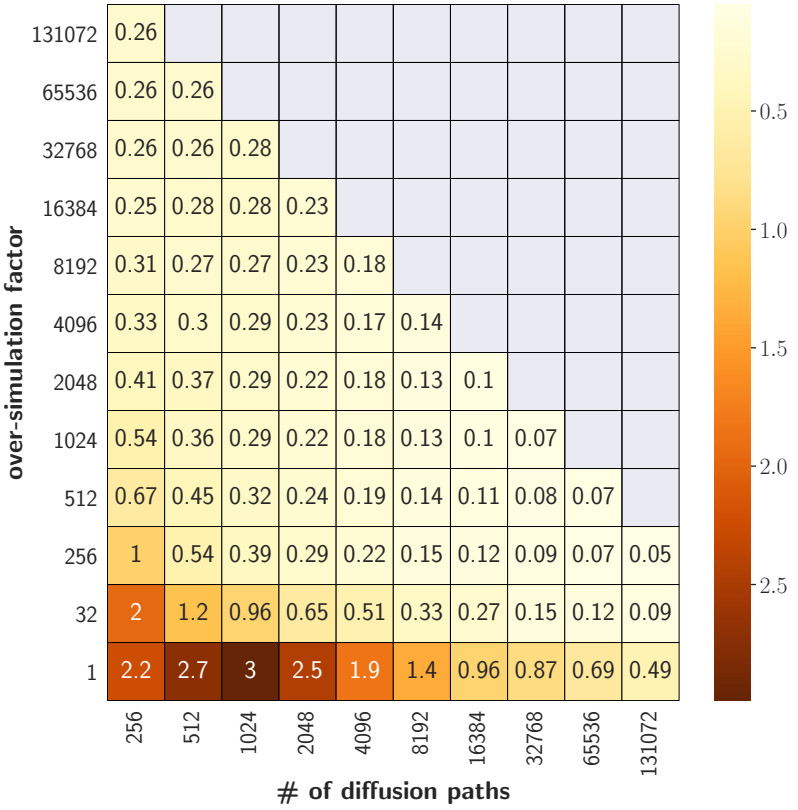


Figure 19: Relative RMSE of the prediction against a nested Monte Carlo benchmark at the pricing time $i = 5$ years, for different combinations of the number of market paths μ and of the hierarchical simulation factor ν.

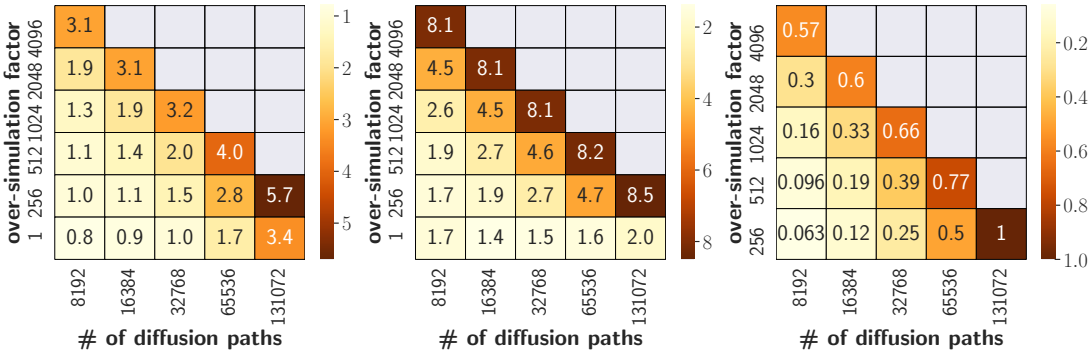


Figure 20: Simulation times in seconds (left) and training times in minutes (center) and RAM usage as a % of its maximum usage over all the displayed experiments (right), for different combinations of the number of market paths μ and of the hierarchical simulation factor ν.

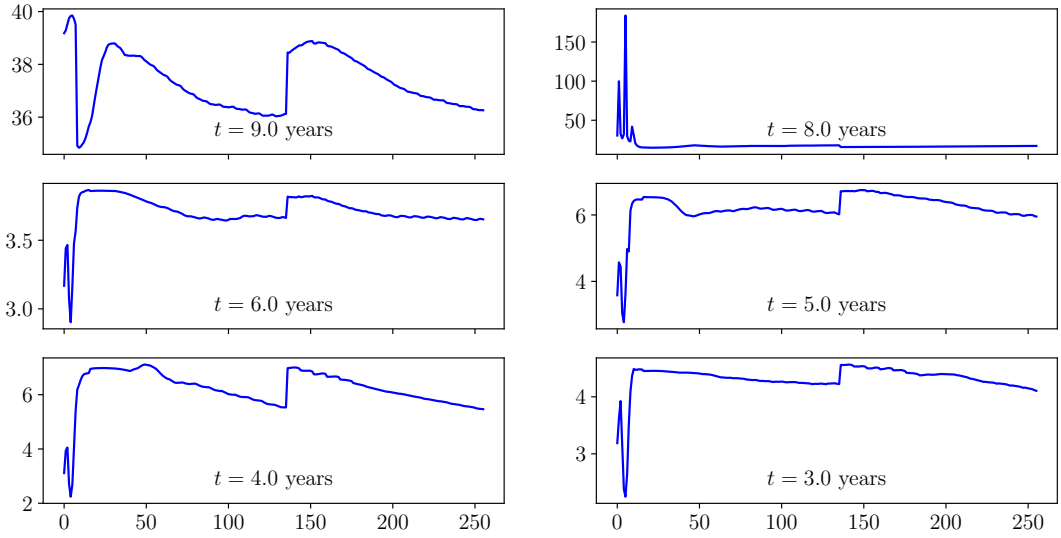


Figure 21: $\sqrt{\frac{Q_i^\theta}{R_i^\theta}}$ at different pricing time steps i (panels) and SGD iterations (x axes).

with a very high hierarchical simulation factor ($N = 2048$). Moreover, going to higher XVA layers such as the FVA and the KVA, a nested Monte Carlo approach would become $\sqrt{\mu}$ times slower per each new layer (Abbas-Turki et al., 2018, Section 3.3), whereas a regression approach would just become slower by a constant each time. In addition, learned XVA metrics can be used in prediction at a very low cost (inference times are fast as inference involves no automatic differentation or stochastic gradient descent), whereas nested Monte Carlo numbers need be recomputed from scratch every time.

D.1 A note on validation

As part of the validation of our approach, we computed a benchmark estimator and compared the learning approach against it by computing L^2 error estimates. In fact, one can compute such L^2 error estimates without necessarily computing a benchmark and thus without performing a slow nested Monte Carlo run. At a given time step i, let $\xi_{i,n}^{(1)}$ and $\xi_{i,n}^{(2)}$ denote two independent copies of $\xi_{i,n}$ conditional on (X_i, Y_i)[13]. For any Borel function $\varphi : \mathbb{R}^p \times \mathbb{R}^q \to \mathbb{R}$ such that $\varphi(X_i, Y_i)$ is square integrable (e.g. a neural net estimate of $\mathbb{E}[\xi_{i,n}|X_i, Y_i]$), we have:

$$\mathbb{E}[(\varphi(X_i, Y_i) - \mathbb{E}[\xi_{i,n}|X_i, Y_i])^2] = \mathbb{E}[\varphi(X_i, Y_i)^2 - (\xi_{i,n}^{(1)} + \xi_{i,n}^{(2)})\varphi(X_i, Y_i) + \xi_{i,n}^{(1)}\xi_{i,n}^{(2)}].$$

The equality follows from the fact that, by conditional independence,

$$\mathbb{E}[\xi_{i,n}|X_i, Y_i]^2 = \mathbb{E}[\xi_{i,n}^{(1)}|X_i, Y_i]\mathbb{E}[\xi_{i,n}^{(2)}|X_i, Y_i] = \mathbb{E}[\xi_{i,n}^{(1)}\xi_{i,n}^{(2)}|X_i, Y_i],$$

followed by an application of the tower rule. Thus, one is able to approximate the L^2 error against the ground truth conditional expectation, without ever observing it, using only two inner paths. This can be used as a very fast validation procedure and as a safeguard in a production environment before using the learned values. A slower but more complete nested Monte Carlo approach is then only needed periodically, e.g. after significant changes in the risk factor models, or to perform more elaborate checks (e.g. tail behavior).

---

are sufficiently accurate *point-wise* and be able to get accurate tail estimates, and nested Monte Carlo simulation thus takes 8 times more computation time.

[13]The conditional independence means that for any Borel bounded functions ϕ_1 and ϕ_2, we have $\mathbb{E}[\phi_1(\xi_{i,n}^{(1)})\phi_2(\xi_{i,n}^{(2)})|X_i, Y_i] = \mathbb{E}[\phi_1(\xi_{i,n}^{(1)})|X_i, Y_i]\mathbb{E}[\phi_2(\xi_{i,n}^{(2)})|X_i, Y_i]$.

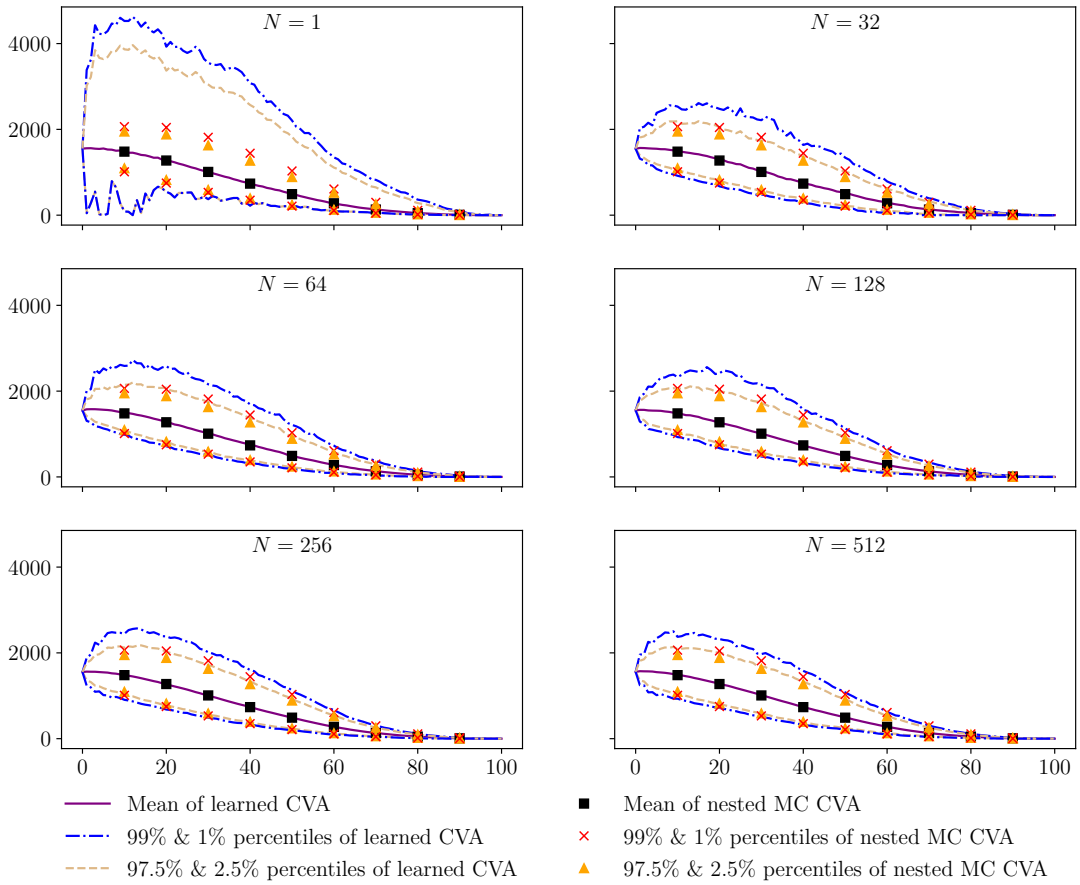


Figure 22: Learned and nested Monte Carlo CVA processes for various hierarchical simulation factors N (x-axis: pricing times, y-axis: CVA levels; $\mu = 16384$). Statistics computed using out-of-sample paths.

D.2 Industry viewpoint

Having in mind a portfolio of the order of one million trades spread over maturities ranging over 50 years and involving a few thousands of clients, the computational CPU ressources typically available in banks hardly allow considering a mark-to-market cube with more than 10^4 paths. Switching to GPU ressources (as required anyway if training path-wise XVA metrics is envisioned) could allow computing a mark-to-market cube with 10^5 to 10^6 paths (in about one hour of computations spread over a few GPUs). Moreover, while we performed our computations using only one GPU, we expect a bank to have access to more than just a single GPU, which would drastically reduce the computation times given that Monte Carlo simulations and stochastic gradient descent can easily be adapted to multi-GPU setups.

However, without the hierarchical simulation technique, the bottom row of Figure 19 confirms the message of the first plot ($\nu = 1$) of Figure 22, according to which a path-wise CVA cannot be learned based on the hybrid market and defaults formulation (31): for $\mu = 16384$ and 131072, the corresponding errors are 96% and 49%. But increasing ν from 1 (bottom row) to 256 brings these errors down to 11% and to 5% (and for $\mu = 1024$ and $\nu = 512$ the error is 1%). As visible from Figure 20, the simulation times are only marginally increased when increasing the hierarchical simulation factor ν (while increasing the number of diffusion paths M increases the simulation time approximately by the same factor of increase in M). These results show that the hierarchical simulation technique is key to the success of a learning approach involving a combination of mark-to-market and default data.

Chapter VII

Calibration Methods

An important financial engineering issue is model calibration. Calibrating a model means finding numerical values of its parameters such that the prices of market instruments computed within the model, at a given time, coincide with their market prices. The simplest example of a calibration problem was encountered in II.§5, when we discussed the notions of the implied volatility of an option and the implied correlation of a CDO tranche. In these cases the calibration problem is easy since there is only one parameter to calibrate to only one market quote.

Calibration thus corresponds to estimation of a model. However, in finance the term "estimation" specifically refers to statistical estimation, i.e. estimation based on historical data by maximum likelihood or any other statistical procedure. Statistical estimation is thus backward looking, whereas calibration is forward looking, since derivative prices at the current time are based on the pricing views of the market regarding the future dynamics of the underlyings.

Statistical estimation and calibration are complementary to each other. Statistical procedures should at least be used "negatively" for invalidating econometrically unrealistic models. Models exhibiting excessive recalibration leakage, hence requiring too frequent recalibration, or unrealistic reverse stress testing scenarios in the sense of Albanese, Crépey, and Iabichino (2022), should be invalidated as well. The residual model risk in the sense of uncertainty between equally valid, co-calibrated models can be handled through the Bayesian-robust approach sketched in Albanese, Crépey, and Iabichino (2022, Section 4.3).

The Ill-Posed Inverse Calibration Problem The calibration problem is the inverse of the pricing problem: instead of computing prices in a model for given values of its parameters, we compute values of model parameters consistent with observed prices. It is well-known to physicists that inverse problems are ill-posed, where a problem is said to be well-posed if its solution exists, is unique and depends continuously on its input data. Hence a problem is ill-posed for any of the following reasons:

- it admits no solution,
- it admits multiple solutions,
- it has a solution that doesn't depend continuously on the input data.

Except for trivial situations, there exists no model achieving a perfect fit with a full calibration data set, including a zero-coupon curve, expected dividend yield curves on the underlyings and a number of vanilla option prices (sometimes also a few exotics, see Remark 2). But there are typically many models that fit the data within the bid–ask spread. Then, if one perturbs the data (e.g., if the observed prices move from some small amount between today and tomorrow), a numerical solution to the calibration problem tends to switch from one locally best fit solution to another, resulting in numerical instability of the calibrated parameters. In order to get a well-posed problem, we need to introduce some regularization.

§1 Approximate Calibration by Regularized Nonlinear Least Square Methods

The most widely known and applicable stabilization method is Tikhonov regularization. The monograph by Engl, Hanke, and Neubauer (1996b) is the general reference for this subsection.

Let there be given a closed convex nonvoid subset $\mathcal{C}$ of a Hilbert space $\mathcal{H}$, a direct operator

$$\mathcal{H} \supseteq \mathcal{C} \ni \varrho \xrightarrow{\Pi} \Pi(\varrho) \in \mathbb{R}^d,$$

noisy data π^δ, and a prior $\varrho_* \in \mathcal{H}$. In the financial interpretation, Π, ϱ, π^δ and ϱ_* correspond respectively to a pricing functional at the current time (time at which the calibration is performed, say $t = 0$), a set of model parameters, current market prices known up to the bid-ask spread δ and an a priori guess for the set of model parameters. The Tikhonov regularization method for inverting Π at π^δ, or for calibrating the model parameter ϱ given the observation π^δ, consists of:

- reformulating the inverse problem as the following nonlinear least squares problem:

$$\min_{\varrho \in \mathcal{C}} \|\Pi(\varrho) - \pi^{ma}\|^2 \tag{1}$$

  to ensure existence of a solution,

- selecting the solution to this nonlinear least squares problem that minimizes $\|\varrho - \varrho_*\|$, to grant uniqueness, and

- introducing a trade-off between accuracy and regularity, parameterized by a level of regularization $\alpha > 0$, to ensure stability.

More precisely, let us introduce the following cost criterion:

$$J_\alpha^\delta(\varrho) \equiv \left\|\Pi(\varrho) - \pi^\delta\right\|^2 + \alpha \|\varrho - \varrho_*\|_{\mathcal{H}}^2 . \tag{2}$$

Definition 1 *Given α, δ and an additional parameter η, which represents an error tolerance on the minimization, a regularized solution to the inverse problem for Π at π^δ means any model parameter $\varrho_\alpha^{\delta,\eta} \in \mathcal{C}$ such that*

$$J_\alpha^\delta\left(\varrho_\alpha^{\delta,\eta}\right) \leq J_\alpha^\delta(\varrho) + \eta\,,\ \varrho \in \mathcal{C}.$$

We will now see that under suitable assumptions, the regularized inverse problem is well-posed. We first postulate the following continuity assumption on the direct operator Π.

Assumption 1 (Compactness) *$\Pi(\varrho_n)$ converges to $\Pi(\varrho)$ in $\mathbb{R}^d$ if ϱ_n converges to ϱ weakly in $\mathcal{H}$.*

Then regularized solutions exist and we have the following stability result.

Proposition 1 (Stability) *Let $\pi^{\delta_n} \to \pi^\delta$, $\eta_n \to 0$ as $n \to \infty$. Then any sequence of regularized solutions $\varrho_\alpha^{\delta_n,\eta_n}$ admits a subsequence which converges on $\mathcal{H}$ toward a regularized solution $\varrho_\alpha^{\delta,\eta=0}$ as $n \to \infty$.*

An interesting feature of Tikhonov regularization is that the calibration input data do not need to belong to the range of the direct operator for the method to be applicable. However, assuming that the data lie in the range of the model leads to convergence properties of regularized solutions to the inverse problem as $\alpha \to 0$. We thus make the following additional

Assumption 2 (Range property) $\pi \in \Pi(\mathcal{C})$.

By a ϱ_*-solution of the inverse problem for Π at π, we mean any ϱ in the set $\underset{\{\Pi(\varrho)=\pi\}}{\text{Argmin}} \|\varrho - \varrho_*\|_{\mathcal{H}}$ (which is nonempty by the assumption 2). For a proof of the following result, see Theorem 2.3 in Engl, Kunisch, and Neubauer (1989).

Proposition 2 (Convergence) *Let the perturbed parameters $\alpha_n, \delta_n, \eta_n$ and the perturbed data $\pi_n \in \mathbb{R}^d$ satisfy*

$$\begin{array}{ll} (n \in \mathbb{N}) & \|\pi - \pi_n\| \leq \delta_n \\ (n \to \infty) & \alpha_n \,,\ \delta_n^2/\alpha_n \,,\ \eta_n/\alpha_n \quad \longrightarrow \quad 0. \end{array}$$

Then any sequence of regularized solutions $\varrho_{\alpha_n}^{\delta_n,\eta_n}$ admits a subsequence which converges toward a ϱ_*-solution ϱ of the inverse problem for Π at π as $n \to \infty$. In particular, in case a ϱ_*-solution ϱ is unique, then $\varrho_{\alpha_n}^{\delta_n,\eta_n}$ converges to ϱ as $n \to \infty$.

Assuming further regularity of Π, we can get convergence rates uniform over all data $\pi \in \Pi(\mathcal{C})$ "sufficiently close" to the prior ϱ_*, in the sense of the additional condition (3) below. We thus make the following additional regularity assumption on Π.

Assumption 3 (Twice Gateaux differentiability) There exist linear and bilinear forms $d\Pi(\varrho)$ on $\mathcal{H}$ and $d^2\Pi(\varrho)$ on $\mathcal{H} \times \mathcal{H}$, such that

$$\Pi(\varrho + \epsilon h) = \Pi(\varrho) + \epsilon d\Pi(\varrho) \cdot h + \frac{\epsilon^2}{2} d^2\Pi(\varrho)(h,h) + \mathrm{o}\left(\epsilon^2\right)$$

holds for every $\varrho, \varrho + h \in \mathcal{C}$, with, for every $\varrho \in \mathcal{C}$ and $h, h' \in \mathcal{H}$;

$$\|d\Pi(\varrho) \cdot h\| \leq C \|h\|, \ \left\|d^2\Pi(\varrho) \cdot (h,h')\right\| \leq C \|h\|_{\mathcal{H}} \|h'\|_{\mathcal{H}},$$

where the C constant is uniform in $\varrho \in \mathcal{C}$.

In the next statement, the operator

$$\mathbb{R}^d \ni \lambda \overset{d\Pi(\varrho)^*}{\longmapsto} d\Pi(\varrho)^* \cdot \lambda \in \mathcal{H},$$

denotes the adjoint of

$$\mathcal{H} \ni h \overset{d\Pi(\varrho)}{\longmapsto} d\Pi(\varrho) \cdot h \in \mathbb{R}^d,$$

in the sense that $\langle h,\, d\Pi(\varrho)^\star \cdot \lambda \rangle_{\mathcal{H}} = \lambda^\top d\Pi(\varrho) \cdot h$ for every $(\lambda, h) \in \mathbb{R}^d \times \mathcal{H}$. For a proof of the following result, see Theorem 10.4 of Engl, Hanke, and Neubauer (1996b)

Proposition 3 (Convergence Rates) *Assume*

$$\begin{array}{ll} (n \in \mathbb{N}) & \|\pi - \pi_n\| \leq \delta_n \\ (n \to \infty) & \alpha_n \longrightarrow 0 \,,\ \alpha_n \sim \delta_n \,,\ \eta_n = \mathrm{O}\left(\delta_n^2\right). \end{array}$$

Then $\|\varrho_{\alpha_n}^{\delta_n,\eta_n} - \varrho\| = \mathrm{O}(\sqrt{\delta_n})$ for every ϱ_*-solution ϱ of the inverse problem for Π at π such that

$$\varrho - \varrho_* = d\Pi(\varrho)^* \cdot \lambda \tag{3}$$

for λ sufficiently small in $\mathbb{R}^d$. In particular, there exists at most one such ϱ_*-solution ϱ.

An important practical issue is the choice of the regularization parameter α that determines the trade-off between accuracy and regularity in the method. To set α, the main approaches are:

- a priori methods, in which the choice of α only depends on the data noise δ (size of the bid–ask spread),

- more general a posteriori methods, in which α may depend on the data in a less specific way.

In financial applications, the most commonly used method for choosing α is an a posteriori method based on the so-called discrepancy principle, which computes iteratively the greatest α for which $\left\|\Pi\left(\varrho_\alpha^{\delta,\eta}\right) - \pi^\delta\right\|$ doesn't exceed the data noise δ (for given δ, η).

Regularization is effectively used in practice for calibrating nonparametric models. In the case of parametric pricing models, i.e. models with a small number of scalar parameters such as the Heston or the Merton model, the need for regularization is less stringent and the choice of a regularization term is not obvious; the industry standard is then to solve the unregularized nonlinear least squares problem (1).

In any case, with Tikhonov regularization, the calibration of a model is effectively reduced to a nonlinear least squares problem of the form (2), with $\alpha = 0$ if no regularization is used. When it comes to implementation, the minimization problem (2) is discretized, which results in a nonlinear minimization problem on (some subset of) $\mathbb{R}^k$, where k is the number of model parameters to be estimated.

In the case of a convex and differentiable cost criterion J in (2), various gradient descent algorithms are proven to be convergent to the unique minimum of J. Gradient descent algorithms consist of moving at each step by some amount in a direction defined by the gradient ∇J at the current step of the algorithm, in combination with, in conjugate gradient or quasi-Newton algorithms, the gradient(s) ∇J at the previous step(s).

In the context of calibration problems in finance, J is typically not convex in ϱ. Sometimes too, as in the American calibration problem mentioned at the end of A.1, J is only almost everywhere differentiable. In such cases gradient descent algorithms can still be used, but they typically converge to one among many local minima of J.

When the gradient ∇J doesn't exist, or is not computable explicitly or numerically with the required accuracy, an alternative to gradient descent methods is the nonlinear simplex method[1], which uses only the values (as opposed the gradient) of J.

A Extracting the Local Volatility

In what follows, we consider the problem of the stable inference of a local volatility function $\sigma(t, S)$[2] from observed vanilla option prices. The local volatility function thus inferred may then be used for various purposes, such as pricing exotic options and/or greeking consistent with the market, or calibrating more general stochastic volatility models[3].

To test different calibration methods, we consider the DAX index options data set of May 2, 2001, consisting of about 300 European vanilla option prices distributed throughout 6 maturities with moneyness $K/S_0 \in [0.8, 1.2]$, corresponding to the implied volatility surface displayed in the bottom panel of Figure 1.

The local volatility calibration problem is an ill-posed inverse problem. A naive approach, based on numerical differentiation, using the Dupire's formula I.(60), gives a local volatility surface that is both very irregular at a fixed calibration time t (see the top panel in Figure 1) and unstable in t. Since market prices are only available for a finite set of strikes and maturities, the calibration problem is also under-determined.

[1] Not to be confused with Danzig's simplex linear programming algorithm.
[2] see I.§2.B.
[3] see II.§4 and Gatheral (2011).

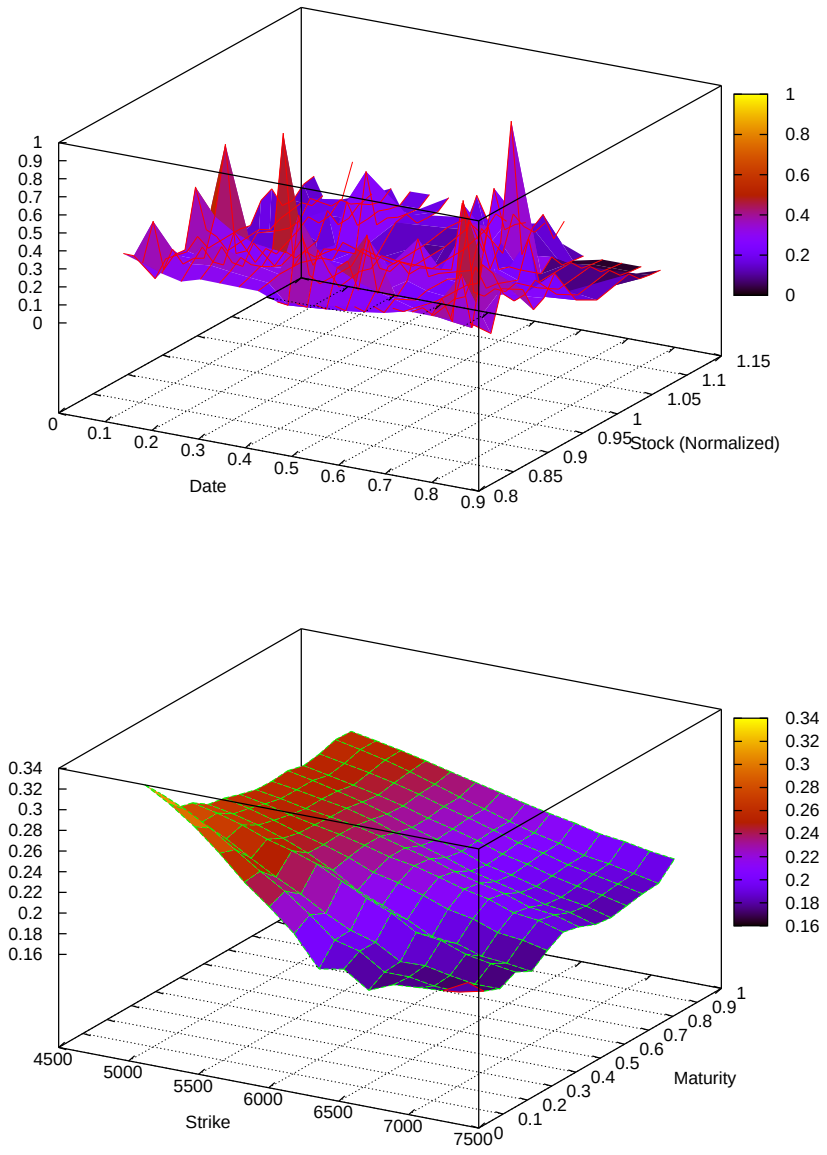


Figure 1: (top) *Local volatility surface obtained from the Dupire formula by numerical differentiation using the implied volatility surface below as input data;* (bottom) *Implied volatility surface corresponding to the DAX index options data set of May 2, 2001.*

A.1 Tikhonov Regularization

To stabilize the local volatility calibration problem, we can reformulate it in the form of the following nonlinear minimization problem:

$$\min_{\{\sigma\equiv\sigma(\cdot,\cdot);\underline{\sigma}\le\sigma\le\overline{\sigma}\}} J(\sigma) = \|\Pi(\sigma) - \pi\|^2 + \alpha\|\sigma - \sigma_*\|^2_{\mathcal{H}^1}, \tag{4}$$

where:

- α is a positive regularization parameter,
- the bounds $\underline{\sigma}$ and $\overline{\sigma}$ are positive constants,
- π is a vector of vanilla option prices observed in the market at the calibration time $t = 0$,
- $\Pi(\sigma)$ is the corresponding vector of prices in the local volatility model with volatility function σ,
- σ_* is a prior on σ and
- $\|u\|^2_{\mathcal{H}^1} \equiv \int_0^\infty \int_0^\infty (u(t,S)^2 + (\partial_t u(t,S))^2 + (\partial_S u(t,S))^2)\, dtdS$.

Stability, convergence and convergence rates for this formulation of the local volatility calibration problem are established in Crépey (2003a). A trinomial tree implementation developed in Crépey (2003b) draws its efficiency from an exact computation of the gradient of the (discretized) cost criterion J in (4). Figure 2 displays the local volatility surface thus calibrated, along with the accuracy[4] of the calibration, using the implied volatility surface of Figure 1 (bottom panel) as calibration input data.

This approach can also be extended to the calibration of a local volatility function using American option quotes (Crépey, 2003b).

A.2 Entropic Regularization

As an alternative, Samperi (2002) uses entropic regularization, rewriting the calibration problem as the following nonlinear minimization problem[5]:

$$\min_{\{\sigma\equiv\sigma(\cdot,\cdot);\underline{\sigma}\le\sigma\le\overline{\sigma}\}} J(\sigma) = \|\Pi(\sigma) - \pi\|^2 + \alpha\mathbb{E}\int_0^\infty (\sigma(t,S_t) - \sigma_*(t,S_t))^2\, dt. \tag{5}$$

Using a dual formulation, the minimization problem (5) can be solved in time $\mathrm{O}(d)$, where d is the number of options in the calibration data set, versus $\mathrm{O}(n^2)$ in the case of Tikhonov regularization implemented on a trinomial tree with n time steps. The numerical solution is thus typically faster than by Tikhonov regularization. However, it is also less stable, since the regularization term doesn't involve the gradient, but only the values, of $(\sigma - \sigma_*)$. See Figure 3, which displays the results obtained by this method for the data set of Figure 1.

This approach is developed further and the related regularization issue is solved in §2.C.1.

§2 Exact Calibration by Martingale Optimal Transport Methods

This Section is the arxiv paper by Guo et al. (2021), with minimal modifications made for consistency of presentation with the present notes. This paper provides a survey of recent results

[4] represented as the differences between market implied volatilities and the implied volatilities recomputed in the calibrated local volatility model.

[5] least squares penalization à la Avellaneda, Buff, Friedman, Grandechamp, Kruk, and Newman (2001, Section 4) of the original exact calibration method of Avellaneda, Friedman, Holmes, and Samperi (1997).

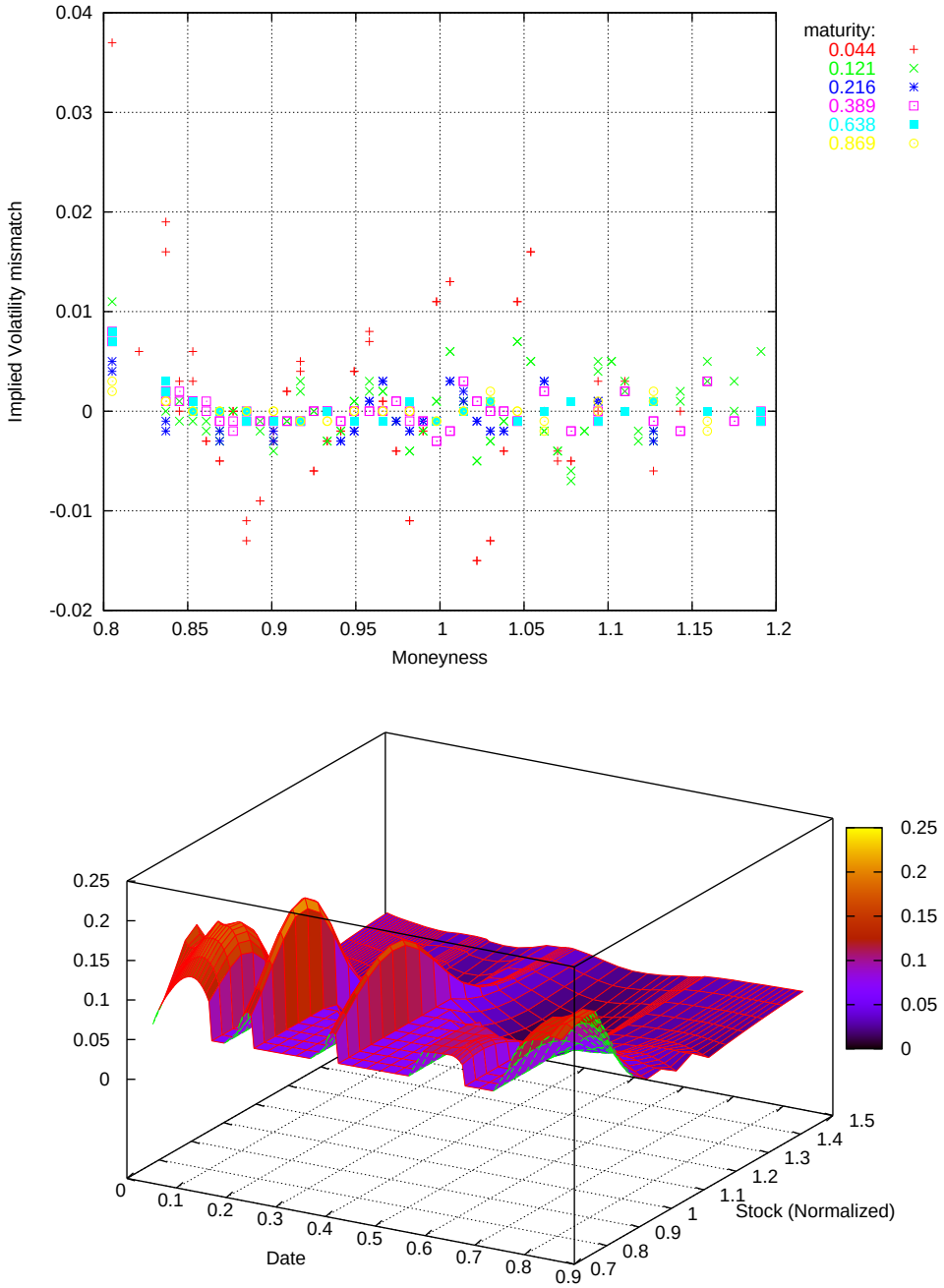


Figure 2: *Calibration by Tikhonov regularization to the implied volatility surface of Figure 1:* (bottom) *calibrated local volatility surface and* (top) *calibration accuracy.*

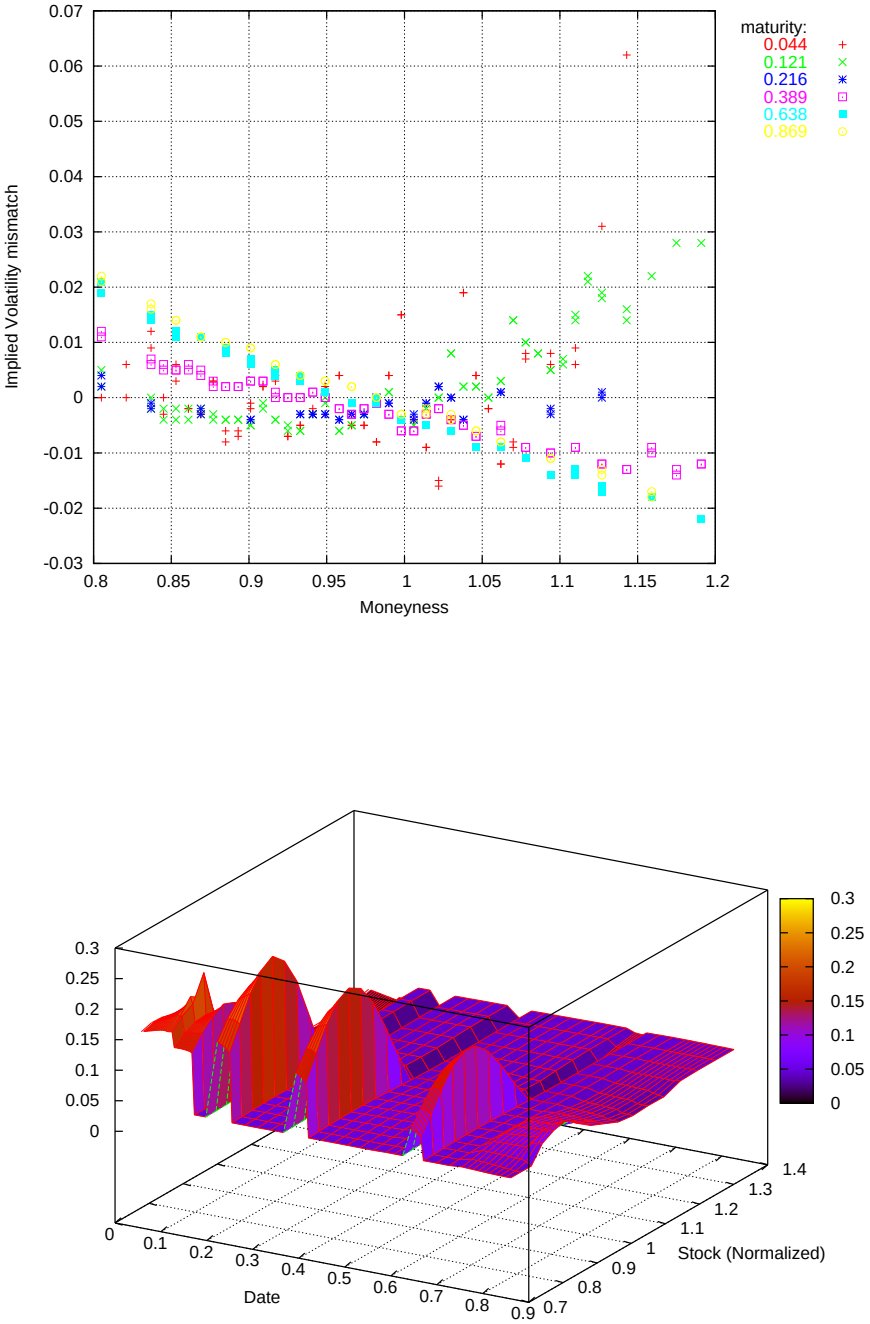


Figure 3: *Calibration by entropic regularization to the implied volatility surface of Figure 1: (bottom) calibrated local volatility surface; (top) calibration accuracy.*

on model calibration by optimal transport. The interest of the method is its genericity, as illustrated by the broad range of applications below. As the method is PDE based, it is subject to the curse of dimensionality[6]. However, this practical limitation is mitigated by the recent developments of machine learning methods for solving non-linear PDEs in high dimension[7].

A Introduction

In recent years, optimal transport theory has attracted the attention of many researchers. The problem was first formulated by Monge (1781) in the context of civil engineering and was later given a rigorous mathematical treatment by Kantorovich (1948) through the introduction of linear programming (some years before it was advanced by George Dantzig, and for which Kantorovich was awarded the Nobel Prize in economics in 1970). Brenier (1991) then revisited the subject (having in mind applications to the famous Euler equations of fluid dynamics) and, later on, Benamou and Brenier (2000) introduced a time-continuous formulation of the problem, which gave rise to a massive amount of applications and mathematical results[8]. Two recent Fields medallists (Villani 2010 and Figalli 2018) are world specialists of optimal transport, which says a lot about the importance that the topic has taken nowadays.

Recently, the theory of optimal transport has been adapted to solve problems in robust hedging and pricing both in discrete and in continuous-time models[9], when it was discovered that pricing bounds on path-dependent derivatives with fixed European options could be formulated as a *martingale optimal transport* problem. Martingale optimal transport then became a subject of study in itself. The theory has been further used to calibrate the non-parametric discrete-time model proposed by Guyon (2020)[10] to solve the so-called VIX/SPX joint calibration problem. The Schrödinger bridge problem, which is highly related to optimal transport, has been recently applied by Henry-Labordère (2019) to introduce a new class of stochastic volatility models. These models can be calibrated by modifying only the drift while keeping the volatility of volatility unchanged.

In this part, we give a synthetic overview of recent results on the continuous-time optimal transport for model calibration, obtained in Guo and Loeper (2021); Guo, Loeper, Oblój, and Wang (2020); Guo, Loeper, and Wang (2019a,b). The results allow for exact calibration in the spirit of the Dupire's formula, albeit without requiring the knowledge of prices for a continuum of European options. We review the calibration of

- local volatility models to European options (Guo, Loeper, and Wang, 2019b),
- local-stochastic volatility models to European options (Guo, Loeper, and Wang, 2019a),
- the joint VIX/SPX calibration problem (Guo, Loeper, Oblój, and Wang, 2020).

B The Semimartingale Optimal Transport Problem

The problem of optimal transport by semimartingales was studied by Tan and Touzi (2013). Later in Guo, Loeper, and Wang (2019a), motivated by financial applications, the authors extended this problem by replacing the terminal distribution constraint with a finite number of discrete constraints.

Let $\Omega := \mathcal{C}([0,T],\mathbb{R}^d), T > 0$ be the set of continuous paths, X be the canonical process and $\mathfrak{F} = (\mathfrak{F}_t)_{0\le t\le T}$ be the canonical filtration generated by X. Let $\mathcal{Q}^0$ be the collection of all probability measures $\mathbb{Q}$ on $(\Omega,\mathfrak{F}_T)$, under which X is an $(\mathfrak{F},\mathbb{Q})$ semimartingale such that

$$dX_t = \beta_t^{\mathbb{Q}}\,dt + (\alpha_t^{\mathbb{Q}})^{\frac{1}{2}}\,dW_t^{\mathbb{Q}}, \tag{6}$$

[6] see V.§4.A.

[7] see e.g., E, Han, and Jentzen (2017); Huré, Pham, and Warin (2020).

[8] see (Villani, 2021, 2009) for an account of these results.

[9] see (Beiglböck, Henry-Labordère, and Penkner, 2013; Henry-Labordère and Touzi, 2014; Tan and Touzi, 2013; Marco and Henry-Labordère, 2015).

[10] see Guyon (2021) for an extended version.

where $W^{\mathbb{Q}}$ is a $\mathbb{Q}$ standard d-variate Brownian motion, while $\beta^{\mathbb{Q}}$ and $\alpha^{\mathbb{Q}}$ are $\mathfrak{F}$ progressive processes with $\alpha^{\mathbb{Q}}$ valued in $\mathbb{S}^d_+$, the subset of the positive semi-definite elements in the space $\mathbb{S}^d$ of the symmetric real matrices of order d. We say that $\mathbb{Q}$ is *characterised* by $(\beta^{\mathbb{Q}}, \alpha^{\mathbb{Q}})$. Let $\mathcal{Q}^1 \subset \mathcal{Q}^0$ be a subset of probability measures $\mathbb{Q}$ characterised by $(\beta^{\mathbb{Q}}, \alpha^{\mathbb{Q}})$ that are $\mathbb{Q}$ integrable on $[0, T]$, i.e.

$$\mathbb{E}^{\mathbb{Q}}\left(\int_0^T |\beta_t^{\mathbb{Q}}| + |\alpha_t^{\mathbb{Q}}|\, dt\right) < +\infty,$$

where $|\cdot|$ is the Euclidean norm.

$\mathcal{Q}^1$ corresponds to the set of feasible market dynamics. Throughout, for simplicity, we will take the interest rates and the dividend yield to be zero[11]. To consider the subset of calibrated models, we fix $x_0 \in \mathbb{R}^d$ and a finite number m of constraints: market prices $\pi \in \mathbb{R}^m$ corresponding to options with bounded continuous payoffs $\psi \in (\mathcal{C}_b(\mathbb{R}^d))^m$ and maturities $\tau \in (0, T]^m$. We assume that the longest maturity coincides with the time horizon, i.e. $\max_k \tau_k = T$. We are then interested in:

$$\mathcal{Q}(x_0, \pi, \tau, \psi) := \{\mathbb{Q} \in \mathcal{Q}^1 : \mathbb{Q} \circ X_0^{-1} = \delta_{x_0} \text{ and } \mathbb{E}^{\mathbb{Q}}\psi_i(X_{\tau_i}) = \pi_i,\ i = 1, \ldots, m\}. \tag{7}$$

Remark 1 *ψ are required to be bounded continuous functions due to technical reasons. In practice, many options do not have bounded payoffs (e.g. call options). This can be fixed by either converting them into options with bounded payoffs via arbitrage arguments (e.g., put options via put-call parity), or by truncating the domain at some extremely large value. Options that do not have continuous payoffs (e.g. digital options, barrier options, etc.) can be approximated by uniformly continuous functions.*

We may have further restrictions on the $\mathbb{Q}$ pricing measures, e.g., some assets may have to be $\mathbb{Q}$ martingales. This, as well as other desirable properties, e.g., proximity to a reference model, are encoded through a cost function $F : [0, T] \times \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{S}^d \to \mathbb{R} \cup \{+\infty\}$. $F = F(t, x, \beta, \alpha)$ is taken convex in (β, α). Finding a suitable calibrated market model corresponds to solving

$$U_0 := \inf_{\mathbb{Q} \in \mathcal{Q}(x_0, \pi, \tau, \psi)} \mathbb{E}^{\mathbb{Q}} \int_0^T F(t, X_t, \beta_t^{\mathbb{Q}}, \alpha_t^{\mathbb{Q}})\, dt, \tag{8}$$

where $\inf \emptyset = +\infty$ and, in particular, a finite value indicates that a perfectly calibrated model was found.

The mimicking properties of diffusions (Brunick and Shreve, 2013) allow us to restrict the optimisations to *local* diffusions, i.e., to $(\beta^{\mathbb{Q}}, \alpha^{\mathbb{Q}})$ which are functions of time t and the state variables X_t. Therefore, the problem in (8) can be studied via PDE methods.

Lemma 1 *For $\mathbb{Q} \in \mathcal{Q}^0$ with associated representation (6) of X such that $(\beta_t^{\mathbb{Q}}, \alpha_t^{\mathbb{Q}}) = (\beta^{\mathbb{Q}}, \alpha^{\mathbb{Q}})(t, X_t)$[12], the following Feynman-Kac representation formula holds:*

$$\mathbb{E}_t^{\mathbb{Q}}\Big(\int_t^T -F(s, X_s, \beta_s^{\mathbb{Q}}, \alpha_s^{\mathbb{Q}})\, ds + \sum_{t < \tau_i} \lambda_i \psi_i(X_{\tau_i})\Big) = u^{\mathbb{Q}}(t, X_t), \tag{9}$$

where the function $u^{\mathbb{Q}}$ solves

$$\partial_t u^{\mathbb{Q}}(t, x) + f^{\mathbb{Q}}(t, x, \partial_x u(t, x), \partial^2_{x^2} u(t, x)) = -\lambda \cdot \big(\psi_\cdot(x)\delta(t - \tau_\cdot)\big) \tag{10}$$

with the terminal condition $u(T, \cdot) = 0$, where

$$f^{\mathbb{Q}}(t, x, b, a) = b\beta^{\mathbb{Q}} + \frac{1}{2}\mathrm{tr}(a\alpha^{\mathbb{Q}}) - F(t, x, \beta^{\mathbb{Q}}, \alpha^{\mathbb{Q}}). \tag{11}$$

[11] in applications with market data we then work out the forward prices.

[12] to which we restrict ourselves hereafter, without loss of generality as just said.

Proof. (starting from $u^{\mathbb{Q}}$ defined by (9), which we then show satisfies (10)-(11)[13]) By application of the Itô formula and $\mathbb{Q}$ martingale property of the process

$$\mathbb{E}_\cdot^{\mathbb{Q}}\Big(\int_0^T -F(s,X_s,\beta_s^{\mathbb{Q}},\alpha_s^{\mathbb{Q}})\,ds+\sum_i \lambda_i\psi_i(X_{\tau_i})\Big),$$

used on the time intervals delimited by the successive maturities of the data (by decreasing order, starting from T). ∎

B.1 Primal and Dual Formulations

Following the (Benamou and Brenier, 2000) formulation of the classical optimal transport, we introduce the following:

Primal formulation Solve

$$U_0=\inf_{\gamma,\beta,\alpha}\int_0^T\int_{\mathbb{R}^d} F(t,x,\beta(t,x),\alpha(t,x))\gamma(t,x)\,dxdt, \tag{12}$$

where the infimum is taken among all (γ,β,α) satisfying (in the distributional sense)

$$\gamma(0,\cdot)=\delta_{x_0} \text{ and } \partial_t\gamma(t,x)+\sum_i \partial_{x_i}\big(\gamma(t,x)\beta_i(t,x)\big)$$
$$-\frac{1}{2}\sum_{i,j}\partial^2_{x_i,x_j}\big(\gamma(t,x)\alpha_{ij}(t,x)\big)=0, \tag{13}$$

$$\int_{\mathbb{R}^d}\psi_i(x)\gamma(\tau_i,x)\,dx=\pi_i,\quad \forall i=1,\dots,m,. \tag{14}$$

In this primal formulation, the objective function is convex and all constraints are linear in $(\gamma,\gamma\beta,\gamma\alpha)$. Applying the classical tools of convex analysis[14], we obtain

$$\begin{aligned}
U_0 &= \inf_{\gamma,\beta,\alpha;(13)-(14)}\int_0^T\int_{\mathbb{R}^d} F(t,x,\beta(t,x),\alpha(t,x))\gamma(t,x)\,dxdt\\
&= \inf_{\gamma,\beta,\alpha;(13)}\sup_{\lambda\in\mathbb{R}^m}\Big(\int_0^T\int_{\mathbb{R}^d} F(t,x,\beta(t,x),\alpha(t,x))\gamma(t,x)\,dxdt\\
&\quad +\lambda\cdot\big(\pi-\int_{\mathbb{R}^d}\psi_\cdot(x)\gamma(\tau_\cdot,x)\,dx\big)\Big)\\
&= \sup_{\lambda\in\mathbb{R}^m}\inf_{\gamma,\beta,\alpha;(13)}\Big(\int_0^T\int_{\mathbb{R}^d} F(t,x,\beta(t,x),\alpha(t,x))\gamma(t,x)\,dxdt\\
&\quad +\lambda\cdot\big(\pi-\int_{\mathbb{R}^d}\psi_\cdot(x)\gamma(\tau_\cdot,x)\,dx\big)\Big)\\
&= \sup_{\lambda\in\mathbb{R}^m}\Big(\lambda\cdot\pi+\inf_{\gamma,\beta,\alpha;(13)}\big(\int_0^T\int_{\mathbb{R}^d} F(t,x,\beta(t,x),\alpha(t,x))\gamma(t,x)\,dxdt-\int_{\mathbb{R}^d}\lambda\cdot\psi_\cdot(x)\gamma(\tau_\cdot,x)\,dx\big)\Big)\\
&= \sup_{\lambda\in\mathbb{R}^m}\Big(\lambda\cdot\pi-\sup_{\gamma,\beta,\alpha;(13)}\big(\int_0^T\int_{\mathbb{R}^d}(-F)(t,x,\beta(t,x),\alpha(t,x))\gamma(t,x)\,dxdt\\
&\quad +\int_{\mathbb{R}^d}\lambda\cdot\psi_\cdot(x)\gamma(\tau_\cdot,x)\,dx\big)\Big).
\end{aligned}$$

[13] equivalently one can start from $u^{\mathbb{Q}}$ defined by (10)-(11) and then show that it satisfies (9).

[14] the proof of duality mainly relies on the Fenchel–Rockafellar theorem. We refer the reader to Guo, Loeper, and Wang (2019a) for the full proof.

One then has the following "nonlinear extension" of Lemma 1. We omit the proof, which would require an advanced course in stochastic control:

Dual formulation Solve

$$U_0 = \sup_{\lambda \in \mathbb{R}^m} (\lambda \cdot \pi - u(0, x_0)), \tag{15}$$

where u solves the following HJB equation (in the viscosity sense[15]):

$$\partial_t u(t,x) + f(t, x, \partial_x u(t,x), \partial^2_{x^2} u(t,x)) = -\lambda \cdot \big(\psi_\cdot(x)\delta(t - \tau_\cdot)\big) \tag{16}$$

with the terminal condition $u(T, \cdot) = 0$, where f is the convex conjugate of F defined by

$$f(t, x, b, a) = \sup_{\beta,\alpha}\{b\beta + \frac{1}{2}\mathrm{tr}(a\alpha) - F(t, x, \beta, \alpha)\}. \tag{17}$$

The dual formulation can be solved by gradient descent methods, and each component of the gradient vector can be calculated by solving a linear PDE. In fact, given a $\lambda \in \mathbb{R}^m$, denote by u^λ the associated solution to (16). Define

$$L(\lambda) := \lambda \cdot \pi - u^\lambda(0, x_0), \tag{18}$$

then $\partial_{\lambda_i} L(\lambda) = \pi_i - \partial_{\lambda_i} u^\lambda(0, x_0)$. By taking functional derivatives of (16) with respect to λ_i, we can formulate the gradients as

$$\partial_{\lambda_i} L(\lambda) = \pi_i - u'_i(0, x_0), \quad i = 1, \ldots, m, \tag{19}$$

where u'_i solves

$$\begin{cases} \partial_t u'_i + (\partial_x u'_i)\beta^\lambda + \dfrac{1}{2}\mathrm{tr}\left((\partial^2_{x^2} u'_i)\alpha^\lambda\right) = 0 \text{ in } [0, \tau_i) \times \mathbb{R}^d, \\ u'_i(\tau_i, \cdot) = \psi_i, \end{cases} \tag{20}$$

in which $(\beta^\lambda, \alpha^\lambda)$ denote the maximisers in the definition (17) of f for $(b, a) = (\partial_x u^\lambda, \partial^2_{x^2} u^\lambda)$. Hence $u'_i(0, x_0) = \mathbb{E}^{\mathbb{Q}(\lambda)}\psi_i(X_{\tau_i})$, where $\mathbb{Q}(\lambda)$ is characterised by $(\beta^\lambda, \alpha^\lambda)$, so that the gradient

$$\partial_{\lambda_i} L = \pi_i - u'_i(0, x_0) = \pi_i - \mathbb{E}^{\mathbb{Q}(\lambda)}\psi_i(X_{\tau_i})$$

can be interpreted as the difference between the model option prices given by the current optimisation iteration and the market option prices. The optimum is reached when the gradient is zero, in other words, the market option prices are attained exactly.

Remark 2 *The problem of allowing F and ψ to be path-dependent was studied in Guo and Loeper (2021). In that case, we need to solve path-dependent PDEs instead of HJB equations.*

B.2 Numerical Method

A numerical method for solving the dual formulation (15)–(17), i.e. for maximizing L in (18), was proposed in Guo, Loeper, and Wang (2019a). The method can be described as follows:

i. set an initial λ, e.g., $\lambda = \mathbf{0} \in \mathbb{R}^m$,

ii. obtain $u^\lambda(0, x_0)$ by solving the related HJB equation (15) and obtain $(\beta^\lambda, \alpha^\lambda)$ by solving the supremum of f for $(b, a) = (\partial_x u^\lambda, \partial^2_{x^2} u^\lambda)$ in (17),

[15] see (Guo, Loeper, and Wang, 2019a) for the definition of the viscosity solution to (16).

iii. solve the linear pricing PDEs (20) with $(\beta^\lambda, \alpha^\lambda)$, and then calculate the gradients by (19),

iv. update λ by a gradient descent algorithm for the cost criterion (18),

v. repeat step ii.-iv. until the all components of gradients are close to zero.

In Guo, Loeper, and Wang (2019a) and Guo, Loeper, Oblój, and Wang (2020), the HJB equation (16) was solved by the standard implicit finite difference method[16] with the so-called policy iteration technique to handle the nonlinearity, while the linear pricing PDEs (20) were solved by an alternating direction implicit finite difference method[17] that is faster than the standard implicit method. For the gradient descent algorithm, the L-BFGS algorithm was employed and showed good convergence in both works.

It should be mentioned that the dual formulation and the numerical method were also studied in Avellaneda, Friedman, Holmes, and Samperi (1997) much earlier in the context of local volatility calibration via entropy minimisation[18], although the connection to optimal transport and the proof of the duality result was not established at that time.

C Applications in Model Calibration

From now on, we will refer to the proposed calibration method simply as *OT framework*.

C.1 Local Volatility Calibration

The application of optimal transport to calibrate the local volatility model of Dupire (1994b) was explored in Guo, Loeper, and Wang (2019b). In Guo, Loeper, and Wang (2019b), as an extension of the seminal work of Benamou and Brenier (2000), an augmented Lagrangian method was developed to solve the primal formulation (8) / (12)–(14). In this part, we resolve the local volatility calibration problem by the OT framework.

Let X_t be the logarithm of the underlying stock price S at time t. We are interested in finding a probability measure $\mathbb{Q} \in \mathcal{Q}^1$ with characteristics $(-\frac{1}{2}\sigma^2, \sigma^2)$ where σ is some progressive process. In other words, we want X to be a $\mathbb{Q}$ semimartingale in the form of

$$dX_t = -\frac{1}{2}\sigma_t^2\, dt + \sigma_t\, dW_t^{\mathbb{Q}}. \tag{21}$$

To ensure that X solves the above SDE, we consider a cost function of the form

$$F(t, x, \beta, \alpha) = \begin{cases} c_1(\alpha/\bar{\sigma}^2)^p + c_2(\alpha/\bar{\sigma}^2)^{-q} + c_3 & \text{if } -2\beta = \alpha > 0, \\ +\infty & \text{otherwise,} \end{cases} \tag{22}$$

where $\bar{\sigma}$ is some reference volatility level, p, q are constants greater than 1, and the c_i are constants chosen so that the function reaches its minimum at $\alpha = \bar{\sigma}^2$[19] with $\min F = 0$.

Given a vector of m (discounted) European option payoff functions ψ, a vector of maturities τ and a vector of European option prices π, we want to further restrict $\mathbb{Q}$ so that $\mathbb{E}^{\mathbb{Q}}\psi_i(X_{\tau_i}) = \pi_i$, $i = 1, \dots, m$ are satisfied. Let x_0 be the logarithm of the current stock price. Then the corresponding local volatility calibration problem (8) / (12)–(14) admits the dual formulation

$$U_0^{lo} = \sup_{\lambda \in \mathbb{R}^m} (\lambda \cdot \pi - u(0, x_0)), \tag{23}$$

[16] see III.§3.B.2.
[17] see III.§3.F.
[18] cf. VII.A.2.
[19] cf. Figure 7.

where u is a solution[20] to the HJB equation

$$\partial_t u + \sup_{\alpha>0}\left\{-\frac{1}{2}\alpha\partial_x u + \frac{1}{2}\alpha\partial^2_{x^2}u - \left(c_1\left(\frac{\alpha}{\bar{\sigma}^2}\right)^p + c_2\left(\frac{\alpha}{\bar{\sigma}^2}\right)^{-q} + c_3\right)\right\} = -\sum_{i=1}^{m}\lambda_i\psi_i\delta(t-\tau_i), \tag{24}$$

with the terminal condition $u(T,\cdot) = 0$.

Numerical example We give here an example in which a local volatility model is calibrated to the prices of 5 European put options at 5 different strikes and maturity $T = 1$. The option prices are generated by another local volatility model with the volatility given in Figure 4. The value of $\bar{\sigma}$ in (22) is set to 0.2.

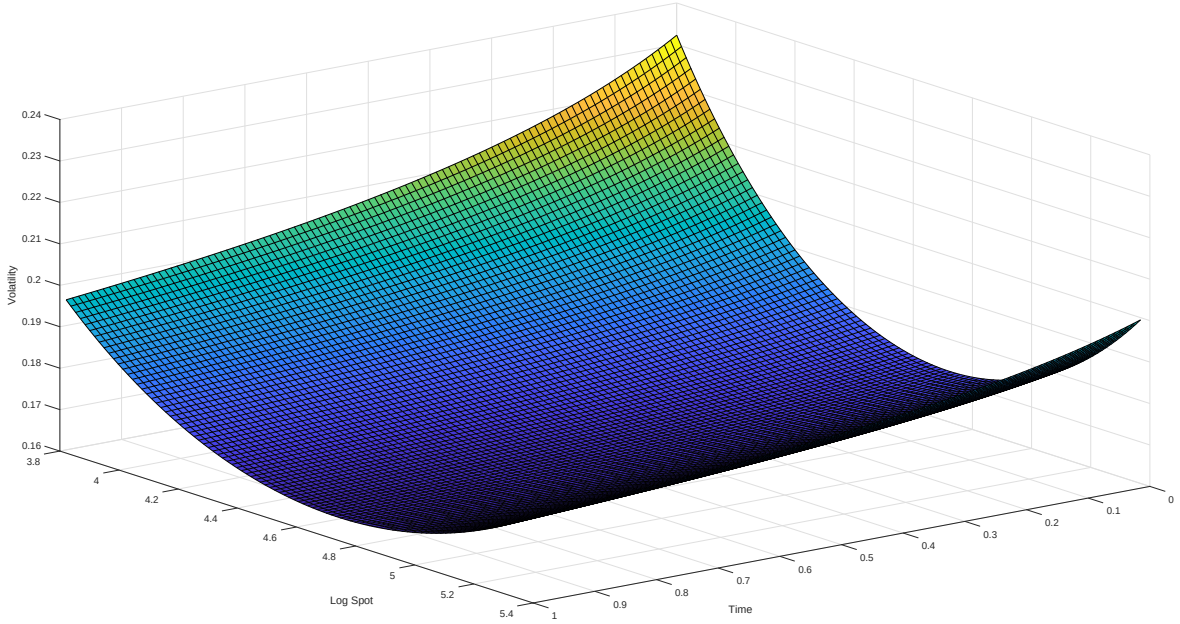


Figure 4: The local volatility surface used for generating European put option prices.

Figure 5 shows the calibrated local volatility surface and the model implied volatility skew. The humps between strikes in the volatility skew are caused by the spikes in the volatility surface. These spikes were also observed in Avellaneda, Friedman, Holmes, and Samperi (1997)[21].

To smooth the volatility surface and hence the volatility skew, we suggest a *reference iteration* method. We start smoothing the spiky volatility surface by a simple moving average method. Next, we set the smoothed surface as the reference value $\bar{\sigma}$ and recalibrate the model by solving again the dual formulation (23) (for the new $\bar{\sigma}$). After iterating the above steps 8 times, we obtain a local volatility model that has a smooth volatility skew and is also fully calibrated to the given option prices. The results are shown in Figure 6.

C.2 Local Stochastic Volatility Calibration

The local stochastic volatility (*lsv*) model was first introduced in Jex, Henderson, and Wang (1999). It incorporates a nonparametric local factor (also called *leverage*) into a classical stochastic volatility

[20] in the viscosity sense.
[21] cf. Figure 3.

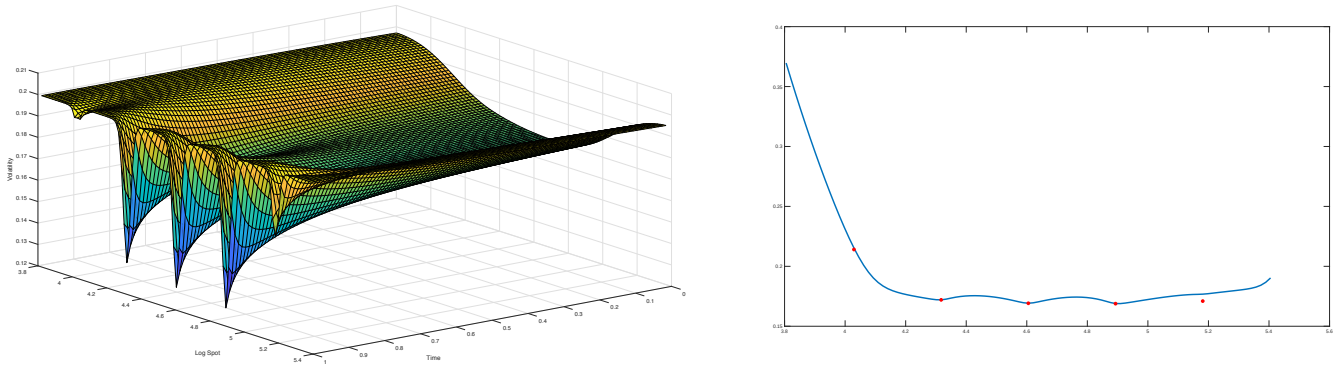


Figure 5: The (unsmoothed) calibrated local volatility surface (left), the model volatility skew (right, blue) and the implied volatility of the calibrating options (right, red).

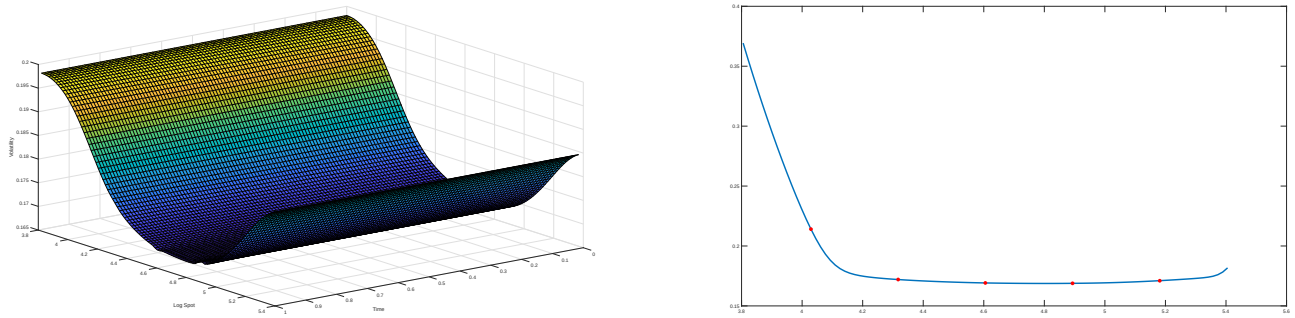


Figure 6: The smoothed local volatility surface (left), the model volatility skew after 8 iterations (right, blue) and the implied volatility of the calibrating options (right, red).

model. Thus, while keeping consistent dynamics, the *lsv* model can match all observed market option prices, as long as one restricts to European products. In this section, we apply the OT framework to solve the *lsv* model calibration problem, as done in Guo, Loeper, and Wang (2019a).

Consider probability measures $\mathbb{Q} \in \mathcal{Q}^1$ under which $X = (X^1, X^2)$ are two-dimensional $\mathbb{Q}$ semi-martingales. Let X^1 be the logarithm of the underlying price and let X^2 be a mean-reverting stochastic factor. We are interested in the following *lsv* model:

$$\begin{cases} dX_t^1 = -\frac{1}{2}\sigma_t^2\, dt + \sigma_t\, dW_t^1, \\ dX_t^2 = \kappa(\theta - X_t^2)\, dt + \eta\sqrt{X_t^2}\, dW_t^2, \\ dW_t^1 dW_t^2 = \nu\dfrac{\sqrt{X_t^2}}{\sigma_t}\, dt, \end{cases} \tag{25}$$

where σ is some progressive process, while $(\kappa, \theta, \eta, \nu)$ are assumed constant given parameters. The above model dynamics can be captured by probability measures $\mathbb{Q}$ characterised by $(\beta^{\mathbb{Q}}, \alpha^{\mathbb{Q}})$ such that

$$(\beta_t^{\mathbb{Q}}, \alpha_t^{\mathbb{Q}}) = \left(\begin{bmatrix} -\frac{1}{2}\sigma_t^2 \\ \kappa(\theta - X_t^2) \end{bmatrix}, \begin{bmatrix} \sigma_t^2 & \nu\eta X_t^2 \\ \nu\eta X_t^2 & \eta^2 X_t^2 \end{bmatrix} \right), \quad 0 \le t \le T.$$

The model we consider here is slightly different from the standard *lsv* model from the literature. In the standard *lsv* model, the correlation between W^1 and W^2 is a constant and $\sigma_t = L(t, X_t^1)\sqrt{X_t^2}$, where L is known as the leverage function. Our simple modification allows that if a function is convex in α, then it is convex in σ^2, which makes it easier to define a suitable cost function. Note that if $\sigma_t = \sqrt{X_t^2}$, the correlation is simply ν and X reduces to a Heston model. If we define a cost function to penalise σ_t away from $\sqrt{X_t^2}$, and we obtain $(\kappa, \theta, \eta, \nu)$ by calibrating a Heston model to the market prices, then σ_t will be close to $\sqrt{X_t^2}$ and hence the correlation will be close to ν. Moreover, if σ_t is independent of X_t^2,

then X is indeed a local volatility model, and X can be exactly calibrated to the option prices generated by any arbitrage-free implied volatility surface. Our goal is to calibrate σ_t with given $(\kappa, \theta, \eta, \nu)$ so that X is fully calibrated to the observable market European option prices.

In the spirit of (22), let us first define a convex function

$$H(x, \bar{x}, s) := \begin{cases} c_1 \left(\frac{x-s}{\bar{x}-s}\right)^p + c_2 \left(\frac{x-s}{\bar{x}-s}\right)^{-q} + c_3 & \text{if } x > s \text{ and } \bar{x} > s, \\ +\infty & \text{otherwise,} \end{cases}$$

where p, q are constants greater than 1, and the c_i are constants chosen so that the function reaches its minimum at $x = \bar{x} > s$ with $\min H = 0$ (see Figure 7). To ensure that X has the dynamics (25), we define the cost function

$$F(t, x, \beta, \alpha) = \begin{cases} H(\alpha_{11}, x_2, \nu^2 x_2) & \text{if } (\beta, \alpha) \in \Gamma(t, x), \\ +\infty & \text{otherwise,} \end{cases}$$

where the convex set

$$\Gamma(t, x) := \{(\beta, \alpha) \mid \beta_1 = -\alpha_{11}/2, \beta_2 = \kappa(\theta - x_2), \alpha_{12} = \alpha_{21} = \nu\eta x_2, \alpha_{22} = \eta^2 x_2\}.$$

In the function H, we set $s = \nu^2 x_2$ to keep $\sigma_t^2 > \nu^2 X_t^2$ so that α_t remains positive semidefinite for all $t \leq T$. We set $\bar{x} = x_2$ to regularise α_{11} (or σ^2) by penalising deviations of X from a standard Heston model.

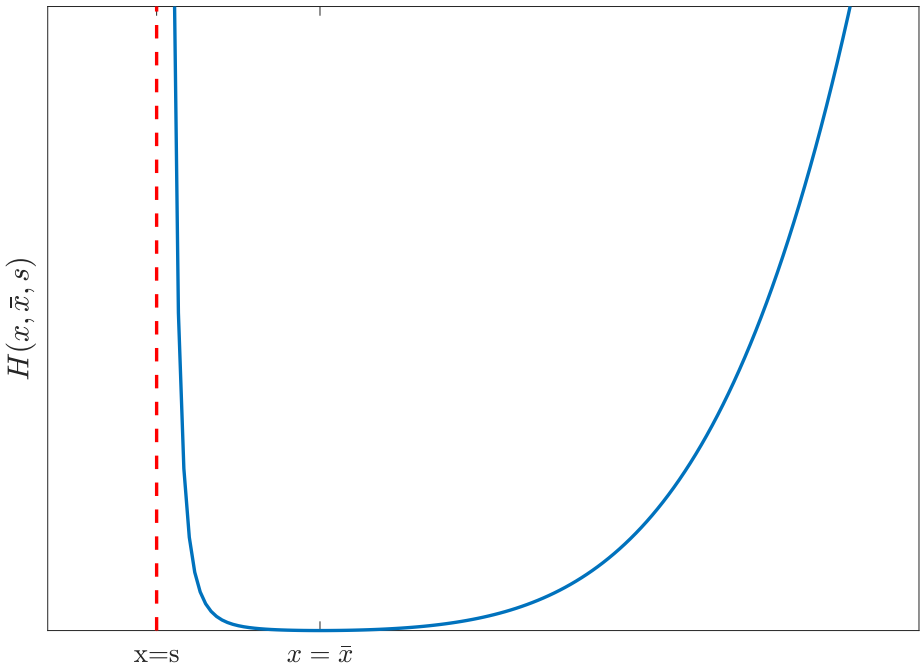


Figure 7: The function $H(x, \bar{x}, s)$ for a given $\bar{x}$ and a given $s < \bar{x}$. *Source:* Guo, Loeper, and Wang (2019a, Figure 1 page 13).

Given a vector of m (discounted) European option payoff functions ψ[22], a vector of maturities τ and a vector of European option prices π, we want to further restrict $\mathbb{Q}$ so that $\mathbb{E}^{\mathbb{Q}}\psi_i(X_{\tau_i}) = \pi_i$, $i = 1, \ldots, m$ are satisfied. Assume that $x_0 \in \mathbb{R}^2$ is given. Its first element is the logarithm of the current stock price, which is observed from the market, and its second element is the initial value of the instantaneous

[22]Note that ψ are functions of X. For example, if ψ_i is the payoff function of an European call option, $\psi_i(x) = \max(\exp(x_1) - K, 0), K > 0$.

variance, which is a parameter but can be obtained by calibrating a Heston model. Applying the arguments developed in B.1, we can derive the following dual formulation of the corresponding *lsv* calibration problem:

$$U_0^{lsv} = \sup_{\lambda \in \mathbb{R}^m} (\lambda \cdot \pi - u(0, x_0)),$$

where the HJB equation (15) for u[23] is

$$\partial_t u + \sup_{\alpha_{11}>0} \Big\{ -\frac{1}{2}\alpha_{11}\partial_{x_1} u + \kappa(\theta - x_2)\partial_{x_2} u + \frac{1}{2}\alpha_{11}\partial^2_{x_1^2} u + \frac{1}{2}\eta^2 x_2 \partial^2_{x_2^2} u + \nu\eta x_2 \partial^2_{x_1,x_2} u - H(\alpha_{11}, x_2, \nu^2 x_2) \Big\} = -\sum_{i=1}^m \lambda_i \psi_i \delta(t - \tau_i),$$

with the terminal condition $u(T, \cdot) = 0$.

Numerical example In the numerical example provided in Guo, Loeper, and Wang (2019a), the process X in (25), also called the *ot-lsv* model, was calibrated to the FX options market data provided in Tian, Zhu, Lee, Klebaner, and Hamza (2015). The data contains 10 maturities ranging from 1 month to 5 years. At each maturity, there are 5 European options at different strikes. The parameters are $(\kappa, \theta, \eta, \nu) = (0.8721, 0.0276, 0.5338, -0.3566)$ which are obtained by (roughly) calibrating a standard Heston model to the market option prices. Since $2\kappa\theta/\eta^2 = 0.168 \ll 1$, the Feller condition is strongly violated in this case. The initial position $X_0 = (0.2287, 0.012)$.

Figures 8 and 9 compare the implied volatility skews of both the calibrated and uncalibrated *ot-lsv* model. The results show that the *ot-lsv* model can be accurately calibrated to both short-maturity and long-maturity market option prices. Unlike the local volatility example in C.1, the volatility skews are very smooth even without iterating the reference values.

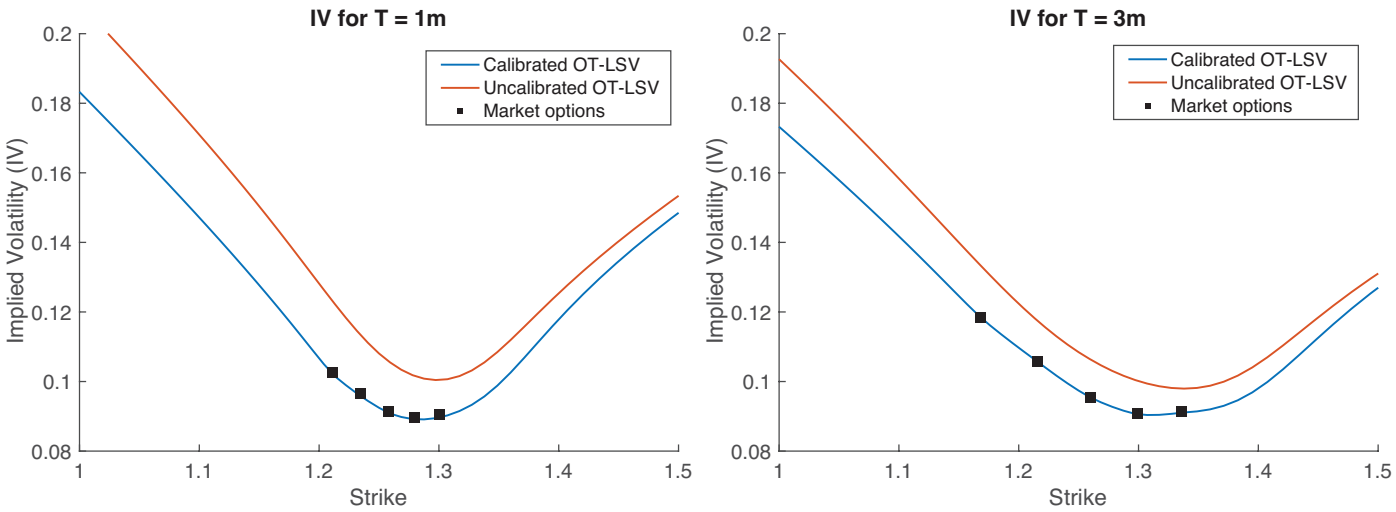


Figure 8: The implied volatility skews generated by both the uncalibrated and the calibrated OT-LSV model for 1 month and 3 months maturities in the FX market data example.

C.3 VIX/SPX Joint Calibration

Since it was first reported in Gatheral (2008), the joint calibration on SPX and VIX has been known to be a challenging problem. More specifically, we want to build a stochastic volatility model that could be jointly calibrated to the options and futures of SPX and VIX. We refer the reader to Guyon (2020) for a comprehensive discussion of the literature and a martingale optimal transport approach with a

[23] in the viscosity sense.

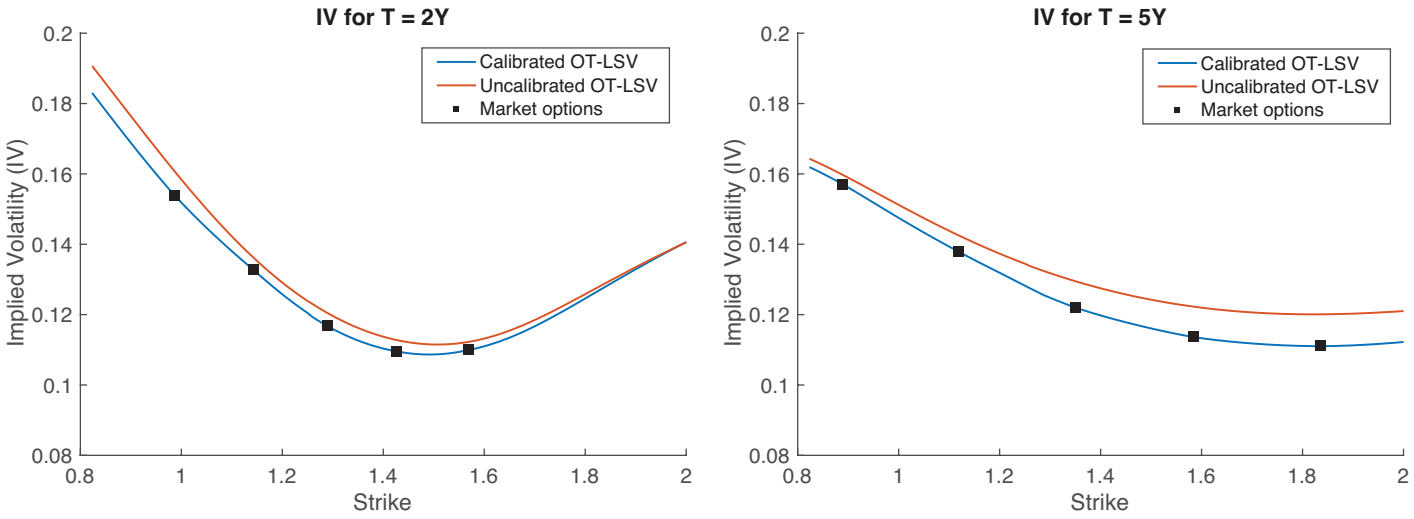


Figure 9: The implied volatility skews generated by both the uncalibrated and the calibrated OT-LSV model for 2 years and 5 years maturities.

discrete-time model. In this paper, we introduce the work of Guo, Loeper, Oblój, and Wang (2020) in which the OT framework was applied to solve the joint calibration problem.

Consider probability measures $\mathbb{Q} \in \mathcal{Q}^1$ under which $X = (X^1, X^2)$ are two-dimensional $\mathbb{Q}$ semi-martingales. We want X^1 to be the logarithm of the SPX price that takes the form

$$X_t^1 = X_0^1 - \frac{1}{2}\int_0^t \sigma_s^2\, ds + \int_0^t \sigma_s\, dW_s, \quad 0 \le t \le T. \tag{26}$$

For such X^1, we then use X_t^2 (or $X_{t,T}^2$ when emphasizing the dependence on T) to represent a half of the expectation of the forward quadratic variation of X^1 on $[t, T]$ observed at time t, that is[24]

$$X_{t,T}^2 = \mathbb{E}^{\mathbb{Q}}\left(\frac{1}{2}\int_t^T \sigma_s^2\, ds \,\middle|\, \mathfrak{F}_t\right) = X_t^1 - \mathbb{E}^{\mathbb{Q}}(X_T^1 \mid \mathfrak{F}_t), \quad 0 \le t \le T. \tag{27}$$

Note that the the second term on the right-hand side of (27) is a martingale. It follows that the modelling setting we just described is captured by probability measures $\mathbb{Q} \in \mathcal{Q}^1$ characterised by (β, α) such that

$$(\beta_t, \alpha_t) = \left(\begin{bmatrix} -\frac{1}{2}\sigma_t^2 \\ -\frac{1}{2}\sigma_t^2 \end{bmatrix}, \begin{bmatrix} \sigma_t^2 & (\alpha_t)_{12} \\ (\alpha_t)_{12} & (\alpha_t)_{22} \end{bmatrix}\right), \quad 0 \le t \le T, \tag{28}$$

where $(\alpha_t)_{12} = d\langle X^1, X^2\rangle_t \,/\, dt$,$(\alpha_t)_{22} = d\langle X^2\rangle_t \,/\, dt$, and with the additional property that $X_{T,T}^2 = 0$.

In order to restrict the probability measures to those characterised by (β, α) of the form (28), we can define a cost function that penalises characteristics that are not in the following convex set:

$$\Gamma := \left\{(\beta, \alpha) \in \mathbb{R}^2 \times \mathbb{S}_+^2 : \beta_1 = \beta_2 = -\frac{1}{2}\alpha_{11}\right\},$$

where $\mathbb{S}_+^2$ is the set of positive semidefinite matrices of order two. Define the convex cost function F as follows:

$$F(t, x, \beta, \alpha) = \begin{cases} \displaystyle\sum_{i,j=1}^{2} (\alpha_{ij} - \bar{\alpha}_{ij})^2 & \text{if } (\beta, \alpha) \in \Gamma, \\ +\infty & \text{otherwise,} \end{cases} \tag{29}$$

where $\bar{\alpha}$ is a matrix of some reference values for α. Note that $\bar{\alpha}$ may depend on (t, x) as well.

[24] assuming $\int_0^{\cdot} \sigma_s\, dW_s$ a true martingale.

The calibration instruments we consider are SPX European options, VIX options and VIX futures. The market prices of these products can be imposed as constraints on X. Let ψ^{spx} be a vector of m number of SPX option (discounted) payoff functions. For example, if the i^{th} option is a put option with a strike $K_i > 0$, then the payoff function $\psi_i : \mathbb{R}^2 \to \mathbb{R}$ is given by $\psi_i^{spx}(x) = \max(K_i - \exp(x_1), 0)$. Let $\pi^{spx} \in \mathbb{R}^m$ be the market SPX option prices and $\tau^{spx} \in [0,T]^m$ be the vector of their maturities. The prices π^{spx} can be imposed on X by restricting $\mathbb{Q}$ to probability measures that satisfy

$$\mathbb{E}^{\mathbb{Q}}\psi_i^{spx}(X_{\tau_i}) = \pi_i^{spx}, \qquad \forall i = 1, \dots, m.$$

The VIX index at t_0 is defined using a synthetic log-payoff option[25]. In our setting, it can be equivalently re-written as the square root of the expected realised variance over the next 30 days (i.e., $T - t_0 = 30$ days), that is[26]

$$\mathrm{VIX}_{t_0} = \sqrt{\mathbb{E}^{\mathbb{Q}}\left(\frac{100^2}{T-t_0}\int_{t_0}^{T}\sigma_t^2\,dt \,\middle|\, \mathfrak{F}_{t_0}\right)} = 100\sqrt{\frac{2}{T-t_0}X_{t_0}^2},$$

by (27). Consider VIX options and futures both with maturity t_0. Let $\pi^{vix,f} \in \mathbb{R}$ be the market VIX futures price and let $\pi^{vix} \in \mathbb{R}^n$ be the market VIX option prices. Let ψ^{vix} be a vector of n number of VIX option payoff functions. Similarly to ψ^{spx}, if the i^{th} VIX option is a put option with a strike $K_i > 0$, then the payoff function $\psi_i^{vix} : \mathbb{R} \to \mathbb{R}$ is given by $\psi_i^{vix}(x) = \max(K_i - x, 0)$. Let $\psi^{vix,f} : \mathbb{R}^2 \to \mathbb{R}$ be given by $\psi^{vix,f}(x) := 100\sqrt{2x_2/(T-t_0)}$. Then, we want to further restrict $\mathbb{Q}$ to those under which X also satisfies the following constraints:

$$\begin{aligned}\mathbb{E}^{\mathbb{Q}}\psi^{vix,f}(X_{t_0}) &= \pi^{vix,f},\\ \mathbb{E}^{\mathbb{Q}}(\psi_i^{vix}\circ\psi^{vix,f})(X_{t_0}) &= \pi_i^{vix}, \qquad \forall i = 1,\dots,n.\end{aligned}$$

Finally, to ensure that $X_{T,T}^2 = 0$, one additional constraint is imposed on the model. Let $\psi^{sg} : \mathbb{R}^2 \to \mathbb{R}$ be a function such that $\psi^{sg}(x) = 0$ if and only if $x_2 = 0$. Here, we choose $\psi^{sg}(x) := 1 - \exp(-(x_2)^2)$ and add constraint $\mathbb{E}^{\mathbb{Q}}\psi^{sg}(X_T) = 0$. This constraint can be interpreted as a contract that has a payoff $\psi^{sg}(X_T)$ at time T, and its price is always null. We will call it the *singular contract*.

We assume that $X_0 = (X_0^1, X_{0,T}^2) \in \mathbb{R}^2$ is known, and the initial marginal of X is a Dirac measure on X_0. The value of X_0^1 is the logarithm of the current SPX price. In practice, $X_{0,T}^2$ can be inferred if the market prices of SPX call and put options maturing at T are available over a continuous spectrum of strikes, as, by Proposition I.7:

$$X_{0,T}^2 = \mathbb{E}^{\mathbb{Q}}\left(\frac{1}{2}\int_0^T \sigma_s^2\,ds\right) = \int_0^{F_0^T}\frac{\mathbb{E}^{\mathbb{Q}}(k-S_T)^+}{k^2}\,dk + \int_{F_0^T}^{\infty}\frac{\mathbb{E}^{\mathbb{Q}}(S_T-k)^+}{k^2}\,dk,$$

where $F_0^T = \mathbb{E}^{\mathbb{Q}}(S_T)$ is the T futures price of the SPX index. Now, to group all constraints together, we define

$$\begin{aligned}\pi &:= (\pi_1^{spx},\dots,\pi_m^{spx}, \quad \pi_1^{vix},\dots,\pi_n^{vix} \quad, \quad \pi^{vix,f} \quad, \quad 0 \quad),\\ \tau &:= (\tau_1^{spx},\dots,\tau_m^{spx}, \quad t_0,\dots,t_0 \quad, \quad t_0 \quad, \quad T \quad),\\ \psi &:= (\underbrace{\psi_1^{spx},\dots,\psi_m^{spx}}_{m\text{ SPX options}}, \underbrace{\psi_1^{vix}\circ\psi^{vix,f},\dots,\psi_n^{vix}\circ\psi^{vix,f}}_{n\text{ VIX options}}, \underbrace{\psi^{vix,f}}_{\text{VIX futures}}, \underbrace{\psi^{sg}}_{\text{singular contract}}).\end{aligned}$$

With F as per (28), the joint calibration problem can then be reformulated as solving

$$U_0^{joint} := \inf_{\mathbb{Q}\in\mathcal{Q}(X_0,\pi,\tau,\psi)}\mathbb{E}^{\mathbb{Q}}\int_0^T F(t,X_t,\beta_t^{\mathbb{Q}},\alpha_t^{\mathbb{Q}})\,dt, \tag{30}$$

[25] cf. Proposition I.7.

[26] see Chicago Board Options Exchange (2009, page 10).

where

$$\mathcal{Q}(X_0,\pi,\tau,\psi) := \{\mathbb{Q}\in\mathcal{Q}^1 : \mathbb{Q}\circ X_0^{-1} = \delta_{X_0} \text{ and } \mathbb{E}^{\mathbb{Q}}\psi_i(X_{\tau_i}) = \pi_i, \\ i = 1,\dots,m+n+2\}.$$

Applying the arguments developed in B.1, we can derive a dual formulation of (30):

$$U_0^{joint} = \sup_{\lambda\in\mathbb{R}^{m+n+2}} (\lambda\cdot\pi - u(0,X_0)),$$

where u is a solution to the following HJB equation (in the viscosity sense):

$$\partial_t u + \sup_{\alpha\in\mathbb{S}_+^2}\Big\{ -\frac{1}{2}\alpha_{11}\partial_{x_1}u - \frac{1}{2}\alpha_{11}\partial_{x_2}u + \frac{1}{2}\alpha_{11}\partial^2_{x_1^2}u + \frac{1}{2}\alpha_{22}\partial^2_{x_2^2}u \\ + \alpha_{12}\partial^2_{x_1,x_2}u - \sum_{i,j=1}^{2}(\alpha_{ij}-\bar\alpha_{ij})^2\Big\} = -\sum_{i=1}^{m+n+2}\lambda_i\psi_i\delta(t-\tau_i),$$

with the terminal condition $u(T,\cdot)=0$.

Numerical example In Guo, Loeper, Oblój, and Wang (2020), the process X, also called the *ot*-calibrated model, was calibrated to market data as of September 1st, 2020. The data consists of monthly SPX options maturing at 17 days and 45 days and monthly VIX futures and options maturing at 15 days. We also add the singular contract as a calibrating instrument (i.e., $\mathbb{E}^{\mathbb{Q}}\psi^{sg}(X_T)=0$) to ensure that the additional property $X^2_{T,T}=0$, $\mathbb{Q}$ a.s. is satisfied.

Define $A(t,\kappa) := (1-e^{-\kappa(T-t)})/\kappa$ and $\nu(t,x,\kappa,\theta) := A(t,\kappa)^{-1}(2x-\theta(T-t))+\theta$. The $\bar\alpha$ in (29) was set to

$$\bar\alpha(t,x) = \begin{bmatrix} \nu(t,x_2,\bar\kappa,\bar\theta) & \frac{1}{2}\bar\rho\bar\eta A(t,\bar\kappa)\nu(t,x_2,\bar\kappa,\bar\theta) \\ \frac{1}{2}\bar\rho\bar\eta A(t,\bar\kappa)\nu(t,x_2,\bar\kappa,\bar\theta) & \frac{1}{4}\bar\eta^2 A(t,\bar\kappa)^2\nu(t,x_2,\bar\kappa,\bar\theta)\end{bmatrix}, \tag{31}$$

where $(\bar\kappa,\bar\theta,\bar\eta,\bar\rho) = (4.99, 0.038, 0.52, -0.99)$. Such $\bar\alpha$ was derived by reformulating a standard Heston model in terms of X^1 and X^2 defined in (26) and (27). The parameters $(\bar\kappa,\bar\theta,\bar\eta,\bar\rho)$ have the usual interpretations as in the Heston model and are obtained by (roughly) calibrating a Heston model to the SPX option prices. The initial position is $X_0 = (8.1673, 0.0048)$. In addition, a *reference iteration* method was used for smoothing the volatility surfaces and skews, which is similar to iterating $\bar\sigma$ in the local volatility example presented in Section C.1. We refer the reader to Guo, Loeper, Oblój, and Wang (2020) for more details.

Figure 10 shows the model volatility skews of the *ot*-calibrated model. The simulation of X is given in Figure 11. The results show that the model accurately attains the market prices while keeping the property $X^2_{T,T}=0$, $\mathbb{Q}$ a.s. satisfied.

§3 Machine Learning Approaches: Learning the Local Volatility Via Shape Constraints

In this section we explore the abilities of two machine learning approaches for no-arbitrage interpolation of European vanilla option prices[27], which jointly yield the corresponding local volatility surface: a finite dimensional Gaussian process (GP) regression approach under no-arbitrage constraints based on prices, and a neural net (NN) approach with penalization of arbitrages based on implied volatilities[28]. We demonstrate the performance of these approaches relative to the SSVI industry standard. The GP

[27] cf. VI.§2.

[28] and for completeness we also include numerical results obtained by NN interpolation of the prices as in VI.§2.

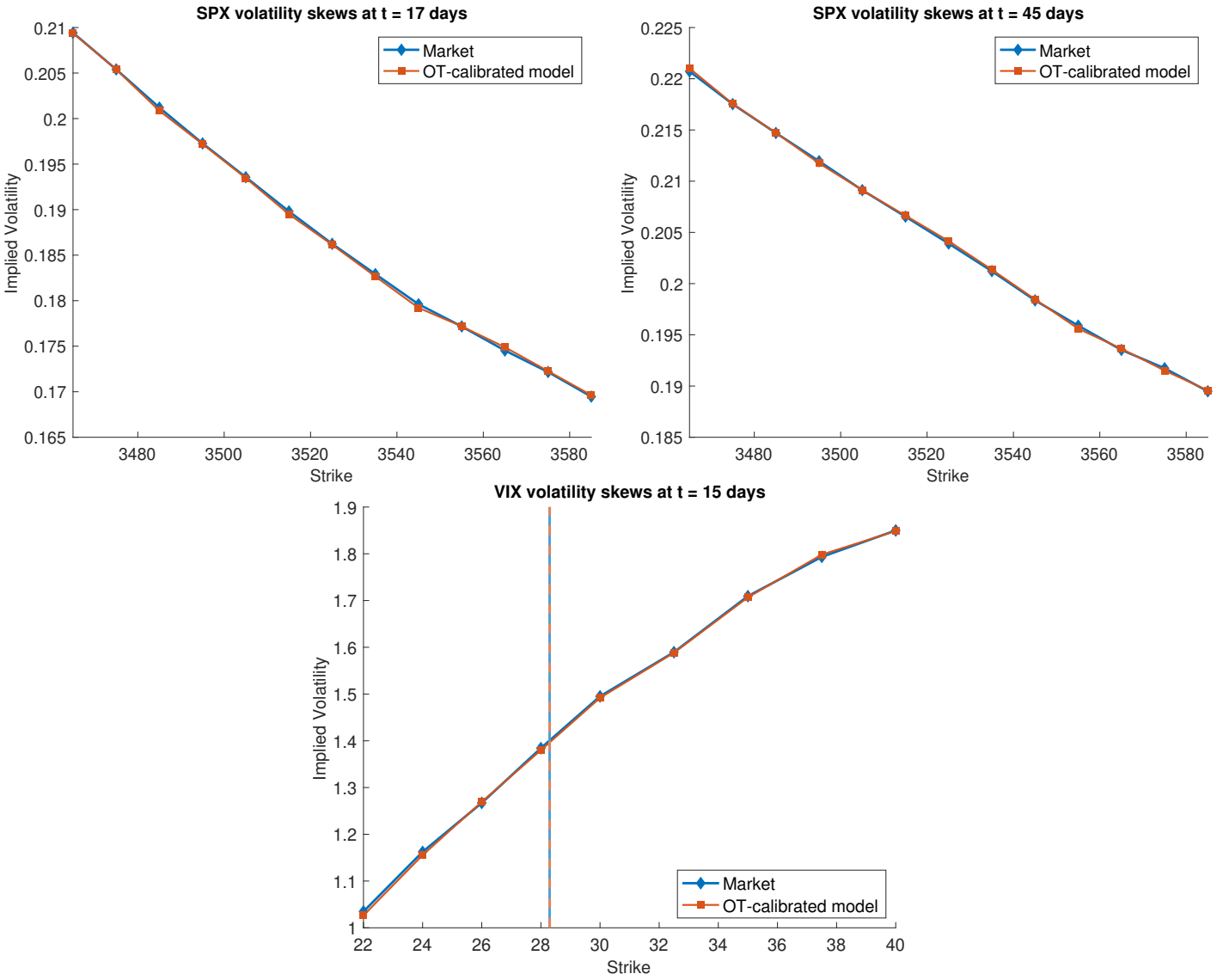


Figure 10: Approximated *ot*-calibrated model volatility skews of SPX options at 17 days, SPX options at 45 days and VIX options at 15 days in the joint calibration numerical example. The vertical lines are VIX futures prices. Markers correspond to computed prices which are then interpolated with a piece-wise linear function.

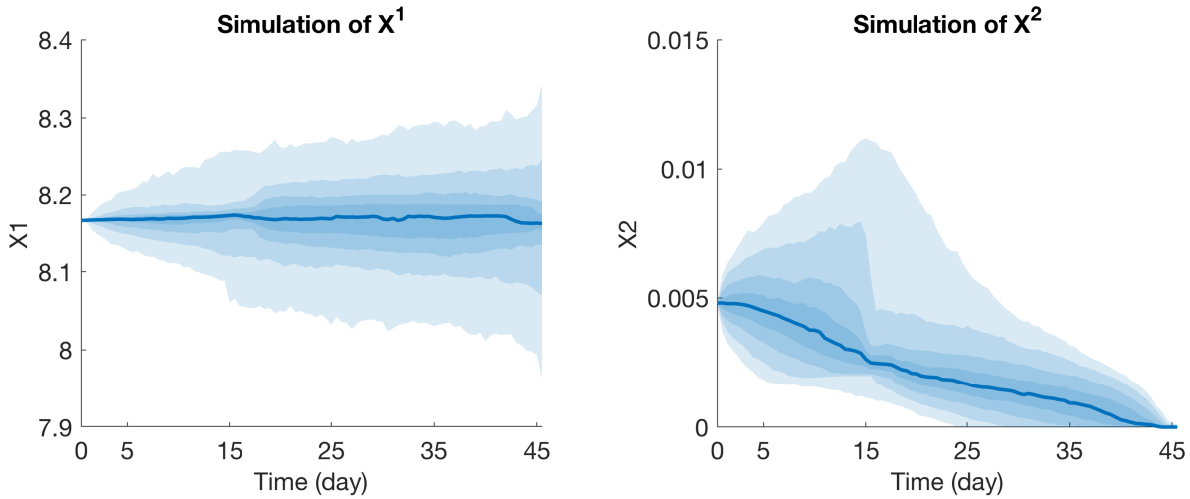


Figure 11: The simulations of the *ot*-calibrated model X in the joint calibration numerical example.

and SSVI approaches are arbitrage-free, whereas arbitrages are only penalized under the NN approach (the arbitrage-free NN approach of VI.§2.B.1 was found insufficiently expressive in VI.§2.E). The GP approach obtains the best out-of-sample calibration error and provides uncertainty quantification. The NN approach yields a smoother local volatility and a better backtesting performance, as its training criterion incorporates a local volatility regularization term.

A python notebook, compatible with Google colab, matlab files, and accompanying data are available in https://github.com/mChataign/Beyond-Surrogate-Modeling-Learning-the-Local-Volatility-Via-Shape-Constraints. Due to file size constraints, the notebook must be run to reproduce the figures and results in this section.

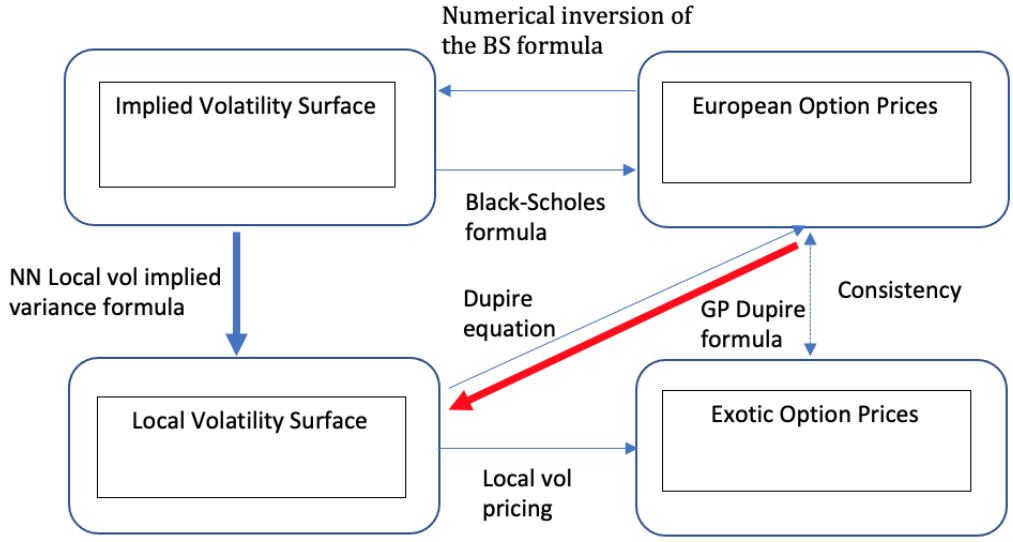


Figure 12: *Mathematical connections between option prices, implied, and local volatility, and our goal of this section, namely to either use the Dupire formula with Gaussian processes to jointly approximate the vanilla price and local volatility surfaces, or use the Dupire formula (or its reformulation in terms of implied volatility rather than prices) with neural networks to jointly approximate the implied volatility and local volatility surfaces.*

The setup is the one of VI.§2.A.

A Gaussian Process Regression for Learning Arbitrage-Free Price Surfaces

Our first goal is to construct, by Gaussian process regression, an arbitrage-free and continuous put price surface $P : \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R}_+$ interpolating the market prices P_* up to some error term, and to retrieve the corresponding local volatility surface $\sigma(\cdot, \cdot)$ by the Dupire (1994a) formula (assuming P of class $\mathcal{C}^{1,2}$ on $\{T > 0\}$)

$$\frac{\sigma^2(T,K)}{2} = \frac{\partial_T P(T,K) + (r-q)K\partial_K P(T,K) + qP(T,K)}{K^2 \partial^2_{K^2} P(T,K)}.$$

In terms of the reduced prices $p(T,k) = e^{qT}P(T,K)$, where $k = Ke^{-(r-q)T}$, the formula reads

$$\frac{\sigma^2(T,K)}{2} = \frac{\partial_T p(T,k)}{k^2 \partial^2_{k^2} p(T,k)} =: \mathrm{dup}(T,k). \tag{32}$$

Obviously, for the Dupire formula to be meaningful, its output must be nonnegative. This holds, in particular, whenever the interpolating map p exhibits nonnegative derivatives w.r.t. T and second derivative w.r.t. k, i.e. (as already considered in VI.(12))

$$\partial_T p(T,k) \geq 0\,,\ \partial^2_{k^2} p(T,k) \geq 0. \tag{33}$$

In this part, we construct by a constrained GP regression reduced put price surfaces $(T,k) \mapsto p(T,k)$ satisfying the conditions (33) from n noisy observations $\boldsymbol{y} = [y_1, \ldots, y_n]^\top$ of function p at input points $\mathbf{x} = [x_1, \ldots, x_n]^\top$. The input points $x_i = (T_i, k_i)$ correspond to observed maturities and strikes. The market fit condition is written as

$$\boldsymbol{y} = p(\mathbf{x}) + \boldsymbol{\varepsilon}, \tag{34}$$

where $p(\mathbf{x}) = [p(x_1), \ldots, p(x_n)]^\top$ is the vector composed of the put prices at the observation points. The additive noise term $\boldsymbol{\varepsilon} = [\varepsilon_1, \ldots, \varepsilon_n]^\top$ is assumed to be a zero-mean Gaussian vector, independent from $p(\mathbf{x})$, and with an homoscedastic covariance matrix given as $\varsigma^2 I_n$, where I_n is the identity matrix of dimension n.

Remark 3 *Tegnér and Roberts (2021, see their Eq. (10)) first attempt the use of GPs for local volatility modeling by placing a Gaussian prior directly on the local volatility surface (to guarantee the positivity of the local volatility, they assign a positive function on the prior). Their approach leads to a nonlinear least squares training loss function, as it involves the nonlinear transformation of the local volatility into the corresponding vanilla option prices. The resulting loss function is not obviously amenable to gradient descent (stochastic or not), so the authors resort to a MCMC optimization.*

A.1 Classical Gaussian process regression

We consider a zero-mean Gaussian process prior on the mapping $p = p(x)_{x\in\Omega}$ with covariance function (or kernel function) c. Then, the output vector $p(\mathbf{x})$ has a normal distribution with zero mean and covariance matrix $\mathbf{C}$ with components $\mathrm{cov}(p(x_i), p(x_j)) = c(x_i, x_j)$. We use the 2-dimensional isotropic covariance kernel given, for any $x = (T,k), x' = (T',k') \in \Omega$, as

$$c(x,x') = \sigma^2 c_T(T - T', \theta_T) c_k(k - k', \theta_k). \tag{35}$$

Here $(\theta_T, \theta_k) = \theta$ and σ^2 correspond to the length scale and the variance hyper-parameters of the kernel function c and the functions c_T and c_k are kernel correlation functions. By Gaussian conditioning, the conditional process $p \mid p(\mathbf{x}) + \boldsymbol{\varepsilon} = \boldsymbol{y}$ is Gaussian with mean function $\eta_{\boldsymbol{y}}$ and covariance function $c_{\boldsymbol{y}}$ such that

$$\eta_{\boldsymbol{y}}(x) = \boldsymbol{c}(x)^\top (\mathbf{C} + \varsigma^2 I_n)^{-1} \boldsymbol{y}, \quad x \in \Omega \tag{36}$$

$$c_{\boldsymbol{y}}(x,x') = c(x,x') - \boldsymbol{c}(x)^\top (\mathbf{C} + \varsigma^2 I_n)^{-1} \boldsymbol{c}(x'), \quad x, x' \in \Omega \tag{37}$$

where $\boldsymbol{c}(x) = [c(x,x_1), \ldots, c(x,x_n)]^\top$.

Without consideration of the conditions (33), (unconstrained) kriging prediction and uncertainty quantification are made using the conditional distribution $p \mid p(\mathbf{x}) + \boldsymbol{\varepsilon} = \boldsymbol{y}$. The best linear unbiased estimator of p is given as the conditional mean function (36). The conditional covariance function (37) can then be used to obtain confidence bands around the predicted price surface. The hyper-parameters of the kernel function c as well as the variance ς^2 of the noise can be estimated using a maximum likelihood estimator (MLE).

A.2 Imposing the no-arbitrage conditions

To deal with the constraints (12), we adopt the solution of Cousin et al. Cousin et al. (2016) that consists in constructing a finite dimensional approximation p^h of the Gaussian prior p for which these constraints can be imposed in the entire domain Ω with a finite number of checks. One then recovers the (non Gaussian) constrained posterior distribution by sampling a truncated Gaussian process.

Remark 1 *Switching to a finite dimensional approximation can also be viewed as a form of regularization, which is also required to deal with the ill-posedness of the (numerical differentiation) Dupire formula.*

We first consider a discretized version of the (rescaled) input space $\Omega = [0,1]^2$ as a regular grid $(\imath h)_\imath$, where $\imath = (i,j)$, for a suitable mesh size h and indices i, j ranging from 0 to $1/h$ (taken in $\mathbb{N}^\star$). For each knot $\imath = (i,j)$, we introduce the hat basis functions $\phi_\imath$ with support $[(i-1)h,(i+1)h]\times[(j-1)h,(j+1)h]$ given, for $x = (T,k)$, by

$$\phi_\imath(x) = \max(1 - \frac{|T - ih|}{h}, 0)\max(1 - \frac{|k - jh|}{h}, 0).$$

We take $V = H^1(\Omega) = \{u \in L_2(\Omega) : D^\alpha u \in L_2(\Omega), |\alpha| \leq 1\}$, where $D^\alpha u$ is a weak derivative of order $|\alpha|$, as the space of (the realizations of) p. Let $V^h \subset V$ denote the finite dimensional linear subspace spanned by the M linearly independent basis functions $\phi_\imath$. The (random) surface p in V is projected onto V^h as

$$p^h(x) = \sum_\imath p(\imath h)\phi_\imath(x),\ \forall x \in \Omega. \tag{38}$$

If we denote $\varrho_\imath = p(\imath h)$, then $\boldsymbol{\varrho} = (\varrho_\imath)_\imath$ is a zero-mean Gaussian column vector (indexed by $\imath$) with $M \times M$ covariance matrix Γ^h such that $\Gamma^h_{\imath,\jmath} = c(\imath h, \jmath h)$, for any two grid nodes $\imath$ and $\jmath$. Let $\boldsymbol{\phi}(x)$ denote the vector of size M given by $\boldsymbol{\phi}(x) = (\phi_\imath(x))_\imath$. The equality (38) can be rewritten as $p^h(x) = \boldsymbol{\phi}(x) \cdot \boldsymbol{\varrho}$. Denoting by $p^h(\mathbf{x}) = [p^h(x_1), \ldots, p^h(x_n)]^\top$ and by $\boldsymbol{\Phi}(\mathbf{x})$ the $n \times M$ matrix of basis functions where each row ℓ corresponds to the vector $\boldsymbol{\phi}(x_\ell)$, one has $p^h(\mathbf{x}) = \boldsymbol{\Phi}(\mathbf{x}) \cdot \boldsymbol{\varrho}$. By application of the results of Maatouk and Bay (2017):

Proposition 2 (i) *The finite dimensional process p^h converges uniformly to p on Ω as $h \to 0$, almost surely,*

(ii) *$p^h(T,k)$ is a nondecreasing function of T if and only if $\varrho_{i+1,j} \geq \varrho_{i,j}, \forall (i,j)$,*

(iii) *$p^h(T,k)$ is a convex function of k if and only if $\varrho_{i,j+2} - \varrho_{i,j+1} \geq \varrho_{i,j+1} - \varrho_{i,j}, \forall (i,j)$.* ∎

In view of (i), denoting by $\mathcal{I}$ the set of 2d continuous positive functions which are nondecreasing in T and convex in k, we choose as constrained GP metamodel for the put price surface the law of p^h conditional on

$$\begin{cases} p^h(\mathbf{x}) + \boldsymbol{\varepsilon} = \boldsymbol{y} \\ p^h \in \mathcal{I}. \end{cases}$$

In view of (ii)-(iii), $p^h \in \mathcal{I}$ if and only if $\boldsymbol{\varrho} \in \mathcal{I}^h$, where $\mathcal{I}^h$ corresponds to the set of ($\imath$ indexed) vectors $\boldsymbol{\rho} = (\rho_\imath)_\imath$ such that $\rho_{i+1,j} \geq \rho_{i,j}$ and $\rho_{i,j+2} - \rho_{i,j+1} \geq \rho_{i,j+1} - \rho_{i,j}$ $\forall (i,j)$. Hence, our GP metamodel for the put price surface can be reformulated as the law of $\boldsymbol{\varrho}$ conditional on

$$\begin{cases} \boldsymbol{\Phi}(\mathbf{x}) \cdot \boldsymbol{\varrho} + \boldsymbol{\varepsilon} = \boldsymbol{y} \\ \boldsymbol{\varrho} \in \mathcal{I}^h. \end{cases} \tag{39}$$

A.3 Hyper-parameter learning

Hyper-parameters consist in the length scales θ and the variance parameter σ^2 in (35), as well as the noise variance ς. Up to a constant, the so called marginal log likelihood of $\boldsymbol{\varrho}$ at $\lambda = [\theta, \sigma, \varsigma]^\top$ can be expressed as (see e.g. (Murphy, 2012, Section 15.2.4, p. 523)):

$$\mathcal{L}(\lambda) = -\frac{1}{2}\boldsymbol{y}^\top\left(\boldsymbol{\Phi}(\mathbf{x})\Gamma^h\boldsymbol{\Phi}(\mathbf{x})^\top + \varsigma^2 I_n\right)^{-1}\boldsymbol{y} - \frac{1}{2}\log\Big(\det\left(\boldsymbol{\Phi}(\mathbf{x})\Gamma^h\boldsymbol{\Phi}(\mathbf{x})^\top + \varsigma^2 I_n\right)\Big).$$

We maximize $\mathcal{L}$ for learning the hyper-parameters λ (MLE estimation).

Remark 3 *The above expression does not take into account the inequality constraints in the estimation. However, Bachoc et al. (Bachoc et al., 2019, see e.g. their Eq. (2)) argue (and we observed empirically) that, unless the sample size is very small, conditioning by the constraints significantly increases the computational burden with negligible impact on the MLE.*

A.4 The most probable response surface and measurement noises

We compute the joint MAP $(\widehat{\boldsymbol{\rho}}, \widehat{\boldsymbol{e}})$ of the truncated Gaussian vector $\boldsymbol{\varrho}$ and of the Gaussian noise vector $\boldsymbol{\varepsilon}$,

$$(\widehat{\boldsymbol{\rho}}, \widehat{\boldsymbol{e}}) = \underset{(\boldsymbol{\rho}, \boldsymbol{e})}{\text{Argmax}}\ \text{Prob}\left(\boldsymbol{\varrho} \in [\boldsymbol{\rho}, \boldsymbol{\rho} + \mathrm{d}\boldsymbol{\rho}], \boldsymbol{\varepsilon} \in [\boldsymbol{e}, \boldsymbol{e} + \mathrm{d}\boldsymbol{e}] \mid \boldsymbol{\Phi}(\mathbf{x}) \cdot \boldsymbol{\varrho} + \boldsymbol{\varepsilon} = \boldsymbol{y},\ \boldsymbol{\varrho} \in \mathcal{I}^{\mathrm{h}}\right)$$

(for the probability measure Prob underlying the GP model). As $(\boldsymbol{\varrho}, \boldsymbol{\varepsilon})$ is Gaussian centered with block-diagonal covariance matrix with blocks Γ^h and $\varsigma^2 I_n$, this implies that the MAP $(\widehat{\boldsymbol{\rho}}, \widehat{\boldsymbol{e}})$ is a solution to the following quadratic problem :

$$\underset{\boldsymbol{\Phi}(\mathbf{x}) \cdot \boldsymbol{\rho} + \boldsymbol{e} = \boldsymbol{y},\ \boldsymbol{\rho} \in \mathcal{I}^h}{\text{Argmin}} \left(\boldsymbol{\rho}^\top (\Gamma^h)^{-1} \boldsymbol{\rho} + \boldsymbol{e}^\top (\varsigma^2 I_n)^{-1} \boldsymbol{e}\right). \tag{40}$$

We define the most probable measurement noise to be $\widehat{\boldsymbol{e}}$ and the most probable response surface $\widehat{p}^h(\mathbf{x}) = \boldsymbol{\Phi}(\mathbf{x}) \cdot \widehat{\boldsymbol{\rho}}$. Distance to the data can be an effect of arbitrage opportunities within the data and/or misspecification / lack of expressiveness of the kernel.

A.5 Sampling finite dimensional Gaussian processes under shape constraints

The conditional distribution of $\boldsymbol{\varrho} \mid \boldsymbol{\Phi}(\mathbf{x}) \cdot \boldsymbol{\varrho} + \boldsymbol{\varepsilon} = \boldsymbol{y}$ is multivariate Gaussian with mean $\boldsymbol{\eta}_{\mathbf{y}}(\mathbf{x})$ and covariance matrix $\mathbf{C}_{\mathbf{y}}(\mathbf{x})$ such that

$$\boldsymbol{\eta}_{\mathbf{y}}(\mathbf{x}) = \Gamma^h \boldsymbol{\Phi}(\mathbf{x})^\top (\boldsymbol{\Phi}(\mathbf{x}) \Gamma^h \boldsymbol{\Phi}(\mathbf{x})^\top + \varsigma^2 I_n)^{-1} \boldsymbol{y} \tag{41}$$

$$\mathbf{C}_{\mathbf{y}}(\mathbf{x}) = \Gamma^h \boldsymbol{\Phi}(\mathbf{x})^\top (\boldsymbol{\Phi}(\mathbf{x}) \Gamma^h \boldsymbol{\Phi}(\mathbf{x})^\top + \varsigma^2 I_n)^{-1} \boldsymbol{\Phi}(\mathbf{x}) \Gamma^h. \tag{42}$$

In view of (39), we thus face the problem of sampling from this truncated multivariate Gaussian distribution, which we do by Hamiltonian Monte Carlo, using the MAP $\widehat{\boldsymbol{\varrho}}$ of $\boldsymbol{\varrho}$ as the initial vector (which must verify the constraints) in the algorithm.

A.6 Local volatility

Due to the shape constraints and to the ensuing finite-dimensional approximation with basis functions of class $\mathcal{C}^0$ (for the sake of Proposition 2), p^h is not differentiable. Hence, exploiting GP derivatives analytics, as done for the mean in (Crépey and Dixon, 2020, cf. Eq. (10)) and also for the covariance in Ludkovski and Saporito (2020), is not possible for deriving the corresponding local volatility surface here. Computation of derivatives involved in the Dupire formula is implemented by finite differences with respect to a coarser grid (than the grid of basis functions). Another related solution would be to formulate a weak form of the Dupire equation and construct a local volatility surface approximation using a finite element method.

See Algorithm 2 for the main steps of the GP approach.

Algorithm 2: The GP algorithm for local volatility surface approximation.

Data: Put price training set $p_\star$
Result: M realizations of the local volatility surface $\{\mathrm{dup}_i^h\}_{i=1}^M$
$\widehat{\lambda} \leftarrow$ Maximize the marginal log-likelihood of the put price surface p^h w.r.t. λ `// Hyperparameter fitting`;
$(\widehat{\boldsymbol{\rho}}, \widehat{\boldsymbol{e}}) \leftarrow$ Minimize quadratic problem (40) based on $\widehat{\lambda}$ `// Joint MAP estimate`;
$\widehat{\boldsymbol{\rho}} \rightarrow$ Initialize a Hamiltonian MC sampler;
$p_1^h, \ldots, p_M^h \leftarrow$ Hamiltonian MC Sampler `// Sampling price surfaces` ;
$\mathrm{dup}_i^h \leftarrow$ Finite difference approximation using each $p_i^h,\ i := 1 \rightarrow M$;

B Neural Networks Implied Volatility Metamodeling

Our second goal is to use neural nets (NN) to construct an implied volatility (IV) put surface $\Sigma : \mathbb{R}_+ \times \mathbb{R} \to \mathbb{R}_+$, interpolating implied volatility market quotes Σ_* up to some error term, both being stated in terms of a put option maturity T and log-(forward) moneyness $\kappa = \log(\frac{k}{S_0}) = \log\left(\frac{K}{S_0}\right) - (r-q)T$. The advantage of using implied volatilities rather than prices (as previously done in Chataigner et al. (2020)), both being in bijection via the Black-Scholes put pricing formula as well known, is their lower variability, hence better performance as we will see.

The corresponding local volatility surface σ is given by the following local volatility implied variance formula, i.e. the Dupire formula stated in terms of the implied total variance[29] $\Theta(T,\kappa) = \Sigma^2(T,\kappa)T$ (assuming Θ of class $\mathcal{C}^{1,2}$ on $\{T>0\}$):

$$\sigma^2(T,K) = \frac{\partial_T \Theta}{1 - \frac{\kappa}{\Theta}\partial_\kappa \Theta + \frac{1}{4}\left(-\frac{1}{4} - \frac{1}{\Theta} + \frac{\kappa^2}{\Theta^2}\right)(\partial_\kappa \Theta)^2 + \frac{1}{2}\partial_{\kappa^2}\Theta}(T,\kappa) \qquad =: \frac{\mathrm{cal}_T(\Theta)}{\mathrm{butt}_k(\Theta)}(T,\kappa). \tag{43}$$

We use a feedforward NN with weights $\mathbf{W}$, biases $\mathbf{b}$ and smooth activation functions for parameterizing the implied volatility and total variance, which we denote by

$$\Sigma = \Sigma_{\mathbf{W},\mathbf{b}}\,,\ \Theta = \Theta_{\mathbf{W},\mathbf{b}}.$$

The terms $\mathrm{cal}_T(\Theta_{\mathbf{W},\mathbf{b}})$ and $\mathrm{butt}_k(\Theta_{\mathbf{W},\mathbf{b}})$ are available analytically, by automatic differentiation, which we exploit below to penalize calendar spread arbitrages, i.e. negativity of $\mathrm{cal}_T(\Theta)$, and butterfly arbitrage, i.e. negativity of $\mathrm{butt}_k(\Theta)$.

The training of NNs is a non-convex optimization problem and hence does not guarantee convergence to a global optimum. We must therefore guide the NN optimizer towards a local optima that has desirable properties in terms of interpolation error and arbitrage constraints. This motivates the introduction of an arbitrage penalty function into the loss function to select the most appropriate local minima. An additional challenge is that maturity-log moneyness pairs with quoted option prices are unevenly distributed and the NN may favor fitting to a cluster of quotes to the detriment of fitting isolated points. To remedy this non-uniform data fitting problem, we re-weight the observations by the Euclidean distance between neighboring points. More precisely, given n observations $\chi_i = (T_i, \kappa_i)$ of maturity-log moneyness pairs and of the corresponding market implied volatilities $\Sigma_*(\chi_i)$, we construct the $n \times n$ distance matrix with general term $d(\chi_i,\chi_j) = \sqrt{(T_j - T_i)^2 + (\kappa_j - \kappa_i)^2}$. We then define the loss weighting w_i for each point χ_i as the distance $w_i = \min_{j, j\neq i} d(\chi_i,\chi_j)$. with the closest point. These modifications aim at reducing error for any isolated points. In addition, in order to avoid linear saturation of the neural network, we apply a further log-maturity change of variables (adapting the partial derivatives accordingly).

Learning the weights $\mathbf{W}$ and biases $\mathbf{b}$ to the data subject to no arbitrage soft constraints (i.e. with penalization of arbitrages) then takes the form of the following (nonconvex) loss minimization problem:

$$\underset{\mathbf{W},\mathbf{b}}{\mathrm{Argmin}} \sqrt{\frac{1}{n}\sum_i \left(w_i \frac{\Sigma_{\mathbf{W},\mathbf{b}}(\chi_i) - \Sigma_*(\chi_i)}{\Sigma_*(\chi_i)}\right)^2} + \frac{\mu_w}{m}\sum_{\rho\in\Omega_h} \lambda^\intercal \mathcal{R}(\Theta_{\mathbf{W},\mathbf{b}})(\rho), \tag{44}$$

where $\lambda = [\lambda_1,\lambda_2,\lambda_3]^\top \in \mathbb{R}^3_+$ and

$$\mathcal{R}(\Theta) = [\mathrm{cal}_T^-(\Theta), \mathrm{butt}_k^-(\Theta), \left(\frac{\mathrm{cal}_T}{\mathrm{butt}_k}(\Theta) - \overline{a}\right)^+ + \left(\frac{\mathrm{cal}_T}{\mathrm{butt}_k}(\Theta) - \underline{a}\right)^-]^\top$$

is a regularization penalty vector evaluated over a penalty grid Ω_h with m nodes as detailed below. The error criterion is calculated as a root mean square error on relative difference, so that it does not

[29] This follows from the Dupire formula by simple transforms detailed in (Gatheral, 2011, p.13).

discriminate high or low implied volatilities. The first two elements in the penalty vector $\mathcal{R}(\Theta)$ favor the no-arbitrage conditions (12) and the third element favors desired lower and upper bounds $0 < \underline{a} < \overline{a}$ (constants or functions of T) on the estimated local variance $\sigma^2(T, K)$. In order to adjust the weight of penalization, we multiply our penalties by the weighting mean $\mu_w := \frac{1}{m}\sum_i w_i$. Suitable values of the "Lagrange multipliers" λ, ensuring the right balance between fit to the market implied volatilities and the constraints, is then obtained by grid search. Of course a soft constraint (penalization) approach does not fully prevent arbitrages. However, for large λ, arbitrages are extremely unlikely to occur, except perhaps very far from Ω. With this in mind, we use a penalty grid Ω_h that extends well beyond the domain of the IV interpolation. This is intended so that the penalty term penalizes arbitrages outside of the domain used for IV Interpolation.

See Algorithm 3 for the pseudo-code of the NN approach.

Algorithm 3: The NN-IV algorithm for local volatility surface approximation.

Data: Market implied volatility surface Σ_*

Result: The local volatility surface $\sqrt{\frac{\text{cal}_T}{\text{butt}_k}(\Theta_{\widehat{\mathbf{W}},\widehat{\mathbf{b}}})}$

$(\widehat{\mathbf{W}}, \widehat{\mathbf{b}}) \leftarrow$ Minimize the penalized training loss (44) w.r.t. $(\mathbf{W}, \mathbf{b})$;

$\sqrt{\frac{\text{cal}_T}{\text{butt}_k}(\Theta_{\widehat{\mathbf{W}},\widehat{\mathbf{b}}})} \leftarrow$ AAD differentiation of the trained NN implied vol. surface;

C Neural Network Price MetaModeling With Dupire Penalization

We also consider the approach analogous to the above, but based on prices instead of implied volatilities. In other words, we consider the approach of VI.§2.B.2, but with an additional penaly term, namely for a regularization penalty vector, ϕ there, replaced by

$$\phi := [(\partial_T p)^-, (\partial^2_{k^2} p)^-, (\text{dup} - \overline{a})^+ + (\text{dup} - \underline{a})^-],$$

where dup is related to p through (32) (and the vector of Lagrange multipliers λ is now in $\mathbb{R}^3_+$ as in (44)). The purpose of the penalization on the local variance bounds is to improve the overall fit in prices and stabilize the local volatility surface.

Table 1 shows the RMSEs in absolute pricing resulting from repeating the same set of experiments reported in Table VI.2, but with the half-variance bounds included in the penalization. For the sparse network with hard constraints, we set $\lambda = [0, 0, 10]$ and choose $\underline{a} = 0.05^2/2$ and $\overline{a} = 0.4^2/2$. For the sparse and dense networks with soft constraints, we set $\lambda = [1.0 \times 10^5, 1.0 \times 10^3, 10]$. Compared to Table VI.2, we observe improvement in the test error for the hard and soft constraints approaches when including the additional local volatility penalty term. Table 2 is the analog of Table VI.3, with similar conclusions. Note that, here as there, the arbitrage opportunities that arise are not only very few (except in the unconstrained case), but also very far from the money and, in fact, mainly regard the learning of the payoff function, corresponding to the horizon $T = 0$. See for instance Figure 13 for the location of the violations that arise in the unconstrained case with Dupire penalization. Hence such apparent 'arbitrage opportunities' cannot necessarily be monetised once liquidity is accounted for.

	Sparse network		Dense network	
	Hard constraints	Soft constraints	Soft constraints	No constraints
Training dataset	28.04	3.44	2.48	3.48
Testing dataset	27.07	3.33	3.36	4.31
Indicative training times	400s	600s	300s	250s

Table 1: *Price RMSE (absolute pricing errors) and training times with Dupire penalization.*

	Sparse network		Dense network	
	Hard constraints	Soft constraints	Soft constraints	No constraints
Training dataset	0	0	0	30/254
Testing dataset	0	2/360	0	5/360

Table 2: *The fraction of static arbitrage violations with Dupire penalization.*

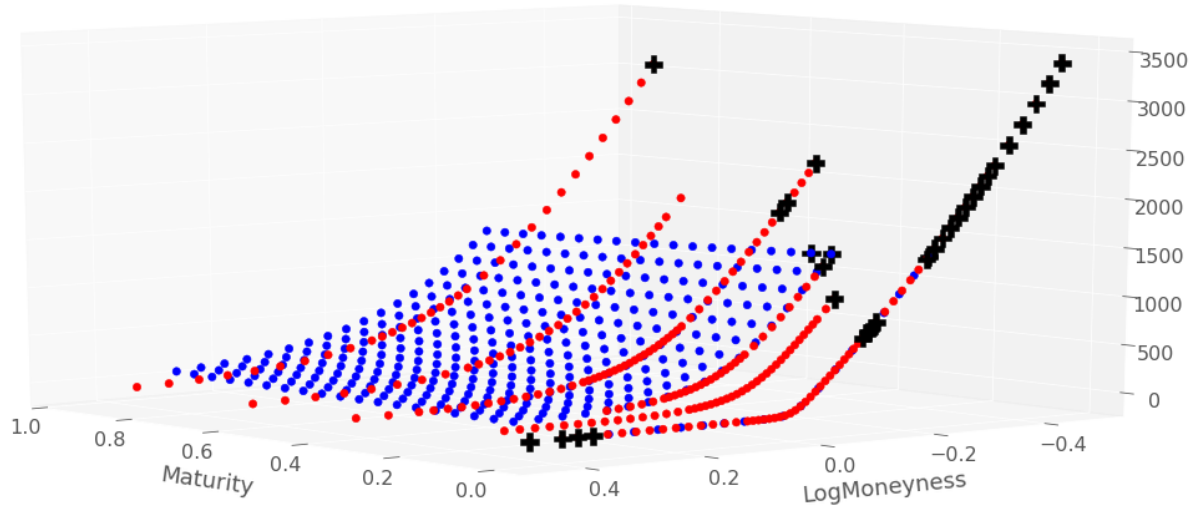


Figure 13: *Location of the violations, denoted by black crosses, corresponding to the right column in Table 2.*

D Benchmarking Results: Price Based Neural Net Approaches vs. Tikhonov Regularization

Regarding the local volatility, our main focus in this section, we first benchmark the price based neural net approaches of VI.§2.B.2 and C with Tikhonov regularization. Namely, after training, a local volatility surface is extracted from the price based neural nets by application of the Dupire formula (32), leveraging on the availability of the corresponding exact sensitivities, i.e., using automatic algorithmic differentiation (AAD) and not finite differences. This local volatility surface is then compared to the one obtained in Crépey (2002) by the Tikhonov regularization approach surveyed in VII.§3.C. Our motivation for this choice as a benchmark here is, first, the theoretical, mathematical justification for this method provided by Theorems 6.2 and 6.3 in (Crépey, 2003a). Second, it is price (as opposed to implied volatility) based, which makes it at par with our focus on *price based* neural network local volatility calibration schemes in this part. Third, it is non parametric ('model free' in this sense), like our neural network schemes again, and as opposed to various parameterizations such as SABR or SSVI that are used as standard in various segments of the industry, but come without theoretical justification for robustness, are restricted to specific industry segments on which they play the role of a market consensus, and are all implied volatility based (SSVI will in fact be included to the benchmarks also including the implied volatility based approaches in E). Fourth, an efficient numerical implementation of the Tikhonov method (as we call it for brevity hereafter), already put to the test of hundreds of real datasets in the context of Crépey (2004), is available through Crépey (2002). Fifth, this method is itself benchmarked to other (spline interpolation and constrained stochastic control) approaches Section 7 of Crépey (2002).

D.1 Numerical Stability Through Recalibration

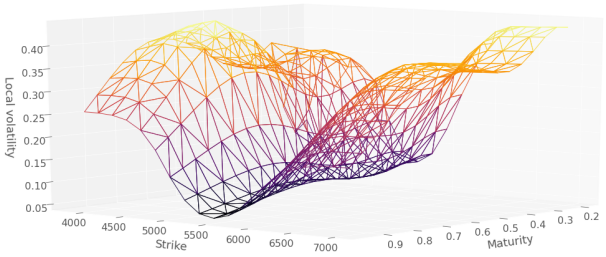
Figures 14, 15 and 16 show the comparison of the local volatility surfaces obtained using hard constraints (sparse network) without Dupire penalization, dense network and soft constraints without and with Dupire penalization, as well as the Tikhonov regularization approach of Crépey (2002), on price quotes listed on August 7^{th}, 8^{th}, and 9^{th}, 2001, respectively. The soft constraint approach without Dupire penalization is both irregular (exhibiting outliers on a given day) and unstable (from day to day). In contrast, the soft constraint approach with Dupire penalization yields a more regular (at least, less spiky) local volatility surface, both at fixed calendar time and in terms of stability across calendar time. From this point of view the results are then qualitatively comparable to those obtained by Tikhonov regularization (which is however quicker, taking of the order of 30s to run).

D.2 Monte Carlo backtesting repricing error

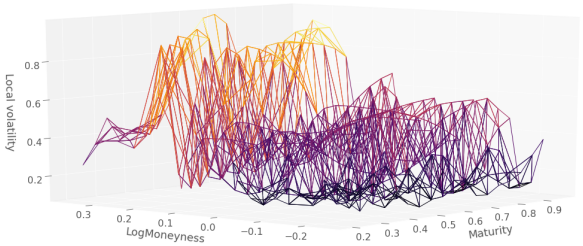
Next we evaluate the performance of the models in a backtesting Monte Carlo exercise. Namely, the options in each testing grid are repriced by Monte Carlo with 10^5 paths of 100 time steps in the model

$$\frac{dS_t}{S_t} = (r(t) - q(t))\,dt + \sigma(t, S_t)dW_t, \tag{45}$$

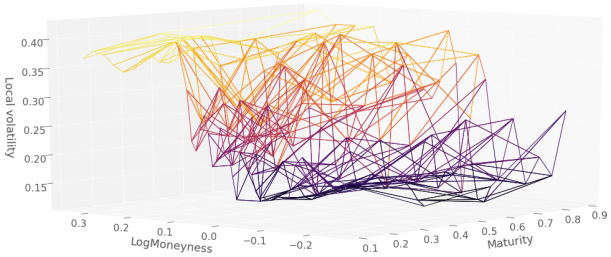
using differently calibrated local volatility functions $\sigma(\cdot,\cdot)$ in (45), for each of the 7th, 8th, and 9th August dataset. Table 3 shows the corresponding Monte Carlo backtesting repricing errors, using the option market prices from the training grids as reference values in the corresponding RMSEs. The neural network approaches provide a full surface of prices and local volatilities, as opposed to values at the calibration trinomial tree nodes only in the case of Tikhonov, for which the Monte Carlo backtesting exercise thus requires an additional layer of local volatility inter-extrapolation, here achieved by a nearest neighbors algorithm. We see from the table that both the benchmark Tikhonov method and the dense network soft constraints approach with Dupire penalization yield very reasonable and acceptable repricing errors (with still a certain advantage to the Tikhonov method), unlike the hard constraints



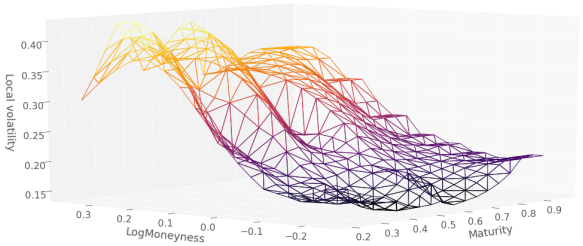
(a) Hard Constraints



(b) Soft constraints (without local vol. constraints)

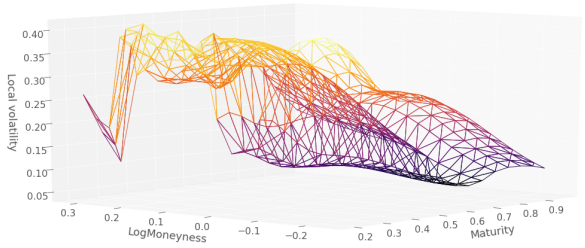


(c) Soft constraints (with local vol. constraints)

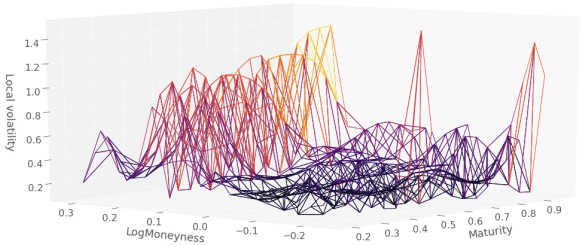


(d) Tikhonov local volatility

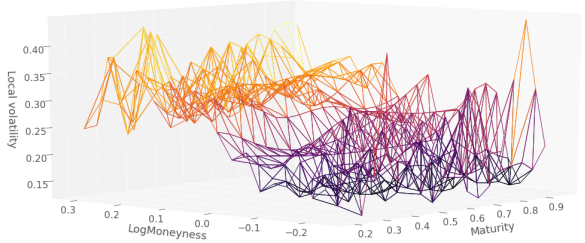
Figure 14: *Local volatility for 07/08/2001.*



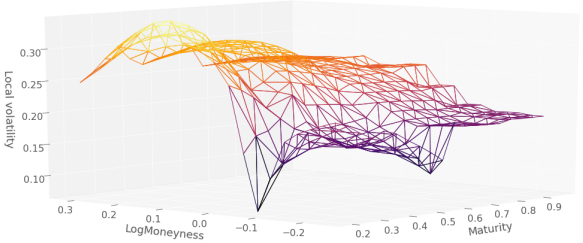
(a) Hard Constraints



(b) Soft constraints (without local vol. constraints)

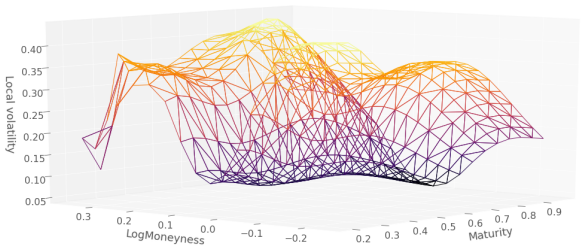


(c) Soft constraints (with local vol. constraints)

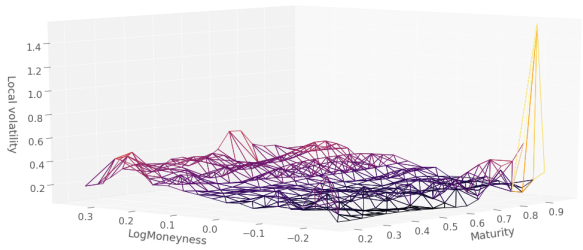


(d) Tikhonov local volatility

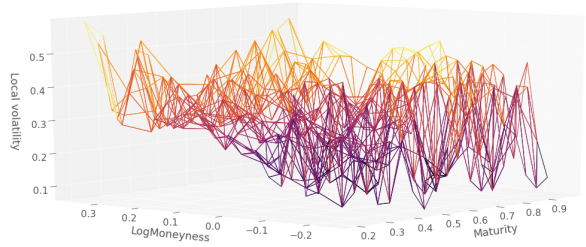
Figure 15: *Local volatility for 08/08/2001.*



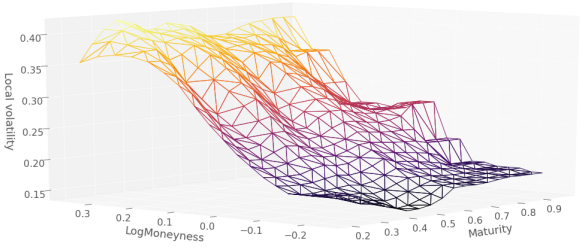
(a) Hard Constraints



(b) Soft constraints (without local vol. constraints)



(c) Soft constraints (with local vol. constraints)



(d) Tikhonov local volatility

Figure 16: *Local volatility for 09/08/2001.*

approaches. Moreover, the Dupire penalization is essential for extracting a decent local volatility function: The dense network with soft constraint but without this penalization yields very poor Monte Carlo repricing RMSEs.

$\sigma(\cdot,\cdot)$	Tikhonov Monte Carlo	Dense network with soft constraints and Dup. penal.	Dense network with soft constraints	Hard constraint with Dup. penal.	Hard constraint without Dup. Pen.
07/08/2001	5.42	10.18	68.48	48.57	50.44
08/08/2001	5.55	7.44	50.82	56.63	56.98
09/08/2001	4.60	8.18	59.39	66.23	65.50

Table 3: Monte Carlo backtesting repricing RMSEs on training grid against market prices.

Yet the residual gap between the Monte Carlo RMSEs of the (even best) price based neural network local volatility and of the Tikhonov local volatility is disappointing. Hence the need for further investigations also involving the approaches of A and B.

E Benchmarking Results: Neural Nets and Gaussian Processes vs. SSVI

We now compare numerically the approaches of parts A, B and C.

E.1 Experimental design

Our training set is prepared using SPX European puts with different available strikes and maturities ranging from 0.005 to 2.5 years, listed on 18th May 2019, with $S_0 = \$2859.53$. Each contract is listed with a bid/ask price and an implied volatility corresponding to the mid-price. The associated interest rate is constructed from US treasury yield curve and dividend yield curve rates are then obtained from call/put

parity applied to the option market prices and forward prices. We preprocess the data by removing the shortest maturity options, with $T < 0.055$, and the numerically inconsistent observations for which the gap between the listed implied volatility and the implied volatility calibrated from mid-price with our interest/dividend curves exceeds 5% of the listed implied volatility. But we do not remove arbitrable observations. The preprocessed training set is composed of 1720 market put prices. The testing set consists of a disjoint set of 1725 put prices.

All results for the GP method are based on using Matern $\nu = 5/2$ kernels over a $[0,1]^2$ domain with fitted kernel standard-deviation hyper-parameter $\widehat{\sigma} = 185.7611$, length-scale hyper-parameters $\widehat{\theta}_k = 0.3282$ and $\widehat{\theta}_T = 0.2211$, and homoscedastic noise standard deviation, $\widehat{\varsigma} = 0.6876$.[30] The grid of basis functions for constructing the finite-dimensional process p^h has 100 nodes in the modified strike direction and 25 nodes in the maturity direction. The Matlab interior point convex algorithm `quadprog` is used to solve the MAP quadratic program (40).

Regarding the NN approach, we use a three layer architecture similar to the one based on prices (instead of implied volatilities in Section B) in Chataigner et al. (2020), to which we refer the reader for implementation details. We use a penalty grid Ω_h with $m = 50 \times 100$ nodes. In the moneyness and maturity coordinates, the domain of the penalty grid is $[0.005, 10] \times [0.5, 2]$.

E.2 Arbitrage-free SVI

We benchmark the machine learning results with the industry standard provided by the arbitrage free stochastic volatility inspired (SVI) model of Gatheral and Jacquier (2014). Under the "natural parameterization" $\text{SVI} = (\Delta, \mu, \rho, \omega, \zeta)$, the implied total variance is given, for any fixed T, by

$$\Theta_{\text{SVI}}(\kappa) = \Delta + \frac{\omega}{2}\left(1 + \rho(\kappa - \mu)\zeta + \sqrt{(\zeta(\kappa - \mu) + \rho)^2 + (1 - \rho^2)}\right). \tag{46}$$

Our SSVI parameterization of a surface corresponds to $\text{SVI}_T = (0, 0, \rho, \Theta_T, \phi(\Theta_T))$ for each T, where Θ_T is the at-the-money total implied variance and we use for ϕ a power law function $\phi(\vartheta) = \frac{\eta}{\vartheta^\gamma(1+\vartheta)^{1-\gamma}}$. (Gatheral and Jacquier, 2014, Remark 4.4) provides sufficient conditions on SSVI parameters ($\eta(1+|\rho|) \leq 2$ with $\gamma = 0.5$) that rule out butterfly arbitrage, whereas SSVI is free of calendar arbitrage when Θ_T is nondecreasing.

We calibrate the model as in Gatheral and Jacquier (2014):[31] First, we fit the SSVI model; Second, for each maturity in the training grid, the five SVI parameters are calibrated, (starting in each case from the SSVI calibrated values. The implied volatility is obtained for new maturities by a weighted average of the parameters associated with the two closest maturities in the training grid, T and U, say, with weights determined by Θ_T and Θ_U. The corresponding local volatility is extracted by finite difference approximation of (43).

As, in practice, no arbitrage constraints are implemented for SSVI by penalization (see (Gatheral and Jacquier, 2014, Section 5.2)), in the end the SSVI approach is in fact only practically arbitrage-free, much like our NN approach, whereas it is only the GP approach that is proven arbitrage-free.

E.3 Calibration results

Training times for SSVI, GP, and NNs are reported in the last row of Table 4 which, for completeness, also includes numerical results obtained by NN interpolation of the prices as per Chataigner et al. (2020). Because price based NN results are outperformed by IV based NN results we only focus on the IV based NN in the figures that follow, referring to Chataigner et al. (2020) for every detail on the price based NN approach. We recall that, in contrast to the SSVI and NNs which fit to mid-quotes, GPs fit to the bid-ask prices.

[30] When re-scaled back to the original input domain, the fitted length scale parameters of the 2D Matern $\nu = 5/2$ are $\widehat{\theta}_k = 973.1901$ and $\widehat{\theta}_T = 0.5594$.

[31] Building on https://www.mathworks.com/matlabcentral/profile/authors/4439546.

IV RMSE (Price RMSE)	SSVI	GP	IV based NN	Price based NN	SSVI Unconstr.	GP Unconstr.	IV based NN Unconstr.	Price based NN Unconstr.
Calibr. fit on the training set	1.37% (2.574)	0.58% (0.338)	1.23% (2.897)	13.70% (9.851)	1.04% (2.691)	0.60% (0.321)	0.84% (2.163)	5.65 % (2.456)
Calibr. fit on the testing set	1.52% (2.892)	0.57% (0.355)	1.29% (2.966)	14.27% (10.347)	1.09% (2.791)	0.57% (0.477)	0.86% (2.045)	6.14% (2.888)
MC backtest	8.69% (22.826)	19.76% (74.017)	2.95% (4.989)	6.37% (11.764)	N/A	N/A	N/A	N/A
CN backtest	6.88% (33.545)	7.86% (35.270)	3.43% (11.976)	5.56% (26.785)	N/A	N/A	N/A	N/A
Comput. time (seconds)	33	856	191	185	1	16	76	229

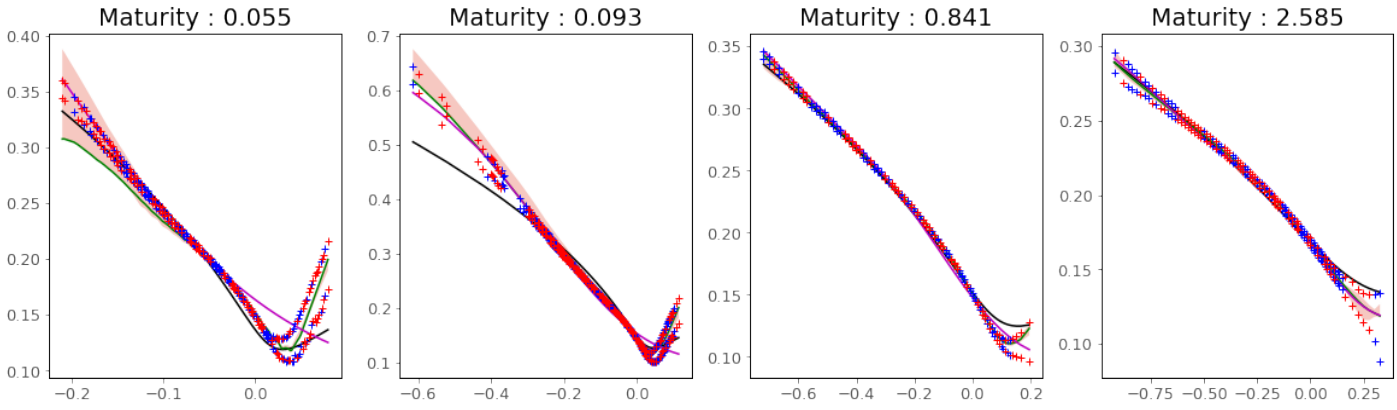
Table 4: The IV and price RMSEs of the SSVI, GP and NN approaches. Last row: computation times (in seconds).

The GP implementation is in Matlab whereas the SSVI and NN approaches are implemented in Python. On our (large) dataset, the constrained GP has the longest training time. Training is longer for constrained SSVI than for unconstrained SSVI because of the ensuing amendments to the optimization routine. There are no arbitrage violations observed for any of the constrained methods in neither the training or the testing grid. Unconstrained methods yield 18 violations with NN and 177 with SSVI on the testing set, out of a total of 1725 testing points, i.e. violations in 1.04% and 10.26% of the test nodes. The unconstrained GP approach yields constraint violations on 12.5% of the basis function nodes $\imath h$. The NN penalizations $(\mathrm{cal}_T)^-$ and $(\mathrm{butt}_k)^-$ vanish identically on the penalty grid Ω_h in the constrained case, whereas in the unconstrained case their averages across grid nodes in Ω_h are $(\mathrm{cal}_T)^- = 3.91 \times 10^{-6}$ and $(\mathrm{butt}_k)^- = 1.60 \times 10^{-2}$ with the IV based NN.

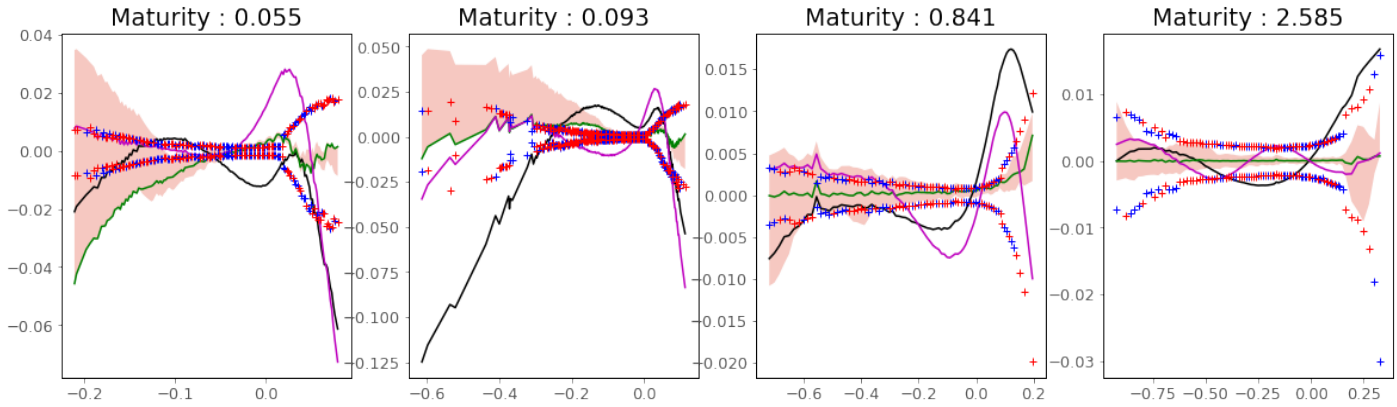
Fig. 17(a-b) respectively compare the fitted IV surfaces and their errors with respect to the market mid-implied volatilities, among the constrained methods. The surface is sliced at various maturities (more slices are available in the github) and the IVs corresponding to the bid-ask price quotes are also shown – the blue and red points respectively denote training and test observations.

We generally observe good correspondence between the models and that each curve typically falls within the bid-ask spread, except for the shortest maturity contracts where there is some departure from the bid-ask spreads for observations with the lowest log-moneyness values. We see on Fig. 17(b) that the GP IV errors are small and mostly less than 5 volatility points, whereas NN and SSVI exhibit IV error that may exceed 15 volatility points. The green line and the red shaded envelopes respectively denote the GP MAP estimates and the posterior uncertainty bands under 100 samples per observation. The support of the posterior GP process assessed on the basis of 100 simulated paths of the GP captures the majority of bid-ask quotes. The GP MAP estimate occasionally corresponds to the boundary of the support of the posterior simulation. This indicates that the posterior truncated Gaussian distribution is heavily skewed for some points, and that the MAP estimate consequently saturates the arbitrage constraints. This indicates a tension between these constraints and the calibration requirement, which cannot be fully reconciled, most likely because some of the (short maturity) data are arbitrable (they are at least illiquid and hence noisy). See notebook for location of arbitrages in the unconstrained approach.

Fig. 17(a-b) suggest that the data may exhibit arbitrage at the lowest maturities where the methods depart from the bid-ask spreads. This is further supported in Fig. 18(a-b) which shows the corresponding methods without the no-arbitrage constraints. In Fig. 18(a-b) we observe that the estimated IVs now fall within close proximity of the bid-ask spreads–all methods exhibit an error typically less than 5 volatility points. Note that the y-axis has been scaled for each plot in Fig. 18(b) to accommodate the wide uncertainty band of the posterior for the unconstrained GP. Whereas the uncertainty band of the constrained GP spanned at most 10 volatility points, the uncertainty band of the unconstrained GP is an order of magnitude larger, sometimes spanning more than 100 volatility points.

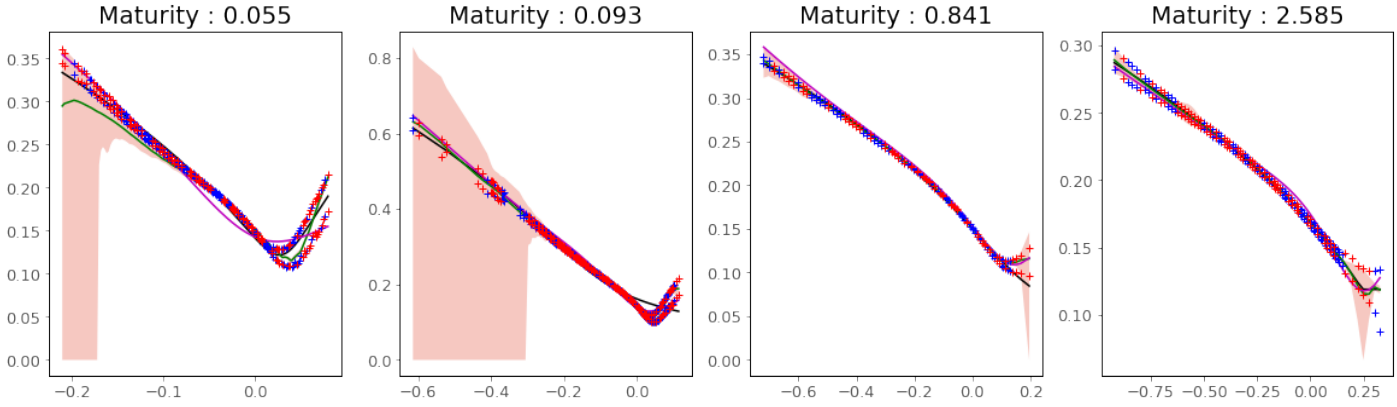


(a) Implied volatilities.

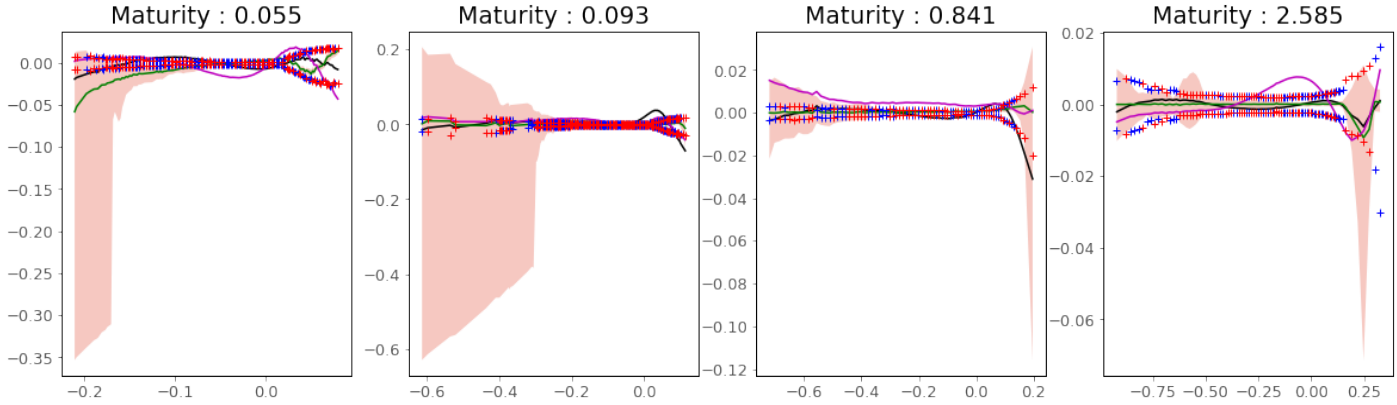


(b) Fitted IV errors with respect to mid-price IVs.

Figure 17: *Slices of constrained GP (green), NN (purple), and SSVI (black) models of SPX puts with training bid-asks IVs (+) and testing bid-asks IVs as a function of log forward moneyness (+)(the bid-ask IVs are reconstructed numerically from the corresponding bid-ask market prices). The shaded envelopes show 100 paths of the constrained GP's posterior.*

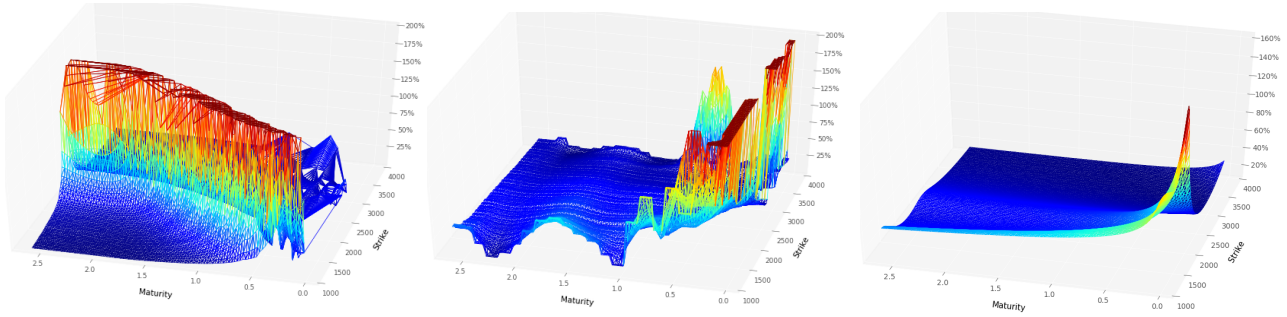


(a) Implied volatilities.



(b) Fitted IV errors with respect to mid-price IVs.

Figure 18: *Same as Figure 17 but for unconstrained GP, NN and SSVI.*



(a) The local volatility surface generated by SSVI with finite differences, capped at the 200% level.

(b) The MAP estimate of the GP local volatility surface, capped at the 200% level.

(c) The implied volatility based NN local volatility surface (with the local volatility penalization).

Figure 19: *The GP, SSVI, and NN local volatility estimate.*

Fig. 19 shows the local volatility surfaces that stem from the three constrained approaches. Fig. 19(a) shows the spiky local volatility surface generated by SSVI, capped at the 200% level for scaling convenience. Fig. 19(b) shows the capped local volatility surface constructed from the GP MAP price estimate. Fig. 19(c) shows the (complete) NN local volatility surface.

E.4 In-sample and out-of-sample calibration errors

The error between the prices of the calibrated models and the market data are evaluated on both the training and the out-of-sample data set. The first two rows of Table 4 compare the in-sample and out-of-sample RMSEs of the prices and implied volatilities across the different approaches. The differences between the training and testing RMSEs are small, suggesting that all approaches are not over-fitting the training set. The GP exhibits the lowest price RMSEs.

E.5 Backtesting results

The first repricing backtest estimates the prices of the European options corresponding to the testing set, by Monte Carlo sampling in each calibrated local volatility model (same methodology as in (Chataigner et al., 2020, Section 7.2)). The second approach uses finite differences to price the options with the calibrated local volatility surfaces. The pricing PDEs with local volatility are discretized using a Crank-Nicolson (CN) scheme implemented on a 100×100 backtesting grid. The last two rows in Table 4 compare the resulting price backtest RMSEs across the different approaches. The NN fitted to implied volatilities exhibit significantly lower errors in the backtests, followed by NN based on prices, SSVI and GP. To quantify discretization error in these backtesting results (as opposed to the part of the error stemming from a wrong local volatility), we ran the same backtests in a Black-Scholes model with 20% volatility and the associated prices. The corresponding Monte Carlo and Crank-Nicholson backtesting IV(price) RMSEs are 2.90%(1.56) and 0.846%(4.10), confirming the significance of the above results.

Conclusion We approach the option quote fitting problem from two perspectives: (i) the GP approach assumes noisy data and hence the existence of a latent function. The mid-prices are not considered, rather the GP calibrates to bid-ask quotes; and (ii) the NN and SSVI approaches fit to the mid-prices under a noise-free assumption. While these two approaches are important to distinguish on theoretical grounds, in practice there are other factors which are more important for, in particular, local volatility modeling. In line with classical inverse problems theory, we find that regularization of the local volatility is critical for backtesting performance.

Chapter VIII

Financial Nowcasting

In this chapter, we explore other possible applications of machine learning in finance, complementing the ones of Chapter VI in that they involve time series of historical data, as opposed to simulated data before. This raises new and arduous challenges: non-stationarity of financial data, high dimensionality, data size (sometimes limited) and missing data, issues related to extremes and dependence (thinking of, in particular, risk measures), frequent absence of labels.

Specifically, we devise a neural network based compression/completion methodology for financial nowcasting. The latter is meant in a broad sense encompassing completion of gridded values, interpolation, or outlier detection, in the context of financial time series of curves or surfaces (also applicable in higher dimensions, at least in theory). We will see (and this is meant as a general message) that in finance it is very hard, even if sometimes possible, to beat the well established traditional techniques—in the statistical context of this section: PCA, whenever applicable, but we already saw in the numerical probability setup of §3 that the performance of the best machine learning technology is bounded by the quality and quantity of the data that you are feeding it with.

Any notation of the form $\min_x \Lambda(x, y)$ means that we minimize in x a loss Λ given the value y of additional parameters; $x^\star$ then refers to a numerical minimizer of $\Lambda(x, y)$ (which is typically nonconvex in x), for this given y.

§1 Problems

We consider a data set consisting of a time series of observations, each consisting of m points, or features, structured as a multivariate tensor. By the latter, we mean a discretized tensor of values of homogenous quantities, such as rates of different terms, implied volatilities of different strikes and maturities, etc., defined at each tensor grid node.

A Compression

The compression problem is mainly a pre-processing stage that aims at reducing the dimensionality m of a feature space, i.e. the number of grid nodes in each tensor (here assumed constant across observations ω, see C.C regarding the variant of the functional approach with a possibly variable m_ω). Assume that each observation takes its values in (a subset of) $\mathbb{R}^m$. We call encoder E any injective map from a relevant subset $\mathcal{S}$ of $\mathbb{R}^m$ to a space $\mathbb{R}^f$ of factors, where $f \ll m$ is the number of factors. Conversely, one would like to be able to reconstruct the m values of a tensor from any set of factors, or code, thanks to a map, called decoder, $D : \mathbb{R}^f \to \mathcal{S}$. The compression challenge is to build D and E such that $D \circ E : \mathcal{S} \to \mathcal{S}$ is bijective and "as close as possible to identity" (cf. Bengio, Goodfellow, and Courville (2017, Chapter 14)).

The inspection of common financial time series of tensors suggests that, in their case, this challenge is somehow not unreasonable. Indeed, structural constraints often exist between the values at different

tensor nodes, e.g. arbitrage pricing relationships throughout the option chain. Moreover, usual financial tensors exhibit some spatial regularity, in the sense that values at grid nodes vary smoothly with respect to node location (think of interest rates with respect to their term or implied volatilities with respect to the maturity and strike of an option). In addition, some coordinates may have a regularizing effect. For instance, in the region of large expiries, the at-the-money swaption implied volatility surface is mostly affected by translation moves (and not so much by steepening, etc.) as time passes. (see Trolle and Schwartz (2010)). Last, some (monotonicity, convexity,...) patterns are often apparent (e.g. the well-known volatility smile in equity derivative, and some similar features in interest rate swaption implied volatility surfaces, cf. Figure 11).

Both maps E and D are sought within classes of neural networks with respective parameters ε and δ, collectively denoted by θ. The motivation for using neural networks in this context is their nonparametric (or, at least, very expressive) and nonlinear features. Gaussian processes for instance would be much less flexible, with only a few, e.g. two, kernel hyperparameters for squared exponential kernel to calibrate a full data set of thousands of tensors.

We include into θ weights, biases, as well as any variable calibrated during the compression stage. Denoting $E = E_\varepsilon$ and $D = D_\delta$ in reference to this parameterization, the compression stage is the training of the neural networks according to the following optimization problem:

$$\min_{\vartheta=(\delta,\varepsilon)} \sum_{\omega\in\Omega} \sum_{(n,y)\in\omega} \Big(y - \Big(D_\delta\big(E_\varepsilon(\omega)\big)\Big)_n\Big)^2, \tag{1}$$

where Ω stands for the training data set.

Certain additional properties are desirable for D and E. The parameterization θ should allow for a robust and fast numerical solution to the problem (1). This may be harder to achieve for some deep neural networks too sensitive to the initialization of their parameters. In particular, two similar tensors should give rise to similar codes and vice versa, i.e. we want D and E to be "sufficiently smooth" in such way as to preserve distance in the subspace.

B Completion

Having found a parametrization $\theta^\star = (\delta^\star, \varepsilon^\star)$ that ensures a satisfying reconstruction loss in (1), the completion task consists in the exploitation of $D_{\delta^\star}$ in order to find the missing values of an incomplete observation ω (of the current day, say, to be completed based on the complete observations of the previous days, used as training set).

Toward this end, we introduce the following optimization problem:

$$\min_c \sum_{(n,y)\in\omega} \Big(y - \big(D_\delta(c)\big)_n\Big)^2, \tag{2}$$

considered for $\delta = \delta^\star$. The completed tensor is then defined as the image $D_{\delta^\star}(c^\star)$ of the code $c^\star$ by the decoder $D_{\delta^\star}$. Obviously, the more missing values, the harder the completion task (higher overfitting risk, unless some appropriate regularization is used).

Note that, thanks to the compression step, the number of variables to estimate is drastically reduced in (2), to some reference number, i.e. the dimensionality of c (e.g. 4, 15, or 8 in our repo, equity index derivative and interest-rate swaption case studies), independent of the number of unknowns in the native, "uncompressed" completion problem (such as the number of missing implied volatility values in a to-be-completed surface). Moreover, a factorial representation with $f \ll m$ filters out the unlikely tensors (as outlined by the reconstruction error from our neural networks, cf. C) that could otherwise arise from a decoding due to the ill-posedness of large-scale arg-minimization problems. The regularity of the map $D_{\delta^\star}$ can sometimes be exploited to ease the completion, by initializing the numerical solution of (2) with the encoding of the last fully observed (e.g. already completed) tensor.

Literature Review The literature on completion primarily deals with data structured on a fixed grid. This means that columns in the data set refer to the same feature (in our case: financial instrument). This is not consistent with most financial nowcasting applications, for which, in particular, the time-to-maturity decreases with calendar time. Only naive interpolation methods on a given tensor, without possible exploitation of a data set, are available in the case of a moving grid.

The standard completion framework relies on a low rank representation of the data set (see Nguyen, Kim, and Shim (2019)). Along this line but, via the functional approach of C.C, on a possibly moving grid, we compress each observation in a code which can be seen as a latent vector. However, in contrast with methods such as SVD, alternating least squares (see Hastie, Mazumder, Lee, and Zadeh (2015)), or denoising autoencoders (see Strub and Mary (2015)), which learn a user matrix, we do not consider the interaction between the observations (i.e. the dynamics): we focus on the interaction between the variables (instruments).

Finally, standard completion methods in recommender systems assume missing completely at random (MCAR) values dispersed throughout the whole data set. In our case studies missing values are located completely at random but only for the current observation.

C Outlier Detection

Hawkins (1980) defines outliers as "observation which deviates so much from other observations as to arouse suspicion it was generated by a different mechanism". Outlier detection is of course a crucial issue in finance. For instance, investment banks receive market information from a data provider. Sometimes, the data can be polluted with errors of various sources, or "different mechanism", whether it is data feed bugs, fat finger of other market participants, or failure from computation processes (for instance, implicitation of volatility surface from option prices). It can be either a punctual outlier, i.e. a single value of the tensor is too far away from what it should be, or the whole tensor may have a shape that is very unlikely.

To detect the punctual outliers, many simple methods are available, based on smoothness metrics or on historical percentile ranges of the values. To detect shape outliers, some criteria can be checked for very specific data sets, e.g. non-arbitrage butterfly/calendar spread conditions in the case of option prices.

Here we propose a general method to detect both punctual outliers and shape aberration. The functional variant of our method works on an unstructured grid and is amenable to high dimension.

To say that the tensors generated from the "normal mechanism" is of a certain form is equivalent to say that the mechanism generates values that lie in a sub-manifold $\mathcal{S}$ of the initial feature space (cf. A). Finding this sub-manifold is equivalent to detecting anomalies. From this point of view, anomaly detection and compression/decompression are two sides of the same coin. Indeed, from an information theory point of view, there is an equivalence between being an anomaly and being hard to reconstruct (large reconstruction error in a lossy data compression setup, low or even negative compression rate in a lossless data compression setup): See the seminal paper by Shannon (1948) or Chapter 4 in MacKay and Mac Kay (2003). That is, a compression/decompression setup provides a natural anomaly detection tool.

Specifically, we identify an outlier as an observation whose reconstruction error (cf. (1)) is above a predefined threshold.

Some key practical questions in outlier detection are how a threshold for outlier detection should be chosen or how one can validate the method. In principle this can only be addressed by human expertise. An expert would gradually diminish the threshold until the newly detected "outliers" are no longer considered such by the expert. The method is valid and performs well if the outlier detection in a validation set is consistent with the expert view (so that, in particular, the threshold is stable through time and does not need to be reassessed too frequently).

However, our compression methodology also provides a validation tool for the quality of our outlier detection method. Namely, one can corrupt some of the data (manually or in an automated fashion) and check whether the outlier detection procedure identifies the corrupted data.

Our approach also provides guidance to a human expert for anomaly correction. Currently experts only rely on naive heuristics, such as interpolation between different points of a surface, who cannot automatically exploit the overall data set of surfaces. In the outlier detection validation framework of the previous paragraph, one can also check whether the correction that our approach provides is closer to the true data than to the corrupted ones.

Literature Review Among many related references on outlier detection:

- Patcha and Park (2007), Chandola, Banerjee, and Kumar (2009), Omar, Ngadi, and Jebur (2013), or Anandakrishnan, Kumar, Statnikov, Faruquie, and Xu (2018) provide surveys, the last one specialized in finance and the next-to-last one on machine learning techniques;
- Lakhina, Joseph, and Verma (2010) use PCA, An and Cho (2015) variational autoencoders, Schlegl, Seeböck, Waldstein, Langs, and Schmidt-Erfurth (2019) generative adversarial networks, Lakhina, Joseph, and Verma (2010) and Cappozzo, Greselin, and Murphy (2020) semi-supervised learning. Chaloner and Brant (1988) and Cansado and Soto (2008) resort to Bayesian methodologies;
- Ro, Zou, Wang, and Yin (2015) is about high-dimensional data, Anandakrishnan, Kumar, Statnikov, Faruquie, and Xu (2018) about high dimensional big data, Rocke and Woodruff (1996) about multivariate data, Goix, Sabourin, and Clémençon (2017) and Goix, Sabourin, and Clémençon (2015) about detection of anomalies among extremes.

§2 Models

A The Convolutional (Autoencoder) Approach

Typical autoencoder architectures are composed of two successive feedforward neural networks E and D, the encoder and the decoder. Both networks can be constituted of several layers, intermediated by nonlinear activation functions, with an overall bottleneck structure (to enforce compression in the middle).

Convolutional layers have been introduced for image processing and, more generally, any data structure represented as a tensor. These networks aim to model the interactions between close points (whereas dense layers bind any output unit to all input units). Spatial regularity properties are handled by a convolutional structure of the neural network architectures, whereby the only (non-zero) connections are between units corresponding to adjacent (in a suitable sense) grid nodes (cf. Figure 13). The network then also uses fewer parameters, which reduces the complexity of the corresponding compression problem. For implementation details such as kernels and padding, we refer to Chapter 9 in Bengio, Goodfellow, and Courville (2017).

B The Linear Projection Approach

It is well known that an autoencoder with linear activation functions and an L_2 reconstruction error is equivalent to a PCA (see Chapter 14 in Bengio, Goodfellow, and Courville (2017)). As a limiting case of the above, we consider a linear, PCA kind of benchmark, but one itself implemented as an autoencoder with linear activation functions (as opposed to spectral decomposition for classical PCA implementation). With respect to classical PCA (which will also be included in our case studies), this approach involves an additional bias parameter. Moreover, it allows benefiting from the implicit

regularization provided by early stopping in the related training procedure, as opposed to a regularization provided by truncation of the lowest eigenvalues in spectral decomposition based PCA implementation.

C The Functional Approach

We introduce a variant of the above, especially suited to interpolation purposes (without reference to a fixed grid of nodes). This approach relies on a parameterized function $D = D_\delta(c, n)$ of a code c and a node location n, where the latter no longer needs belong to a pre-determined grid. Here δ corresponds to the parameters of the decoder D, whereas the approach does not entail any encoder (at least, not explicitly).

The compression is written as (compare with (1), using a similar notation as well as $C = (C_\omega)_{\omega\in\Omega}$)

$$\min_{\delta,C} \sum_{\omega\in\Omega} \sum_{(n,y)\in\omega} \Big(y - D_\delta\big(C_\omega, n\big)\Big)^2. \tag{3}$$

Then, given a single, possibly partial observation ω, the completion is given as (similar to (2))

$$\min_{c} \sum_{(n,y)\in\omega} \Big(y - D_\delta\big(c, n\big)\Big)^2, \tag{4}$$

considered for $\omega = \omega^\star$ and $\delta = \delta^\star$. Importantly, for each given δ, the minimization (3) decouples into one (full observation) minimization (4) for each $\omega \in \Omega$. Hence, the larger compression problem (3) can be solved numerically as a succession of smaller problems (4), in conjunction with gradient iterations in the direction of δ. This ensures the scalability of the approach. It also makes it amenable to online learning. The above observation also shows the consistency between (3) and (4) in the sense that, if a full observation ω is used in (4), it should yield $c^\star = C^\star_\omega$ (assuming global and unique minima to all problems for the sake of the argument).

Under this approach, dubbed functional, the decoder takes as input the location n of the point, in addition to the factors c (see Figures 1, 3 and 12). It rebuilds each point individually, as per $n \to D_\delta(c, n)$. The network is thus able to interpolate between the nodes of the data grid. The concept of neighborhood intervenes through the argument n of D, but the parameterization δ as well as the code c are common to all locations n. The compression (3) can also accommodate incomplete data or discretization changes, i.e. varying grids in the training data. This feature allows training the functional network with "missing completely at random data" (MCAR, in the statistical missing data terminology).

By comparison, under the convolutional approach of A, the concept of neighborhood intervenes through $\vartheta = (\delta, \varepsilon)$, since each point of the grid is only sensitive to a subset of connections (the convolutional architecture only connects neighbouring points, cf. Figure 13). The encoding c is obtained directly thanks to E, when the observation is complete, or by numerical completion (as always under the functional approach) otherwise.

D Synthesis

To conclude this section, Tables 1 and 2 summarize and put into perspective the different approaches referred to in the above.

Also note that, from a numerical complexity point of view, the functional approach is less sensitive to the dimension than, say, a classical autoencoder on a fixed grid (including our convolutional approach), for which the size of the grid typically grows exponentially with the dimension.

§3 Experimental Methodology and Setting

In this section, we devise an experimental methodology and the learning procedures, so that all

Encoder	Implicit and non-linear $\widehat{c} = \underset{c}{\text{Argmin}} \sum_{(n,y)\in\omega} \Big(y - D_\delta\big(c,n\big)\Big)^2$
Decoder	Analytic and non-linear $\widehat{y} = D_\delta\big(c,n\big)$
Compression (training) step	Optimization w.r.t. (δ, c) $\min_{\delta,C} \sum_{\omega\in\Omega} \sum_{(n,y)\in\omega} \Big(y - D_\delta\big(C_\omega,n\big)\Big)^2$
Reconstructed surface Reconstruction	Implicit $\widehat{y} = D\left(\underset{c}{\text{Argmin}} \sum_{(n,y)\in\omega} \Big(y - D_\delta\big(c,n\big)\Big)^2, n\right)$
Completed surface	$\widehat{y} = D\left(\underset{c}{\text{Argmin}} \sum_{(n,y)\in\omega} \Big(y - D_\delta\big(c,n\big)\Big)^2, n\right)$

Table 1: The functional approach.

	PCA	Convolutional
Encoder	Analytic and Linear $\widehat{c} = E_\varepsilon(y)$	Analytic and non-linear $\widehat{c} = E_\varepsilon(y)$
Decoder	Analytic and linear $\widehat{y} = D_\delta\big(c\big)$	Analytic and non-linear $\widehat{y} = D_\delta\big(c\big)$
Compression (training) step	Optimization w.r.t. (δ, ε) $\min_{\vartheta=(\delta,\varepsilon)} \sum_{\omega\in\Omega} \sum_{(n,y)\in\omega} \Big(y - \Big(D_\delta\big(E_\varepsilon(\omega)\big)\Big)_n\Big)^2$	Optimization w.r.t. (δ, ε) $\min_{\vartheta=(\delta,\varepsilon)} \sum_{\omega\in\Omega} \sum_{(n,y)\in\omega} \Big(y - \Big(D_\delta\big(E_\varepsilon(\omega)\big)\Big)_n\Big)^2$
Reconstructed surface Reconstruction	Explicit/analytic $\widehat{y} = D\left(E(y)\right)$	Explicit/analytic $\widehat{y} = D\left(E(y)\right)$
Completed surface	$\widehat{y} = D\left(\underset{c}{\text{Argmin}} \sum_{(n,y)\in\omega} \Big(y - D_\delta\big(c\big)\Big)^2\right)$	$\widehat{y} = D\left(\underset{c}{\text{Argmin}} \sum_{(n,y)\in\omega} \Big(y - D_\delta\big(c\big)\Big)^2\right)$

Table 2: PCA and convolutional approaches.

models are set on comparable grounds.

All the optimization (compression or completion) problems are solved with the Adam adaptive learning rate stochastic gradient algorithms of Kingma and Ba (2014). The output of a neural network is by construction non-convex with respect to its parameters. So are therefore all our loss functions. The Adam algorithm has proven its robustness in non-convex optimization context. With the help of automatic adjoint differentiation, it provides fast training for most neural networks architectures. However, no convergence is guaranteed theoretically.

For the compression stage, we make a 80 : 20 split of a full data set into a training set and a test set. The split is chronological in order to avoid look-ahead bias (cf. Ruf and Wang (2020)). The training set is further split into a calibration and a validation data set. The former is used for computing the gradients driving the numerical optimization in the training problem, whereas the latter is used for determining an early stopping rule that provides implicit regularization, as detailed below.

The learning rate of the Adam optimizer is set to 0.001. Mini-batch learning is used in the repo and equity index derivative case studies, whereas batch-learning is employed with swaption volatilities. The gradient descent is driven by the loss computed on the calibration set, but the validation error is the loss function computed on the validation data set. The learning procedure is stopped when we do not observe any decrease of the validation error during a certain number of iterations, called patience. The parametrization returned by the compression is the one that minimizes the validation error. Early stopping in this sense limits the generalization error (cf. Engl, Hanke, and Neubauer (1996a)), i.e. the gap between the reconstruction errors computed on the calibration data set and a new, unobserved data set, the role of which is played by the test set. Sometimes, as detailed later, a penalization term is added to the compression loss function in order to provide a more regular and stable minimization. A maximum number of iterations is fixed to 10^4 at compression stage and 10^3 at completion stage, in order to cap the length of the optimizations.

All approaches are implemented in python, using the tensorflow package in the swaption case

study and pytorch in the two others. Note that all hyperparameters are chosen manually, rather than by grid search or random search techniques. Grid search is not possible because we have too many hyperparameters. Exploring different neural net architectures would be too demanding computationally. However, some of the hyperparameters can be fixed based on human expertise. For instance, 15 factors in our case study of B is the number of factors that equity derivative traders commonly use in PCAs (after interpolation on a fixed grid, as they are faced with moving grids).

A Performance Metrics

We want to assess, for each approach, the performance of the corresponding compression and completion procedures, as well as the behavior (distribution and dynamics) of the resulting factors. For the compression, we consider the average root mean square reconstruction error RMSE_ω on the test set Ω', where

$$\mathrm{RMSE}_\omega \tag{5}$$

(or the analogous quantity with m_ω and $D_{\delta^\star}\left(C^\star_\omega, n\right)$ instead of m and $\left(D_{\delta^\star}\left(E_{\varepsilon^\star}(\omega)\right)\right)_n$, as relevant), i.e. RMSE_ω is the root mean square error between the values at the nodes of the tensor ω and their reconstructed counterparts. In the case of the functional approach the encoder E is implicit and its definition is detailed in table 1. We refer to (5) as the reconstruction loss in the compression stage of our case studies given that ω is a complete surface.

In constrast with (5),

$$\sqrt{\frac{1}{m}\sum_{(n,y)\in\omega}\left(y-\left(D_{\delta^\star}\left(\widehat{c}\right)\right)_n\right)^2} \tag{6}$$

(or m_ω rather than m in the case of the functional approach) is called the completion loss when we compare the complete original observation with the completed observation. This completed observation is given by the decoder for code values $\widehat{c}$ which are calibrated on the incomplete view provided by ω.

In the case of interpolation benchmarks, there is no compression stage and no code is involved at the completion stage: the completion loss is then defined by the RMSE between the interpolated surface (from an incomplete ω) and the original complete ω.

We provide a focus on the observation ω leading to the worst RMSE_ω over the test set, in order to identify the locations that are less well handled (e.g. short option maturities). In addition, we display the time series of the codes. A good compression should exploit each factor in the code (we should not observe factors stuck at zero).

The quality of the completion is assessed by a backtest on the test set. Each day of Ω', we solve the problem (2) or (4), initialiazing the factors with the fully informed encoding of the previous day. We then mask 90 % of the points in each tensor of the test set. For each such observation $\omega \in \Omega'$, we check the reconstruction RMSE_ω between the completed surface and the true one. Like for compression, we plot the worst completion obtained on the test set Ω'.

B Introduction to the Case Studies

We provide numerical results on three daily time series of real financial data: repurchase agreement yield rates, equity implied volatility surfaces and at-the-money swaption implied volatilities. However, the swaption implied volatilities have been preprocessed by our data provider to fit a fixed grid (whereas the native, raw data had a moving time-to-expiry). A preprocessing entails an unquantifiable bias and our recommendation would be to apply the functional approach to the original data (whenever available). The main motivation for the third example is that one can then benchmark the functional approach against PCA and the convolutional approach.

The advantage of working with yield rates or implied volatilities, instead of the corresponding option prices, is that these are scaled quantities, exempt from first order dependence on contract characteristics

such as nominal, time-to-maturity, actual level of the underlying in at-the-money option data, etc., which should otherwise be added to the set of explanatory variables in all learning procedures. The ensuing arbitrage issue is discussed in the next subsection.

C Discussion of the Arbitrage Issue

Arbitrage constraints can be expressed naturally in terms of options prices using calendar spread and butterfly. But in terms of implied volatility, they are non-trivial, even in the simplest case of equity derivatives (for which they are fully stated in Roper (2010)). No compression/completion method applied to implied volatility surfaces provides a way to deal with those constraints without coming back inherently to option prices. In order to circumvent that problem, one could apply our approach to the coefficients of a (e.g. local vol) model, from which non arbitrable prices and implied volatilities could be derived in a second step. However, we do not choose this route because:

- the market practitioners, who play both the roles of human experts and users, have built intuitions over decades on implied volatilities. They think of option prices directly in terms of implied volatilities. Providing them with a good recommendation tool in terms of a quantity that is familiar to them is of great value and the primary purpose of our approach;

- most of the times, the starting point for calibrating a model (e.g. Dupire) is nothing else than the implied volatilities. Therefore the trader must correct the anomalies *before* the implied volatility surface can be plugged as an input to model calibration. Hence one of the requirements of our proposed approach is that it should be model-free;

- Having said this, if one assumes that, on the one hand, most of the surfaces in our database are arbitrage-free and, on the other hand, a more regular surface is less prone to arbitrage opportunities, then one concludes that our model should tend to remove part of the arbitrages present in the data. This can actually be seen empirically on some of the examples in B. This is a natural by-product of anomaly correction and it also eases the calibration process.

Similar comments apply on most markets (beyond equity implied volatility), including the ones of our three case studies, i.e. repo contracts, handled by traders in terms of yield curves, and equity index derivatives and swaptions, which are handled in terms of implied volatility.

§4 Repo Curves

Our first case study bears on the nowcasting of repo rates, based on an 2013–2019 daily time series of repo yield curves (repo rates, where repo is a shorthand for repurchase agreement).

The grid of nodes in the data is unstructured, in the sense that the corresponding dates (time-to-maturities of bonds with standardized maturity dates) vary, in both number and location, from day to day (with as little as two or three points on particularly idle days), see e.g. Figure 2. Indeed, as the expiration dates used to compute the repo curve are fixed, and the variable of interest for the repo curve shape is rather time to expiry, the latter decreases as the expiry date approaches. For a given repo curve, the times to expiry for which the repo value is available is not known in advance for that reason. Therefore, there is no canonical way to have a systematic representation of repo curves on a fixed grid, one would need to introduce artificial time to expiry of interest and interpolate/extrapolate (which poses issues of its own) the repo curve to get the values, and then working on transformed data. This is the situation the functional approach is tailored for. By not making any assumptions on the domain of input (time to expiry), the functional approach enables to handle unaltered data, by treating

the time-to-maturity of a transaction as an input value (cf. Figure 1).

A Functional Network Architecture

Our functional approach is implemented by a single feed-forward neural network composed of three fully-connected layers with 20, 20 and 1 units (see Figure 1). Hyperbolic tangent activation is applied

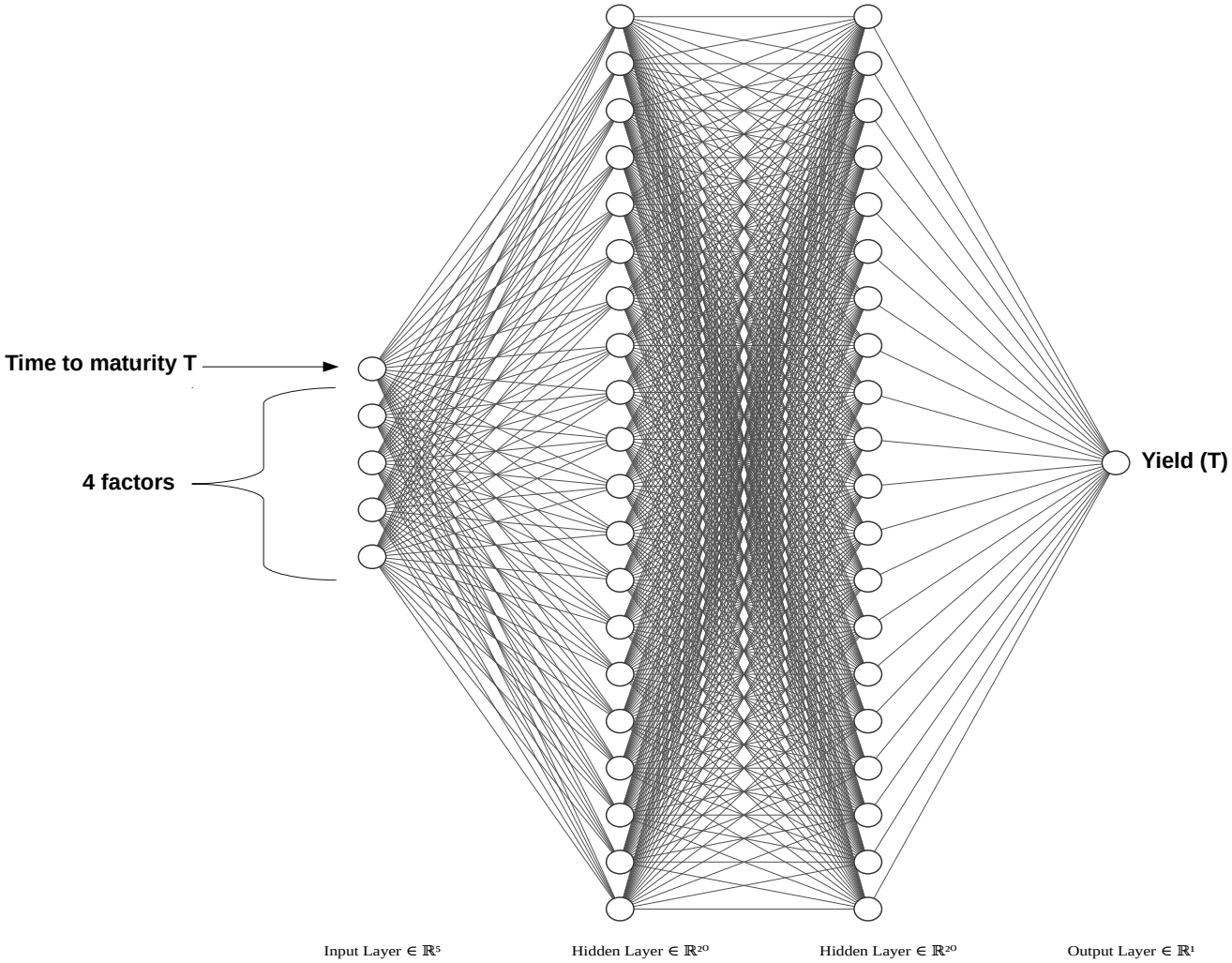


Figure 1: Network of the functional approach used in the repo case study. Here and in Figures 3 and 12 below, the graphs have been produced using the style FCNN of the NN-SVG software: the units and the connections between them are represented by circles and edges.

to each but the output layer for the same reasons as above (and the output layer is linear).

B Numerical Results

As the bottom panels of Figure 2 illustrate, the parameterization is flexible and can accomodate different curve shapes or node localizations.

As explained in §1.C, the compression stage can be used for detecting an abnormal curve and correcting it with a more likely one. The distinction between inliers and outliers is determined by a threshold on the reconstruction error. A bad reconstruction is taken as a signal that the codebook is not able to explain the corresponding observation. We then conclude that the latter does not lie in the manifold $\mathcal{S}$ of the "usual" curves, hence we classify it as an outlier (see §1.C). We can then correct (replace) these data by the curve reconstructed from the decoder with the factors calibrated on the current values, i.e. by the output of the corresponding completion (4).

The lower panels of Figure 2 show the gap between the observed data points and the reconstructed ones. The upper left panel spots the outliers at a 0.035 absolute RMSE threshold. The upper right panel gives an example of outlier correction.

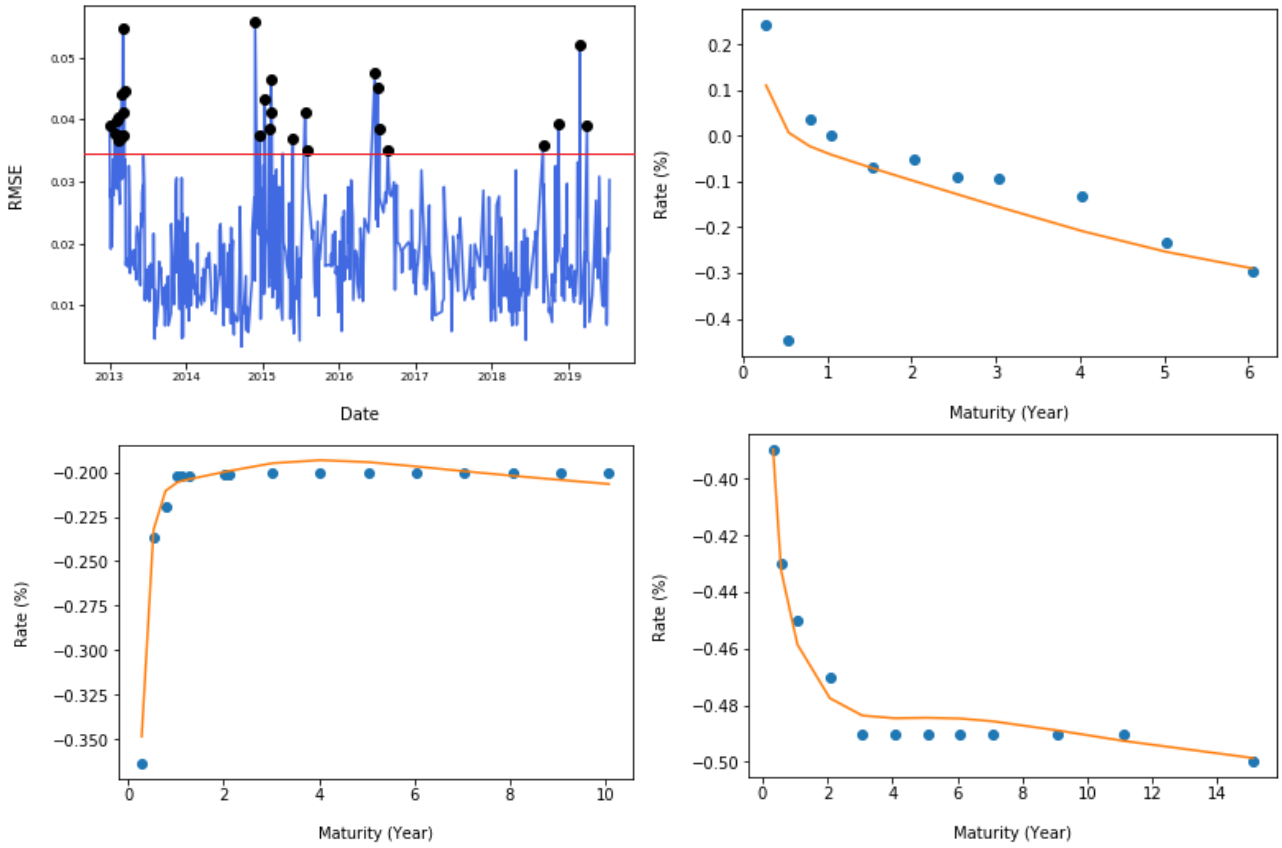


Figure 2: *(Bottom)* Interpolation of two inlier repo curves; *(Top left)* Time series of the (absolute) RMSEs on the repo data and 0.035 RMSE threshold; the spotted values correspond to the outliers at the chosen threshold. *(Top right)* Interpolation of an outlier repo curve.

§5 Equity Derivative Implied Volatility Surfaces

As a second experiment, we apply our functional approach to Black–Scholes implied volatilities surfaces of equity index derivatives. The corresponding volatilities price options on the Nikkei 225 index from 2015 to 2018 (included), corresponding to 1544 observable surfaces. The order of magnitude of implied volatilities fluctuates between 0.15 and 1.2. We include the forward rate as an exogenous variable that can be plugged into the functional network (1) along with log-maturity and log-moneyness.

As in the repo case study, the grid of nodes in the data is unstructured, in the sense that the corresponding dates (time-to-maturities of equity index options with standardized maturity dates) But, again, this is the situation the functional approach is tailored for (cf. Figure 1). The corresponding architecture of the functional approach is then similar to the one used for repo curves in the previous section, except that the log-time-to-maturity and the log-moneyness are used as the (two dimensional) localization inputs, and that 15 latent variables are used (instead of only 4 previously): see Figure 3. Moreover, one can also easily incorporate the forwards as exogenous variables. For taking them into account, it suffices to add to the network of Figure 3 an additional feature (input unit) containing the level of the forward swap rate with maturity T. Hence, the units for the maturity T indicate the common

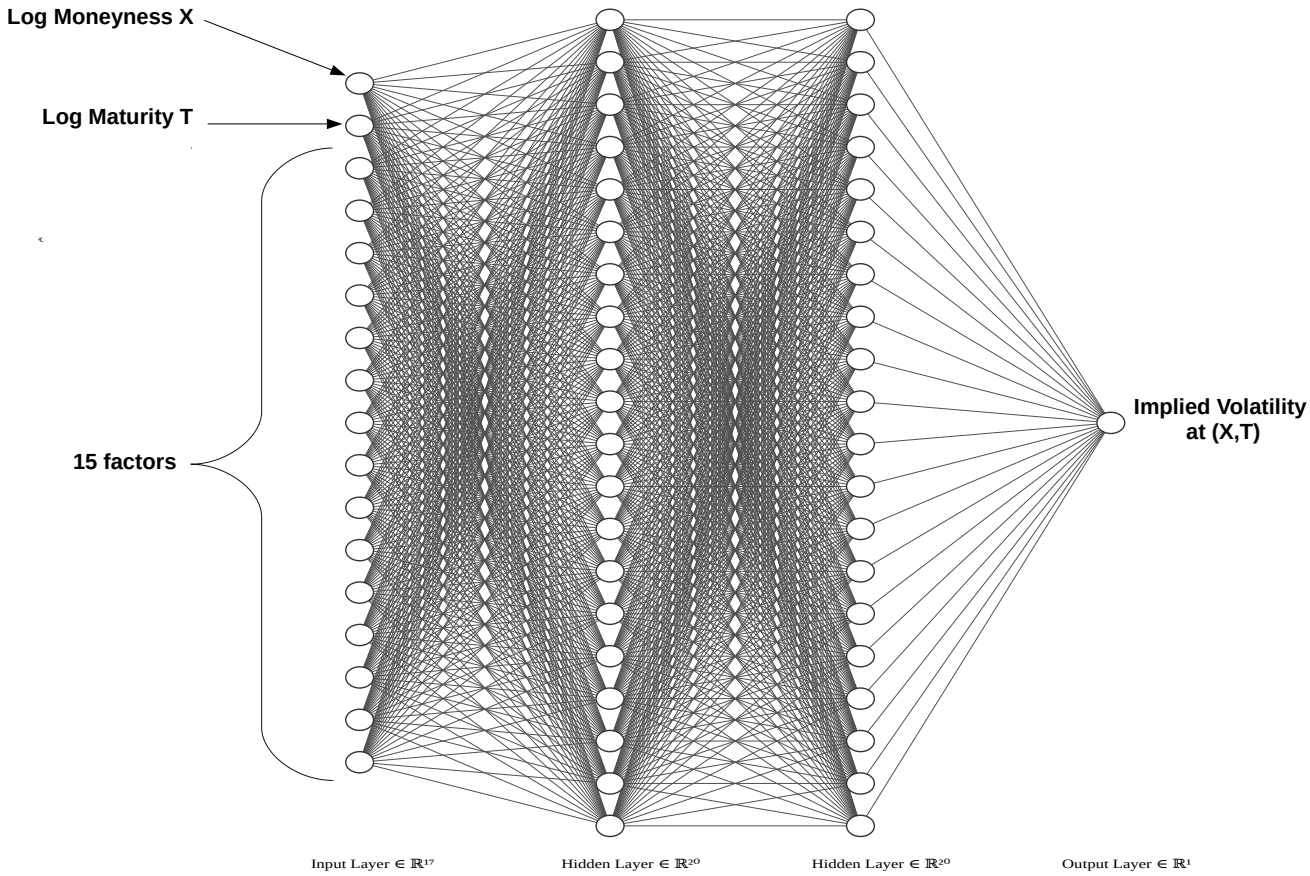


Figure 3: Network of the functional approach used in the equity case study (style FCNN of the NN-SVG software, cf. Figure 1).

location of the corresponding volatilities and forward rates.

A Compression

We first calibrate our functional approach with the compression stage. Toward this end, we execute the optimization (1) on the training set and then calibrate codes with (2) for each observation in both testing and training data sets. The quality of the compression is assessed through the reconstruction errors reported in Table 3. By reconstruction error we mean the gap between the original surface and the surface induced by the code calibrated from (2).

We emphasize the difference between a reconstucted surface (as above) and a completed surface (considered later): the code leading to the completed surface is calibrated from an incomplete surface whereas the one for the reconstructed surface is obtained from a complete real surface.

	Functional	Functional with Forward
Training set	0.0070	0.0063
Testing set	0.0058	0.0064

Table 3: RMSEs for reconstructed implied volatilities.

In all four cases, the RMSEs in Table 3 are very small compared to the order of magnitude of implied volatilities (between 0.15 and 1.2). The results show no sign of overfitting (the reconstructions error are similar on the training set and the testing set). Moreover the comparison between the two columns of the table indicates that there is no benefit in including the forward price as an exogenous variable in our network.

Another way to assess the performance of the compression stage is to consider the worst compression,

i.e. the surface yielding the highest reconstruction error. This worst reconstruction corresponds to a RMSE of 0.0096. It is represented in Figure 4, with the real surface on the top-left corner, the reconstructed couterpart on the top-right corner and the pointwise absolute difference between the two at the bottom.

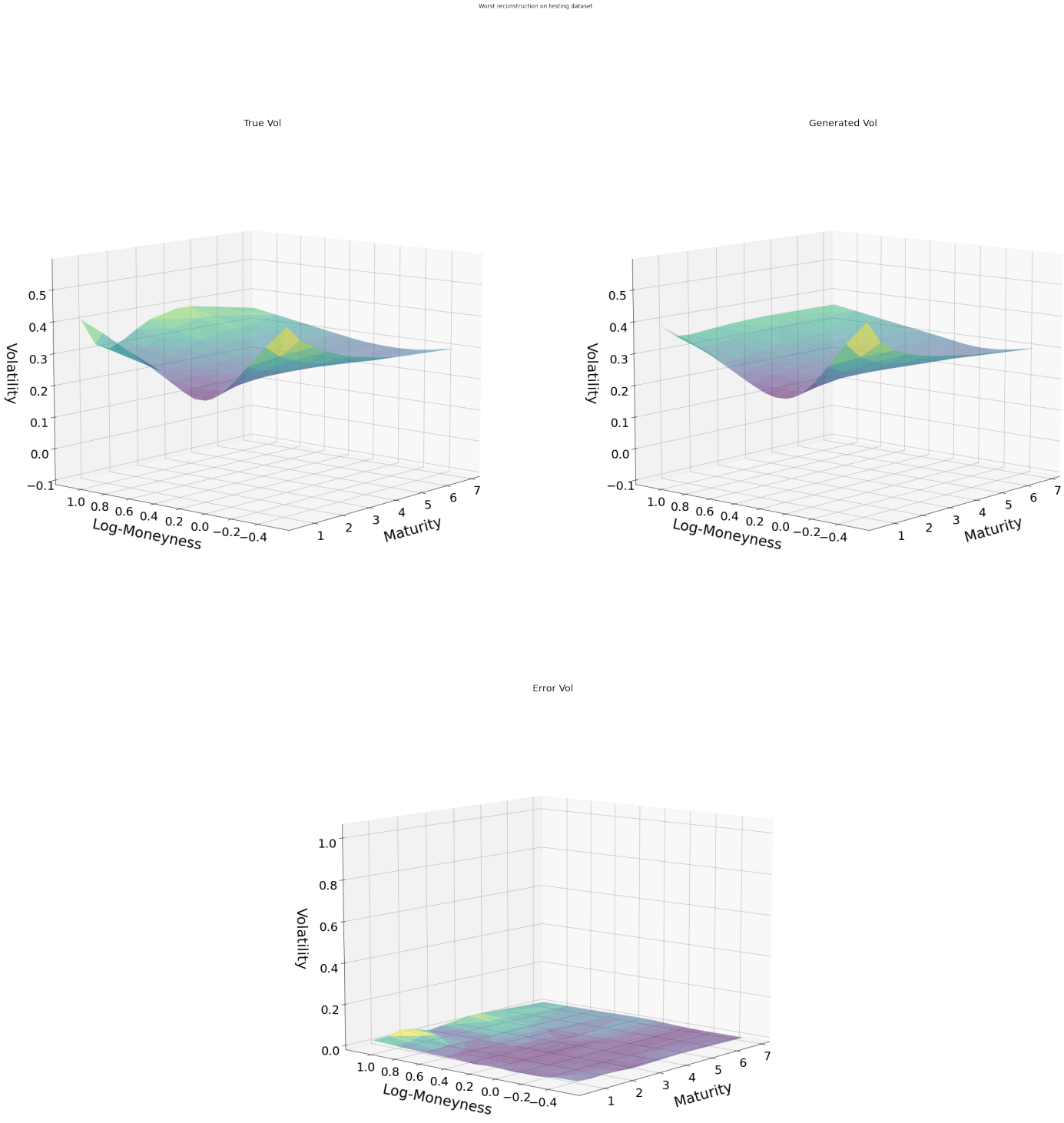


Figure 4: Original surface vs compressed surface yielding worst RMSE.

We notice that the errors are concentrated on the upper tail (deep in the money call options) and for short maturities, which corresponds to illiquid options.

A bad reconstruction of a surface can also be used for qualifying it as an outlier. For instance, Figure 5 shows the implied volatility values corresponding to the most extreme strikes in Figure 4: original data points as dots and curves from the reconstructed surface. The left panel corresponding to the illiquid upper tail shows around the maturity 1.5 year a very low point that an expert would indeed

qualify as an anomaly. The correction (i.e. the reconstructed surface) ignores this anomaly and has a more reasonable shape from a practitioner of view.

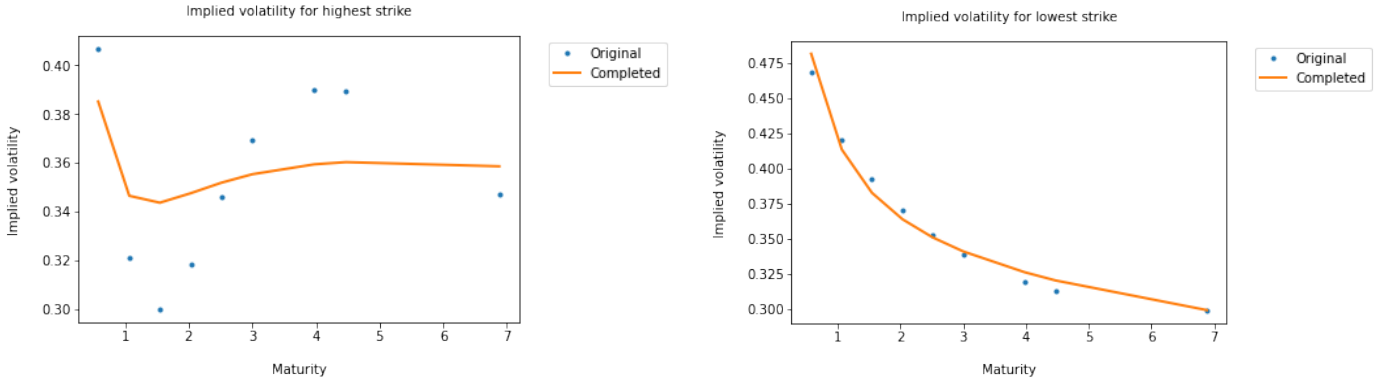


Figure 5: Tails of compressed surface vs original implied volatilities.

The left part of Figure 7 shows that the corrected surface is not prone to calendar arbitrage: the sensitivity to the maturity of the corresponding implied total variance is positive for every maturity T.[1] Sensitivity is computed thanks to adjoint automatic differentiation from neural network.

The above example shows that the functional neural network is indeed apt to learn from the compression stage a low-dimensional representation of likely observations. The low-dimensional representation gives large reconstruction errors to the surfaces of the testing set atypical with respect to the past observations (the training set in our experiments) and their latent structure.

B Outlier Detection and Correction

To confirm our views on outliers, we propose the following sanity check. An observation (first volatility surface in the testing set) is chosen and articially corrupted by doubling the values on four randomly chosen points: see the top-left corner in Figure 6.

Then we run the optimization (2) on this corrupted surface and obtain recalibrated codes. These code produce with the decoder the reconstructed surface (called correction) on the top-right corner. The correction is a smooth surface in which the corrupted values have been overwritten by values close to the original (non corrupted) ones. The bottom-left panel shows that only the corrupted values have been modifed significantly by the correction stage. The bottom-right figure indicates that the corrected surface is very close to the original one. The RMSE between the corrupted and the corrected surface is 0.0446 whereas the one between the correction and the original surface is 0.0151.

Note that the calendar arbitrage condition is still respected (see Figure 7) for the correction, which exhibits a positive sensitivity of the implied total variance with respect to the maturity of the option.

This experiment confirms that a high reconstruction error is a good indicator of an outlier. The calibrated latent structure of the functional network smoothes the corresponding surface by identifying and correcting its anomalous points.

C Completion

We now want to leverage on the calibrated low-dimensional latent structure of the functional network to recover a complete surface from partial information. Our hope is that this procedure will generate likely surfaces while approaching the available values (including on moving grids).

[1] Regarding butterfly arbitrages, Durrleman's condition on the density (involving sensitivity with respect to forward log-moneyness, cf. Roper (2010)) can unfortunately not be checked for lack of data regarding dividends and discounting.

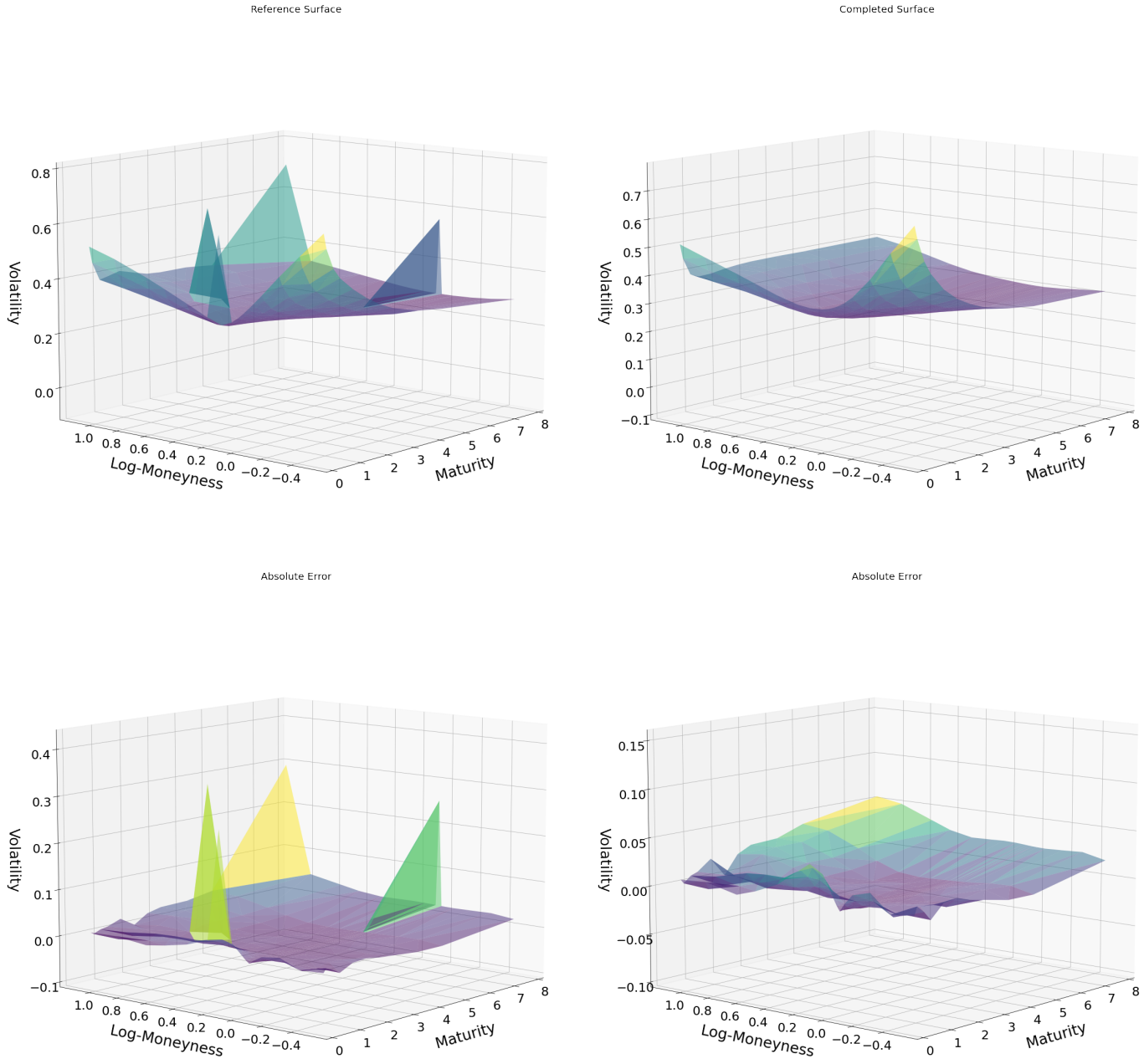


Figure 6: Outlier correction : Corrupted surface (Top-left), Corrected surface (Top-right), absolute error between corruption and correction (bottom-left), absolute error between correction and original surface before corruption (bottom-right)

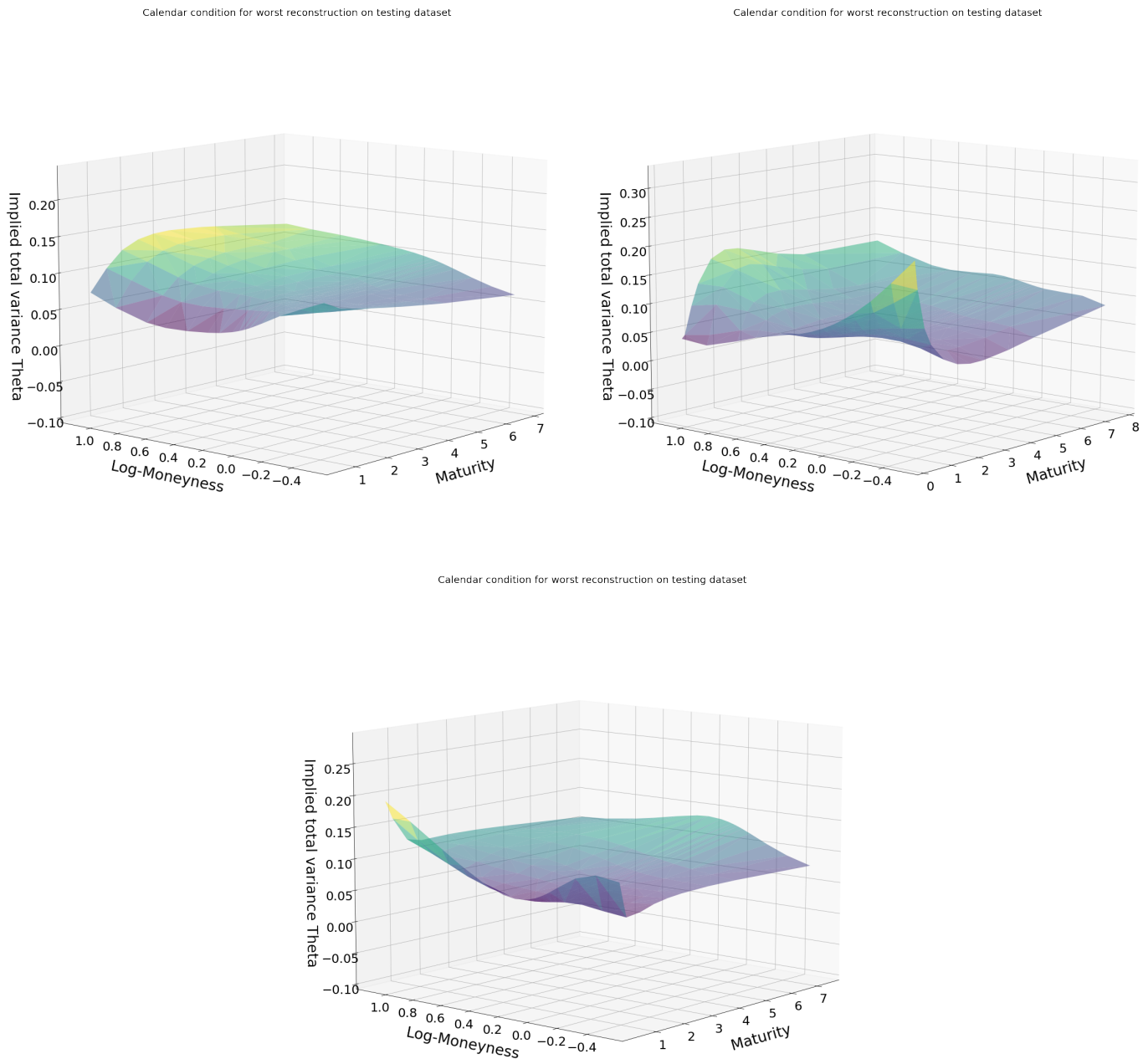


Figure 7: Implied total variance theta for worst reconstruction on top-left, outlier correction on top-right and worst completion at the bottom.

For each observation (surface) in the testing set, we select 40 points among the 255 points and remove all the others. Then we calibrate the latent variables by solving numerically the problem (2) with loss corresponding to these 40 points.

In order to benchmark the functional approach and assess the contribution of the historical data to the performance of the method, we report average completion errors[2] on the testing set for standard interpolation procedures (within each given surface, without exploitation of the information provided by the others):

1. Linear interpolation: given a triangulation of the 2D maturity and log-moneyness space base on the locations of the 40 available points, the interpolated value is taken as the barycenter on each triangle;

2. Spline interpolation: uses in each triangle as above a piecewise cubic interpolating Bezier polynomial (see Alfeld (1984) and the scipy documentation of the CloughTocher2DInterpolator method);

3. Gaussian process regression and squared exponential kernel: denoting by X the observed locations (maturity and log-moneyness), by Y the observed lognormal volatilities at locations X, by $X^\star$ the locations without values and by $Y^\star$ the unknown (looked for) implied volatilities, a Gaussian process regression assumes a Gaussian distribution

$$(Y, Y^\star) \sim \mathcal{N}(0, \begin{pmatrix} K(X,X) & K(X,X^\star) \\ K(X^\star,X) & K(X^\star,X^\star) \end{pmatrix})$$

with

$$K(X,X^\star)_{ij} = \sigma \exp\left(-\frac{\|x_i - x_j\|^2}{l^2}\right), \tag{7}$$

where σ and l are two hyperparameters calibrated by log-likelihood to the available values. In (7),

$$\|x_i - x_j\|^2 = (T_i - T_j)^2 + (\ln(m_i) - \ln(m_j))^2,$$

where T denotes a maturity and $\ln(m)$ a log-moneyness;

4. Gaussian process regression with flat extrapolation; similar to 3, except that the Gaussian process predictor is only used for interpolation purposes; extrapolation whenever required is performed by the nearest neighbour method.

Again, a major difference between our functional (or neural net more generally) approach and these interpolation benchmarks is that, in order to interpolate a given surface, the neural network takes into account the information contained in all the surfaces of the data set, which is used as training set at the compression stage. In contrast, the above interpolation benchmarks only use the information provided by the available points of the currently interpolated surface, without consideration of the other surfaces in the data set. In particular, by Gaussian process regression in 3. and 4., we just mean interpolation within a given surface, using the available points in this surface as training set (unrelated to the potential use of Gaussian processes as an alternative to neural networks in our compression/completion approaches, which would be unrealistic as discussed in §1.A).

Accordingly, the functional approach exhibits significantly smaller completion errors. In Table 4, we reported these errors for two different choices of the 40 visible points :

- Less correlated points, i.e. locations for which the implied volatilities are the less correlated;

- Uniformly spread points, i.e.a random selection of at least 2 points per maturity. The lowest maturity can be assigned 3 visible points in order to reach a total number of 40 points.

	Functional	Functional with Forward	Linear interpolation	Spline interpolation	Gaussian process no extrapolation	Gaussian process flat extrapolation
Less correlated points	0.0262	0.0265	0.0632	0.0462	0.0555	0.0459
Uniformly spread points	0.0076	0.0091	0.0211	0.0168	0.0201	0.0208

Table 4: RMSEs for completed implied volatilities.

As the loss in (2) is now computed on much fewer points (partial information in this sense), the compression errors of the functional approach are obviously higher than the reconstruction errors from Table 3. Smaller error are reported in the second case above because less correlated points are rather located on short maturities, so that, in the first case little information, is available for the long maturities.

All the completion results reported hereafter correspond to the case of uniformly spread visible points.

The completion method provided by the functional approach is also robust: even the worst completion does not produce an outlier, i.e.

- the completed surface is smooth,
- the completed surface has a shape similar to the one of the original surface (the pointwise errors between the original and the completed surfaces are uniformly distributed),
- the implied total variance sensitivity with respect to the maturity is still positive (see Figure 7), inducing no calendar arbitrage opportunity,
- tails are consistent with the original points (see Figure 9) and not irregular.

Such robustness is not provided by the interpolation benchmarks. For instance, in the case of the worst completion with the spline interpolation, the completed surface (top-right corner of Figure 10) is irregular in the tails.

§6 At-the-Money Swaption Surfaces

The previous section was showing a case where the functional approach outperforms elementary interpolation benchmarks in an situation (in fact, the most common in the context of financial nowcasting applications) involving a moving grid.

We now consider an application where the grid is constant (after a preprocessing by our data provider) so that PCA or more classical autoencoder approaches are also available. The results show that the functional approach then performs as well as these classical benchmarks (which, however, would not be available on the original data with variable time-to-maturity).

A swaption is a financial contract allowing a client to enter into an interest rate swap with some strike K at some future expiry date U, for some tenor length T. A large body of literature deals with the swaption implied volatility as a function of the strike parameter.

By contrast, very few works are dealing with the swaption implied volatility as a function of the expiry and tenor parameters (see Figure 11). One exception is Trolle and Schwartz (2010), who, based on a time series of swaption cubes, investigate how the conditional moments of the underlying swap rate distributions vary with expiry, tenor, and calendar time. One possible reason for this relative lack of literature may be that swaption arbitrage pricing relationships are mainly known along the strike direction. Across expiries and tenors, one only has "statistical arbitrage" relations, reflecting the overlap between the cash flow streams of the underlying swaps.

[2]Gap between the original surface and the completed one.

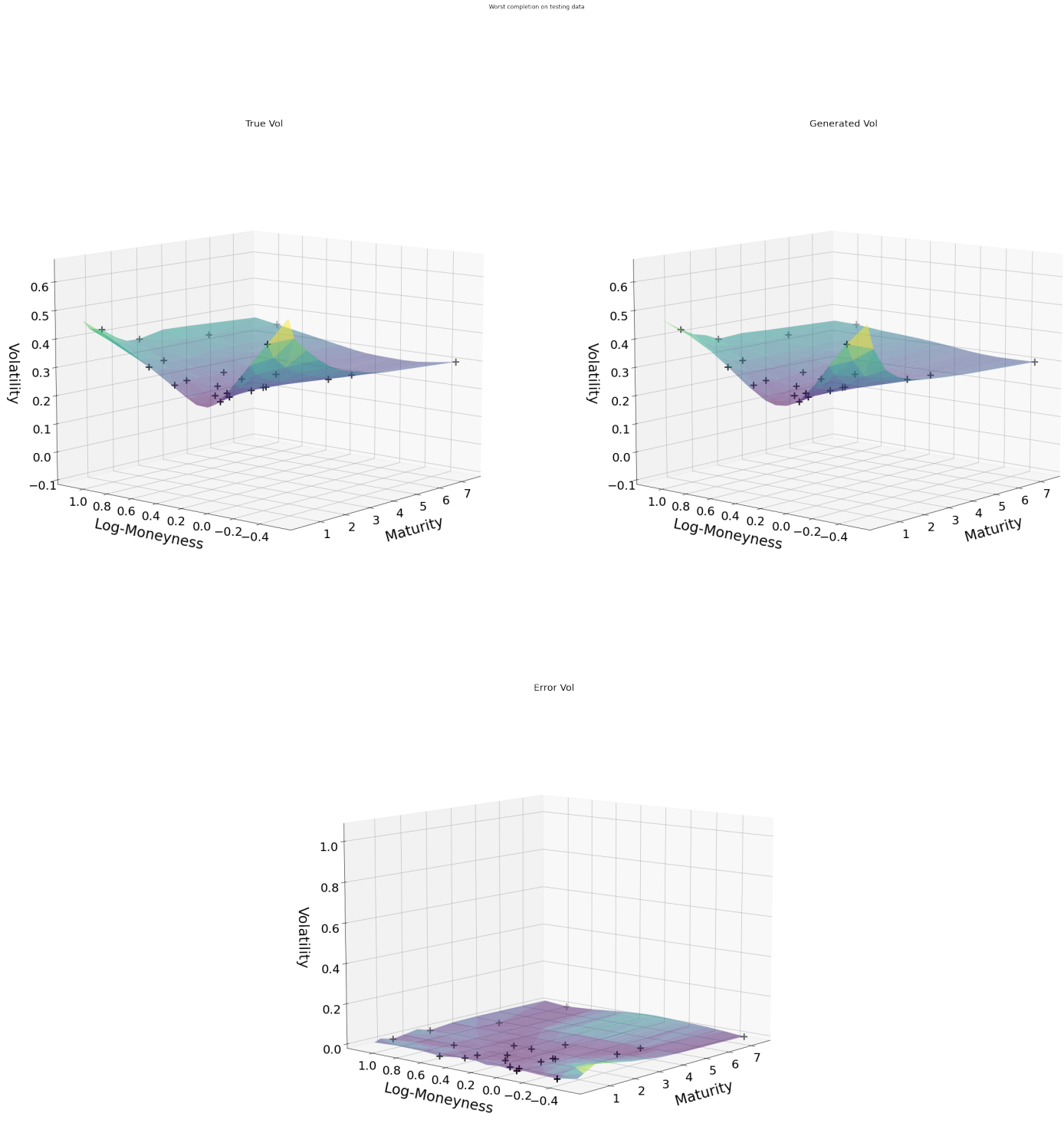


Figure 8: Original surface vs. completed surface yielding the worst RMSE. Black crosses mark visible points.

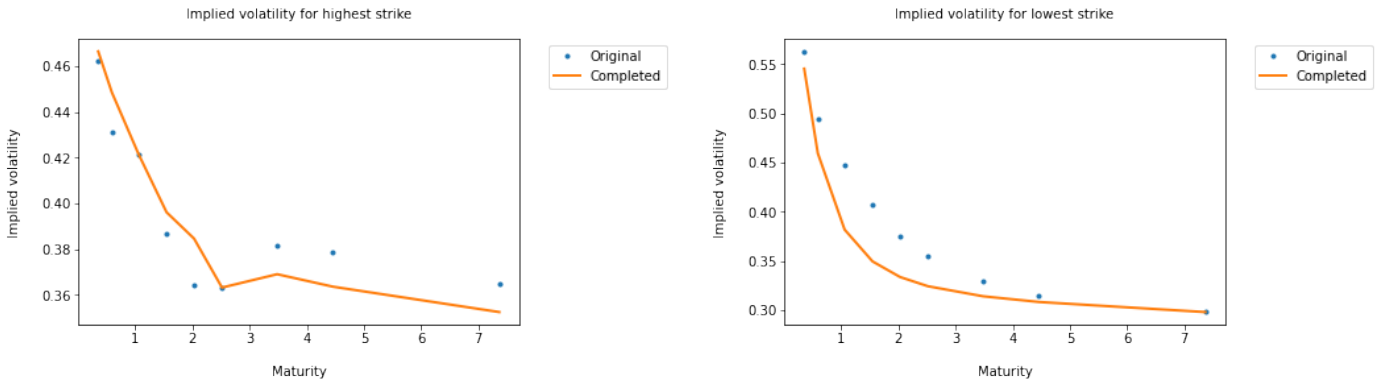


Figure 9: Tails of completed surface vs. original implied volatilities.

In the following case study, we focus on at-the-money (ATM, which are also the most liquid) swaption implied volatilities as a function of U and T. The approach is model free in the sense that we do not formulate or use any hypothesis on the underlying forward swap rate processes.

Our study is conducted on a daily database of monocurrency (euro) ATM swaption normal[3] implied volatilities, covering 2400 business days corresponding to the period from 2007 to 2017. The training calibration and validation set Ω covers the 2007 to 2014 sub-period (1900 first observation days of the data set), whereas the test set Ω' ranges from 2015 to 2017 (500 subsequent ones). The data have been preprocessed by our provider so that all the ATM implied volatility surfaces are defined on a common rectangular grid of eighty (U,T) nodes, without missing implied volatility values at any day or node, corresponding to the ten expiries (with M for month and Y for year)

$$U \in (1M, 3M, 6M, 1Y, 2Y, 5Y, 7Y, 10Y, 20Y, 30Y)$$

and the eight tenors

$$T \in (3M, 1Y, 2Y, 5Y, 10Y, 15Y, 20Y, 30Y).$$

For testing our completion approach, we mask 90% of the points in each surface of the test set Ω', only keeping the volatility points corresponding to the grid nodes (U,T) in

$$\begin{aligned}&(1M,3M),(1M,10Y),(1M,30Y),(6M,2Y),\\&(6M,15Y),(5Y,1Y),(5Y,20Y),(10Y,5Y).\end{aligned} \tag{8}$$

Such specification is in line with the reality of a market where the shortest expiries are the most liquidly traded ones (as well as the most volatile). Hence, our completion exercise corresponds to the intraday situation of a swaption trader facing mostly short expiry ATM implied volatility data, and left with the task of guessing the "most likely values" of the remaining implied volatilities.

A Network Architectures

The corresponding architecture of the functional approach is then similar to the one used for equity derivatives in C.A, except that the expiry U and tenor T are used as the localization inputs, and only 8 latent variables are used (instead of 15 previously): see Figure 12. Moreover, one can also incorporate the forward swap rates as exogenous variables. These are the underlyings of the swaptions and they are structured similarly to the ATM implied volatilities of the latter, located by an expiry and a tenor. For taking them into account, it suffices to add to the network of Figure 12 an additional feature (input unit) containing the level of the forward swap rate with expiry U and tenor T. Hence, the units for the expiry U and the tenor T indicate the common location of the corresponding ATM volatilities and forward swap rates.

The convolutional autoencoders use feed-forward neural networks for the encoder and the decoder, with four hidden layers each: one dense layer is applied on top of three convolutional layers for the

[3] rather than Black–Scholes, because of the negative rates environment.

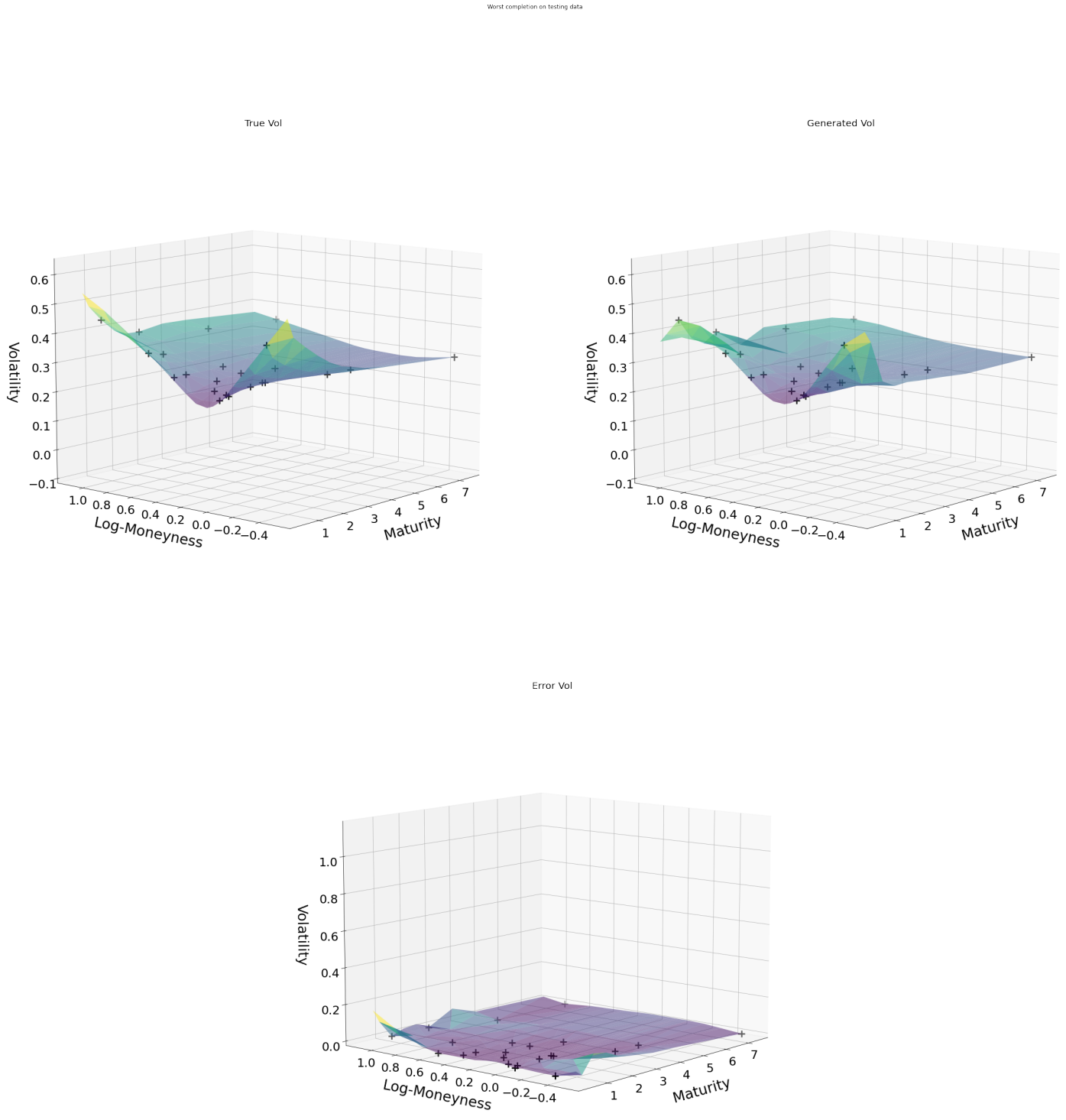


Figure 10: Original surface vs. completed surface yielding the worst RMSE with spline interpolation. Black crosses mark visible points.

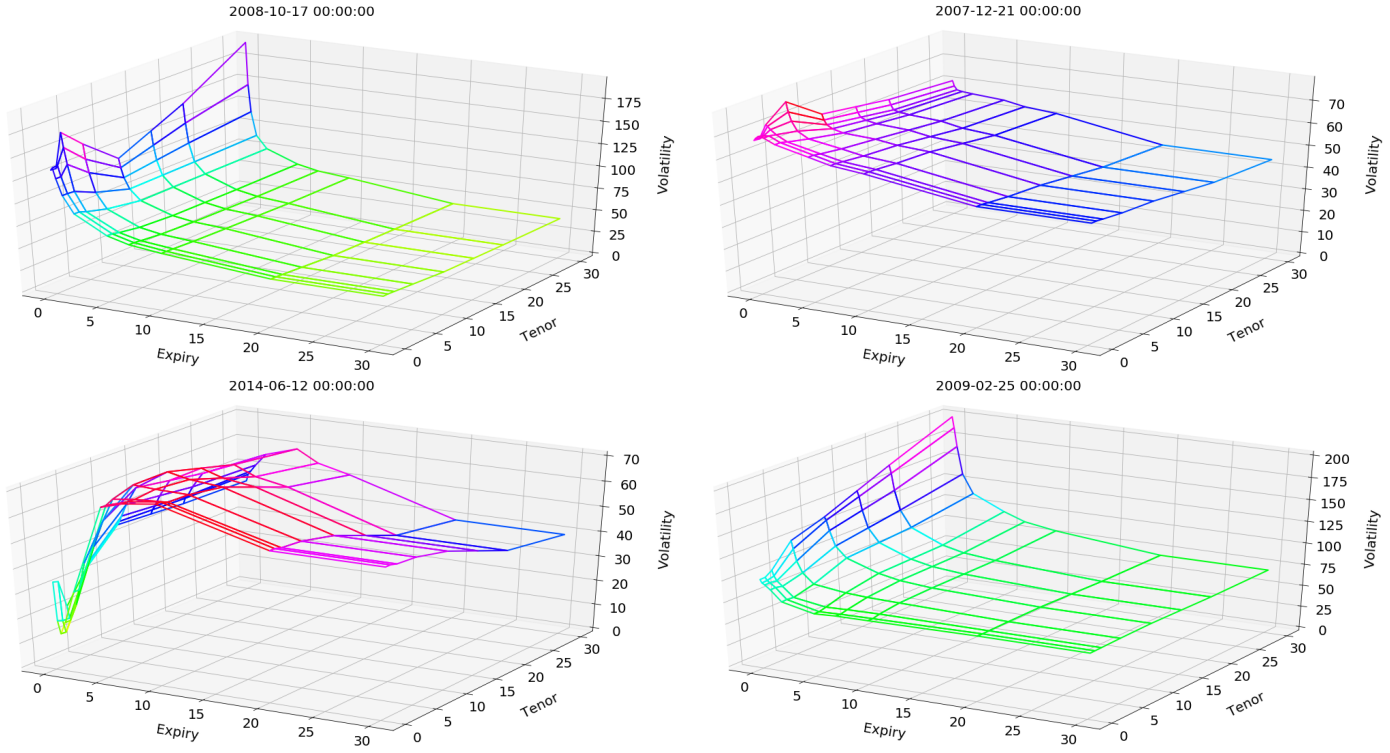


Figure 11: Different patterns of at-the-money swaption volatility surface.

encoder and, symmetrically, three deconvolutional layers are built on top of one dense layer. The data set is reshaped as a $(10, 8)$ tensor per day. The convolution layers are built with the respective kernels (used for specifying the localization of the weights) $(5, 4)$, $(4, 3)$, and $(3, 3)$. Each convolution layer produces 3 channels (see Figure 13) and, symmetrically, each deconvolution layer has in input 3 times more channels than in output. Padding is set as VALID in order to reduce the size of the hidden units after each convolution layer. As output of the three convolution layers, we have a hidden layer of 27 units, corresponding to 27 channels of size $(1, 1)$. A softplus (regularized ReLU) activation function is chosen after each convolution layer. This results in sparsity of the calibrated network (the compression stage sets very negative biases on the intermediate units that the neural network wants to ignore, cf. Bengio (2012)), as well as positivity and regularity of the ensuing implied volatility surface. The dense layers between the factors and the (de)convolution layers are linear. Hence, the convolution layers can be seen as a kernel that linearly separates the features.

Following a divide-and-conquer, sequential training strategy, we train the convolutional layers by pairs, from the most outer to the most inner ones, i.e. the layers surrounding the latent variables (greedy layer-wise pre-training as per Hinton, Osindero, and Teh (2006) and Bengio, Lamblin, Popovici, and Larochelle (2007)). A final optimization fine-tunes the weights of all the layers together. This also allows exploiting any hierarchical structure of the data (cf. Masci, Meier, Cireşan, and Schmidhuber (2011)): The outer layers detect the greatest patterns, while inner layers detect the finest ones.

In the case of the fully connected networks that are used in the linear projection and in the functional approaches, we use the Glorot and Bengio (2010) initialization rule for the weights, with a centered normal distribution of standard deviation $\sqrt{\frac{4}{n_{inputs}+n_{outputs}}}$. In the case of the convolutional layers we use a truncated normal distribution with 0.1 standard deviation. All biases are initialized to zero.

Each iteration leads to the computation of the loss gradient on the whole calibration data set. Indeed, given the relatively small size of our data sets, full gradient evaluation is not an issue in practice. Moreover, mini-batch would require that each batch sample has approximately the same distribution, which is notoriously violated in the case of (non-stationary) financial time series.

Penalization is used at the compression stage for regularizing the calibrated parameters. More precisely, ridge regularization is used for the kernel weights of the fully-connected layers of the convolutional and of the functional approaches, with a penalization coefficient of 0.1 intended to balance the

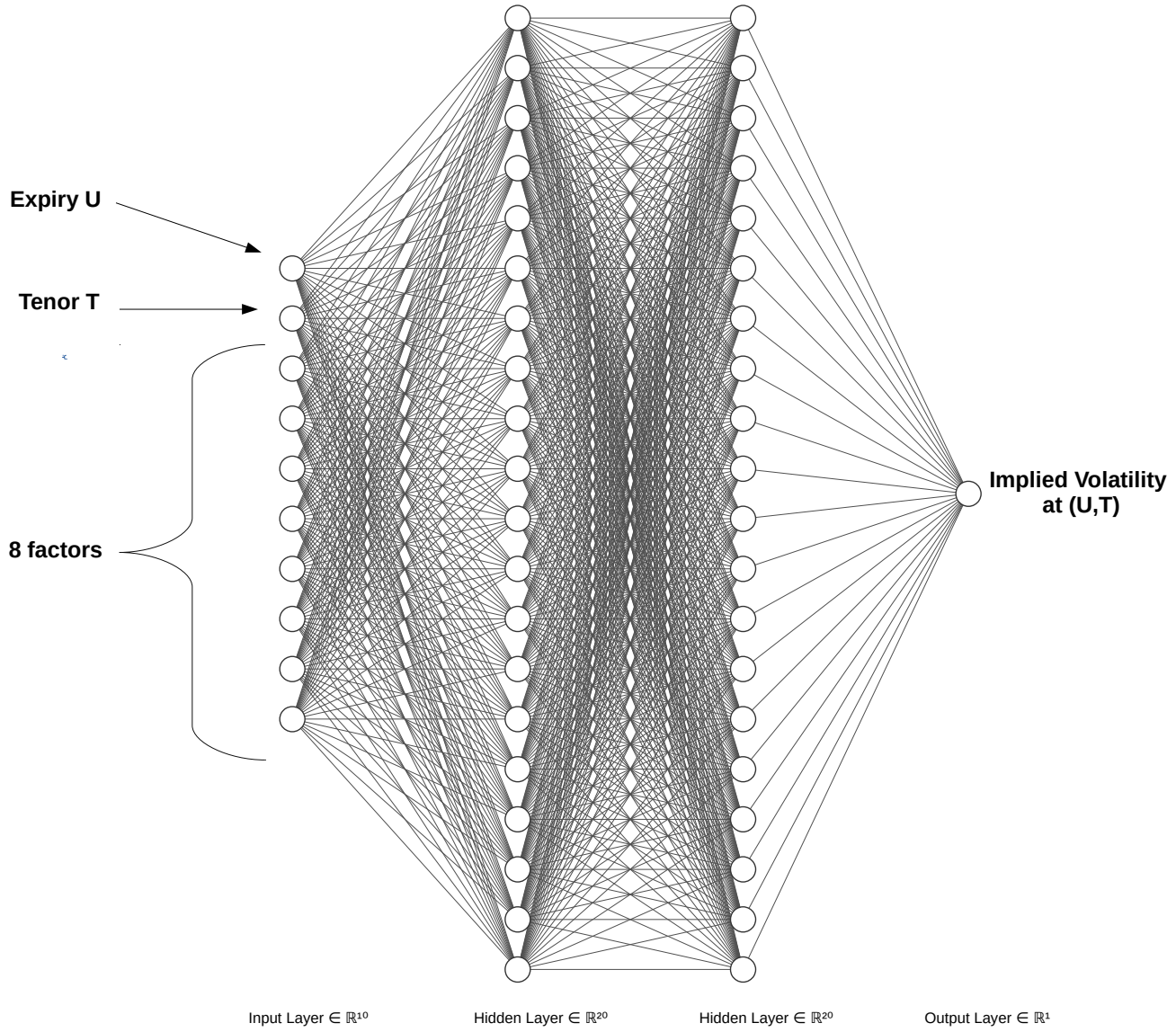


Figure 12: Network of the functional approach used in the swaption case study (style FCNN of the NN-SVG software, cf. Figure 1).

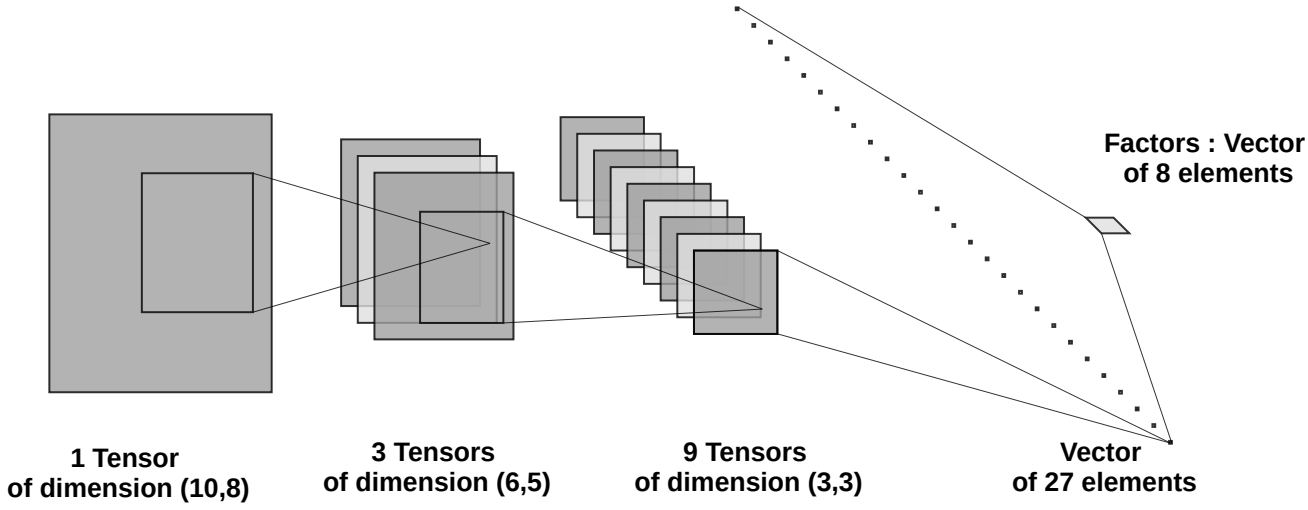


Figure 13: Architecture of the convolutional encoder used in our ATM swaption case study. Graph produced using the style LeNet of the NN-SVG software: Each of the four layers is represented by a triangle; The inputs of each of the three convolutional layers are displayed as collections of tensors; The ones of the last, dense layer are represented as a series of dots.

reconstruction loss and the penalization term at the minimum.

B Benchmarking

Table 5 is a report on the errors of all our approaches. It is based on the absolute daily RMSEs (cf. (5) and (6)).

	Standard PCA	Linear projection	Convolutional autoencoder	Functional approach	Functional approach with forward rate
Average compression error on Ω	1.23	1.58	1.97	1.85	2.29
Average compression error on Ω'	3.71	3.54	6.19	3.77	3.02
Worst compression error on Ω [day] ([day])	4.15 [2008-12-03]	3.98 [2008-12-09]	7.18 [2008-12-08]	8.32 [2008-10-09]	6.93 [2008-10-10]
Worst compression error on Ω' [day] ([day])	5.76 [2016-04-28]	5.18 [2016-04-28]	12.0 [2015-07-07]	6.34 [2015-12-21	5.16 [2015-12-18]
Average completion error on Ω'	6.19	4.07	5.03	6.41	5.19
Worst completion error on Ω' [day] ([day])	12.6 [2015-06-30]	6.50 [2015-07-10]	9.89 [2015-07-10]	12.8 [2015-03-09]	9.09 [2016-01-14]
Training time in seconds	$\emptyset$	9	411	1287	276

Table 5: RMSEs in the sense of (5) and (6)

The last row of Table 5 displays the corresponding training times for all but the standard PCA approach, which involves no training and is in fact much faster than all the others (as it essentially reduces to the inversion of an $m \times m$ matrix, with $m = 80$). The dates in brackets in the tables identify the observations corresponding to the worst errors.

At the completion stage, we take as initial factor values the volatility encoding of the previous day. Figure 14 shows the stability through calendar time of the codes obtained by the linear projection approach.

As shown by Figure 15 in the case of the linear projection approach (but this is also true of the nonlinear approaches), the dominant errors are concentrated on the shortest expiries. This is because the implied volatilities corresponding to these shortest expiries are the more volatile. Hence, their spatial dependence structure is less informative. To recover these points better, one could think of providing extra information through exogenous variables, such as the level of the underlying forward swap rates. Under the functional approach, this can easily be done in the way explained in A. However, the last columns in Table shows that this only has a minor positive impact.

The linear approaches are as accurate as the nonlinear ones and the convolutional approach is typically outperformed by at least the linear projection or the functional approach.

Figure 17 illustrates that the functional approach enables to interpolate smoothly the surface over an arbitrarily fine grid, in this case 10^4 points obtained by the corresponding interpolation of the tensor of Figure 16.

§7 Conclusions and Perspectives

In this chapter we have presented a generic neural network based curve or surface (or more general tensor) compression/completion methodology, for which we have proposed two concrete specifications: the functional approach, amenable to the treatment of unstructured data with varying grid nodes (as

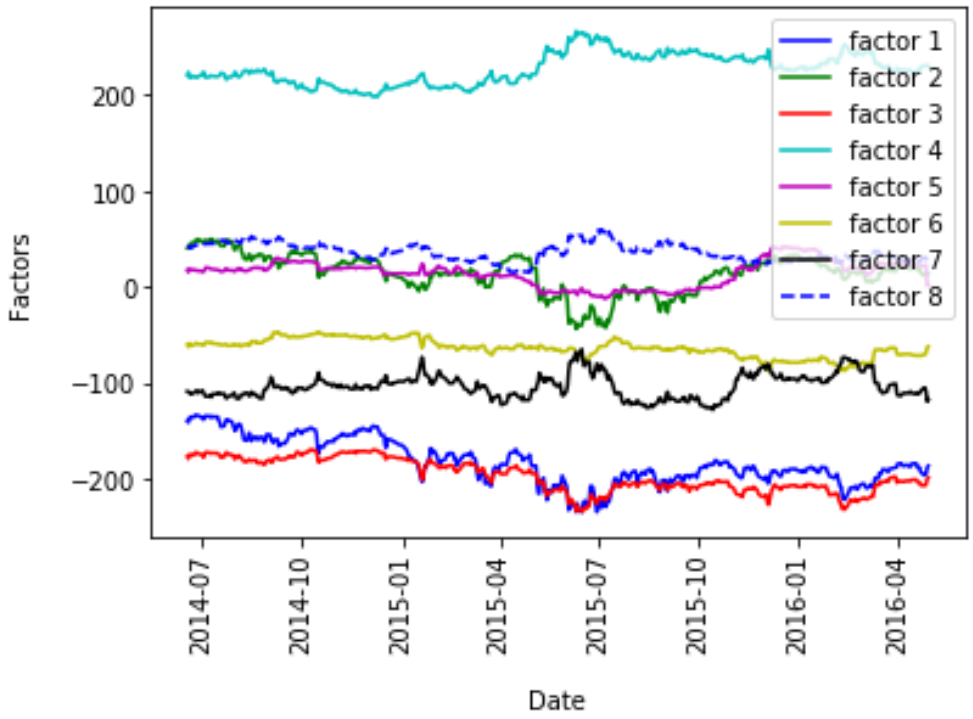


Figure 14: Time series of the factors obtained by encoding of the training observations under the linear projection approach.

natively the case in most financial nowcasting applications), and a convolutional autoencoder approach, including PCA or PCA-like projections as linear special cases, applicable in the special case of a constant grid (natively or possibly after some preprocessing). The compression stage also allows for outlier detection and correction by generating surfaces or curves in line with training samples.

The analysis of the corresponding reconstruction errors suggests that linear methods are sufficient to compress structured tensors, corresponding to a constant grid of nodes, into few factors coefficients. The completion stage allows recovering with success about 90% values of the data, starting from about 10% of known values. But the functional approach is the only one that is able to directly deal (without preprocessing) with the most common situation of unstructured tensors. The only alternative is then naive interpolation benchmarks that do not exploit the data set, and which the functional approach is shown to outperform in our equity derivative case study.

All approaches suffer from non-stationarities occurring during extreme events or change of market regimes. This can be seen as an advantage with respect to anomaly detection. For other purposes, it would plead in favor of further modeling of the factor dynamics, whether this relies on times series machine learning or Markov chain Monte Carlo (filtering) techniques. More generally, it would be interesting to extend this study in several directions, such as the introduction of backtesting hedging criteria (Garcia and Gençay, 2000), scenario simulation in a context of variational networks (Tschannen et al., 2018), application of the method to the whole swaption volatility cube, strike dimension included (Trolle and Schwartz, 2010), or specification of dynamics on the factors (for instance by Kalman filters).

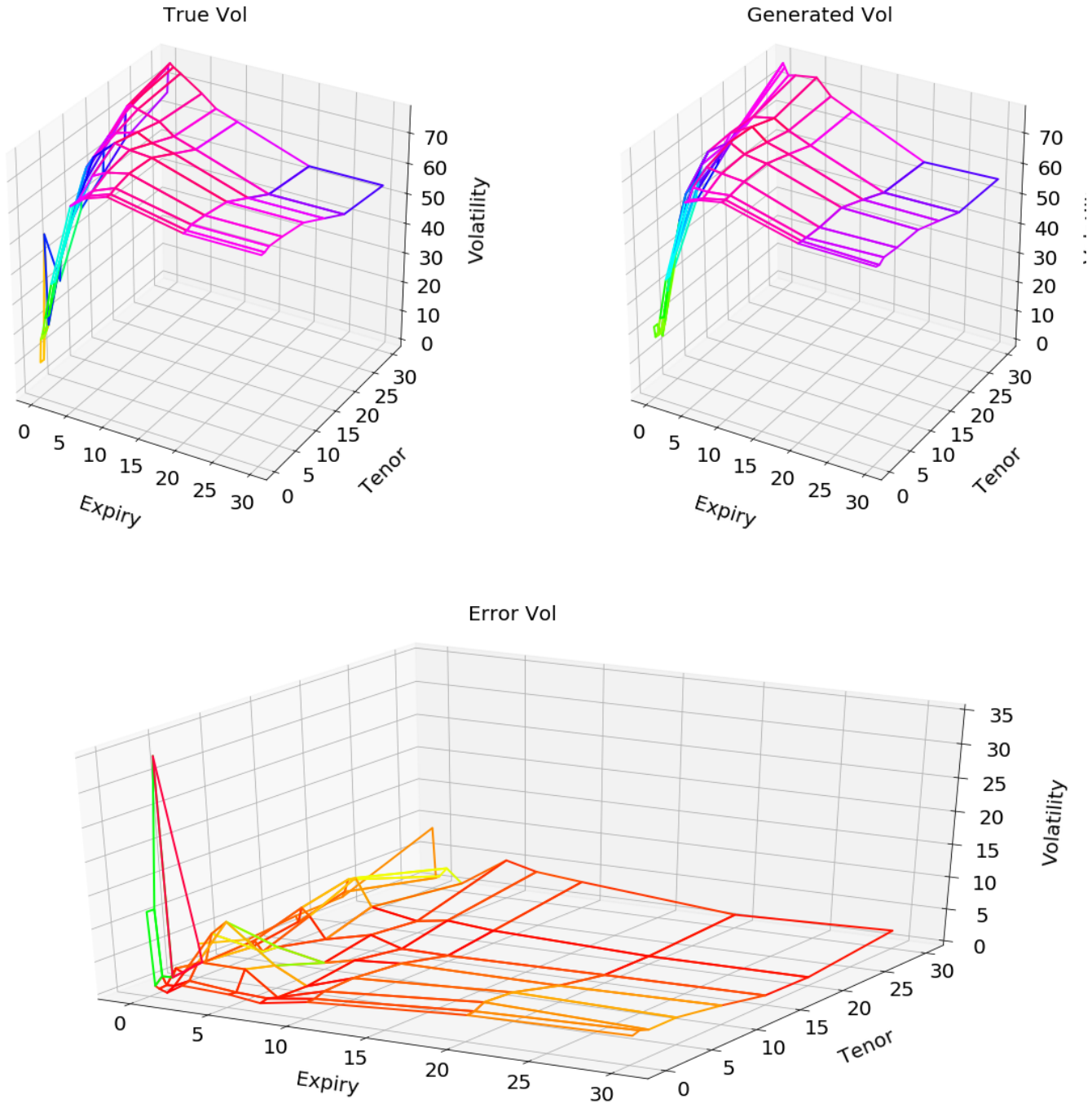


Figure 15: Linear projection approach: *(Top left)* Original (full) tensor; *(Top right)* Tensor $D_{\delta^\star}(c^\star)$ completed based on the 8 points of the latter given by (8); *(Bottom)* Pointwise absolute error between the two, for the worst observation in Ω'.

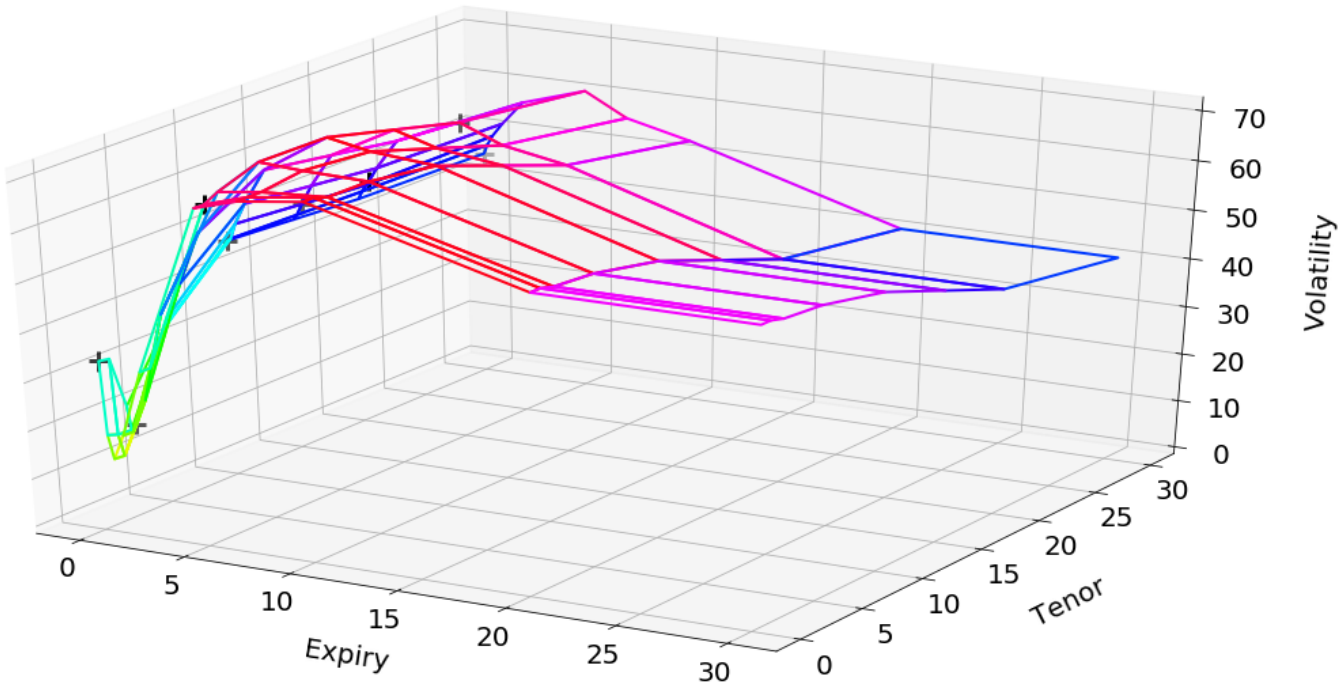


Figure 16: Complete tensor corresponding to the first observation in Ω'. The black crosses designate the "available points", specified by (8), that are used in the completion exercise.

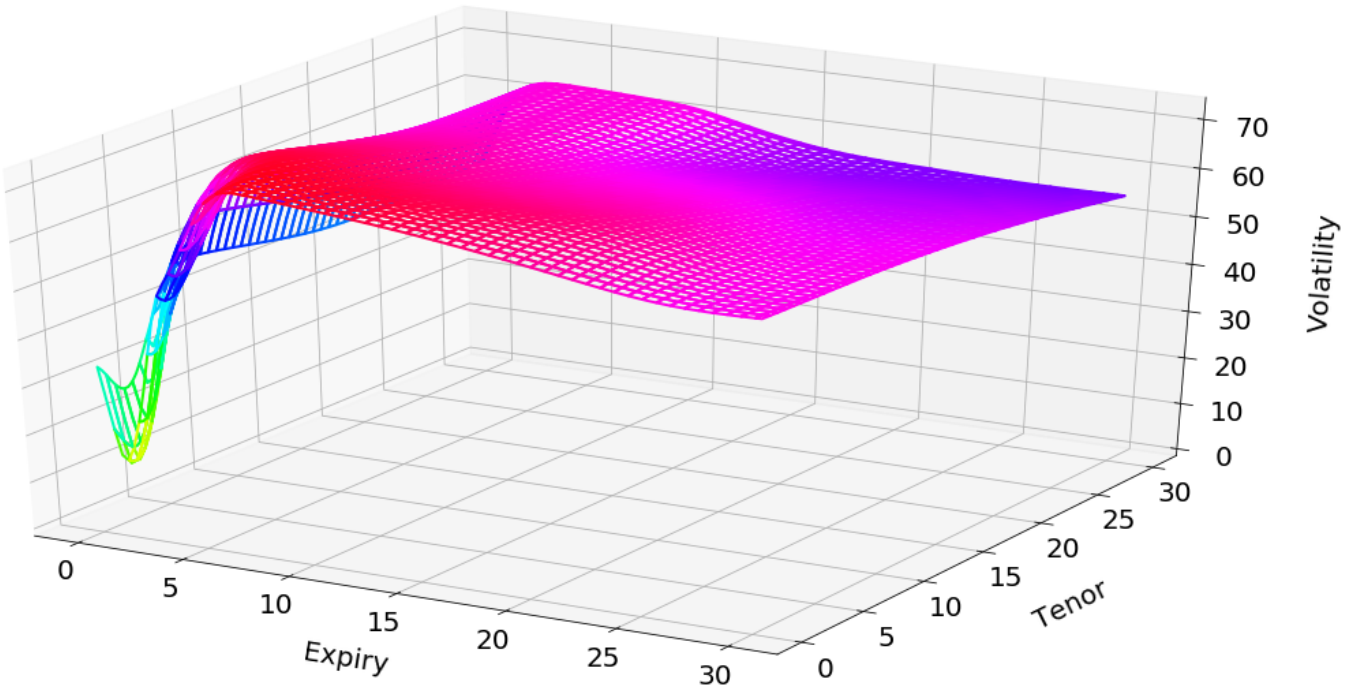


Figure 17: Surface with 10^4 points obtained by the functional approach applied to the first observation in Ω'.

COMPLEMENTS

Chapter IX

Mathematical Tools

We provide a toolbox of results in stochastic analysis, which are used in the main body of the notes.

We use the French acronym càdlàg for "left limited and right continuous" (a.s., with finite left and right limits everywhere). We use càglàd for the left-limit processes of càdlàg processes[1].

§1 Local Martingales

On a probability space $(\Omega, \mathcal{A}, \mathbb{Q})$, with $\mathbb{Q}$ expectation denoted by $\mathbb{E}$, and given a fixed horizon $T \in (0, +\infty)$, we consider a continuous-time filtration $\mathfrak{F} = (\mathfrak{F}_t)_{t\in[0,T]}$, with respect to which all our *processes* are adapted.

Definition 1 *A process Y is said to be a martingale (resp. supermartingale) with respect to the filtration $\mathfrak{F}$ if (shortening $\mathbb{E}[\cdot|\mathfrak{F}_t]$ into $\mathbb{E}_t$):*
(i) *(Y is $\mathfrak{F}$ adapted and) $\mathbb{E}|Y_t| < +\infty$, $\forall t$;*
(ii) *$\forall s \leq t$, we have $Y_s = \mathbb{E}_s Y_t$ (resp. $Y_s \geq \mathbb{E}_s Y_t$).*

We assume that the model filtration $\mathfrak{F}$ satisfies the so-called usual conditions of completeness[2] and right-continuity[3]. Then every martingale (or more general local martingale below) admits a càdlàg modification[4]. Modifying a process does not modify the family of its finite-dimensional distributions, hence preserves its law. In these notes all local martingales are assumed càdlàg.

Definition 2 *A $[0,T]$ valued random variable ϑ is said to be a stopping time with respect to $\mathfrak{F}$ if, for each t, the indicator function $\mathbb{1}_{\{\vartheta\leq t\}}$ of the event $\{\vartheta \leq t\}$ is measurable with respect to $\mathfrak{F}_t$.*

Lemma 1 [5] **(i)** *The hitting time of an open set $\mathcal{O}$ by an adapted càdlàg process X, i.e. $\vartheta = \inf\{t > 0;\ X_t \in \mathcal{O}\} \wedge T$ (restricting attention to $[0,T]$), is a stopping time.*
(ii) *The hitting time of a closed set $\mathcal{C}$ by an adapted continuous process X, i.e. $\vartheta = \inf\{t > 0;\ X_t \in \mathcal{C}\} \wedge T$ (likewise), is a stopping time.*
(iii) *For any $a \in \mathbb{R}$ and adapted nondecreasing process X, $\vartheta = \inf\{t > 0;\ X_t \geq a\} \wedge T$ is a stopping time.*

Proof (sketched). Given a denumerable dense subset $\mathbb{D}$ of $\mathbb{R}$, e.g. the rational numbers: **(i)** Show that $\{\tau < s\} = \cup_{\mathbb{D}\ni r<s}\{X_r \in \mathcal{O}\} \in \mathfrak{F}_s$. Deduce that $\{\tau \leq t\} \in \mathfrak{F}_{t+} = \mathfrak{F}_t$.

[1] more restrictive than the common use of càglàd for "left limited and right continuous" processes.
[2] i.e. $\mathfrak{F}_0$ contains all the null sets of $(\Omega, \mathbb{Q})$.
[3] i.e. $\mathfrak{F}_{t+} = \mathfrak{F}_t$, where $\mathfrak{F}_{t+} = \cap_{s>t}\mathfrak{F}_s$ represents the model information "right after time t".
[4] see He, Wang, and Yan (1992, Corollary 2.48 page 56).
[5] see Karatzas and Shreve (1991, Corrected problems 2.6 and 2.7 p.39), He, Wang, and Yan (1992, Problems and Complements 3.1 through 3.3 pages 106-107), and Revuz and Yor (1999, 1.§4).

(ii) Noting that $X_s \in \mathcal{C} \iff Y_s := d(X_s, \mathcal{C}) = 0$, show using the continuity of X that $\{\tau \leq t\} = \{(\inf_{s\in(\mathbb{D}\cap[0,t])\cup\{t\}} Y_s) = 0\}$. Deduce that $\{\tau \leq t\} \in \mathfrak{F}_t$.
(iii) Show that $\{\tau \geq s\} = \cap_{\mathbb{D}\ni r<s}\{X_r < a\}$. Deduce that $\{\tau < s\} \in \mathfrak{F}_s$, hence $\{\tau \leq t\} \in \mathfrak{F}_{t+} = \mathfrak{F}_t$. ∎

Lemma 2 [6] *Let M be a martingale on $[0,T]$ and ϑ be a $[0,T]$ valued stopping time, then*

$$\mathbb{E}M_\vartheta = \mathbb{E}M_0.$$

Conversely:

Lemma 3 *If an $\mathfrak{F}$ adapted process Y is such that Y_ϑ is integrable for any $[0,T]$ valued stopping time ϑ, with $\mathbb{E}Y_\vartheta$ independent of ϑ, then Y is a martingale on $[0,T]$.*

Proof. Then, for any $t \in [0,T]$ and $A \in \mathfrak{F}_t$, $t_A = t\mathbb{1}_A + T\mathbb{1}_{A^c}$ is a $[0,T]$ valued stopping time[7] and we have $Y_{t_A} = Y_t\mathbb{1}_A + Y_T\mathbb{1}_{A^c}$, hence as $\mathbb{E}(Y_{t_A}) = \mathbb{E}(Y_T)$:

$$\mathbb{E}(Y_t\mathbb{1}_A) = \mathbb{E}(Y_{t_A}) - \mathbb{E}(Y_T\mathbb{1}_{A^c}) = \mathbb{E}(Y_T) - \mathbb{E}(Y_T\mathbb{1}_{A^c}) = \mathbb{E}(Y_T\mathbb{1}_A).$$

Therefore $\mathbb{E}_tY_T = Y_t$ holds a.s., for each $t \in [0,T]$, and Y is a martingale. ∎

The following definition is the variant on $[0,T]$, obtained "by stopping at T", of the usual definition on $\mathbb{R}_+$.

Definition 3 *The process Y is said to be a local martingale on $[0,T]$ with respect to the filtration $\mathfrak{F}$ if it admits a nondecreasing (dubbed* localizing*) sequence of $[0,T]$ valued stopping times τ_n such that $\mathbb{Q}(\cup_n\{\tau_n = T\}) = 1$ and every stopped process $Y_{\cdot\wedge\tau_n}$ is a martingale on $[0,T]$.*

Lemma 4 [8] *A local martingale M satisfying $\mathbb{E}\sup_{0,T}|M| < +\infty$ is a martingale over $[0,T]$.*

Proof. Let (τ_n) denote a localizing sequence of $[0,T]$ valued stopping times for M. If $t \leq T$, then $\mathbb{E}_tM_{T\wedge\tau_n} = M_{t\wedge\tau_n}$ holds for all n, which implies $\mathbb{E}_tM_T = M_t$, by the dominated convergence theorem. ∎

Lemma 5 **(i)** *Any local martingale $Y \geq$ a martingale M is a supermartingale.*
(ii) *Any local martingale sandwiched between two martingales and, in particular, any local martingale dominated (i.e. in absolute value) by a martingale, is a martingale.*

Proof. **(i)** The general case is readily deduced by application to $Y - M$ of the special case when $M = 0$, to which we therefore restrict ourselves hereafter. Let there be given a local martingale $Y \geq 0$ with a localizing sequence of stopping times (τ_n). For every n,

$$Y_{s\wedge\tau_n} = \mathbb{E}[Y_{t\wedge\tau_n}|\mathfrak{F}_s]\,,\ 0 \leq s \leq t \leq T.$$

Sending n to ∞, we obtain by the (conditional) Fatou lemma that

$$\begin{aligned} Y_s &= \liminf_n Y_{s\wedge\tau_n} = \liminf_n \mathbb{E}[Y_{t\wedge\tau_n}|\mathfrak{F}_s] \\ &\geq \mathbb{E}[\liminf_n Y_{t\wedge\tau_n}|\mathfrak{F}_s] = \mathbb{E}[Y_t|\mathfrak{F}_s]\,,\ 0 \leq s \leq t \leq T. \end{aligned}$$

In addition, $Y_0 = Y_{0\wedge\tau_0}$ is integrable, hence so is Y_t for any fixed t, as $\mathbb{E}Y_t \leq \mathbb{E}Y_0$ holds by the already established inequality $\mathbb{E}_0Y_t \leq Y_0$.

(ii) results from a double application of (i) to $\pm Y$. ∎

[6] cf. He, Wang, and Yan (1992, Theorem 2.58 page 60).
[7] cf. He, Wang, and Yan (1992, Theorem 3.9(i) page 85).
[8] cf. Protter (2004, Theorem I.51 page 38).

Lemma 6 *Any absolutely continuous*[9] *local martingale M on $[0,T]$ is in fact constant.*

Proof. [10] Let us assume $M_0 = 0$ and show that M vanishes (the general case follows by application of this special case to $M - M_0$). Since M is absolutely continuous, $\int_0^t |M'_s| ds$ is finite for all $t \in [0,T]$[11]. The $\tau_n := \inf\{t > 0;\ \int_0^t |M'_s| ds \geq n\} \wedge T$ are a nondecreasing sequence of stopping times such that $\mathbb{Q}(\cup_n \{\tau_n = T\}) = 1$[12]. As

$$|M_t| = |\int_0^t dM_s| \leq \int_0^t |M'_s| ds, \tag{1}$$

for any fixed k the process $K := M_{\cdot \wedge \tau_k}$ is a bounded martingale, by application[13] of Lemma 5(ii). Then, denoting $t_i^n = i\frac{T}{n}$, we have $\mathbb{E}_{t_{i-1}^n} K_{t_i^n} = K_{t_{i-1}^n}$, hence

$$\begin{aligned}
\mathbb{E}(K_{t_i^n} - K_{t_{i-1}^n})^2 &= \mathbb{E}\mathbb{E}_{t_{i-1}^n}(K_{t_i^n}^2 - 2K_{t_i^n} K_{t_{i-1}^n} + K_{t_{i-1}^n}^2) \\
&= \mathbb{E}(K_{t_{i-1}^n}^2 - 2K_{t_{i-1}^n} \mathbb{E}_{t_{i-1}^n} K_{t_i^n} + \mathbb{E}_{t_{i-1}^n} K_{t_i^n}^2) \\
&= \mathbb{E}K_{t_i^n}^2 - \mathbb{E}K_{t_{i-1}^n}^2.
\end{aligned}$$

Therefore,

$$\begin{aligned}
\mathbb{E}K_t^2 = \sum_{i=1}^n \mathbb{E}(K_{t_i^n} - K_{t_{i-1}^n})^2 &\leq \mathbb{E}\big(\max_{1 \leq i \leq n} |K_{t_i^n} - K_{t_{i-1}^n}| \cdot \sum_{i=1}^n |K_{t_i^n} - K_{t_{i-1}^n}| \big) \\
&\leq k\mathbb{E} \max_{1 \leq i \leq n} |K_{t_i^n} - K_{t_{i-1}^n}| \to_{n\infty} 0,
\end{aligned} \tag{2}$$

by the dominated convergence theorem and the fact that (K_t) is continuous a.s.. Hence $K_t = M_{t \wedge \tau_k} = 0$, for all k. Since $\mathbb{Q}(\cup_n \{\tau_n = T\}) = 1$ we conclude that $M_t = 0$ almost surely holds. Then $M = 0$ almost surely holds on $[0,T]$, by continuity of M. ∎

Lemma 7 *Given a positive $\mathbb{Q}$ martingale ν with $\nu_0 = 1$, let $\widetilde{\mathbb{Q}}$ be the probability measure on $(\Omega, \mathfrak{F}_T)$ such that $\frac{d\widetilde{\mathbb{Q}}}{d\mathbb{Q}} = \nu_T$. We denote the $\widetilde{\mathbb{Q}}$ expectation by $\widetilde{\mathbb{E}}$. For any given (adapted) nonnegative process $\widetilde{X}$:*
(i) *For $s \leq t$, $\widetilde{\mathbb{E}}(\widetilde{X}_t \,|\, \mathfrak{F}_s) = \frac{1}{\nu_s}\mathbb{E}\big(\nu_t \widetilde{X}_t \,|\, \mathfrak{F}_s\big)$.*
(ii) *A process $\widetilde{X}$ is a $\widetilde{\mathbb{Q}}$ (resp. local) martingale if and only if $(\nu\widetilde{X})$ is a $\mathbb{Q}$ (resp. local) martingale.*

Proof. For a given (adapted) nonnegative process $\widetilde{X}$, for each time s, we have (shortening $\mathbb{E}[\cdot|\mathfrak{F}_s]$ into $\mathbb{E}_s$)

$$\widetilde{\mathbb{E}}\widetilde{X}_s = \mathbb{E}(\widetilde{X}_s \nu_T) = \mathbb{E}\mathbb{E}_s(\widetilde{X}_s \nu_T) = \mathbb{E}(\widetilde{X}_s \nu_s),$$

by the martingale property of ν. Hence an adapted process $\widetilde{X}$ is $\widetilde{\mathbb{Q}}$ integrable if and only if $(\nu\widetilde{X})$ is $\mathbb{Q}$ integrable. Then the formula in (i) holds because, for any $A \in \mathfrak{F}_s$,

$$\begin{aligned}
\widetilde{\mathbb{E}}\Big(\frac{1}{\nu_s}\mathbb{E}\big(\nu_t \widetilde{X}_t \,|\, \mathfrak{F}_s\big)\mathbb{1}_A\Big) &= \mathbb{E}\Big(\frac{\nu_T}{\nu_s}\mathbb{E}\big(\nu_t \widetilde{X}_t \,|\, \mathfrak{F}_s\big)\mathbb{1}_A\Big) = \mathbb{E}\mathbb{E}_s\Big(\frac{\nu_T}{\nu_s}\mathbb{E}\big(\nu_t \widetilde{X}_t \,|\, \mathfrak{F}_s\big)\mathbb{1}_A\Big) \\
&= \mathbb{E}\Big(\mathbb{E}\big(\nu_t \widetilde{X}_t \,|\, \mathfrak{F}_s\big)\mathbb{1}_A\Big) = \mathbb{E}\big(\nu_t \widetilde{X}_t \mathbb{1}_A\big) = \mathbb{E}\big((\mathbb{E}_t \nu_T)\widetilde{X}_t \mathbb{1}_A\big) \\
&= \mathbb{E}\mathbb{E}_t(\widetilde{X}_t \mathbb{1}_A \nu_T) = \mathbb{E}(\widetilde{X}_t \mathbb{1}_A \nu_T) = \widetilde{\mathbb{E}}(\widetilde{X}_t \mathbb{1}_A).
\end{aligned}$$

This implies the statement in (ii) regarding martingales, from which the one regarding local martingales follows by localization. ∎

[9] i.e. in dt.
[10] adapted from https://faculty.math.illinois.edu/ psdey/MATH562FA21/lec11.pdf accessed on 26 Nov 2021.
[11] see https://en.wikipedia.org/wiki/Absolute_continuity#Definition.
[12] as $\cup_n \{\tau_n = T\}$ contains the set $\{\int_0^T |M'_s| ds < +\infty\}$, which is of $\mathbb{Q}$ probability measure 1.
[13] to K and to $(-K)$.

Lemma 8 *A progressive process Z[14] such that $\int_0^T \|Z_s\|^2 ds < \infty$ is integrable against a standard Brownian motion W (possibly multivariate, with then Z of the same dimension as W) in the sense of local martingales on $[0,T]$.*

Proof. Assuming $\int_0^T \|Z_s\|^2 ds < \infty$, the

$$\tau_n = \inf\{t; \int_0^t \|Z_s\|^2 ds \geq n\} \wedge T$$

are a nondecreasing sequence of stopping times[15] such that $\mathbb{Q}(\cup_n\{\tau_n = T\}) = 1$.[16]. For each n we have

$$\int_0^T \mathbb{1}_{\{s\leq\tau_n\}}\|Z_s\|^2 ds = \int_0^{\tau_n} \|Z_s\|^2 ds \leq n,$$

hence

$$\mathbb{E}\big(\int_0^T \mathbb{1}_{\{s\leq\tau_n\}} Z_s dW_s\big)^2 = \mathbb{E}\big(\int_0^T \mathbb{1}_{\{s\leq\tau_n\}}\|Z_s\|^2 ds\big) \leq n$$

is finite, so that $\int_0^\cdot \mathbb{1}_{\{s\leq\tau_n\}} Z_s dW_s = \int_0^{\cdot\wedge\tau_n} Z_s dW_s$ is a martingale on $[0,T]$, by the standard Itô stochastic integration theory[17]. In conclusion $\int_0^\cdot Z_s dW_s$ is a local martingale on $[0,T]$, with τ_n as a localizing sequence of stopping times. ∎

Lemma 9 *Progressive processes Z such that $\int_0^T \|Z_s\|^2 ds < \infty$ include all the càglàd or càdlàg (adapted) processes on $[0,T]$.*

Proof. Any (adapted) càdlàg process $\widetilde{Z}$ or càglàd process $Z = \widetilde{Z}_-$ is progressive[18]. The

$$\tau_n = \inf\{t; \sup_{s\leq t} \|\widetilde{Z}_s\| \geq n\} \wedge T$$

are a nondecreasing sequence of stopping times[19] such that $\mathbb{Q}(\cup_n\{\tau_n = T\}) = 1$. In fact, setting $\chi = \sup_{t\in[0,T]} \|\widetilde{Z}_t\|$ (which is finite for $\widetilde{Z}$ càdlàg), we have

$$\Big\{\chi < \infty\Big\} = \cup_{\nearrow n\infty}\Big\{\chi < n\Big\},$$

hence $\mathbb{Q}(\{\chi < n\}) \nearrow_{n\infty} 1$ and

$$\mathbb{Q}(\{\tau_n < T\}) \leq \mathbb{Q}(\chi \geq n) \searrow_{n\infty} 0.$$

Therefore $\mathbb{Q}(\cap_n\{\tau_n < T\}) = 0$ and $\mathbb{Q}(\cup_n\{\tau_n = T\}) = 1$, as claimed. Moreover, on $\{\tau_n = T\}$, we have $\|Z\| < n$ on $[0,T]$, hence $\int_0^T \|Z_s\|^2 ds \leq n^2 T < \infty$. In conclusion, $\int_0^T \|Z_s\|^2 ds = \int_0^T \|\widetilde{Z}_s\|^2 ds$ is almost surely finite. ∎

In the case of càglàd integrands Z, the Itô stochastic integration theory can be extended to integration against an arbitrary local martingale M[20], e.g. a compensated Poisson process $M = N - \lambda t$ where, given a Poisson process N of intensity λ with successive jump times denoted by T_l, we have, for any integrand[21] Z:

$$\int_0^\cdot Z_t dN_t := \sum_{T_l\leq\cdot} Z_{T_l}. \tag{3}$$

In case of a Brownian motion $M = W$, the following result is a consequence of Lemmas 8 and 9. We only additionally prove it below in the case of a compensated Poisson process $M = N - \lambda t$.

[14] i.e. such that $Z|_{\Omega\times[0,t]}$ is $\mathfrak{F}_t \times \mathcal{B}([0,t])$ measurable for each $t \geq 0$ (He, Wang, and Yan, 1992, Definition 3.10 page 86).
[15] by Lemma 1(ii).
[16] as $\cup_n\{\tau_n = T\}$ contains the set $\{\int_0^T \|Z_s\|^2 ds < +\infty\}$, which is of $\mathbb{Q}$ probability measure 1.
[17] for progressive integrands ζ such that $\mathbb{E}\int_0^T \zeta_t^2 dt < +\infty$, see e.g. Lamberton and Lapeyre (1996, Section 3.4).
[18] see He, Wang, and Yan (1992, Theorem 3.11 page 86).
[19] by Lemma 1(iii).
[20] see He, Wang, and Yan (1992, Definition 9.1 page 227) or Protter (2004).
[21] adapted, for adaptedness of the resulting integral.

Theorem 1 [22] *Given any local martingale integrator M and càglàd process Z, the stochastic integral $Y = \int_0^\cdot Z_s dM_s$ is a well defined local martingale.*

Proof (only in the case of a compensated Poisson process $M = N - \lambda t$). For Z nonnegative bounded and any piecewise-constant approximation $Z^n = \sum \zeta_{i-1} \mathbb{1}_{(t_{i-1},t_i]}$ of Z such that $t_i = \frac{iT}{n}$ and $\zeta_{i-1} := Z_{t_{i-1}}$, setting $t_i^\vartheta = t_i \wedge \vartheta$ for any $[0,T]$ valued stopping time ϑ and $i = 1 \dots n$, we have

$$\mathbb{E}\int_0^\vartheta Z_t^n dM_t = \mathbb{E}\int_0^T \sum_{i=1}^n \zeta_{i-1}(M_{t_i^\vartheta} - M_{t_{i-1}^\vartheta}) = \int_0^T \mathbb{E}\sum_{i=1}^n \zeta_{i-1}\mathbb{E}_{t_{i-1}^\vartheta}(M_{t_i^\vartheta} - M_{t_{i-1}^\vartheta}) = 0, \tag{4}$$

by the tower rule and the martingale property of M (i.e. the $\mathbb{E}_s(N_t - N_s) = \lambda(t-s)$ property of the Poisson process N, for any constants $s \le t$). Hence

$$\mathbb{E}\int_0^\vartheta Z_t^n dN_t = \mathbb{E}\int_0^\vartheta Z_t^n \lambda dt. \tag{5}$$

Calling C the upper bound of Z, we have that $\int_0^\vartheta Z_t^n \lambda_t dt \le C\lambda T$ and $\int_0^\vartheta Z_t^n dN_t \le C N_T$ with $\mathbb{E}N_T = \lambda T < \infty$. Therefore, we may apply the dominated convergence theorem to both sides of (5) to conclude that $\mathbb{E}\int_0^\vartheta Z_t dM_t = 0$. Hence the process Y is a martingale on $[0,T]$, by Lemma 3. Relaxing the assumption Z nonnegative bounded into $\mathbb{E}\int_0^T |Z_t| dt$ finite, setting $g_K(x) = \mathbb{1}_{x\le K} + \mathbb{1}_{K<x\le K+1}(K+1-x)$, for each $\varepsilon = \pm$, the integrand $Z^{\varepsilon,K} = g_K(Z^\varepsilon)Z^\varepsilon$ is càglàd and bounded, hence the process $Z_t^{\varepsilon,K} dM_t$ is a martingale as just seen. For any $[0,T]$ valued stopping time ϑ, we thus have $\mathbb{E}[\int_0^\vartheta Z_t^{\varepsilon,K} dN_t] = \mathbb{E}[\int_0^\vartheta \lambda Z_t^{\varepsilon,K} dt]$, hence $\mathbb{E}[\int_0^\vartheta Z_t^{\varepsilon,K} dN_t] = \mathbb{E}[\int_0^\vartheta \lambda Z_t^{\varepsilon,K} dt]$ follows by monotone convergence when $K \to +\infty$, for each $\varepsilon = \pm$. Therefore $\mathbb{E}\int_0^\vartheta Z_t dM_t = 0$ holds for any $[0,T]$ valued stopping time ϑ, so that the process Y is again a martingale on $[0,T]$, by Lemma 3.

By Lemma 9, any càglàd integrand Z satisfies $\int_0^T |Z_t| dt < \infty$ a.s. The

$$\tau_n = \inf\{t; \int_0^t |Z_s| ds \ge n\} \wedge T$$

are a nondecreasing sequence of stopping times[23] such that $\mathbb{Q}(\cup_n\{\tau_n = T\}) = 1$.[24]. For each n we have

$$\int_0^T \mathbb{1}_{\{s\le\tau_n\}} |Z_s| ds = \int_0^{\tau_n} |Z_s| ds \le n,$$

hence $\mathbb{E}\int_0^T \mathbb{1}_{\{s\le\tau_n\}} |Z_s| ds$ is finite so that $\int_0^\cdot \mathbb{1}_{\{s\le\tau_n\}} Z_s dMs = \int_0^{\cdot\wedge\tau_n} Z_s dMs$ is a martingale on $[0,T]$, by the first part of the proof. In conclusion $\int_0^\cdot Z_s dM_s$ is a local martingale on $[0,T]$, with τ_n as a localizing sequence of stopping times. ∎

§2 Semimartingales

A finite variation process $D = D^{[+]} - D^{[-]}$ is a difference between two adapted nondecreasing càdlàg processes $D^{[\pm]}$ starting from 0. The variation $V_t^\pi(D)$ of D on any finite partition $\pi = \{t_0, t_1, \dots, t_n\}$ of $[0,T]$ satisfies

$$V_T^\pi(D) := \sum_{i=1}^n |D_{t_i} - D_{t_{i-1}}| \le D_T^{[+]} + D_T^{[-]} < \infty. \tag{6}$$

[22] see e.g. Protter (2004, Theorem IV.29 page 173).

[23] by Lemma 1(ii).

[24] as $\cup_n\{\tau_n = T\}$ contains the set $\{\int_0^T |Z_s| ds < +\infty\}$, which is of $\mathbb{Q}$ probability measure 1.

Semimartingales are a class of integrators giving rise to a flexible theory of stochastic integration encompassing both the Itô integral with respect to the Brownian motion as presented in, for instance, Lamberton and Lapeyre (1996, Section 3.4), and pathwise Lebesgue-Stieltjes integrals (e.g. integral with respect to a Poisson process as per (3)). For detailed treatments, see e.g. Meyer (1976) and He, Wang, and Yan (1992) or, for a renewed pedagogical presentation building on Dellacherie (1980), Protter (2004). In one of these two equivalent characterizations[25]:

Definition 4 *A semimartingale X on $[0,T]$ corresponds to the sum of a finite variation process D and a local martingale M.*

A financial motivation for modeling prices of traded assets as semimartingales is that price processes outside this class give rise to arbitrages unless rather stringent conditions are imposed on the trading strategies (see Delbaen and Schachermayer (2005)).

A representation $X = D + M$ as above is called a Doob-Meyer decomposition of the semimartingale X. Such a Doob-Meyer decomposition is not generally unique. However, by virtue of[26] the following extension of Lemma 6, there is at most one such representation of a process X with D continuous[27]. One then speaks of "the canonical Doob-Meyer decomposition of a special semimartingale X".

Lemma 10 *The only finite variation continuous local martingale M is the null process.*

Proof. [28] As M is continuous and the $M^{[\pm]}$ are càdlàg, we can in fact assume the $M^{[\pm]}$ continuous, by subtracting from them their (necessarily at most countable and common) jumps in the first place. Since M has finite variation, we have

$$|M_t| \leq M_t^{[+]} + M_t^{[-]} < +\infty, \tag{7}$$

for all $t \in [0,T]$[29]. The $\tau_n := \inf\{t > 0;\ M_t^{[+]} + M_t^{[-]} \geq n\} \wedge T$ are a nondecreasing sequence of stopping times[30] such that $\mathbb{Q}(\cup_n\{\tau_n = T\}) = 1$[31]. In view of (7), for any fixed k the process $K := M_{\cdot\wedge\tau_k}$ is a bounded martingale, by application[32] of Lemma 5(ii). The rest of the proof follows verbatim the one (after (1)) of Lemma 6. ∎

The stochastic integral of a càglàd process Z[33] with respect to a semimartingale X is defined as

$$Y_t = \int_0^t Z_s dX_s := \int_0^t Z_s dD_s + \int_0^t Z_s dM_s, \tag{8}$$

where $X = D + M$ is any Doob-Meyer decomposition of X, $\int_0^t Z_s dD_s$ is a pathwise Lebesgue-Stieljes integral, and $\int_0^t Z_s dM_s$ is the stochastic integral in the sense of local martingales introduced before Theorem 1. The corresponding notion of stochastic integral is independent of the Doob-Meyer decomposition of X that is used in (8)[34].

[25] see Meyer (1976, 1 DEFINITION page 32) and Dellacherie (1980, Définition 1 page 119, Théorème 6 page 125).
[26] in fact, equivalently to.
[27] see e.g. He, Wang, and Yan (1992, Theorem 8.5 page 210).
[28] adapted from https://faculty.math.illinois.edu/ psdey/MATH562FA21/lec11.pdf accessed on 26 Nov 2021.
[29] see e.g. He, Wang, and Yan (1992, Theorem 3.44 page 101).
[30] by Lemma 1(ii).
[31] as $\cup_n\{\tau_n = T\}$ contains the set $\{M_T^{[+]} + M_T^{[-]} < +\infty\}$, which is of $\mathbb{Q}$ probability measure 1.
[32] to K and to $(-K)$.
[33] cf. Lemma 9 and He, Wang, and Yan (1992, Theorem 7.7 1) page 192).
[34] see e.g. He, Wang, and Yan (1992, Lemma 9.12 and Definition 9.13 page 234).

A Quadratic Variation

Semimartingale integration by parts formula[35]: For any semimartingales X and Y, the quadratic covariation (or bracket) of X and Y:

$$[X,Y] = XY - X_0Y_0 - \int_0^{\cdot} X_{t-}\, dY_t - \int_0^{\cdot} Y_{t-}\, dX_t \tag{9}$$

(simply denoted by $[X]$ if $X = Y$), satisfies, for each $t \le T$,

$$[X,Y]_t = \mathbb{Q}\lim_{n\to\infty} \sum_{i=1}^{\infty} \left(X_{t_i^n\wedge t} - X_{t_{i-1}^n\wedge t}\right)\left(Y_{t_i^n\wedge t} - Y_{t_{i-1}^n\wedge t}\right), \tag{10}$$

for any sequence $\pi_n = \{t_0^n, t_1^n, \ldots\}$ of finite partitions of $[0,T]$ such that the mesh size $|\pi_n| = \max_i \left(t_i^n - t_{i-1}^n\right)$ tends to zero in probability as $n \to \infty$.

An application of (9) shows that

$$[X+Y] = [X] + [Y] + 2[X,Y] \tag{11}$$

holds for any semimartingales X, Y. Another useful property is[36]

$$[\int_0^{\cdot} Z_t dX_t, Y] = \int_0^{\cdot} Z_t d[X,Y]_t, \tag{12}$$

for any semimartingales X, Y and càdlàg process Z.

Let $\Delta Y = Y - Y_-$ denote the jump process of any càdlàg process (e.g. semimartingale) Y.

Lemma 11 *Let X be a semimartingale and V be a finite variation process. Their covariation is*

$$[X,V]_t = \int_0^t \Delta X_s\, dV_s = \sum_{s\le t} \Delta X_s \Delta V_s. \tag{13}$$

In particular, if either of X or V is continuous, then $[X,V] = 0$.

Proof (from https://almostsuremath.com/2010/01/19/properties-of-quadratic-variations). Expressing the covariation as the limit along equally spaced partitions of $[0,t]$ gives

$$\begin{aligned}
[X,V]_t &= \lim_{n\to\infty} \sum_{k=1}^{n} (X_{kt/n} - X_{(k-1)t/n})(V_{kt/n} - V_{(k-1)t/n}) \\
&= \lim_{n\to\infty} \sum_{k=1}^{n} \int_0^t 1_{\{(k-1)t/n < s \le kt/n\}} (X_{kt/n} - X_{(k-1)t/n})\, dV_s \\
&= \int_0^t \lim_{n\to\infty} \sum_{k=1}^{n} 1_{\{(k-1)t/n < s \le kt/n\}} (X_{kt/n} - X_{(k-1)t/n})\, dV_s \\
&= \int_0^t \Delta X_s\, dV_s.
\end{aligned}$$

The third equality here makes use of the bounded convergence theorem to commute the limit with the integral sign. Then, as X is cadlag, on each sample path there is only a countable set of times S at which $\Delta X_s \neq 0$. Bounded convergence can again be used to evaluate the integral

$$\begin{aligned}
\int_0^t \Delta X_s\, dV_s &= \int_0^t \sum_{s\in S} \Delta X_s 1_{\{s=u\}}\, dV_u \\
&= \sum_{s\in S} \Delta X_s \int_0^t 1_{\{s=u\}}\, dV_u \\
&= \sum_{s\in S, s\le t} \Delta X_s \Delta V_s,
\end{aligned}$$

[35] see https://almostsuremath.com/2010/01/18/quadratic-variations-and-integration-by-parts/#scn_ibp_thm3.
[36] cf. He, Wang, and Yan (1992, Theorem 9.15 3) page 235).

as required. ∎

Corollary 1 *If X, Y are semimartingales and V, W are continuous finite variation processes then, $[X+V, Y+W] = [X, Y]$.*

That is, when calculating covariations, we can disregard any continuous finite variation terms added to the processes.

Example 1 *Given a Poisson process N, a time-integrable process U and a process V, let*[37]

$$dX_t = U_t\, dt + V_t dN_t. \tag{14}$$

Then

$$[X] = \int_0^{\cdot} V_t^2 dN_t. \tag{15}$$

Proof. By application of Corollary 1 and (9) applied to $Y = X$,

$$[X] = \Big[\int_0^{\cdot} V_t dN_t\Big] = \Big(\int_0^{\cdot} V_t dN_t\Big)^2 - 2\int_0^{\cdot}\Big(\int_0^{\cdot} V_s dN_s\Big)_{t-} V_t dN_t = (\sum_{T_l\le\cdot} V_{T_l})^2 - 2\sum_{T_k<T_l\le\cdot} V_{T_k}V_{T_l} = \sum_{T_l\le\cdot} V_{T_l}^2,$$

which yields (15). ∎

By application of Corollary 1 and (9) again:

Corollary 2 *The following* ***elementary integration by parts formula***[38] *holds, for any semimartingale X and $\beta = e^{-\int_0^{\cdot} r_s ds}$ (with r time-integrable):*

$$d(\beta_t X_t) = \beta_t(dX_t - r_t X_t dt). \tag{16}$$

We also state the Lévy's characterization of Brownian motion[39] (for one-dimension):

Theorem 2 *Let X be a local martingale with $X_0 = 0$. Then, the following are equivalent:*
(i) *X is a standard Brownian motion.*
(ii) *X is continuous and $X_t^2 - t$ is a local martingale.*
(iii) *X is continuous and $\langle X\rangle_t = t$, $t \ge 0$*[40].

Proof of (iii)⇒(i)[41]. Let $M_t = e^{i\lambda X_t + \frac{1}{2}\lambda^2 t}$, where $i^2 = -1$. By using Itô's formula, we obtain that for $s \le t$, $M_t = M_s + i\lambda\int_s^t M_u dX_u$. Hence the process M is a local martingale, by Theorem 1. As it is also bounded, it is a martingale, by Lemma 4, and the above equality yields $\mathbb{E}_s e^{i\lambda(X_t - X_s)} = e^{-\frac{1}{2}\lambda^2(t-s)}$. The process $(X_t)_{t\ge 0}$ is therefore a continuous process with stationary and independent increments such that X_t is normally distributed with mean 0 and variance t. It is thus a Brownian motion. ∎

From https://almostsuremath.com/2010/03/29/quadratic-variations-and-the-ito-isometry/:

[37] with $\int_0^{\cdot} V_t dN_t = \sum_{T_l\le\cdot} V_{T_l}$, cf. (3).
[38] cf. He, Wang, and Yan (1992, Theorem 9.33 page 243, Definition 8.2 page 209, and Theorem 7.25 page 198).
[39] see He, Wang, and Yan (1992, Corollary 11.39 and its proof pages 314-315).
[40] where, for any continuous semimartingale X, $\langle X\rangle_t = X^2 - 2\int_0^{\cdot} X_t dX_t$ satisfies the Itô formula $du(t, X_t) = \partial_t u(t, X_t)dt + \partial_x u(t, X_t)dX_t + \frac{1}{2}\partial_{x^2}^2 u(t, X_t)d\langle X\rangle_t$, for any function u of class $\mathcal{C}^{1,2}$ with respect to (t, x).
[41] from https://fabricebaudoin.wordpress.com/2012/09/27/lecture-23-time-changed-martingales-and-planar-brownian-motion.

Lemma 12 *If X and Y are local martingales, then $XY - [X,Y]$ is a local martingale. In particular, $X^2 - [X]$ is a local martingale for all local martingales X.*

Proof. Integration by parts gives

$$XY - [X,Y] = X_0Y_0 + \int_0^{\cdot} X_{t-}\, dY_t + \int_0^{\cdot} Y_{t-}\, dX_t,$$

which, by Theorem 1, is a local martingale. ∎

For square integrable martingales[42], it is possible to drop the "local" from the statement above. That is, $X^2 - [X]$ will be a true martingale. Before proving this, the following inequality[43] will be useful. Let $X_t^* \equiv \sup_{s\le t} |X_s|$ denote the maximum process of X.

First we recall the following (classical and not too difficult to show[44]) Doob martingale inequality.

Lemma 13 *Let X be a (cadlag) martingale and $t > 0$. Then*

$$\mathbb{E}\big((X_t^*)^2\big) \le 4\mathbb{E}(X_t^2)(\le +\infty). \tag{17}$$

Lemma 14 *Let X be a local martingale $X_0 = 0$. Then,*

$$\mathbb{E}\left([X]_t\right) \le \mathbb{E}\left((X_t^*)^2\right) \le 4\mathbb{E}\left([X]_t\right)(\le +\infty). \tag{18}$$

Proof. By Lemma 12 there exist a localizing sequence of stopping times τ_n such that the stopped processes $(X^2 - [X])^{\tau_n}$ and X^{τ_n} are martingales. So, $\mathbb{E}([X]_{t\wedge\tau_n}) = \mathbb{E}(X^2_{t\wedge\tau_n})$. Also, $\mathbb{E}[(X^*_{t\wedge\tau_n})^2] \le 4\mathbb{E}([X^2_{t\wedge\tau_n}])$ by Doob's inequality (17), giving

$$\mathbb{E}\left([X]_{t\wedge\tau_n}\right) = \mathbb{E}[X^2_{t\wedge\tau_n}] \le \mathbb{E}\left((X^*_{t\wedge\tau_n})^2\right) \le 4\mathbb{E}\left(X^2_{t\wedge\tau_n}\right) = 4\mathbb{E}\left([X]_{t\wedge\tau_n}\right).$$

Letting n increase to infinity and applying monotone convergence to $[X]_{t\wedge\tau_n}$ and $(X^*_{t\wedge\tau_n})^2$ gives (18). ∎

We can now prove that for square integrable martingales X, $X^2 - [X]$ is a true martingale. We also obtain a necessary and sufficient condition for any local martingale to be a square integrable martingale.

Lemma 15 *A local martingale X is a square integrable martingale if and only if $\mathbb{E}[X_0^2] < \infty$ and $[X]$ is integrable, in which case $X^2 - [X]$ is a martingale. In particular, for any square integrable martingale X starting from 0,*

$$\mathbb{E}([X].) = \mathbb{E}(X_.^2). \tag{19}$$

Proof. Replacing X by $X - X_0$, we may suppose that $X_0 = 0$. If $[X]$ is integrable, then Lemma 14 gives $\mathbb{E}((X_.^*)^2) < +\infty$, so $X^2 - [X]$ is a local martingale (by Lemma 12) dominated (i.e. in absolute value) by the integrable random variable $(X_T^*)^2 + [X]_T$ (for $t \le T$). Hence it is a true martingale, by Lemma 5(ii). Therefore X is a square integrable martingale. Conversely, supposing the latter, then Doob's inequality (17) shows that $\mathbb{E}[(X_t^*)^2] \le 4\mathbb{E}[X_t^2]$ is finite and, by Lemma 14, $[X]$ is integrable. ∎

[42] in the sense that $\mathbb{E}X_t^2 < +\infty$ for all t.

[43] a special case of the much more general Burkholder-Davis-Gundy inequalities, which are beyond the level of these notes.

[44] for a proof see https://almostsuremath.com/2009/12/21/martingale-inequalities.

§3 Itô and Markov Processes

Let there be given an $\mathbb{R}^d$ valued drift coefficient b, and $\mathbb{R}^{d\times d}$ diffusion coefficient σ, and an $\mathbb{R}^d$ valued jump size process δ, given as càdlàg processes (hence progressive and such that $\int_0^T(|b_t|+|\delta_t|+|\sigma|_t^2)dt < +\infty$ a.s., by Lemma 9). We consider an Itô process in the sense of a d-variate process X obeying the following dynamics[45]: $X_0 = x$ and, for $t \geq 0$,

$$dX_t = b_t\,dt + \sigma_t\,dW_t + \delta_{t-}dN_t, \tag{20}$$

for some d-variate standard Brownian motion W and a Poisson process N with intensity λ.

Lemma 16 *The local martingale*

$$\mu = \int_0^{\cdot}(\delta_{t-}dN_t - \lambda\delta_t dt) \tag{21}$$

is
(i) *a martingale if* $\mathbb{E}\int_0^T|\delta_t|dt < +\infty$, *in which case*

$$\mathbb{E}\int_0^T \delta_{t-}dN_t = \lambda\mathbb{E}\int_0^T \delta_t dt; \tag{22}$$

(ii) *a square integrable martingale if* $\mathbb{E}\int_0^T \delta_t^2 dt < +\infty$, *in which case*

$$\mathbb{V}\mathrm{ar}(\mu_T) = \mathbb{E}(\mu_T^2) = \lambda\mathbb{E}\int_0^T \delta_t^2 dt. \tag{23}$$

Proof. Part (i) was already established in the proof of Theorem 1. Moreover, we have $[\mu] = \int_0^{\cdot}\delta_{t-}^2 dN_t$, by (15). Hence, assuming $\mathbb{E}\int_0^T \delta_t^2 dt$ finite,

$$\mathbb{E}[\mu]_T = \mathbb{E}\int_0^T \delta_{t-}^2 dN_t = \lambda\mathbb{E}\int_0^T \delta_t^2 dt, \tag{24}$$

by part (i) applied to δ^2 here in the role of δ there. Thus μ is a square integrable martingale, by Lemma 15, so that

$$\mathbb{V}\mathrm{ar}(\mu_T) = \mathbb{E}(\mu_T^2) = \mathbb{E}[\mu]_T = \lambda\mathbb{E}\int_0^T \delta_t^2 dt,$$

by (19) and (24). ∎

The corresponding Markov (jump-diffusion) setup is when

$$b_t = b(t, X_t)\,,\ \ \sigma_t = \sigma(t, X_t)\,,\ \ \delta_{t-} = \delta(t, X_{t-}) \tag{25}$$

for continuous *functions* $b(\cdot,\cdot), \sigma(\cdot,\cdot), \delta(\cdot,\cdot)$, so that (20) is in effect a stochastic differential equation (forward SDE). This SDE has a unique solution[46] X provided the coefficients b and σ are Lipschitz with linear growth in x, uniformly in t[47]. A notable feature of the solution is the so-called ***Markov property***, i.e.[48]

$$\mathbb{E}_t(\Phi(X_s, s \in [t,T]) = \mathbb{E}(\Phi(X_s, s \in [t,T])\,|\,X_t) \tag{26}$$

[45] cf. (3).
[46] strong solution, i.e. for a priori given driving processes W and N.
[47] this can be shown iteratively on larger and larger time intervals $[0, T_l]$, where the T_l are the increasing jump times of N, by using Karatzas and Shreve (1991, Theorems 2.5 page 287 and 2.9 page 289) between jumps; see also Élie (2006, Section II.1.5 page 125) for direct computations with jumps.
[48] see Protter (2004, theorems I.45 page 35 and V.32 page 300).

holds for every functional Φ of X that makes sense on both sides of the equality. Thus the past of X doesn't influence its future; the present of X provides all the relevant information.

Given a real valued and $\mathcal{C}^{1,2}$ function $u = u(t,x)$, an application of the diffusive Itô formula between jumps combined with a direct inspection of the impact of the jumps yield the Itô formula with jumps (of finite intensity λ) : for any $t \in [0,T]$,

$$\begin{aligned} du(t,X_t) = \\ (\partial_t + \mathcal{A}_x)\, u(t,X_t)dt + \partial_x u(t,X_t)\sigma_t\, dW_t + \delta u(t,X_{t-})dM_t, \end{aligned} \tag{27}$$

where $\partial_x u$ is the row gradient of u with respect to x, where in the compensated jump local martingale of the last term

$$\delta u(t,x) = u(t,x+\delta(t,x)) - u(t,x),$$

and where the infinitesimal generator $\mathcal{A}_x$ of X acts on u as

$$(\mathcal{A}_x u)(t,x) = \partial_x u(t,x)b(t,x) + \frac{1}{2}\mathrm{tr}\left(\partial^2_{x^2} u(t,x)a(t,x)\right) + \lambda\delta u(t,x), \tag{28}$$

with $a = \sigma\sigma^\top$. In addition, for all $\mathcal{C}^{1,2}$ functions u and v of (t,x) such that

$$\mathbb{E}\int_0^T \left(\partial u(t,X_t)a_t(\partial v(t,X_t))^\top\right)^2 dt + \lambda\mathbb{E}\int_0^T \left(\delta u(t,X_t)\delta v(t,X_t)\right)^2 dt < \infty,$$

we have with $Y_t = u(t,X_t)$ and $Z_t = v(t,X_t)$ the following "carré du champ" formula:

$$\begin{aligned} \frac{d\langle Y,Z\rangle_t}{dt} &= \partial u(t,X_t)a_t(\partial v(t,X_t))^\top + \lambda\delta u(t,X_t)\delta v(t,X_t) \\ &= \left(\mathcal{A}_x(uv) - u\mathcal{A}_x v - v\mathcal{A}_x u\right)(t,X_t), \end{aligned} \tag{29}$$

where

$$\frac{d\langle Y,Z\rangle_t}{dt} = \lim_{h\to 0} h^{-1}\mathbb{C}\mathrm{ov}(Y_{t+h} - Y_t, Z_{t+h} - Z_t|\mathcal{F}_t). \tag{30}$$

A Extensions

Let now $\delta = \delta_t(y)$ represent an $\mathbb{R}^d$ valued càdlàg process parameterized by $y \in \mathbb{R}^d$ and let $J_{(t)}$ denote a family of i.i.d. d-variate random variables with distribution denoted by $w(dy)$, with each $J_{(T_l)}$ independent from $\mathfrak{F}_{T_l-}$[49], where the T_l denote the increasing jump times of N. We can extend the jump setup in the above to a compound Poisson kind of term $\delta_{t-}(J_{(t)})$ in (20). Let $\bar\delta_t = \int_{\mathbb{R}^d}\delta_t(y)w(dy)$ and $\bar\delta^2_t = \int_{\mathbb{R}^d}\delta^2_t(y)w(dy)$ (with $\bar\delta$ and $\bar\delta^2$ assumed well defined and càdlàg).

Lemma 16 holds mutatis mutandis in the thus-extended jump setup, just replacing δ_{t-} by $\delta_{t-}(J_{(t)})$, $\delta_t dt$ by $\bar\delta_t dt$, $|\delta_t|dt$ by $|\bar\delta_t|dt$ and $\delta^2_t dt$ by $\bar\delta^2_t dt$ everywhere. Namely:

Lemma 17 *The local martingale*

$$\mu = \int_0^\cdot (\delta_{t-}(J_{(t)})dN_t - \lambda\bar\delta_t dt) \tag{31}$$

is
(i) *a martingale if* $\mathbb{E}\int_0^T |\bar\delta_t|dt < +\infty$, *in which case*

$$\mathbb{E}\int_0^T \delta_{t-}(J_{(t)})dN_t = \lambda\mathbb{E}\int_0^T \bar\delta_t dt; \tag{32}$$

(ii) *a square integrable martingale if* $\mathbb{E}\int_0^T \bar\delta^2_t dt < +\infty$, *in which case*

$$\mathbb{V}\mathrm{ar}(\mu_T) = \mathbb{E}(\mu^2_T) = \lambda\mathbb{E}\int_0^T \bar\delta^2_t dt. \tag{33}$$

[49] where, for any stopping time ϑ, $\mathcal{F}_{\vartheta-} := \mathcal{F}_0 \vee \sigma\{A \cap \{t \le \vartheta\} : A \in \mathfrak{F}_t, t \in \mathbb{R}_+\}$, see He, Wang, and Yan (1992, Eq. (3.3) page 80).

Proof. [50] Since the process $\int_0^\cdot |\bar\delta_t| dt$ is continuous, there exists a nondecreasing sequence of $[0,T]$ stopping times $(\tau_n)_{n\in\mathbb{N}}$ such that $\mathbb{Q}(\cup_n\{\tau_n = T\}) = 1$ and $\int_0^{\cdot\wedge\tau_n} \bar\delta_t dt$ is bounded[51]. For any $[0,T]$ valued stopping time ϑ, we can then write, with $\vartheta_n = \vartheta\wedge\tau_n$ (using the independence between $J_{(T_l)}$ and $\mathfrak{F}_{T_l-}$ to pass to the second line),

$$\begin{aligned}\mathbb{E}[\int_0^{\vartheta_n} \delta_{t-}(J_{(t)})dN_t] &= \mathbb{E}[\sum_{T_l\le\vartheta_n} \delta_{T_l-}(J_{(T_l)})] = \mathbb{E}[\sum_{T_l\le\vartheta_n} \mathbb{E}_{T_l-}\delta_{T_l-}(J_{(T_l)})]\\ &= \mathbb{E}[\sum_{T_l\le\vartheta_n} \bar\delta_{T_l-}] = \mathbb{E}[\int_0^{\vartheta_n} \bar\delta_{t-}dN_t] = \mathbb{E}[\int_0^{\vartheta_n} \lambda\bar\delta_t dt],\end{aligned} \tag{34}$$

by Lemma 16(i). Therefore, by Lemma 3, the process (31) stopped at τ_n is a martingale on $[0,T]$, for each n. Hence, the process μ in (31) is a local martingale on $[0,T]$, and a martingale with

$$\mathbb{E}\int_0^T \delta_{t-}(J_{(t)})dN_t = \mathbb{E}\int_0^T \lambda\bar\delta_t dt$$

if $\mathbb{E}\int_0^T |\bar\delta_t| dt < \infty$, so that the computation (34) is in fact valid for ϑ_n replaced by ϑ itself, without localization by the τ_n.

Moreover, we have

$$[\mu] = \int_0^\cdot \delta^2_{t-}(J_{(t)})dN_t, \tag{35}$$

by (15) applied to $X = \mu$ as per (31). Hence, if $\mathbb{E}\int_0^T \lambda\bar\delta_t^2 dt < +\infty$, then, by Lemma 16(ii) applied to $\bar\delta^2$ here in the role of δ there (and using the independence between $J_{(T_l)}$ and $\mathfrak{F}_{T_l-}$ to pass to the second line),

$$\begin{aligned}+\infty > \mathbb{E}\int_0^T \lambda\bar\delta_t^2 dt = \mathbb{E}\int_0^T \bar\delta^2_{t-}dN_t = \mathbb{E}\sum_{T_l\le T} \bar\delta^2_{T_l-} =\\ \mathbb{E}\sum_{T_l\le T} \mathbb{E}_{T_l-}\delta^2_{T_l-}(J_{(t)}) = \mathbb{E}\sum_{T_l\le T} \delta^2_{T_l-}(J_{(t)}) = \mathbb{E}\int_0^T \delta^2_{t-}(J_{(t)})dN_t = \mathbb{E}[\mu]_T,\end{aligned} \tag{36}$$

by (35). Thus μ is a square integrable martingale, by Lemma 15, and (19) then yields

$$\mathbb{V}\mathrm{ar}(\mu_T) = \mathbb{E}(\mu_T^2) = \mathbb{E}([\mu]_T) = \mathbb{E}\int_0^T \lambda\bar\delta_t^2 dt,$$

by (36). ∎

Proposition 1 *For any process δ as in Lemma 17, with related processes μ as per* (31) *and $\bar\delta$ in Lemma 17 such that $\mathbb{E}\int_0^T \bar\delta_t^2 dt < +\infty$, and càdlàg process σ such that $\mathbb{E}\int_0^T \sigma_t^2 dt < +\infty$, we have*

$$\begin{aligned}\mathbb{V}\mathrm{ar}\big(\int_0^T \sigma_t\, dW_t + \mu_T\big) = \mathbb{E}\Big(\big(\int_0^T \sigma_t\, dW_t + \mu_T\big)^2\Big) =\\ \mathbb{E}\int_0^T (\sigma_t^2 + \lambda\bar\delta_t^2)dt.\end{aligned} \tag{37}$$

Proof. Under the stated assumptions, the processes $\int_0^\cdot \sigma_t\, dW_t$ (cf. Lemma 9 and the last footnote in the proof of Lemma 8) and μ (by Lemma 17(ii)) are martingales, which implies the first identity in (37). Moreover,

$$\mathbb{E}\Big(\big(\int_0^T \sigma_t\, dW_t + \mu_T\big)^2\Big) = \mathbb{E}\Big(\big(\int_0^T \sigma_t\, dW_t\big)^2\Big) + \mathbb{E}(\mu_T^2) + 2\mathbb{E}\big(\mu_T \int_0^T \sigma_t\, dW_t\big), \tag{38}$$

[50] see also Lamberton and Lapeyre (1996, Lemma 7.3.3 p.179) for a longer but elementary proof (without semimartingale Itô calculus).

[51] cf. He, Wang, and Yan (1992, Theorem 7.7 3) p.192).

where the Itô isometry (cf. also again Lemma 9) yields $\mathbb{E}\big(\int_0^T \sigma_t\, dW_t\big)^2 = \mathbb{E}\big(\int_0^T \sigma_t^2 dt\big)$, whereas $\mathbb{E}(\mu_T^2) = \lambda\mathbb{E}\int_0^T \bar{\delta}_t^2 dt$ holds by Lemma 17(ii). It remains to show that the last term vanishes in (38). By Lemma 11 (case where X is continuous), $[\mu, \int_0^\cdot \sigma_t\, dW_t] = 0$. The integration by parts formula (9), Lemmas 8-9 and 17(i) then show that the process $\mu\int_0^\cdot \sigma_t\, dW_t$ is a local martingale. Moreover,

$$\mathbb{E}\sup_{[0,T]}\Big|\mu\int_0^\cdot \sigma_t\, dW_t\Big| \le \mathbb{E}\Big((\sup_{[0,T]}|\mu|)\big(\sup_{[0,T]}\Big|\int_0^\cdot \sigma_t\, dW_t\Big)\Big) \le$$
$$\Big(\mathbb{E}\big(\sup_{[0,T]}\mu^2\big)\Big)^{\frac{1}{2}}\Big(\mathbb{E}\big(\sup_{[0,T]}(\int_0^\cdot \sigma_t\, dW_t)^2\big)\Big)^{\frac{1}{2}} \le 4\Big(\mathbb{E}\big([\mu]_T\big)\Big)^{\frac{1}{2}}\Big(\mathbb{E}\big(\big[\int_0^\cdot \sigma_t\, dW_t\big]_T\big)\Big)^{\frac{1}{2}},$$

by Lemma 14. But the expression in the right-hand side is finite, by Lemma 17(ii) and 15. Hence the local martingale $\mu\int_0^\cdot \sigma_t\, dW_t$ is a martingale, by Lemma 4, so that $\mathbb{E}(\mu_T\int_0^T \sigma_t\, dW_t)$ in (38) vanishes, by Lemma 3 applied for $\vartheta = 0$ and T. ∎

The corresponding Markovian specification is $\delta_{t-}(J_{(t)}) = \delta(t, X_{t-}, J_{(t)})$ instead of $\delta_{t-} = \delta(t, X_{t-})$ in (25), for a function $\delta = \delta(t, x, y)$ depending on an additional d-variate argument y. Then the compensated jump local martingale (last term) in (27) becomes of the compensated compound Poisson type

$$\delta u(t, X_{t-}, J_{(t)})dN_t - \lambda\bar{\delta}u(t, X_t)dt, \tag{39}$$

where

$$\begin{aligned}\delta u(t,x,y) &= u(t, x+\delta(t,x,y)) - u(t,x),\\ \bar{\delta}u(t,x) &= \int_{\mathbb{R}^d}\delta u(t,x,y)w(dy).\end{aligned} \tag{40}$$

The nonlocal (last) terms in (28) and (29) become

$$\lambda\bar{\delta}u(t,x) \text{ and } \lambda\int_{\mathbb{R}^d}\delta u(t,X_t,y)\delta v(t,X_t,y)w(dy). \tag{41}$$

Moreover, by Girsanov transform based measure change, one can extend further the above to models with a random intensity λ_t and/or a random distribution $w_t(dy)$ of jumps or, in a Markovian specification, $\lambda(t, X_t)$ and/or $w(t, X_{t-}, dy)$[52]. This allows designing models with dependent driving noises W and N, where the latter is a point process (increasing by one at increasing random times) with intensity *process* λ.

Further extension outside the scope of these notes would be to "infinite intensity" jumps[53].

§4 Fourier and Laplace Transform Formulas

The Fourier transform $\mathcal{F}f$ of an absolutely integrable function f from $\mathbb{R}$ to itself is defined, for every $z \in \mathbb{R}$, by (with $i^2 = -1$)

$$\mathcal{F}f(z) = \int_{-\infty}^{\infty} e^{izx} f(x)dx. \tag{42}$$

By integration by parts, the differentiation operator translates into multiplication by iz in the Fourier space, so $\mathcal{F}f'(z) = (-iz)\mathcal{F}f(z)$. The inverse Fourier transform formula[54] states that, for $x \in \mathbb{R}$,

$$f(x) = \tfrac{1}{2\pi}\int_{-\infty}^{\infty} e^{-izx}\mathcal{F}f(z)dz. \tag{43}$$

[52] see e.g. Crépey, Bielecki, and Brigo (2014, Section 13.6) and Crépey (2013, Section 12.3.2).
[53] see e.g. Crépey, Bielecki, and Brigo (2014, Sections 13.2-13.3) or Cont and Tankov (2003b).
[54] see e.g. Rudin (1987, Theorem 9.11 page 185).

The Fourier transform $\mathcal{F}f$ may be extended to complex values of its argument, resulting in the so-called complex Fourier transform[55] of f, for z in suitable strips of analyticity of $\mathcal{F}f$ parallel to the real axis. In particular, for sufficiently small $\eta \geq 0$,

$$f(x) = \tfrac{1}{2\pi} \int_{-\eta i-\infty}^{-\eta i+\infty} e^{-izx} \mathcal{F}f(z) dz. \tag{44}$$

Proposition 2 *In case f represents the density of the law of a real random variable X, the Fourier transform of f coincides with the characteristic function $\Phi(z) = \mathbb{E}[\exp(izX)]$ of X and, for any $k \in \mathbb{R}$,*

$$\mathbb{Q}\Big(X > k\Big) = \frac{1}{2} + \frac{1}{\pi} \int_0^\infty \Re e\left[\frac{ie^{-izk}\Phi(z)}{iz}\right] dz. \tag{45}$$

Proof (Sketched). (44) implies for $\eta > 0$ that

$$\begin{aligned} G(k) = \mathbb{Q}\Big(X > k\Big) = \int_k^\infty f(y)dy = \\ \frac{1}{2\pi} \int_{-\eta i-\infty}^{-\eta i+\infty} \Phi(z) \left(\int_k^\infty e^{-izy} dy \right) dz = \frac{1}{2\pi i} \int_{-\eta i-\infty}^{-\eta i+\infty} \frac{e^{-izk}}{z} \Phi(z)\, dz. \end{aligned} \tag{46}$$

Letting $\eta \to 0+$, one can then show, by application of the Cauchy residue formula (see Remark 1 below), that

$$G(k) = \frac{1}{2} + \frac{1}{2\pi} \lim_{\varepsilon \to 0+} \int_{z \in (-\infty, -\varepsilon) \cup (\varepsilon, \infty)} \frac{e^{-izk}\Phi(z)}{iz}\, dz. \tag{47}$$

Since, for $z \in \mathbb{R}$, $\frac{e^{izk}\Phi(-z)}{-iz}$ is the complex conjugate of $\frac{e^{-izk}\Phi(z)}{iz}$, we have

$$\lim_{\varepsilon \to 0+} \int_{z \in (-\infty, -\varepsilon) \cup (\varepsilon, \infty)} \frac{e^{-izk}\Phi(z)}{iz}\, dz = 2 \lim_{\varepsilon \to 0+} \int_\varepsilon^\infty \Re e\left[\frac{e^{-izk}\Phi(z)}{iz}\right] dz,$$

which leads to the formula (45) for $G(k)$. ∎

Remark 1 *Here is a sketched proof of* (47)*. On the one hand, the Cauchy residue formula*[56] *yields, for each $\eta > 0$,*

$$\int_{-\eta i-\infty}^{-\eta i+\infty} \frac{e^{-izk}}{z} \Phi(z)\, dz - \int_{\eta i-\infty}^{\eta i+\infty} \frac{e^{-izk}}{z} \Phi(z)\, dz = 2\pi i. \tag{48}$$

On the other hand, the integral in the sense of principal values in the second line below is such that

$$\begin{aligned} \lim_{\eta \to 0+} \left(\int_{-\eta i-\infty}^{-\eta i+\infty} \frac{e^{-izk}}{z} \Phi(z)\, dz + \int_{\eta i-\infty}^{\eta i+\infty} \frac{e^{-izk}}{z} \Phi(z)\, dz \right) \\ = 2\ p.v. \int_{\mathbb{R}} \frac{e^{-izk}\Phi(z)}{z}\, dz = 2 \lim_{\varepsilon \to 0+} \int_{z \in (-\infty, -\varepsilon) \cup (\varepsilon, \infty)} \frac{e^{-izk}\Phi(z)}{z}\, dz. \end{aligned} \tag{49}$$

As the integrals in (48)-(49) *do in fact not depend on η, the formula* (47) *for $G(k) = \frac{1}{2\pi i} \int_{-\eta i-\infty}^{-\eta i+\infty} \frac{e^{-izk}}{z} \Phi(z)\, dz$*[57] *follows by summation between* (48) *and* (49)*.*

[55] see e.g. Rudin (1987, Chapter 19).
[56] see e.g. Rudin (1987, Theorem 10.42 page 224).
[57] by (46).

A Affine Diffusions

Let $\Phi_{y,z}$ and $\Psi_{y,z}$ satisfy the following Riccati system of ODEs, parameterized by real numbers $a, c, y, z \geq 0 : \Phi_{y,z}(0) = y, \Psi_{y,z}(0) = 0$ and, for $t \geq 0$,

$$\Phi_{y,z}(0) = y, \Psi_{y,z}(0) = 0 \text{ and, for } t \geq 0, \begin{cases} \dot{\Phi}_{y,z}(t) = -\frac{c^2}{2}\Phi^2_{y,z}(t) - a\Phi_{y,z}(t) + z \\ \dot{\Psi}_{y,z}(t) = a\Phi_{y,z}(t). \end{cases} \tag{50}$$

By the classical method already result used to solve I.(79):

Lemma 18 [58] *The above ODEs are solved explicitly as*

$$\begin{aligned} \Phi_{y,z}(t) &= \frac{y\big(\gamma + a + e^{\gamma t}(\gamma - a)\big) + 2z(e^{\gamma t} - 1)}{c^2 y(e^{\gamma t} - 1) + \gamma - a + e^{\gamma t}(\gamma + a)}, \\ \Psi_{y,z}(t) &= -\frac{2a}{c^2}\log\Big(\frac{2\gamma e^{\frac{t(\gamma+a)}{2}}}{c^2 y(e^{\gamma t} - 1) + \gamma - a + e^{\gamma t}(\gamma + a)}\Big), \end{aligned} \tag{51}$$

where $\gamma = \sqrt{a^2 + 2c^2 z}$.

Let X be an extended CIR process such that

$$dX_t = a(b(t) - X_t)dt + c\sqrt{X_t}dW_t, \tag{52}$$

where a and c are nonnegative constants as before and $b(\cdot)$ is a nonnegative and càdlàg function. We recall from Remark I.8 that, for b constant, (52) has a unique nonnegative strong solution, which is is therefore also the case for b piecewise constant.

The following (time-inhomogenous) affine Laplace transform formulas yield convenient pricing (and Greeking formulas) for equity derivatives in affine stochatic volatility models[59], bond and bond derivatives in affine short interest rate models [60], and credit derivatives in reduced-form affine default intensity models:

Proposition 3 *For any $s \geq t$ and $y, z \geq 0$, we have:*

$$\mathbb{E}\Big(e^{-yX_s - z\int_t^s X_u du}|X_t = x\Big) = e^{-J(t,x;s,y,z)}, \tag{53}$$

where

$$J(t, x; s, y, z) = x\Phi_{y,z}(s - t) + a\int_t^s \Phi_{y,z}(s - u)b(u)du. \tag{54}$$

Also, we have:

$$\begin{aligned} &\mathbb{E}\Big(X_s|X_t = x\Big) = e^{-J(t,x;s,0,0)}\partial_y J(t, x; s, 0, 0), \\ &\mathbb{E}\Big(\int_t^s X_u du|X_t = x\Big) = e^{-J(t,x;s,0,0)}\partial_z J(t, x; s, 0, 0), \\ &\mathbb{E}\Big(X_s e^{-\int_t^s X_u du}|X_t = x\Big) = e^{-J(t,x;s,0,1)}\partial_s J(t, x; s, 0, 1), \end{aligned} \tag{55}$$

where the function $\dot{\Phi}_y$ that is implicit in $\partial_s J$ in (55) can be computed explicitly via the first line in (51).

[58] see Lamberton and Lapeyre (1996, Proposition 6.2.4 page 162) or Jeanblanc, Yor, and Chesney (2009, Proposition 6.3.4.1 page 361).

[59] cf. I.§3.D and the references there

[60] see (Filipovic, 2009).

If $b(\cdot)$ is piecewise-constant, such that $b(t) = b_k$ on every interval $[T_{k-1}, T_k)$ of a time-grid (T_k), and if $i \leq j$, such that $t \in [T_{i-1}, T_i)$ and $s \in [T_{j-1}, T_j)$, the second term in (54) reduces to

$$a\int_t^s \Phi_y(s-u)b(u)du = (\Psi_y(s-t) - \Psi_y(s-T_i))\, b_i + \sum_{k=i+1}^{j-1} (\Psi_y(s-T_{k-1}) - \Psi_y(s-T_k))\, b_k + \Psi_y(s-T_{j-1})b_j \tag{56}$$

if $i < j$; otherwise $a\int_t^s \Phi_y(s-u)b(u)du = \Psi_y(s-t)b_i$.

Proof. Let $\mathcal{A}_x = a(b(t)-x)\partial_x + \frac{1}{2}c^2x\partial^2_{x^2}$. The function J in (54) satisfies $e^{-J(s,x;s,y,z)} = e^{-yx}$ and, for $t < s$, using $J(t,x)$ or sometimes even J as a shorthand for $J(t,x;s,y,z)$:

$$\begin{aligned}
&\partial_x e^{-J} = (-e^{-J})\partial_x J\,,\ \partial^2_{x^2}(e^{-J}) = (-e^{-J})\big(\partial^2_{x^2}J - (\partial_x J)^2\big),\\
&(-e^J)(\partial_t + \mathcal{A}_x - zxI)e^{-J} = \partial_t J + a(b-x)\partial_x J + \frac{1}{2}c^2x\big(\partial^2_{x^2}J - (\partial_x J)^2\big) + zx\\
&\quad = -x\dot\Phi_y(s-t) - a\Phi_y(s-t)b + a(b-x)\Phi_y(s-t) - \frac{1}{2}c^2x\Phi^2_y(s-t) + zx\\
&\quad = -x\big(\dot\Phi_y(s-t) + a\Phi_y(s-t) + \frac{1}{2}c^2\Phi^2_y(s-t) - z\big) = 0,
\end{aligned}$$

by the equations (54) for J and (50) (first line) for Φ_y. An application of the Itô formula then shows that $de^{-J(t,X_t)} - zX_te^{-J(t,X_t)}dt = \partial_x e^{-J(t,X_t)}c\sqrt{X_t}dW_t$. (16) then yields $d(e^{-z\int_0^t X_u du}e^{-J(t,X_t)}) = e^{-z\int_0^t X_u du}\partial_x e^{-J(t,X_t)}c\sqrt{X_t}dW_t$, a local martingale M by Lemmas 8-9. But $X, \Phi \geq 0$ (for $z \geq 0$)) hence $J \geq 0$. As a consequence M is bounded, so that M is a true martingale, by Lemma 4. Therefore

$$e^{-z\int_0^t X_u du}e^{-J(t,X_t)} = \mathbb{E}_t e^{-z\int_0^s X_u du}e^{-J(s,X_s)} = \mathbb{E}_t e^{-z\int_0^s X_u du}e^{-yX_s},$$

hence

$$e^{-J(t,X_t)} = \mathbb{E}_t e^{-z\int_t^s X_u du}e^{-yX_s},$$

which implies (53). This proves that J defined by (54) satisfies (53). The first, second and third line in (55) then follow from (53)-(54) by differentiating $(-e^{-J(t,x;s,y,0)})$ in y and evaluating it at $y = 0$, resp. $(-e^{-J(t,x;s,0,z)})$ in z and evaluating it at $z = 0$, resp. resp. $(-e^{-J(t,x;s,y,1)})$ in y and evaluating it at $y = 0$. Finally, in the case of a piecewise-constant $b(\cdot)$, (56) follows from (54) in view of the second line in (50). ∎

§5 Convergence of Stochastic Approximation Algorithms

Let $H : \mathbb{R}^d \times \mathbb{R}^q \to \mathbb{R}^d$ be a measurable function and let $\{\gamma_k, k \geq 1\}$ be a sequence of positive numbers. Let $\mathbb{R}^q$-valued random variables $\{V_k, k \geq 0\}$ and $\theta_0 \in \mathbb{R}^d$ be defined on a probability space $(\Omega, \mathcal{A}, \mathbb{Q})$. Theorem 3 provides sufficient conditions for the almost-sure convergence and the L^p-convergence, $p \in (0,2)$, of the sequence $\{\theta_k, k \geq 0\}$ given by

$$\theta_{k+1} = \theta_k - \gamma_{k+1}\, H(\theta_k, V_{k+1}). \tag{57}$$

These conditions are general enough to cover the case when the r.v. $\{V_k, k \geq 1\}$ are not i.i.d. but have a distribution converging, in some sense, to the distribution of a r.v. $V_\star$.

We write

$$H(\theta_k, V_{k+1}) = h(\theta_k) + e_{k+1} + r_{k+1}, \tag{58}$$

where

$$\begin{aligned}
h(\theta) &:= \mathbb{E}\left[H(\theta, V_\star)\right],\\
e_{k+1} &:= H(\theta_k, V_{k+1}) - \mathbb{E}\left[H(\theta_k, V_{k+1})|\mathfrak{F}_k\right],\\
r_{k+1} &:= \mathbb{E}\left[H(\theta_k, V_{k+1})|\mathfrak{F}_k\right] - h(\theta_k),
\end{aligned}$$

and where the (discrete-time) filtration $\{\mathfrak{F}_k, k \geq 1\}$ is defined by $\mathfrak{F}_k := \sigma\{V_1, \ldots, V_k\}$.

Theorem 3 *Suppose that*

(i) $\{\gamma_k, k \geq 1\}$ *is a deterministic positive sequence such that* $\sum_k \gamma_k = +\infty$ *and there exists* $\kappa \in (0, 1]$ *such that* $\sum_{k\geq 1} \gamma_k^{1+\kappa} < \infty$,

(ii) $H : \mathbb{R}^d \times \mathbb{R}^q \to \mathbb{R}^d$ *is measurable and* $h : \mathbb{R}^d \to \mathbb{R}^d$ *is continuous,*

(iii) the set $\mathcal{T} := \{h = 0\}$ *is a non-empty compact subset of* $\mathbb{R}^d$ *and for any* $\theta^* \in \mathcal{T}$ *and* $\theta \notin \mathcal{T}$, *we have* $\langle \theta - \theta^*, h(\theta)\rangle > 0$.

Let $\{\theta_k, k \geq 0\}$ *be given by* (57) *where the r.v.* $\{V_k, k \geq 0\}$ *satisfy*

(iv) $\sum_{k\geq 1} \gamma_k^{1-\kappa} |r_k|^2 < +\infty$ $\mathbb{Q}$*-a.s.*

(v) There exist non-negative constants $C_{H,1}, C_{H,2}$ *such that, for any* $\theta \in \mathbb{R}^d$,

$$\sup_{k\geq 1} \mathbb{E}\left[|H|^2(\theta, V_k)\right] \leq C_{H,1} + C_{H,2}|\theta|^2.$$

Then there exists a $\mathcal{T}$*-valued random variable* θ_∞ *such that* $\mathbb{Q}(\lim_k \theta_k = \theta_\infty) = 1$. *If, in addition,*

(vi) $\sum_{k\geq 1} \gamma_k^{1-\kappa} \mathbb{E}[|r_k|^2] < +\infty$,

then $\sup_{k\geq 0} \mathbb{E}[|\theta_k - \theta_\infty|^2] < +\infty$ *and for any* $p \in (0, 2)$, $\lim_k \mathbb{E}\left[|\theta_k - \theta_\infty|^p\right] = 0$.

Proof from Barrera, Crépey, Diallo, Fort, Gobet, and Stazhynski (2019, Section A.2).

Step 1. Almost-sure boundedness and convergence. Let $\theta^* \in \mathcal{L}$. We have, by (58),

$$\begin{aligned}|\theta_{k+1} - \theta^*|^2 &= |\theta_k - \theta^* - \gamma_{k+1}\left(h(\theta_k) + e_{k+1} + r_{k+1}\right)|^2 \\ &= |\theta_k - \theta^*|^2 - 2\gamma_{k+1}\langle \theta_k - \theta^*, h(\theta_k)\rangle \\ &\quad - 2\gamma_{k+1}\langle \theta_k - \theta^*, e_{k+1}\rangle - 2\gamma_{k+1}\langle \theta_k - \theta^*, r_{k+1}\rangle + \gamma_{k+1}^2 |H|^2(\theta_k, V_{k+1}).\end{aligned}$$

Since $\{e_k, k \geq 1\}$ is a martingale-increment w.r.t. the filtration $\{\mathcal{G}_k, k \geq 1\}$ and θ_k is $\mathcal{G}_k$-measurable, we have for any k,

$$\begin{aligned}\mathbb{E}\left[|\theta_{k+1} - \theta^*|^2 | \mathcal{G}_k\right] \leq |\theta_k - \theta^*|^2 - 2\gamma_{k+1}\langle \theta_k - \theta^*, h(\theta_k)\rangle + \gamma_{k+1}^{1+\kappa}|\theta_k - \theta^*|^2 + \gamma_{k+1}^{1-\kappa}|r_{k+1}|^2 \\ + \gamma_{k+1}^2 C_{H,1} + \gamma_{k+1}^2 C_{H,2}|\theta_k|^2,\end{aligned}$$

where we used the inequality $-2\gamma\langle a, b\rangle \leq \gamma^{1+\kappa}|a|^2 + \gamma^{1-\kappa}|b|^2$, the equality $\mathbb{E}[r_{k+1}|\mathcal{G}_k] = r_{k+1}$ and the assumption (v). Hence, by using $|\theta_k|^2 \leq 2|\theta_k - \theta^*|^2 + 2|\theta^*|^2$,

$$\mathbb{E}\left[|\theta_{k+1} - \theta^*|^2 | \mathcal{G}_k\right] \leq \left(1 + 2\gamma_{k+1}^2 C_{H,2} + \gamma_{k+1}^{1+\kappa}\right)|\theta_k - \theta^*|^2 - \tag{59}$$

$$2\gamma_{k+1}\langle \theta_k - \theta^*, h(\theta_k)\rangle + \gamma_{k+1}^{1-\kappa}|r_{k+1}|^2 + \gamma_{k+1}^2 C', \tag{60}$$

where $C' := C_{H,1} + 2C_{H,2}|\theta^*|^2$. From the assumptions (i), (iii), and (iv), we have that, $\mathbb{Q}$-a.s.,

$$\forall k \geq 0 \quad \gamma_{k+1}\langle \theta_k - \theta^*, h(\theta_k)\rangle \geq 0, \qquad \sum_{k\geq 0} (\gamma_{k+1}^{1-\kappa}|r_{k+1}|^2 + \gamma_{k+1}^2 + \gamma_{k+1}^{1+\kappa}) < +\infty.$$

By the Robbins-Siegmund lemma (see Robbins and Siegmund (1971)), $\mathbb{Q}$-a.s. (for an almost-sure set depending upon θ^*)

$$\lim_k |\theta_k - \theta^*| \text{ exists}, \qquad \sum_{k\geq 0} \gamma_{k+1}\langle \theta_k - \theta^*, h(\theta_k)\rangle < +\infty.$$

Since $\mathcal{L}$ is bounded and $\theta^* \in \mathcal{L}$, this implies that the sequence $\{\theta_k, k \geq 0\}$ is bounded with probability one. Using the separability of $\mathbb{R}^d$ and since $\theta' \mapsto \lim_k |\theta_k - \theta'|$ is continuous, we have $\mathbb{Q}$-a.s.:

$$\forall\, \theta' \in \mathcal{L}, \quad \text{the limit } \lim_k |\theta_k - \theta'| \text{ exists.} \tag{61}$$

Set $\varsigma := \liminf_{k\to+\infty} \langle \theta_k - \theta^*, h(\theta_k)\rangle$. By (iii), $\varsigma \geq 0$. As $\sum_{k\geq 0} \gamma_k = +\infty$, we have $\{\varsigma > 0\} \subseteq \sum_{k\geq 0} \gamma_{k+1} \langle \theta_k - \theta^*, h(\theta_k)\rangle = +\infty$; the probability of the second event is zero. Hence $\mathbb{Q}(\varsigma = 0) = 1$.

Therefore, with probability one, there exists a subsequence $\{n_k, k \geq 1\}$ such that $\lim_k \langle \theta_{n_k} - \theta^*, h(\theta_{n_k})\rangle = 0$. Since the sequence $\{\theta_{n_k}, k \geq 1\}$ is bounded a.s., we can still assume (up to extraction of another subsequence) that $\{\theta_{n_k}, k \geq 1\}$ converges to some limit θ_∞. By assumption (ii), we have $\langle \theta_\infty - \theta^*, h(\theta_\infty)\rangle = 0$, and by assumption (iii), this implies that $\theta_\infty \in \mathcal{L}$. But using (61) we get $\lim_k |\theta_k - \theta_\infty| = \lim_k |\theta_{n_k} - \theta_\infty| = 0$. This implies that $\lim_k \theta_k = \theta_\infty$.

Step 2. Uniform boundedness in L^2. Let a (deterministic) point $\theta^* \in \mathcal{L}$ be given. By taking expectation in (59), we have

$$\mathbb{E}\left[|\theta_{k+1} - \theta^*|^2\right] \leq \left(1 + 2\gamma_{k+1}^2 C_{H,2} + \gamma_{k+1}^{1+\kappa}\right) \mathbb{E}\left[|\theta_k - \theta^*|^2\right] + \gamma_{k+1}^{1-\kappa} \mathbb{E}[|r_{k+1}|^2] + C'\gamma_{k+1}^2.$$

Applying again the Robbins-Siegmund lemma with the assumptions (i) and (vi), we deduce that the sequence $\lim_k \mathbb{E}\left[|\theta_k - \theta^*|^2\right]$ exists and thus $\sup_k \mathbb{E}\left[|\theta_k|^2\right] < \infty$ since $\mathcal{L}$ is bounded. This implies $\sup_k \mathbb{E}\left[|\theta_k - \theta_\infty|^2\right] < +\infty$ for any $\mathcal{L}$-valued random variable θ_∞, using again that $\mathcal{L}$ is bounded.

Step 3. Convergence in L^p. Let $C > 0$ and $p \in (0, 2)$. We write

$$\mathbb{E}\left[|\theta_k - \theta_\infty|^p\right] = \mathbb{E}\left[|\theta_k - \theta_\infty|^p \, \mathbb{1}_{|\theta_k - \theta_\infty| < C}\right] + \mathbb{E}\left[|\theta_k - \theta_\infty|^p \, \mathbb{1}_{|\theta_k - \theta_\infty| \geq C}\right].$$

The first term converges to zero by the dominated convergence theorem. For the second term, Hölder's and Markov's inequalities give that

$$\mathbb{E}[|\theta_k - \theta_\infty|^p \mathbb{1}_{|\theta_k - \theta_\infty| \geq C}] \leq \frac{\mathbb{E}[|\theta_k - \theta_\infty|^2]}{C^{2-p}} \leq \frac{\sup_{l\geq 0} \mathbb{E}[|\theta_l - \theta_\infty|^2]}{C^{2-p}};$$

which is lower than $\epsilon > 0$ for some C large enough. This holds true for any ϵ, thus concluding the proof. ∎

Stochastic gradient descent correspond to the special case where $H = \partial_\theta L$ for some locally convex function

$$\mathbb{R}^d \ni \theta \mapsto \ell(\theta) = \mathbb{E}\left[L(\theta, V_\star)\right] \in \mathbb{R},$$

so that root solving h is equivalent to minimizing ℓ, i.e. $\mathcal{T} = \operatorname*{Argmin}_\theta \mathbb{E}\left[L(\theta, V_\star)\right]$. This algorithm or batch variants of it where θ is updated using several draws of V[61] is the core machine learning[62] training scheme.

[61] as opposed to only one in (57).
[62] neural networks, in particular.

Chapter X

Problem Sets

§1 Exit of a Brownian Motion From a Corridor

Exercise

Let N_t denote a Poisson process with intensity $\lambda > 0$, and let $M_t = N_t - \lambda t$ be the compensated martingale of N.

1. Verify that the process Y given as $Y_t = \int_0^t N_{s-} dM_s$ is a martingale:
   (a) First, by invocation of a general theorem of stochastic analysis;
   (b) Second, by a direct computation.
2. Compute $J_t = \int_0^t N_s dM_s$. Is process J_t a martingale?

Problem

Let $X_t = \mu t + \sigma W_t$, where W_t is SBM with respect to a filtration $\mathfrak{F}$ (with $\mathfrak{F}_0$ trivial).

A. Basic properties

Let $u = u(t, x)$ denote a function of class $\mathcal{C}^{1,2}(\mathbb{R}_+ \times \mathbb{R})$.

1. State the Itô formula which is relevant for $u(t, X_t)$.
2. Justify that X is a Markov process.
3. Recalling the characterization $\mathbb{E}_t du(t, X_t) = (\partial_t + \mathcal{A})u(t, X_t)dt$ of the generator $\mathcal{A}$ of a Markov process X, deduce from the above Itô formula the expression of the generator $\mathcal{A}$ of the Brownian motion X, and rewrite the above Itô formula in terms of $\mathcal{A}$.

B. Application

Given reals $a < 0 < b$, let

$$\vartheta = \min\{t \geq 0 : X_t = a \text{ or } b\} \in [0, +\infty].$$

We want to compute $\mathbb{Q}(X_\vartheta = b) = \mathbb{E}\mathbb{1}_{\{X_\vartheta = b\}}$ (with the convention that $\mathbb{1}_{\{X_\vartheta = b\}} = 0$ if ϑ is infinite).

1. Justify that on the random set $\{t \leq \vartheta\}$ we have $\mathbb{Q}(X_\vartheta = b|\mathfrak{F}_t) = u(t, X_t)$ for some function $u = u(t, x)$, admitted to be of class $\mathcal{C}^{1,2}(\mathbb{R}_+ \times \mathbb{R})$.
2. Show that $(\partial_t + \mathcal{A})\, u(t, X_t)dt$ is constant on $[0, \vartheta]$.
3. Derive a partial differential equation satisfied by u on the domain $[0, +\infty) \times [a, b]$, along with suitable boundary conditions at a and b.
4. Admitting uniqueness for a bounded solution to the equation derived in the question 3, show that for $\mu \neq 0$ we have
$$u(t, x) = v(x) := \frac{\exp(\eta x) - \exp(\eta a)}{\exp(\eta b) - \exp(\eta a)},$$
with $\eta = -2\mu/\sigma^2$.
5. Deduce the final formula for $\mathbb{Q}(X_\vartheta = b)$.
6. Proceed as in questions 1-5 above to compute $\mathbb{Q}(X_\vartheta = a)$.
7. Prove from the above that $\vartheta < +\infty$ with probability one.
8. In the case $\mu = 0$, show that $u(t, x) = \alpha + \beta x$, where the values of α and β for $\mathbb{Q}(X_\vartheta = b)$ will be obtained by verification into the equation for u derived in Question 3. What is the probability that $\vartheta = +\infty$?

§2 Jump-to-Ruin

Exercise

1. Describe and justify the inverse method for simulating a real random variable X with a numerically invertible c.d.f. F.
2. Describe the rejection-acceptance method for simulating i.i.d. uniform points in the unit disk D.
3. Describe the Marsaglia method to simulate a pair of independent Gaussian random variables.
4. How to simulate a lognormal random variable X with mean m and standard deviation $s = 20\%m$?

Problem

In an economy with zero interest rates, we consider the following Merton Jump-to-Ruin (jr) model of a non-dividend paying stock S in a zero interest rates economy, stated with respect to a reference filtration $\mathcal{F}$ and a probability measure $\mathbb{Q} \sim$ the physical one:

$$dS_t = \lambda S_t dt + \sigma S_t dW_t - S_{t-} dN_t = \sigma S_t dW_t - S_{t-} dM_t, \tag{1}$$

where W is a standard Brownian motion, $\sigma > 0$ is a constant volatility parameter, and N is a $\mathcal{F}$ Poisson process with intensity $\lambda > 0$ and compensated martingale $M = N - \lambda t$, with W and N independent[1].

One considers a derivative with an $\mathcal{F}_T$ measurable payoff ξ at time T. We denote by $\mathcal{N}$ the Gaussian cumulative distribution function and, given the maturity T and strike K of an option, we let

$$d_\pm = \frac{\ln(\frac{S_0}{K}) + \lambda T}{\sigma\sqrt{T}} \pm \frac{1}{2}\sigma\sqrt{T}.$$

[1] as in fact always the case for a Brownian motion and a Poisson process with respect to a common stochastic basis (He et al., 1992, Theorem 11.43 page 316).

An application of (37) yields

$$\mathbb{V}\mathrm{ar}\left(\int_0^T \alpha_t dW_t + \int_0^T \beta_{t-} dM_t\right) = \mathbb{E}\int_0^T \left(\alpha_t^2 + \lambda\beta_t^2\right) dt, \tag{2}$$

for every càdlàg adapted processes α and β making the right-hand side finite in (2).

0. Preliminary

1. Introducing an auxiliary Black–Scholes model $S^{bs}t$ such that $S_0^{bs} = S_0$ and for $t \in [0,T]$:
$$dS_t^{bs} = \lambda S_t^{bs} dt + \sigma S_t^{bs} dW_t, \tag{3}$$
write an explicit representation of S_t in terms of S_t^{bs} and N_t.
2. Show that S is a $\mathbb{Q}$ martingale and justify that the model is nonarbitrable.

A. Call Option

In this part we consider the pricing of a vanilla call option, so that $\xi = \phi(S_T) = (S_T - K)^+$.

1. Prove that the *jr* price $\mathbb{Q}$ process C of this option can be represented, for $t \in [0,T]$, by:
$$C_t = u(t, S_t), \tag{4}$$
for some pricing function $u = u(t,S)$ over $[0,T] \times \mathbb{R}_+$. Hereafter the function u is admitted to be of class $\mathcal{C}^{1,2}\big([0,T) \times (0,+\infty)\big) \cap \mathcal{C}^0\big([0,T] \times (0,+\infty)\big)$.
2. Show that we have for $t \in [0,T]$
$$\begin{aligned} dC_t = du(t,S_t) = \\ (\partial_t u + \mathcal{A}u)(t,S_t)dt + \partial_S u(t,S_t)\sigma S_{t-} dW_t + \delta u(t, S_{t-}) dM_t, \end{aligned} \tag{5}$$
where
$$\begin{aligned} \delta u(t,S) &= u(t,0) - u(t,S) \\ \mathcal{A}u(t,S) &= \lambda S \partial_S u(t,S) + \frac{\sigma^2 S^2}{2}\partial^2_{S^2} u(t,S) + \lambda \delta u(t,S). \end{aligned}$$
3. Prove that $u(t,0) = 0$ and that the pricing function u satisfies the following pricing equation:
$$\begin{cases} u(T,S) = (S-K)^+, \ S > 0 \\ \partial_t u(t,S) + \lambda S \partial_S u(t,S) + \frac{\sigma^2 S^2}{2}\partial^2_{S^2} u(t,S) - \lambda u(t,S) = 0, \ t < T, \ S > 0. \end{cases} \tag{6}$$
4. Deduce the expression of the price C_0 of the option at time 0.
5. Use the representation of question 0.1 to compute C_0 directly, by probabilistic arguments.

B. Put Option

Now consider the pricing of a put option, in two forms: either a vanilla put with payoff $\xi = (K - S_T)^+$, or a vulnerable put with payoff $\xi^* = \mathbb{1}_{\{\tau > T\}}(K - S_T)^+$. A vulnerable put option thus only delivers its promised payoff $(K - S_T)^+$ if $\tau > T$. This corresponds to the case of a warrant issued by the firm with stock modeled by S. In this case, if the issuing firm is defaulted by T it will not be able[2] to deliver the promised payoff $(K - S_T)^+ = K$.

Note that for a call option, vulnerable or not makes no difference in *jr*, since in this model $S_T = (S_T - K)^+ = 0$ on $\tau < T$.

[2]Up to some recovery, assumed here to be zero for simplicity.

1. Following the lines of questions A.1-2, prove that the *jr* price process P of the vanilla put can be represented, for $t \in [0,T]$, as:
$$P_t = v(t, S_t)$$
for a vanilla put pricing function $v = v(t,S)$ over $[0,T] \times \mathbb{R}_+$. Admitting that the function v is of class $\mathcal{C}^{1,2}\big([0,T) \times (0,+\infty)\big) \cap \mathcal{C}^0\big([0,T] \times (0,+\infty)\big)$, show that
$$\begin{cases} v(T,S) = (K-S)^+, \ S > 0 \\ \partial_t v(t,S) + \lambda S \partial_S v(t,S) + \frac{\sigma^2 S^2}{2} \partial^2_{S^2} v(t,S) - \lambda v(t,S) + \lambda K = 0, \ t < T, \ S > 0. \end{cases} \tag{7}$$

2. Recall and use the vanilla call-put parity relationship to show that
$$P_0 = K e^{-\lambda T} \mathcal{N}(-d_-) - S_0 \mathcal{N}(-d_+) + K(1 - e^{-\lambda T}).$$

3. Prove that the *jr* price process P^* of the vulnerable put can be represented, for $t \in [0,T]$ as:
$$P^*_t = \mathbb{1}_{t<\tau} v^*(t, S^{bs}_t) \tag{8}$$
for a pre-default vulnerable put pricing function $v^* = v^*(t,S)$ over $[0,T] \times \mathbb{R}_+$. Admitting that the function v^* is of class $\mathcal{C}^{1,2}([0,T) \times (0,+\infty) \cap \mathcal{C}^0([0,T] \times (0,+\infty)$, show that
$$\begin{cases} v^*(T,S) = (K-S)^+, \ S > 0 \\ \partial_t v^*(t,S) + \lambda S \partial_S v^*(t,S) + \frac{\sigma^2 S^2}{2} \partial^2_{S^2} v^*(t,S) \\ \qquad\qquad -\lambda v^*(t,S) = 0, \ t < T, \ S > 0. \end{cases} \tag{9}$$

4. Prove directly by probabilistic arguments that
$$\begin{aligned} P^*_0 &= C_0 - S_0 + K e^{-\lambda T} \\ &= P_0 - (1 - e^{-\lambda T}) K. \end{aligned} \tag{10}$$

C. Barrier Option

With a barrier option the right to exercise the payoff at maturity depends on the underlying having crossed a given barrier level on $[0,T]$. We will consider in this part an up-and-out call option with barrier level H corresponding to the following payoff:
$$\xi = \mathbb{1}_{\{\vartheta = T, \, S_T < H\}} (S_T - K)^+,$$
where
$$\vartheta = \inf\{t \geq 0; S_t \geq H\} \wedge T.$$

1. What is the interest of this barrier option as compared to the corresponding vanilla call? Which of the two should be cheaper?

2. Show that the price process B_t of this option is given over $[0, \vartheta]$ by $w(t, S_t)$, for a pricing function $w = w(t,S)$. Hereafter the function w is admitted to be of class $\mathcal{C}^{1,2}\big([0,T) \times (0,H)\big) \cap \mathcal{C}^0\big([0,T] \times (0,H]\big)$. Show that
$$\begin{cases} w(T,S) = (S-K)^+, \ 0 < S < H \\ w(t,H) = 0, \ 0 \leq t \leq T \\ \partial_t w(t,S) + \lambda S \partial_S w(t,S) + \frac{\sigma^2 S^2}{2} w(t,S) \partial^2_{S^2} w(t,S) \\ \qquad\qquad -\lambda w(t,S) = 0, \ t < T, \ 0 < S < H. \end{cases} \tag{11}$$

3. Show that the time-0 *jr* and sprices of the up-and-out call coincide.

D. Local Volatility

We recall the Dupire equation which is satisfied at time 0 by call prices C^{lo} in the (T,K)-variables, in a local volatility model with volatility function $\sigma(t,S)$ and risk-free interest rate r (assuming no dividends on S):

$$\begin{cases} C_0^{lo}(T=0,K)=(S_0-K)^+,\ K>0 \\ \partial_T C_0^{lo}(T,K)+rK\partial_K C_0^{lo}(T,K)-\frac{1}{2}\sigma(T,K)^2K^2\partial^2_{K^2}C_0^{lo}(T,K)=0,\ T,K>0. \end{cases} \tag{12}$$

1. Recall what calibrating a model means.

   What is calibration used for?

   How is model calibration achieved in practice?

2. Using the Dupire equation (12), write the corresponding Dupire formula for $\sigma(T,K)$, and discuss the statement that "the class of local volatility models is calibratable to any reasonable market of vanilla call options", where the meaning of "reasonable" will be specified.

3. Using the Dupire formulas in the *jr* and **s**models, derive the following expression for the local volatility function $\sigma_0^{jr}(\cdot,\cdot)$ calibrated to the *jr* vanilla call prices at time 0 (the *jr* local volatility function σ_0^{jr}):

$$\frac{\sigma_0^{jr}(T,K)^2}{2}=\frac{\sigma^2}{2}+\lambda\sigma\sqrt{2\pi T}\mathcal{N}(d_-)e^{\frac{d_-^2}{2}}. \tag{13}$$

4. Admitting that $\mathcal{N}(x)\sim_{x\to-\infty}\frac{-e^{-x^2}}{\sqrt{2\pi}x}$ and also using the fact that $\mathcal{N}(x)\to 1$ as $x\to+\infty$, compute the limit of $\sigma_0^{jr}(T,K)$ as $T\to 0+$ depending on whether $K<S_0,=S_0$ or $>S_0$. Interpret the result.

E. Local Default Intensity

In this part, we consider a driving point process N with a local intensity $\lambda(t,S_t)$, so that N is no longer a Poisson process (and N and W are no longer independent). All the above Markov and martingale analysis still holds true with λ replaced by $\lambda(t,S)$. But all explicit formulas collapse. So the prices C_0,P_0,P_0^* and B_0 in Parts A, B, C, and the local volatility function $\sigma_0(\cdot,\cdot)$ defined through the relevant Dupire formula (calibrated to the vanilla option prices in the jump-to-ruin model with local jump intensity model of this part) all need to be estimated numerically.

Given uniform time- and space-grids $(t_i)_{1\le i\le n}$ and $(S^j)^{0\le j\le m}$ of respective steps h and k, with the S-grid centered around S_0, we denote

$$\alpha_i^j=\frac{\sigma^2(S^j)^2}{2k^2}-\frac{\lambda_i^jS^j}{2\mathrm{k}},\ \beta_i^j=-\frac{\sigma^2(S^j)^2}{\mathrm{k}^2},\ \gamma_i^j=\frac{\sigma^2(S^j)^2}{2k^2}+\frac{\lambda_i^jS^j}{2\mathrm{k}}, \tag{14}$$

with $\lambda_i^j=\lambda(t_i,S^j)$.

1. In terms of the above coefficients, write an explicit finite difference scheme in the (t,S) variables for computing $C_0=u(0,S_0)$. Derive the associated stability condition.

2. In terms of the above coefficients, write a hybrid scheme, implicit in the differential terms and explicit in the jump term, for computing $C_0=u(0,S_0)$, not subject to the previous stability condition, and discuss the implementation of this scheme.

3. Discuss the changes required in the above schemes for computing P_0,P_0^* and B_0.

4. Propose a numerical method for calibrating the local volatility function $\sigma_0(\cdot,\cdot)$ corresponding to the model of this part (jump-to-ruin model with local jump intensity).

5. Write, in terms of suitable coefficients, finite difference θ-schemes that can be used for pricing the above options in the local volatility model with volatility function $\sigma_0(\cdot,\cdot)$.

   Discuss the particular issues related with explosion of $\sigma_0(t,S)$ (cf. in the situation of Part D. the function $\sigma_0^{jr}(\cdot,\cdot)$ at $t=0+, S<S_0$).

   For which of the options do we expect agreement (up to the numerical noise) between the prices in the jump-to-ruin model with local jump intensity of this part and the prices in the corresponding local volatility model $\sigma_0(\cdot,\cdot)$?

F. Hedging

1. Back at the situation of a constant intensity λ of N_t as in Part D, in this part we consider the issue of dynamically hedging, in continuous-time, one short vanilla call option position by the underlying S and the riskless constant asset. Let a hedging strategy $u=(u_t)_{t\in[0,T]}$ denote the number of units of stock which are held in the hedging portfolio at every point in time.

   Justify that the profit-and-loss process $p=p(u)$ associated with the price-and-hedge strategy (C,u) in S (and the quantity of riskless asset deduced from u by the self-financing condition) evolves following

   $$dp_t = -dC_t + u_t dS_t$$

   (with $p_0=0$).

2. A strategy u is said to replicate the payoff $\phi(S_T)$ if $p_T(u)=0$ almost surely under the physical probability measure $\mathbb{P}$.

   Justify that a strategy u replicates $\phi(S_T)$ under $\mathbb{P}$ if and only if it replicates $\phi(S_T)$ under $\mathbb{Q}$, i.e. $p_T(u)=0$ almost surely under $\mathbb{Q}$.

3. Rewrite dp_t as

   $$dp_t = \alpha_t dW_t + \beta_t dM_t$$

   for integrands α_t and β_t to be specified.

4. Justify that there is no strategy u replicating the payoff $\phi(S_T)$.

5. Using (2), show that the strategy u_t^{va} that minimizes the risk-neutral variance of p_T over the set of càdlàg strategies u making the right-hand side finite in (2), for α and β related to u as per Question 3, is given by

   $$u_t^{va} = \frac{\sigma^2}{\sigma^2+\lambda}\partial_S u(t,S_{t-}) - \frac{\lambda}{\sigma^2+\lambda}\frac{\delta u(t,S_{t-})}{S_{t-}}.$$

6. Show that $u_t^{va} = \frac{d\langle C,S\rangle_t}{d\langle S\rangle_t}$.

7. Does the strategy u^{va} minimize the variance under the physical probability measure?

   Under which measure: risk-neutral or physical, are we interested in minimizing the variance of the hedging error in practice?

   What do you think mathematically of the problem of minimizing the variance of the hedging error under the physical probability measure?

   What do you think of the hedge u^{va}?

8. We now consider the dynamic hedging in continuous-time of the vanilla call option by its underlying S and an auxiliary put option with payoff $(K - S_\Theta)^+$ at $\Theta > T$ (and the constant funding asset)[3]. The auxiliary option price process P is thus written $P_t = v(t, S_t)$, for some related pricing function $v = v(t, S)$. The replication error $e = e(u, \eta)$ associated with the strategy u in S and η in the auxiliary option (and the quantity of riskless asset deduced from the self-financing condition) evolves following
$$dp_t = -dC_t + u_t dS_t + \eta_t dP_t$$
(with $p_0 = 0$).

   Write the replication condition $p_T(u, \eta) = 0$ in the form of a system of equations to be satisfied by the pair (u_t, η_t) for every $t < T$.

   Discuss the well-posedness of this system for every $t \in [0, T]$.

9. Describe simulation algorithms allowing for numerical verification of the hedging properties highlighted in the questions 4 to 7.

10. What changes are needed in this part if the option which is hedged is now a barrier up-and-out call as in Part C?

§3 Pricing With a Regime-Switching Volatility

Exercise

We consider, in the risk-neutral Black-Scholes model $dS_t = \sigma S_t dW_t$, a forward start option with payoff at time T given by
$$\xi = (S_T - K S_\Theta)^+ ,$$
for a fixed time $0 < \Theta < T$. We write
$$d_\pm(t, S_t) = \frac{\ln(\frac{S_t}{K S_\Theta})}{\sigma\sqrt{T-t}} \pm \frac{1}{2}\sigma\sqrt{T-t},$$
for $t \in [0, T]$.

1. Show that the option's price at time Θ is given by
$$C_\Theta = S_\Theta \left(\mathcal{N}(d_+(\Theta)) - K\mathcal{N}(d_-(\Theta))\right), \tag{15}$$
   where $\mathcal{N}$ is the standard Gaussian cumulative distribution function and
$$d_\pm(\Theta) = \frac{-\ln(K)}{\sigma\sqrt{T-\Theta}} \pm \frac{1}{2}\sigma\sqrt{T-\Theta}.$$

2. Show that the option's price at time t is given by
$$C_t = \begin{cases} S_t \left(\mathcal{N}(d_+(\Theta)) - K\mathcal{N}(d_-(\Theta))\right) \text{ if } t < \Theta \\ S_t \mathcal{N}(d_+(t, S_t)) - K S_\Theta \mathcal{N}(d_-(t, S_t)) \text{ if } t \geq \Theta. \end{cases} \tag{16}$$

3. Show that one can replicate this option, using the underlying stock S and the riskless, constant asset as hedging instruments, so there exists a replication strategy Δ_t (number of stocks held at time t in the replication portfolio) such that $dC_t = \Delta_t dS_t$.

4. Compute Δ_t.

[3]Note that the extended market is still nonarbitrable by application of Corollary 0.1.

Problem

In an economy with zero interest rates, we consider the following stochastic volatility model for a non-dividend-paying stock S, stated with respect to a reference filtration $\mathcal{F}$ and a probability measure $\mathbb{Q} \sim$ the physical one:

$$dS_t = \sigma_t S_t dW_t, \tag{17}$$

where W is a standard Brownian motion and the volatility σ follows a two-state Markov chain with constant intensity of transition $\lambda > 0$. In other words, in the time interval $(t, t+dt)$, the instantaneous volatility passes from the "old" value $\sigma_{t-} = \underline{\sigma}$ to the "new" value $\sigma_t = \overline{\sigma}$, and vice versa, with probability λdt, independently of W. Or, in an SDE formulation,

$$d\sigma_t = (\sigma'_{t-} - \sigma_{t-}) dN_t, \tag{18}$$

where:

- $\sigma' = \sigma'(\sigma)$ is a notation for $\underline{\sigma}$ if $\sigma = \overline{\sigma}$ and for $\overline{\sigma}$ if $\sigma = \underline{\sigma}$,
- N is a Poisson process with intensity $\lambda > 0$ and compensated martingale $M = N - \lambda$,

with W and N independent[4].

One considers a derivative with payoff $\phi(S_T)$ at time T, for a measurable and bounded payoff function ϕ.

A. Derivation of the Pricing Equations

1. Denoting $X_t = \ln(S_t)$, derive the SDE satisfied by X.
2. Show that the process S is a $\mathbb{Q}$ martingale and justify that the model is non arbitrable.
3. Show that the $\mathbb{Q}$ process C of the option can be represented as $C_t = u(t, S_t, \sigma_t)$ for a $\mathbb{Q}$ pricing function $u = u(t, S, \sigma)$ over $[0, T] \times (0, +\infty) \times \{\underline{\sigma}, \overline{\sigma}\}$.
4. Hereafter the functions $u(\cdot, \cdot, \underline{\sigma})$ and $u(\cdot, \cdot, \overline{\sigma})$ are assumed to be of class $\mathcal{C}^{1,2}\big([0, T) \times (0, +\infty)\big) \cap \mathcal{C}^0\big([0, T] \times (0, +\infty)\big)$. Show that the dynamics of C are given by:

$$\begin{aligned} du(t, S_t, \sigma_t) = &\left(\partial_t u + \mathcal{A}u\right)(t, S_t, \sigma_t) dt + \\ &\sigma_{t-} S_t \partial_S u(t, S_t, \sigma_{t-}) dW_t + \delta u(t, S_t, \sigma_{t-}) dM_t, \end{aligned} \tag{19}$$

   where we have set

$$\begin{aligned} \delta u(t, S, \sigma) &= u(t, S, \sigma') - u(t, S, \sigma) \\ \mathcal{A}u(t, S, \sigma) &= \frac{\sigma^2 S^2}{2} \partial^2_{S^2} u(t, S, \sigma) + \lambda \delta u(t, S, \sigma). \end{aligned}$$

5. Deduce that the pricing function u satisfies the following system of equations:

$$\begin{cases} \left(\partial_t + \frac{\underline{\sigma}^2 S^2}{2} \partial^2_{S^2}\right) u(t, S, \underline{\sigma}) + \lambda \left(u(t, S, \overline{\sigma}) - u(t, S, \underline{\sigma})\right) = 0 \\ \left(\partial_t + \frac{\overline{\sigma}^2 S^2}{2} \partial^2_{S^2}\right) u(t, S, \overline{\sigma}) + \lambda \left(u(t, S, \underline{\sigma}) - u(t, S, \overline{\sigma})\right) = 0 \end{cases} \tag{20}$$

   for $(t, S) \in [0, T) \times (0, +\infty)$, as well as a terminal condition to be specified at time T.

[4] as in fact always the case for a Brownian motion and a Poisson process with respect to a common stochastic basis (He et al., 1992, Theorem 11.43 page 316).

B. Pricing by PDEs

Given uniform time- and space-grids $(t_i)_{1\leq i\leq n}$ and $(S^j)^{1\leq j\leq m}$ of the respective steps h and k, with the S-grid centered around S_0, let

$$\underline{\alpha}^j = \frac{\underline{\sigma}^2(S^j)^2}{2k^2},\ \underline{\beta}^j = -\frac{\underline{\sigma}^2(S^j)^2}{\mathrm{k}^2},\ \underline{\gamma}^j = \frac{\underline{\sigma}^2(S^j)^2}{2k^2}$$
$$\overline{\alpha}^j = \frac{\overline{\sigma}^2(S^j)^2}{2k^2},\ \overline{\beta}^j = -\frac{\overline{\sigma}^2(S^j)^2}{\mathrm{k}^2},\ \overline{\gamma}^j = \frac{\overline{\sigma}^2(S^j)^2}{2k^2}.$$

1. Write in terms of the above coefficients an explicit finite differences scheme to compute $C_0 = u(0, S_0, \sigma_0)$ and $\Delta_0 = \partial_S u(0, S_0, \sigma_0)$ (in the (t, S) variables, without log-transformation). Derive the related stability condition.

2. Write, in terms of the above coefficients, a hybrid scheme explicit in the jump terms / implicit in the differential terms of (20), not subject to the previous stability condition, and discuss the implementation of this scheme.

3. What do you think about a fully implicit finite difference scheme for this problem?

C. Pricing by Monte Carlo

Now ^{j} refers to the j^{th} simulated trajectory of S.

1. Write an exact simulation algorithm (without time-discretization error) of S_T in (17).

2. Describe the Monte Carlo standard pricing algorithm based on m simulated trajectories of S to compute C_0 and Δ_0. Give the expressions of the estimates $\widehat{C}_0$ and $\widehat{\Delta}_0$ of C_0 and Δ_0, and of the corresponding standard errors, $\widehat{\sigma}_0^C$ and $\widehat{\sigma}_0^\Delta$.

   For the delta, proceed by derivation of the payoff along the trajectory of S, assuming the function ϕ almost everywhere differentiable.

3. Comment on the pros and cons of this Monte Carlo method and of the deterministic schemes of Part B for the computation of C_0 and Δ_0.

   In particular, give the order of magnitude of the errors in both cases (consistency error for deterministic schemes, simulation error for Monte Carlo).

4. Assuming an explicit formula available for pricing the option with payoff function ϕ *in the Black-Scholes model*, describe qualitatively a Monte Carlo pricing scheme with variance reduction based on the control variate $\phi(S_T^{bs})$, where

$$dS_t^{bs} = \sigma_0 S_t^{bs} dW_t, S_0^{bs} = S_0.$$

   What are the properties of the method?

   Describe the simulation scheme to be used for S_T^{bs} in this application.

   Give the expressions of the resulting estimates $\widetilde{C}_0$ and $\widetilde{\Delta}_0$ of C_0 and Δ_0, and of the corresponding standard errors, $\widetilde{\sigma}_0^C$ and $\widetilde{\sigma}_0^\Delta$.

§4 Hedging with a Regime-Switching Volatility

Still in the regime-switching volatility model of Section §3, an application of (37) yields

$$\mathbb{V}\mathrm{ar}\left(\int_0^T \alpha_t dS_t + \int_0^T \beta_{t-} dM_t\right) = \mathbb{E}\left(\int_0^T \alpha_t^2 \sigma_t^2 S_t^2 dt + \lambda \int_0^T \beta_t^2 dt\right), \tag{21}$$

for every càdlàg adapted processes α and β making the right-hand side finite in (21).

A. Derivation of the Pricing Equations

1. Justify that the process C_t can be represented as $C_t = u(t, X_t, \sigma_t)$, with $X_t = \ln(S_t)$ (cf. §3.A.1), for a $\mathbb{Q}$ pricing function $u = u(t, x, \sigma)$ over $[0, T] \times \mathbb{R} \times \{\underline{\sigma}, \overline{\sigma}\}$.

2. Hereafter the functions $u(\cdot, \cdot, \underline{\sigma})$ and $u(\cdot, \cdot, \overline{\sigma})$ are assumed to be of class $\mathcal{C}^{1,2}\big([0, T) \times \mathbb{R}\big) \cap \mathcal{C}^0\big([0, T] \times \mathbb{R}\big)$. Show that the dynamics of C are given by:

$$\begin{aligned} dC_t = du(t, X_t, \sigma_t) = \\ (\partial_t u + \mathcal{A}u)(t, X_t, \sigma_t)dt + \partial_x u(t, X_t, \sigma_t)\sigma_t dW_t + \delta u(t, X_t, \sigma_{t-})dM_t, \end{aligned} \tag{22}$$

   where we have set

$$\begin{aligned} \delta u(t, x, \sigma) &= u(t, x, \sigma') - u(t, x, \sigma) \\ \mathcal{A}u(t, x, \sigma) &= \frac{\sigma^2}{2}\left(\partial^2_{x^2} u - \partial_x u\right)(t, x, \sigma) + \lambda \delta u(t, x, \sigma). \end{aligned}$$

3. Deduce that the pricing function u satisfies the following system of equations:

$$\begin{cases} \left(\partial_t u + \frac{\underline{\sigma}^2}{2}(\partial^2_{x^2} u - \partial_x u)\right)(t, x, \underline{\sigma}) + \lambda\left(u(t, x, \overline{\sigma}) - u(t, x, \underline{\sigma})\right) = 0 \\ \left(\partial_t u + \frac{\overline{\sigma}^2}{2}(\partial^2_{x^2} u - \partial_x u)\right)(t, x, \overline{\sigma}) + \lambda\left(u(t, x, \underline{\sigma}) - u(t, x, \overline{\sigma})\right) = 0 \end{cases} \tag{23}$$

   for $(t, x) \in [0, T) \times \mathbb{R}$, as well as a terminal condition $u = \psi$ to be specified at time T.

B. Pricing

Let there be given uniform time- and space-grids $(t_i)_{1 \le i \le n}$ and $(x^j)^{1 \le j \le m}$ of respective steps h and k, with $t_n = n\mathrm{h} = T$ and with the space-grid centered at $X_0 = x$. We write

$$\begin{aligned} \underline{\alpha} &= \frac{\underline{\sigma}^2}{2k^2} + \frac{\underline{\sigma}^2}{4\mathrm{k}},\ \underline{\beta} = -\frac{\underline{\sigma}^2}{\mathrm{k}^2},\ \underline{\gamma} = \frac{\underline{\sigma}^2}{2k^2} - \frac{\underline{\sigma}^2}{4\mathrm{k}} \\ \overline{\alpha} &= \frac{\overline{\sigma}^2}{2k^2} + \frac{\overline{\sigma}^2}{4\mathrm{k}},\ \overline{\beta} = -\frac{\overline{\sigma}^2}{\mathrm{k}^2},\ \overline{\gamma} = \frac{\overline{\sigma}^2}{2k^2} - \frac{\overline{\sigma}^2}{4\mathrm{k}}. \end{aligned} \tag{24}$$

1. Denoting $\underline{u}_i^j \approx u(t_i, x^j, \underline{\sigma})$, $\overline{u}_i^j \approx u(t_i, x^j, \overline{\sigma})$, write a trinomial trees scheme for u in the form of a suitable condition for $\underline{u}_n$, and then, for $i = n-1, \ldots, 0, j = 1, \ldots, m$:

$$\begin{cases} \underline{u}_i^j = \underline{p}_- \, \underline{u}_{i+1}^{j-1} + \underline{p} \, \underline{u}_{i+1}^j + \underline{p}_+ \, \underline{u}_{i+1}^{j+1} + \underline{q} \, \overline{u}_{i+1}^j \\ \overline{u}_i^j = \overline{p}_- \, \overline{u}_{i+1}^{j-1} + \overline{p} \, \overline{u}_{i+1}^j + \overline{p}_+ \, \overline{u}_{i+1}^{j+1} + \overline{q} \underline{u}_{i+1}^j, \end{cases}$$

   where the weights $\underline{p}_-, \underline{p}, \underline{p}_+, \underline{q}, \overline{p}_-, \overline{p}, \overline{p}_+, \overline{q}$ are expressed in terms of the coefficients in (24).

2. Give the approximations in the tree for $C_0 = u(0, X_0, \sigma_0)$ and $\Delta_0 = S_0^{-1}\partial_x u(0, X_0, \sigma_0)$. Justify this expression for Δ_0.

3. Derive the stability condition of the scheme.

4. Write a fully discrete Markov chain approximation $(\widehat{X}_{t_i}, \widehat{\sigma}_{t_i})$ to (X_t, σ_t) in the above pair of trees, in which $\widehat{X}_t$ evolves at the nodes of the trees and passes from one tree to the other with certain probabilities. More precisely, given that at time t_{i-1} the Markov chain is in the state $(\widehat{X}_{t_{i-1}}, \widehat{\sigma}_{t_{i-1}}) = (x, \underline{\sigma})$, write in terms of $\underline{p}_-, \underline{p}, \underline{p}_+, \underline{q}$ the probabilities that at time t_i we have $(\widehat{X}_{t_i}, \widehat{\sigma}_{t_i}) = (x - \mathrm{k}, \underline{\sigma})$, $(x, \underline{\sigma})$, $(x + \mathrm{k}, \underline{\sigma})$ or $(x, \overline{\sigma})$; express likewise the transition probabilities from $(\widehat{X}_{t_{i-1}}, \widehat{\sigma}_{t_{i-1}}) = (x, \overline{\sigma})$ to $(\widehat{X}_{t_i}, \widehat{\sigma}_{t_i}) = (x - \mathrm{k}, \overline{\sigma})$, $(x, \overline{\sigma})$, $(x + \mathrm{k}, \overline{\sigma})$ in terms of $\overline{p}_-, \overline{p}, \overline{p}_+, \overline{q}$.

5. Using the trees with transition probabilities as developed in the previous question, we simulate $m = 10^4$ different paths of $(\widehat{X}, \widehat{\sigma})$. Describe a Monte Carlo algorithm for C_0 and Δ_0 based on the m trajectories of $(\widehat{X}, \widehat{\sigma})$ thus simulated. Give the expressions of the estimates $\widehat{C}_0$ and $\widehat{\Delta}_0$ of C_0 and Δ_0, and of the corresponding standard errors, $\widehat{\sigma}_0^C$ and $\widehat{\sigma}_0^\Delta$.

   For the delta, proceed by derivation of the payoff function ϕ, assumed almost everywhere differentiable.

6. Is this algorithm the best Monte Carlo solution for this model? Propose a better alternative, without time-discretisation error.

C. Hedging

1. Now consider the issue of dynamically hedging, in continuous-time, one short option position by the underlying S and the riskless constant asset. Let a hedging strategy $u = (u_t)_{t\in[0,T]}$ denote the number of units of stock held in the hedging portfolio at every point in time.

   Justify that the profit-and-loss process $p = p(u)$ associated with the price-and-hedge strategy (C, u) in S (and the quantity of riskless asset deduced from u by the self-financing condition) evolves according to

$$dp_t = -dC_t + u_t dS_t$$

   (with $p_0 = 0$).

2. One says that a strategy u replicates the payoff $\phi(S_T)$ if $p_T(u) = 0$ almost surely under the physical probability measure $\mathbb{P}$.

   Justify that a strategy u replicates $\phi(S_T)$ under $\mathbb{P}$ if and only if it replicates $\phi(S_T)$ under $\mathbb{Q}$, i.e. $p_T(u) = 0$ almost surely under $\mathbb{Q}$.

3. Using (22) and (23), rewrite dp_t as

$$dp_t = \alpha_t dS_t + \beta_t dM_t$$

   for to-be-determined integrands α and β.

4. Exhibit one elementary class of payoff functions ϕ for which there exists a replication strategy u ($p_T(u) = 0$ almost surely) and describe this strategy.

5. Justify that, in general, there is no strategy u replicating the payoff $\phi(S_T)$.

6. Using (21), show that the strategy which minimizes the risk-neutral variance of p_T over the class of càglàd processes u making the right-hand side of (21) finite (for α and β as per Question 3. there) assumes the following form:

$$u_t^{va} = S_t^{-1} \partial_x u(t, X_t, \sigma_{t-}).$$

7. Show that $u_t^{va} = \frac{d\langle C, S\rangle_t}{d\langle S\rangle_t}$.

8. Does the strategy u^{va} minimize the variance under the physical probability measure?

   Under which measure: risk-neutral or physical, are we interested in minimizing the variance of the hedging error in practice?

   What do you think mathematically of the problem of minimizing the variance of the hedging error under the physical probability measure?

   What do you think of the hedge u^{va}?

9. One now considers the dynamic hedging in continuous-time of the option by its underlying S and an auxiliary option with $\mathbb{Q}$ integrable payoff $\varphi(S_\Theta)$ at $\Theta > T$ (and the constant asset), assumed priced by its $\mathbb{Q}$ price process.

   The auxiliary option price process P is thus written $P_t = v(t, X_t, \sigma_t)$, for some related pricing function $v = v(t, x, \sigma)$. The hedging error $p = p(u, \eta)$ associated with the strategy u in S and η in the auxiliary option (and the quantity of riskless asset deduced from the self-financing condition) evolves according to

   $$dp_t = -dC_t + u_t dS_t + \eta_t dP_t$$

   (with $p_0 = 0$).

   Show that the extended hedging market is still nonarbitrable.

   Rewrite de_t as

   $$de_t = \alpha_t dS_t + \beta_t dM_t$$

   for integrands α_t and β_t to be determined.

   Write the replication condition $p_T(u, \eta) = 0$ in the form of a system of equations to be satisfied by the pair (u_t, η_t) for every $t < T$. Discuss the well-posedness of this system (write an explicit condition on the function v for it to have a unique solution) for every $t \in [0, T]$.

10. Describe a simulation algorithm in terms of the tree with transition probabilities as developed in the question B.4, allowing for numerical verification of the statistical properties of the hedging errors studied theoretically in the questions C.4 to 8.

Bibliography

Abbas-Turki, L., S. Crépey, and B. Saadeddine (2022). Pathwise CVA regressions with oversimulated defaults. *Mathematical Finance*, 1–34. DOI: 10.1111/mafi.12368.

Abbas-Turki, L., B. Diallo, and S. Crépey (2018). XVA principles, nested Monte Carlo strategies, and GPU optimizations. *International Journal of Theoretical and Applied Finance 21*, 1850030.

Abi Jaber, E. and N. De Carvalho (2022). Reconciling rough volatility with jumps. Working Paper.

Abi Jaber, E., M. Larsson, and S. Pulido (2019). Affine volterra processes. *The Annals of Applied Probability 29*(5), 3155–3200.

Achdou, Y. and O. Pironneau (2005). *Computational Methods for Option Pricing*. SIAM.

Ackerer, D., N. Tagasovska, and T. Vatter (2019). Deep smoothing of the implied volatility surface. *Available at SSRN 3402942*.

Albanese, C., S. Crépey, R. Hoskinson, and B. Saadeddine (2021). XVA analysis from the balance sheet. *Quantitative Finance 21*(1), 99–123.

Albanese, C., S. Crépey, and S. Iabichino (2021). A Darwinian theory of model risk. *Risk Magazine*, July pages 72–77.

Albanese, C., S. Crépey, and S. Iabichino (2022). Quantitative reverse stress testing, bottom-up. *Quantitative Finance*. Conditionally accepted.

Albanese, C., S. Iabichino, and P. Mammola (2020). Risk Managing the LIBOR Transition. Preprint. Available at SSRN: https://ssrn.com/abstract=3746939.

Alfeld, P. (1984). A trivariate Clough-Tocher scheme for tetrahedral data. *Computer Aided Geometric Design 1*(2), 169–181.

Alvarez, M., L. Rosasco, and N. Lawrence (2012). Kernels for vector-valued functions: A review. *Foundations and Trends in Machine Learning 4*(3), 195–266.

Alvarez, O. and A. Tourin (1996). Viscosity solutions of nonlinear integro–differential equations. *Annales de l'institut Henri Poincaré (C) Analyse non linéaire 13*(3), 293–317.

Amadori, A. L. (2003). Nonlinear integro–differential evolution problems arising in option pricing: a viscosity solutions approach. *Journal of Differential and Integral Equations 16*(7), 787–811.

Amadori, A. L. (2007). The obstacle problem for nonlinear integro–differential operators arising in option pricing. *Ricerche di matematica 56*(1), 1–17.

An, J. and S. Cho (2015). Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE 2*(1), 1–18.

Anandakrishnan, A., S. Kumar, A. Statnikov, T. Faruquie, and D. Xu (2018). Anomaly detection in finance: editors' introduction. In *KDD 2017 Workshop on Anomaly Detection in Finance*, pp. 1–7.

Andersen, L. and D. Bang (2020). Spike Modeling for Interest Rate Derivatives with an Application to SOFR Caplets. Preprint. Available at SSRN: https://ssrn.com/abstract=3700446.

Andersen, L. and R. Brotherton-Ratcliffe (1996, October). Exact exotics. *Risk Magazine*, 85–89.

Andersen, L. and V. Piterbarg (2010). *Interest Rate Modeling*. Atlantic Financial Press.

Andersen, L. and J. Sidenius (2004). Extensions to the Gaussian copula: Random recovery and random factor loadings. *Journal of Credit Risk Volume 1*(1), 05.

Antonov, A., M. Konikov, and M. Spector (2019). *Modern SABR analytics: formulas and insights for quants, former physicists and mathematicians*. SpringerBriefs in Quantitative Finance.

Avellaneda, M., R. Buff, C. Friedman, N. Grandechamp, L. Kruk, and J. Newman (2001). Weighted monte carlo: a new technique for calibrating asset-pricing models. *International Journal of Theoretical and Applied Finance 4*(01), 91–119.

Avellaneda, M., C. Friedman, R. Holmes, and D. J. Samperi (1997). Calibrating volatility surfaces via relative-entropy minimization. *Applied Mathematical Finance 4*(1), 37–64.

Avellaneda, M., A. Levy, and A. Paras (1995). Pricing and hedging derivative securities in markets with uncertain volatilities. *Applied Mathematical Finance 2*(2), 73–88.

Bachoc, F., A. Lagnoux, A. F. López-Lopera, et al. (2019). Maximum likelihood estimation for Gaussian processes under inequality constraints. *Electronic Journal of Statistics 13*(2), 2921–2969.

Balland, P. (2002). Deterministic implied volatility models. *Quantitative Finance 2*, 31–44.

Bally, V., E. Caballero, B. Fernandez, and N. El-Karoui (2002). Reflected bsde's pde's and variational inequalities. Technical Report 4455, INRIA.

Bally, V. and A. Matoussi (2001). Weak solutions for spdes and backward doubly stochastic differential equations. *Journal of Theoretical Probability 14*(1), 125–164.

Bally, V. and G. Pagès (2003). A quantization algorithm for solving multi-dimensional discrete-time optimal stopping problems. *Bernoulli 9*(6), 1003–1049.

Bally, V., G. Pagès, and J. Printems (2001). A stochastic quantization method for nonlinear problems. *Monte Carlo Methods and Applications 7*(1-2), 21–33.

Barles, G., R. Buckdahn, and E. Pardoux (1997). Backward stochastic differential equations and integral–partial differential equations. *Stochastics & Stochastics Reports 60*, 57–83.

Barles, G. and L. Lesigne (1997). SDE, BSDE and PDE. In N. El Karoui and L. Mazliak (Eds.), *Backward Stochastic differential Equations*, pp. 47–80. Pitman. Research Notes in Mathematics Series 364.

Barles, G. and P. E. Souganidis (1991). Convergence of approximation schemes for fully nonlinear second order equations. *Asymptotic Analysis 4*, 271–283.

Barndorff-Nielsen, O. and A. Basse-O'Connor (2011). Quasi Ornstein-Uhlenbeck processes. *Bernoulli 17*(3), 916–941.

Barrera, D., S. Crépey, B. Diallo, G. Fort, E. Gobet, and U. Stazhynski (2019). Stochastic approximation schemes for economic capital and risk margin computations. *ESAIM: Proceedings and Surveys 65*, 182–218.

Bates, D. (1996). Jumps and stochastic volatility: exchange rate processes implicit in deutsche mark. *9*(1), 69–107.

Bayer, C., P. Friz, and J. Gatheral (2016). Pricing under rough volatility. *Quantitative Finance 16*(6), 887–904.

Becker, S., P. Cheridito, and A. Jentzen (2019). Deep optimal stopping.

Beiglböck, M., P. Henry-Labordère, and F. Penkner (2013). Model-independent bounds for option prices—a mass transport approach. *Finance and Stochastics 17*(3), 477–501.

Bellman, R. (1966). Dynamic programming. *Science 153*(3731), 34–37.

Benamou, J.-D. and Y. Brenier (2000). A computational fluid mechanics solution to the Monge–Kantorovich mass transfer problem. *Numerische Mathematik 84*(3), 375–393.

Bengio, Y. (2012). Practical recommendations for gradient-based training of deep architectures. In *Neural networks: Tricks of the trade*, pp. 437–478. Springer.

Bengio, Y., I. Goodfellow, and A. Courville (2017). *Deep Learning*, Volume 1. Citeseer.

Bengio, Y., P. Lamblin, D. Popovici, and H. Larochelle (2007). Greedy layer-wise training of deep networks. In *Advances in neural information processing systems*, pp. 153–160.

Berestycki, H., J. Busca, and I. Florent (2002). Asymptotics and calibration of local volatility models. *Quantitative finance 2*(1), 61–69.

Bergstra, J. and Y. Bengio (2012). Random search for hyper-parameter optimization. *Journal of machine learning research 13*(Feb), 281–305.

Berndt, A., D. Duffie, and Y. Zhu (2020). Across-the-Curve Credit Spread Indices. Preprint. Available at SSRN: https://ssrn.com/abstract=3662770.

Bielecki, T. R., S. Crépey, and M. Jeanblanc (2010). Up and down credit risk. *Quantitative Finance 10*(10), 1137–1151.

Black, F. and M. Scholes (1973). The pricing of options and corporate liabilities. *Journal of Political Economy 81*(3), 637–54.

Bouchard, B. and J.-F. Chassagneux (2016). *Fundamentals and Advanced Techniques in Derivatives Hedging*. Springer Universitext.

Bouchard, B., I. Ekeland, and N. Touzi (2004). On the Malliavin approach to Monte Carlo approximations of conditional expectations. *Finance & Stochastics 8*, 45–71.

Bouchard, B. and N. Touzi (2004). Discrete-time approximation and Monte-carlo simulation of backward stochastic differential equations. *Stochastic Processes and their applications 111*(2), 175–206.

Bouchard, B. and X. Warin (2010). Monte carlo valuation of American options: new algorithm to improve on existing methods. In R. Carmona, D. Moral, H. P., P., and N. Oudjane (Eds.), *Numerical Methods in Finance*, pp. Chapter 7. Springer.

Bozinovski, S. (2020). Reminder of the first paper on transfer learning in neural networks, 1976. *Informatica 44*(3).

Brace, A., D. Gątarek, and M. Musiela (1997). The market model of interest rate dynamics. *Mathematical Finance 7*, 127–155.

Breeden, D. and R. Litzenberger (1978). Prices of state-contingent claims implicit in option prices. *Journal of Business 51*, 621–651.

Brenier, Y. (1991). Polar factorization and monotone rearrangement of vector-valued functions. *Comm. Pure Appl. Math 44*(4), 375–417.

Brézis, H. (2011). *Functional analysis, Sobolev spaces and partial differential equations*. Springer.

Brigo, D. and F. Mercurio (2007). *Interest rate models-theory and practice: with smile, inflation and credit*. Springer.

Broadie, M. and P. Glasserman (1996). Estimating security price derivatives using simulation. *Management Science 42*(2), 269–285.

Broadie, M. and Ö. Kaya (2006). Exact simulation of stochastic volatility and other affine jump diffusion processes. *Operations research 54*(2), 217–231.

Brunick, G. and S. Shreve (2013). Mimicking an Itô process by a solution of a stochastic differential equation. *Annals of Applied Probability 23*(4), 1584–1628.

Buehler, H., L. Gonon, J. Teichmann, and B. Wood (2019). Deep hedging. *Quantitative Finance 19*(8), 1271–1291.

Cansado, A. and A. Soto (2008). Unsupervised anomaly detection in large databases using Bayesian networks. *Applied Artificial Intelligence 22*(4), 309–330.

Cappozzo, A., F. Greselin, and T. B. Murphy (2020). Anomaly and novelty detection for robust semi-supervised learning. *Statistics and Computing*, 1–27.

Carmona, R. and S. Crépey (2010). Particle methods for the estimation of credit portfolio loss distributions. *International Journal of Theoretical and Applied Finance 13*(04), 577–602.

Carr, P. and D. Madan (2001). Towards a theory of volatility trading. *Option Pricing, Interest Rates and Risk Management, Handbooks in Mathematical Finance*, 458–476.

Carr, P. and D. B. Madan (1999). Option valuation using the fast Fourier transform. *Journal of Computational Finance 2*(4), 61–73.

Chaloner, K. and R. Brant (1988). A Bayesian approach to outlier detection and residual analysis. *Biometrika 75*(4), 651–659.

Chandola, V., A. Banerjee, and V. Kumar (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR) 41*(3), 1–58.

Chataigner, M., A. Cousin, S. Crépey, M. Dixon, and D. Gueye (2021). Short communication: Beyond surrogate modeling: Learning the local volatility via shape constraints. *SIAM Journal on Financial Mathematics 12*(3), SC58–SC69.

Chataigner, M., C. Crépey, and J. Pu (2020). Nowcasting networks. *Journal of Computational Finance 24*(3), 1–39.

Chataigner, M., S. Crépey, and M. Dixon (2020). Deep local volatility. *Risks 8*(82), 18 pages.

Chicago Board Options Exchange (2009). The CBOE volatility index - VIX. last accessed on https://cdn.cboe.com/resources/vix/vixwhite.pdf.

Cont, R. and D. Fournié (2013). Functional Itô calculus and stochastic integral representation of martingales. *The Annals of Probability 41*(1), 109–133.

Cont, R. and P. Tankov (2003a). *Financial Modelling with Jump Processes*. Chapman & Hall.

Cont, R. and P. Tankov (2003b). *Financial Modelling with Jump Processes*. Chapman and Hall/CRC Press.

Cont, R. and E. Voltchkova (2005). A finite difference scheme for option pricing in jump-diffusion and exponential Lévy models. *SIAM J. Numer. Anal. 43*, 1596–1626.

Cousin, A., S. Crépey, and Y. H. Kan (2012). Delta-hedging correlation risk? *Review of Derivatives Research 15*(1), 25–56.

Cousin, A., H. Maatouk, and D. Rullière (2016). Kriging of financial term-structures. *European J. Oper. Res. 255*(2), 631–648.

Cox, A. and D. Hobson (2005). Local martingales, bubbles and option prices. *Finance and Stochastics 9*(4), 477–492.

Cox, J., S. Ross, and M. Rubinstein (1979). Option pricing: a simplified approach. *Journal of Financial Economics 7*(3), 229–263.

Cox, J. C., J. E. Ingersoll, and S. A. Ross (1985). A theory of the term structure of interest rates. *Econometrica 53*, 385–407.

Crandall, M., H. Ishii, and P. Lions (1992). User's guide to viscosity solutions of second order partial differential equations. *Bulletin of the American Mathematical Society 27*, 1–67.

Crépey, S. (2001). *Contribution à des méthodes numériques appliquées à la finance et aux jeux différentiels*. Ph. D. thesis, École Polytechnique and INRIA Sophia Antipolis.

Crépey, S. (2002). Calibration of the local volatility in a trinomial tree using Tikhonov regularization. *Inverse Problems 19*(1), 91.

Crépey, S. (2003a). Calibration of the local volatility in a generalized Black–Scholes model using Tikhonov regularization. *SIAM Journal on Mathematical Analysis 34*(5), 1183–1206.

Crépey, S. (2003b). Calibration of the local volatility in a trinomial tree using Tikhonov regularization. *Inverse Problems 19*, 91–127.

Crépey, S. (2004). Delta-hedging vega risk? *Quantitative Finance 4*(5), 559–579.

Crépey, S. (2013). *Financial Modeling: A Backward Stochastic Differential Equations Perspective*. Springer Finance Textbooks.

Crépey, S., T. R. Bielecki, and D. Brigo (2014). *Counterparty Risk and Funding: A Tale of Two Puzzles*. Taylor & Francis, New York. Chapman & Hall/CRC Financial Mathematics Series.

Crépey, S. and M. Dixon (2020). Gaussian process regression for derivative portfolio modeling and application to CVA computations. *Journal of Computational Finance 24*(1), 47–81.

Crépey, S. and R. Douady (2013). LOIS: credit and liquidity. *Risk Magazine* (June), 82–86.

Crépey, S., W. Sabbagh, and S. Song (2020). When capital is a funding source: The anticipated backward stochastic differential equations of X-Value Adjustments. *SIAM Journal on Financial Mathematics 11*(1), 99–130.

Crépey, S. and S. Song (2017). Invariance properties in the dynamic Gaussian copula model. *ESAIM: Proceedings and Surveys 56*, 22–41.

Crisan, D., K. Manolarakis, and N. Touzi (2010). On the Monte Carlo simulation of BSDEs: An improvement on the Malliavin weights. *Stochastic Processes and their Applications 120*(7), 1133–1158.

Davis, M. H. and P. Varaiya (1974). The multiplicity of an increasing family of σ-fields. *Annals of Probability 2*, 958–963.

Delbaen, F. and W. Schachermayer (2005). *The Mathematics of Arbitrage*. Springer Finance.

Dellacherie, C. (1980). Un survol de la théorie de l'intégrale stochastique. In *Measure Theory Oberwolfach 1979*, pp. 365–395. Springer.

Demeterfi, K., E. Derman, M. Kamal, and J. Zou (1999). A guide to volatility and variance swaps. *The Journal of Derivatives 6*(4), 9–32.

Derman, E. (1999). Regimes of volatility: Some observations on the variations of s&p 500 implied volatilities. *Goldman Sachs Quantitative Strategy Research Notes*.

Derman, E. and I. Kani (1994). Riding on a smile. *Risk Magazine*, February 139–145.

Derman, E., I. Kani, and J. Zou (1996). The local volatility surface: Unlocking the information in index option prices. *Financial analysts journal 52*(4), 25–36.

D'Halluin, Y., P. A. Forsyth, and K. R. Vetzal (2005). Robust numerical methods for contingent claims under jump–diffusion processes. *IMA Journal on Numerical Analysis 25*, 87–112.

Dubois, F. and T. Lelievre (2005). Efficient pricing of Asian options by the PDE approach. *Journal of Computational Finance 8*(2), 55–64.

Duffie, D., D. Filipović, and W. Schachermayer (2003). Affine processes and applications in finance. *Ann. Appl. Probab. 13*, 984–1053.

Duffie, D., J. Pan, and K. Singleton (2000). Transform analysis and asset pricing for affine jump-diffusions. *Econometrica 68*(6), 1343–1376.

Duffy, D. (2006). *Financial Difference Methods in Financial Engineering*. Wiley.

Dugas, C., Y. Bengio, F. Bélisle, C. Nadeau, and R. Garcia (2009). Incorporating functional knowledge in neural networks. *Journal of Machine Learning Research 10*(Jun), 1239–1262.

Dupire, B. (1994b). Pricing with a smile. *Risk 7*, 18–20.

Dupire, B. (2019). Functional Itô calculus. *Quantitative Finance 19*(5), 721–729.

Dupire, B. (January 1994a). Pricing with a Smile. *Risk Magazine*, 18–20.

E, W., J. Han, and A. Jentzen (2017). Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. *Communications in Mathematics and Statistics 5*(4), 370–398.

El Karoui, N. (1981). Les aspects probabilistes du contrôle stochastique. In *École d'été de Probabilités de Saint-Flour IX-1979*, pp. 73–238. Springer.

El Karoui, N. and Y. Jiao (2009). Stein's method and zero bias transformation for CDO tranche pricing. *Finance and Stochastics 13*(2), 151–180.

El Karoui, N., Y. Jiao, and D. Kurtz (2008). Gauss and Poisson approximation: applications to CDO tranches pricing. *Journal of Computational Finance 12*(2), 31–58.

Élie, R. (2006). *Contrôle stochastique et méthodes numériques en finance mathématique*. Ph. D. thesis, University Paris-Dauphine. https://pastel.archives-ouvertes.fr/tel-00122883/file/thesis.pdf.

Engl, H., M. Hanke, and A. Neubauer (1996a). *Regularization of Inverse Problems*. Kluwer.

Engl, H. W., M. Hanke, and A. Neubauer (1996b). *Regularization of inverse problems*, Volume 375. Springer Science & Business Media.

Engl, H. W., K. Kunisch, and A. Neubauer (1989). Convergence rates for Tikhonov regularisation of non-linear ill-posed problems. *Inverse problems 5*(4), 523.

Ern, A., S. Villeneuve, and A. Zanette (2004). Adaptive finite element methods for local volatility european option pricing. *International Journal of Theoretical and Applied Finance 7*(6), 659–684.

Ethier, H. and T. Kurtz (1986). *Markov Processes. Characterization and Convergence*. Wiley.

Fang, F. and C. W. Oosterlee (2008). A novel pricing method for european options based on fourier-cosine series expansions. *SIAM J. SCI. COMPUT*.

Filipovic, D. (2009). *Term-Structure Models. A Graduate Course.* Springer.

Filipovic, D. and E. Mayerhofer (2009). Affine diffusion processes: theory and applications. *Advanced financial modelling 8*, 1–40.

Filipović, D. and A. B. Trolle (2013). The term structure of interbank risk. *Journal of Financial Economics 109*(3), 707–733.

Fleming, W. and H. Soner (2006). *Controlled Markov processes and viscosity solutions, second edition*. Springer.

Fournié, E., J.-M. Lasry, J. Lebuchoux, and P.-L. Lions (2001). Applications of Malliavin calculus to Monte Carlo methods in finance II. *Finance & Stochastics 5*, 201–236.

Fournié, E., J.-M. Lasry, J. Lebuchoux, P.-L. Lions, and N. Touzi (1999). An application of malliavin calculus to monte carlo methods in finance. *Finance & Stochastics 3*, 391–412.

Friedman, A. (1983). *Partial Differential Equations of Parabolic Type*. Prentice Hall.

Fukasawa, M., B. Horvath, and P. Tankov (2021). Hedging under rough volatility. arXiv:2105.04073.

Garcia, R. and R. Gençay (2000). Pricing and hedging derivative securities with neural networks and a homogeneity hint. *Journal of Econometrics 94*(1-2), 93–115.

Gardner, J., G. Pleiss, R. Wu, K. Weinberger, and A. Wilson (2018). Product kernel interpolation for scalable gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pp. 1407–1416.

Gaß, M., K. Glau, and M. Mair (2017). Magic points in finance: Empirical integration for parametric option pricing. *SIAM Journal on Financial Mathematics 8*, 766–803.

Gatheral, J. (2008). Consistent modeling of SPX and VIX options. In *Bachelier congress*, Volume 37, pp. 39–51.

Gatheral, J. (2011). *The volatility surface: a practitioner's guide*. Wiley.

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

Gatheral, J., T. Jaisson, and M. Rosenbaum (2018). Volatility is rough. *Quantitative finance 18*(6), 933–949.

Giles, M. and P. Glasserman (2006, January). Smoking adjoints: Fast Monte Carlo Greeks. *Risk Magazine*, 88–92.

Glasserman, P. (2003). *Monte Carlo methods in financial engineering*. Springer.

Glorot, X. and Y. Bengio (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256.

Gobet, E., J.-P. Lemor, and X. Warin (2005). A regression-based Monte Carlo method to solve backward stochastic differential equations. *The Annals of Applied Probability 15*(3), 2172–2202.

Goix, N., A. Sabourin, and S. Clémençon (2015). On anomaly ranking and excess-mass curves. In *Artificial Intelligence and Statistics*, pp. 287–295.

Goix, N., A. Sabourin, and S. Clémençon (2017). Sparse representation of multivariate extremes with applications to anomaly detection. *Journal of Multivariate Analysis 161*, 12–31.

Goodfellow, I., Y. Bengio, and A. Courville (2016). *Deep Learning*. MIT Press.

Gramacy, R. and D. Apley (2015). Local gaussian process approximation for large computer experiments. *Journal of Computational and Graphical Statistics 24*(2), 561–578.

Guo, I. and G. Loeper (2021). Path dependent optimal transport and model calibration on exotic derivatives. *The Annals of Applied Probability 31*(3), 1232–1263.

Guo, I., G. Loeper, J. Oblój, and S. Wang (2020). Joint modelling and calibration of SPX and VIX by optimal transport. *Available at SSRN 3568998*.

Guo, I., G. Loeper, J. Oblój, and S. Wang (2021). Optimal transport for model calibration. arXiv:2107.01978.

Guo, I., G. Loeper, and S. Wang (2019a). Calibration of local-stochastic volatility models by optimal transport. arXiv:1906.06478.

Guo, I., G. Loeper, and S. Wang (2019b). Local volatility calibration by optimal transport. In 2017 MATRIX annals*, volume 2 of* MATRIX Book Series, pp. 51–64.

Gupta, A. and C. Reisinger (2014). Robust calibration of financial models using bayesian estimators. *Journal of Computational Finance 17*(4), 3–36.

Guyon, J. (2020). The joint S&P 500/VIX smile calibration puzzle solved. *Risk Magazine*.

Guyon, J. (2021). Dispersion-constrained martingale Schrödinger problems and the exact joint S&P 500/VIX smile calibration puzzle. *Available at SSRN 3853237*, April.

Guyon, J. and P. Henry-Labordère (2012). *Nonlinear Pricing Methods in Quantitative Finance*. Chapman & Hall/CRC Financial Mathematics Series.

Györfi, L., M. Kohler, A. Krzyzak, and H. Walk (2002). *A Distribution–Free Theory of Nonparametric Regression*. Springer.

Hagan, P., D. Kumar, A. Lesniewski, and D. Woodward (2002). Managing smile risk. *The Best of Wilmott 1*, 249–296.

Hastie, T., R. Mazumder, J. D. Lee, and R. Zadeh (2015). Matrix completion and low-rank svd via fast alternating least squares. *The Journal of Machine Learning Research 16*(1), 3367–3402.

Hawkins, D. M. (1980). *Identification of outliers*, Volume 11. Springer.

He, S.-W., J.-G. Wang, and J.-A. Yan (1992). *Semimartingale Theory and Stochastic Calculus*. Science Press and CRC Press Inc.

Heath, D., R. Jarrow, and A. Morton (1992). Bond pricing and the term structure of interest rates: a new methodology for contingent claims valuation. *Econometrica 60*, 77–105.

Henrard, M. (2014). *Interest rate modelling in the multi-curve framework: Foundations, evolution and implementation*. Applied Quantitative Finance. Palgrave Macmillan.

Henrard, M. (2019). LIBOR Fallback and Quantitative Finance. *Risks 7*(3), 88.

Henry-Labordère, P. (2012). Cutting CVA's complexity. *Risk Magazine*, July 67–73.

Henry-Labordère, P. (2019). From (martingale) Schrödinger bridges to a new class of stochastic volatility models. arXiv:1904.04554.

Henry-Labordere, P., N. Oudjane, X. Tan, N. Touzi, X. Warin, et al. (2019). Branching diffusion representation of semilinear pdes and monte carlo approximation. *55*(1), 184–210.

Henry-Labordere, P., X. Tan, and N. Touzi (2014). A numerical algorithm for a class of BSDEs via the branching process. *Stochastic Processes and their Applications 124*(2), 1112–1140.

Henry-Labordère, P. and N. Touzi (2014, February). A stochastic control approach to no-arbitrage bounds given marginals, with an application to lookback options. *The Annals of Applied Probability 24*(1), 312–336.

Heston, S. (1993). A closed–form solution for options with stochastic volatility with applications to bond and currency options. *The Review of Financial Studies 6*(2), 327–343.

Hilber, N., O. Reichmann, C. Schwab, and C. Winter (2013). *Computational methods for quantitative finance: Finite element methods for derivative pricing.* Springer Science & Business Media.

Hinton, G. E., S. Osindero, and Y.-W. Teh (2006). A fast learning algorithm for deep belief nets. *Neural computation 18*(7), 1527–1554.

Horvath, B., A. Muguruza, and M. Tomas (2021). Deep learning volatility. *Quantitative Finance 21*(1), 11–27.

Hout, K. J. and B. D. Welfert (2007). Stability of ADI schemes applied to convection-diffusion equations with mixed derivative terms. *Applied Numerical Mathematics 57*, 19–35.

Huang, C.-F. (1985). Information structures and viable price systems. *Journal of Mathematical Economics 14*(3), 215–240.

Hull, J. and A. White (1990). Pricing interest rate derivative securities. *Review of Financial Studies 4*.

Hull, J. C. and A. D. White (2000). Forward rate volatilities, swap rate volatilities, and implementation of the LIBOR market model. *The Journal of Fixed Income 10*(2), 46–62.

Huré, C., H. Pham, and C. Warin (2020). Deep backward schemes for high-dimensional nonlinear PDEs. *Mathematics of Computation 89*(324), 1547–1579.

Jacod, J. and A. N. Shiryaev (2003). *Limit Theorems for Stochastic Processes* (2nd ed.). Springer.

Jacquier, A. and M. Keller-Ressel (2018). Implied volatility in strict local martingale models. *SIAM Journal on Financial Mathematics 9*(1), 171–189.

Jaillet, P., D. Lamberton, and B. Lapeyre (1990). Variational inequalities and the pricing of American options. *Acta Applicandae Mathematicae 21*, 263–289.

Jakobsen, E. R. and K. H. Karlsen (2005). Continuous dependence estimates for viscosity solutions of integro–pdes. *Journal of Differential Equations 212*(2), 278–318.

Jakobsen, E. R. and K. H. Karlsen (2006). A "maximum principle for semi–continuous functions" applicable to integro–partial differential equations. *Nonlinear Differential Equations Appl. 13*, 137–165.

Jeanblanc, M., M. Yor, and M. Chesney (2009). *Mathematical methods for financial markets.* Springer.

Jex, M., R. Henderson, and D. Wang (1999). Pricing exotics under the smile. *Risk Magazine* (12), 72–75.

Kahl, C. and P. Jäckel (2005). Not-so-complex logarithms in the Heston model. *Wilmott magazine 19*(9), 94–103.

Kamrad, B. and P. Ritchken (1991). Multinomial approximating models for options with k state variables. *Management Science 37*, 1640–1652.

Kantorovich, L. V. (1948). On a problem of Monge (in Russian). *Uspekhi Matematicheskikh Nauk 3*, 255–226.

Karatzas, I. and S. Shreve (1991). *Brownian Motion and Stochastic Calculus* (2nd ed.). Springer Graduate Texts in Mathematics.

Keller-Ressel, M. (2008). *Affine Processes – theory and applications to finance* . Ph. D. thesis, TU Vienna.

Kemna, G. Z. and A. C. Vorst (1990). A pricing method for options based on average asset values. *Journal of Banking and Finance 14*(1), 113–129.

Kingma, D. P. and J. Ba (2014). Adam: A method for stochastic optimization. arXiv:1412.6980.

Klinglera, S. and O. Syrstadb (2021). Life after LIBOR. Preprint. Forthcoming in *Journal of Financial Economics*.

Kushner, H. and B. Dupuis (1992). *Numerical Methods for Stochastic Control Problems in Continuous Time*. Springer.

Lakhina, S., S. Joseph, and B. Verma (2010). Feature reduction using principal component analysis for effective anomaly–based intrusion detection on NSL-KDD.

Lamberton, D. and B. Lapeyre (1996). *Introduction to Stochastic Calculus Applied to Finance*. Chapman & Hall.

Lapeyre, B., E. Pardoux, and R. Sentis (2003). *Introduction to Monte–Carlo Methods for Transport and Diffusion Equations*. Oxford University Press.

Lapeyre, B. and E. Temam (2001). Competitive Monte Carlo methods for the pricing of Asian options. *Journal of Computational Finance 5*(1), 39–57.

L'Ecuyer, P. (1994). Uniform random number generation. *The Annals of Operations Research 53*, 77–120.

L'Ecuyer, P. (1998). Random number generation. In J. Banks (Ed.), *The Hanbook of Simulation*, pp. 93–137. Wiley. Chapter 4.

Li, D. (2000). On default correlation: A copula function approach. *Journal of Fixed Income 9*(4), 43–54.

Lipton, A. (2002). Assets with jumps. *Risk 15*(9), 149–153.

Livieri, G., S. Mouti, A. Pallavicini, and M. Rosenbaum (2018). Rough volatility: evidence from option prices. *IISE transactions 50*(9), 767–776.

Longstaff, F. A. and E. S. Schwartz (2001). Valuing American options by simulation: A simple least-squares approach. *The Review of Financial Studies 14*(1), 113–147.

Ludkovski, M. and Y. Saporito (2020). Krighedge: Gaussian process surrogates for delta hedging. arXiv:2010.08407.

Lyashenko, A. and F. Mercurio (2019). Libor replacement: a modelling framework for in-arrears term rates. *Risk Magazine*, June. long preprint version ssrn.3482132.

Maatouk, H. and X. Bay (2017). Gaussian process emulators for computer experiments with inequality constraints. *Mathematical Geosciences 49*(5), 557–582. Available as HAL preprint hal-01096751v2.

MacKay, D. J. (1998). Introduction to gaussian processes. In C. M. Bishop (Ed.), *Neural Networks and Machine Learning*. Springer-Verlag.

MacKay, D. J. and D. J. Mac Kay (2003). *Information theory, inference and learning algorithms*. Cambridge university press.

Marco, S. D. and P. Henry-Labordère (2015). Linking vanillas and VIX options: a constrained martingale optimal transport problem. *SIAM Journal on Financial Mathematics 6*(1), 1171–1194.

Masci, J., U. Meier, D. Cireşan, and J. Schmidhuber (2011). Stacked convolutional auto-encoders for hierarchical feature extraction. In *International Conference on Artificial Neural Networks*, pp. 52–59. Springer.

Matache, A.-M., T. Von Petersdorff, and C. Schwab (2004). Fast deterministic pricing of options on lévy driven assets. *Mathematical Modelling and Numerical Analysis 38*(1), 37–72.

Melkumyan, A. and F. Ramos (2011). Multi-kernel gaussian processes. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Two*, IJCAI'11, pp. 1408–1413. AAAI Press.

Mercurio, F. (2010). Interest rates and the credit crunch: new formulas and market models. Technical report, Bloomberg Portfolio Research Paper No. 2010-01-FRONTIERS.

Merton, R. C. (1973). Theory of rational option pricing. *Bell Journal of Economics and Management Science 4*, 141–183.

Merton, R. C. (1976). Option pricing when underlying stock returns are discontinuous. *Journal of Financial Economics 3*(1-2), 125–144.

Metropolis, N. and S. Ulam (1949). The monte carlo method. *Journal of the American Statistical Association 44*, 335–341.

Meyer, P.-A. (1976). Un cours sur les intégrales stochastiques (exposés 1 à 6). *Séminaire de probabilités de Strasbourg 10*, 245–400.

Micchelli, C. A., Y. Xu, and H. Zhang (2006, December). Universal kernels. *J. Mach. Learn. Res. 7*, 2651–2667.

Monge, G. (1781). *Mémoire sur la théorie des déblais et des remblais*.

Morokoff, W. and R. Caflish (1994). Quasi–random sequences and their discrepancies. *SIAM Journal of Scientific Computing 5*(6), 1251–1279.

Morton, K. and D. Mayers (1994). *Numerical Solution of Partial Differential Equations*. Cambridge University Press.

Morton, K. and D. Mayers (2005). *Numerical solution of partial differential equations: an introduction*. Cambridge university press.

Murphy, K. (2012). *Machine Learning: A Probabilistic Perspective*. The MIT Press.

Musiela, M. (2022). My journey through finance and stochastics. *Finance and Stochastics 26*(1), 33–58.

Neal, R. M. (1996). *Bayesian learning for neural networks*, Volume 118 of *Lecture Notes in Statistics*. Springer.

Nguyen, L. T., J. Kim, and B. Shim (2019). Low-rank matrix completion: A contemporary survey. *IEEE Access 7*, 94215–94237.

Niederreiter, H. (1987). Points sets ans sequences with small discrepancy. *Monatshefte Fur Mathematik 104*, 273–337.

Niederreiter, H. (1992). *Random Number Generation and Quasi–Monte Carlo Methods*. SIAM.

NVIDIA Corporation (2020). Programming guide: Cuda toolkit documentation. https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html. Accessed: 2020-04-28.

O'Kane, D. and M. Livesey (2004). Base correlation explained. *Lehman Brothers, Fixed Income Quantitative Credit Research 346*.

Omar, S., A. Ngadi, and H. H. Jebur (2013). Machine learning techniques for anomaly detection: an overview. *International Journal of Computer Applications 79*(2).

Pan, S. J. and Q. Yang (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering 22*(10), 1345–1359.

Patcha, A. and J.-M. Park (2007). An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer networks 51*(12), 3448–3470.

Peaceman, D. and H. Rachford (1955). The numerical solution of parabolic and elliptic differential equations. *SIAM Journal on Applied Mathematics 3*, 28–42.

Pillonetto, G., F. Dinuzzo, and G. D. Nicolao (2010, Feb). Bayesian online multitask learning of gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence 32*(2), 193–205.

Piterbarg, V. (2020a). Arc-sine law and the libor reform. *Available at SSRN 3684535*.

Piterbarg, V. (2020b). Interest rates benchmark reform and options markets. *Available at SSRN 3537925*.

Pleiss, G., J. R. Gardner, K. Q. Weinberger, and A. G. Wilson (2018). Constant-time predictive distributions for gaussian processes. *CoRR abs/1803.06058*.

Protter, P. (2001). A partial introduction to financial asset pricing theory. *Stochastic processes and their applications 91*(2), 169–203.

Protter, P. (2004). *Stochastic Integration and Differential Equations* (3rd ed.). Springer.

Rasmussen, C. E. and Z. Ghahramani (2001). Occam's razor. In *In Advances in Neural Information Processing Systems 13*, pp. 294–300. MIT Press.

Rasmussen, C. E. and C. K. I. Williams (2006). *Gaussian Processes for Machine Learning*. MIT Press.

Rebonato, R. (2005). *Volatility and correlation: the perfect hedger and the fox*. John Wiley & Sons.

Rebonato, R., K. McKay, and R. White (2009). *The SABR/Libor Market Model: Pricing Calibration and Hedging for Complex Interest–Rate Derivatives*. Wiley.

Reisinger, C. (2013). Analysis of linear difference schemes in the sparse grid combination technique. *IMA Journal of Numerical Analysis 33*(2), 544–581.

Revuz, D. and M. Yor (1999). *Continuous Martingales and Brownian Motion* (3rd ed.). Springer.

Richtmyer, R. and K. Morton (1967). *Difference Methods for Initial-Value Problems*. Wiley-Interscience.

Ro, K., C. Zou, Z. Wang, and G. Yin (2015). Outlier detection for high-dimensional data. *Biometrika 102*(3), 589–599.

Robbins, H. and D. Siegmund (1971). A convergence theorem for non negative almost supermartingales and some applications. In *Optimizing methods in statistics (Proc. Sympos., Ohio State Univ., Columbus, Ohio, 1971)*, pp. 233–257. Academic Press, New York.

Rockafellar, R. and S. Uryasev (2000). Optimization of conditional value-at-risk. *Journal of risk 2*, 21–42.

Rocke, D. M. and D. L. Woodruff (1996). Identification of outliers in multivariate data. *Journal of the American Statistical Association 91*(435), 1047–1061.

Rogers, L. (2002). Monte carlo valuation of American options. *Mathematical Finance 12*, 271–286.

Rogers, L. and Z. Shi (1995). The value of an Asian option. *Journal of Applied Probability 32*(4), 1077–1088.

Roper, M. (2010). Arbitrage free implied volatility surfaces. Preprint University of Sydney available at https://talus.maths.usyd.edu.au/u/pubs/publist/preprints/2010/roper-9.pdf.

Rudin, W. (1987). *Real and Complex Analysis* (3rd ed.). McGraw-Hill.

Ruf, J. and W. Wang (2020). Neural networks for option pricing and hedging: a literature review. *Journal of Computational Finance 24*(1).

Saad, Y. and M. Schultz (1986). GMRES: A generalized minimal residual algorithm for solving non-symmetric linear system. *SIAM Journal on Scientific Computing 7*, 856–869.

Samperi, D. J. (2002). Calibrating a diffusion pricing model with uncertain volatility: Regularization and stability. *Mathematical Finance 12*, 71–87.

Savitsky, T., M. Vannucci, and N. Sha (2011, 02). Variable selection for nonparametric gaussian process priors: Models and computational strategies. *Statist. Sci. 26*(1), 130–149.

Schachermayer, W. and J. Teichmann (2008). How close are the option pricing formulas of Bachelier and Black–Merton–Scholes? *Mathematical Finance 18*(1), 155–170.

Schlegl, T., P. Seeböck, S. M. Waldstein, G. Langs, and U. Schmidt-Erfurth (2019). f-anogan: Fast unsupervised anomaly detection with generative adversarial networks. *Medical image analysis 54*, 30–44.

Scholkopf, B. and A. J. Smola (2001). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell system technical journal 27*(3), 379–423.

Shorten, C. and T. Khoshgoftaar (2019). A survey on image data augmentation for deep learning. *Journal of Big Data 6*(1), 60.

Sobol, I. M. (1976). The distribution of points in a cube and the approximate evaluation of integrals. *USSR Computational Mathematics and Mathematical Physics 16*, 236–242.

Soner, H. M., N. Touzi, and J. Zhang (2012). Wellposedness of second order backward SDEs. *Probability Theory and Related Fields 153*(1-2), 149–190.

Strub, F. and J. Mary (2015). Collaborative filtering with stacked denoising autoencoders and sparse inputs. In *NIPS workshop on machine learning for eCommerce*.

Tan, X. and N. Touzi (2013). Optimal transportation under controlled stochastic dynamics. *The Annals of Probability 41*(5), 3201–3240.

Tavella, D. and C. Randall (2000). *Financial Instruments: The Finite Difference Method*. Wiley.

Tegnér, M. and S. Roberts (2021). A probabilistic approach to nonparametric local volatility. *Journal of Computational Finance*. Forthcoming.

Tian, Y., Z. Zhu, G. Lee, F. Klebaner, and K. Hamza (2015). Calibrating and pricing with a stochastic-local volatility model. *Journal of Derivatives 22*, 3.

Trolle, A. B. and E. S. Schwartz (2010, November). An empirical analysis of the swaption cube. Working Paper 16549, National Bureau of Economic Research.

Tschannen, M., O. Bachem, and M. Lucic (2018). Recent advances in autoencoder-based representation learning. *arXiv:1812.05069*.

Tsitsiklis, J. N. and B. Van Roy (2001). Regression methods for pricing complex american-style options. *IEEE Transactions on Neural Networks 12*(4), 694–703.

Vasicek, O. (1977). An equilibrium characterization of the term structure. *Journal of financial economics 5*(2), 177–188.

Vasicek, O. (1991). Limiting loan loss probability distribution. *KMV corporation*.

Villani, C. (2009). *Optimal transport: old and new*. Springer.

Villani, C. (2021). *Topics in optimal transportation*, Volume 58. American Mathematical Soc.

Villeneuve, S. and A. Zanette (2002). Parabolic ADI methods for pricing American option on two stocks. *Mathematics of Operations Research 27*(1), 121–149.

Wilmott, P. (1998). *Derivatives: the theory and practice of financial engineering*. Wiley.

Wilmott, P. (2002, December). Cliquet options and volatility models. *Wilmott Magazine*, 78–83.

Wilmott, P., J. Dewynne, and S. Howison (1993). *Option Pricing*. Oxford Financial Press.

Windcliff, H., P. Forsyth, and K. Vetzal (2006a). Numerical methods for valuing cliquet options. *Applied Mathematical Finance 13*, 353–386.

Windcliff, H., P. Forsyth, and K. Vetzal (2006b). Pricing methods and hedging strategies for volatility derivatives. *Journal of Banking and Finance* (30), 409–431.

Yang, J., T. Hurd, and X. Zhang (2006). Saddlepoint approximation method for pricing CDOs. *Journal of Computational Finance 10*(1), 1.

Zhu, Y., X. Wu, and I. Chern (2004). *Derivative Securities and Difference Methods*. Springer.

Zvan, R., P. A. Forsyth, and K. R. Vetzal (1998). Robust numerical methods for PDE models of Asian option. *Journal of Computational Finance 1*(2), 39–78.