A backward Monte Carlo approach to exotic option pricing†

G. BORMETTI1, G. CALLEGARO2, G. LIVIERI3 and A. PALLAVICINI4,5

1Department of Mathematics, University of Bologna, Piazza di Porta San Donato 5, 40126 Bologna, Italy, email: giacomo.bormetti@unibo.it
2Department of Mathematics, University of Padova, via Trieste 63, 35121 Padova, Italy, email: gcallega@math.unipd.it
3Scuola Normale Superiore, Piazza dei Cavalieri 7, 56126 Pisa, Italy email: giulia.livieri@sns.it
4Department of Mathematics, Imperial College, London SW7 2AZ, UK, email: a.pallavicini@imperial.ac.uk
5Banca IMI, Largo Mattioli 3, 20121 Milano, Italy

(Received 17 December 2015; revised 13 February 2017; accepted 11 March 2017; first published online 12 April 2017)

We propose a novel algorithm which allows to sample paths from an underlying price process in a local volatility model and to achieve a substantial variance reduction when pricing exotic options. The new algorithm relies on the construction of a discrete multinomial tree. The crucial feature of our approach is that – in a similar spirit to the Brownian Bridge – each random path runs backward from a terminal fixed point to the initial spot price. We characterize the tree in two alternative ways: (i) in terms of the optimal grids originating from the Recursive Marginal Quantization algorithm, (ii) following an approach inspired by the finite difference approximation of the diffusion’s infinitesimal generator. We assess the reliability of the new methodology comparing the performance of both approaches and benchmarking them with competitor Monte Carlo methods.

Key words: Monte Carlo methods (65C05), Markov chains (60J10), Computational methods in Markov chains (60J22), Derivative securities (91G20), Numerical methods (including Monte Carlo methods) (91G60).

1 Introduction

Pricing financial derivatives typically requires to solve two main issues. In the first place, the choice of a flexible model for the stochastic evolution of the underlying asset price. At this point, a common trade-off arises as models which describe the historical dynamics of the asset price with adequate realism are usually unable to precisely match volatility smiles observed in the option market (see, for instance, Bouchaud & Potters, 2003; Gatheral, 2011). Second, once a reasonable candidate has been identified, there is the need to develop fast, accurate, and possibly flexible numerical methods (see Clewlow & Strickland, 1996; Hull, 2006; Wilmott et al., 1993) to price financial derivatives. As regards, the former

† GB and GL acknowledge research support from the Scuola Normale Superiore Grant SNS_14_BORMETTI.
A backward Monte Carlo approach to exotic option pricing

point, local volatility (LV) models have become very popular since their introduction by Dupire (1994), and Derman et al. (1996). Even though the legitimate use of LV models for the description of the asset dynamics is highly questionable, the ability to self-consistently reproduce volatility smiles implied by the market motivates their widespread diffusion among practitioners. Since calibration à la Dupire (see Dupire, 1994) of LV models assumes the unrealistic availability of a continuum of vanilla option prices across different strikes and maturities (see Kahale, 2004), recent years have seen the emergence of a growing strand of literature dealing with this problem (see, for instance, Andreasen & Huge, 2011; Coleman et al., 1999; Kahale, 2004; Lipton & Sepp, 2011; Pallavicini, 2016; Reghai et al., 2012). In the present paper, we fix the calibration following the latest achievements and we solely focus on the latter issue. Specifically, our goal is to design a novel pricing algorithm based on the Monte Carlo (MC) approach able to achieve a sizeable variance reduction with respect to competitor approaches.

The main result of this paper is the development of a flexible and efficient pricing algorithm – termed as the backward MC algorithm – which runs backward on top of a multinomial tree. The flexibility of this algorithm permits to price generic payoffs without the need of designing tailor-made solutions for each payoff specification. This feature is inherited directly from the MC approach (see Glasserman, 2004, for an almost exhaustive survey of MC methods in finance). The efficiency, instead, is linked primarily to the backward movement on the multinomial tree. Indeed, our approach combines both advantages of stratified sampling MC and the Brownian Bridge construction (see, for instance, Bormetti et al., 2006; Glasserman, 2004), extending them to more general financial-asset dynamics than the simplistic assumptions of Black, Scholes, and Merton (see Black & Scholes, 1973; Merton, 1973). The second purpose of this paper – minor in relative terms with respect to the first one – is to investigate an alternative scheme for the implementation of the Recursive Marginal Quantization Algorithm (henceforth RMQA). The RMQA is a recursive algorithm which allows to approximate a continuous time diffusion by means of a discrete-time Markov Chain defined on a finite grid of points. The alternative scheme, employed at each step of the RMQA, is based on the Lloyd I method (see Kieffer, 1982) in combination with the Anderson acceleration Algorithm (see Anderson, 1965; Walker & Ni, 2011) developed to solve fixed-point problems. The accelerated scheme permits us to speed up the linear rate of convergence of the Lloyd I method (see Kieffer, 1982), and more importantly, to fix some flaws of previous RMQA implementations highlighted in Callegaro et al. (2015).

In more detail, a discrete-time Markov Chain approximation of the asset price dynamics can be achieved by introducing at each time step two quantities: (i) a grid for the possible values that the price can take, and (ii) the transition probabilities to propagate from one state to another one. Among the approaches discussed in the literature for computing these quantities, in the present paper, we analyse and extend two of them. The first approach quantizes via the RMQA the Euler–Maruyama approximation of the Stochastic Differential Equation (SDE) modelling the underlying asset price. The RMQA has been introduced in Pagès & Sagna (2015) to compute vanilla call and put options prices in a pseudo Constant Elasticity of Variance (CEV) LV model. In Callegaro et al. (2015), authors employ it to calibrate a Quadratic Normal LV model. The alternative approach, instead, discretizes in an appropriate way the infinitesimal Markov generator of the underlying
diffusion by means of a finite difference scheme (see Albanese & Mijatović, 2007; Kushner & Dupuis, 2001, for a detailed discussion of theoretical convergence results). We name the latter approach Large Time Step Algorithm, henceforth LTSA. In Reghai et al. (2012), authors implement a modified version of the LTSA to price discrete look-back options in a CEV model, whereas in Albanese et al. (2009) they employ the LTSA idea to price a particular class of path-dependent payoffs termed Abelian payoffs. More specifically, they incorporate the path-dependency feature – in the specific case whether or not the underlying asset price hits a specified level over the life of the option – within the Markov generator. The joint transition probability matrix is then recovered as the solution of a payoff specific matrix equation. The RMQA and LTSA present two major differences which can be summarized as follows: (i) the RMQA permits to recover the optimal – according to a specific criterion (see Pagès & Printems, 2005) – multinomial grid, whereas the LTSA works with an a priori user-specified grid, (ii) the LTSA necessitates less computational burden than the RMQA when pricing financial derivatives products whose payoff requires the observation of the underlying on a predefined finite set of dates. Unfortunately, this result holds only for a piecewise time-homogeneous LV dynamics.

The usage in both equity and foreign exchange (FX) markets of LV models is largely motivated by the flexibility of the approach which allows the exact calibration to the whole volatility surface. Moreover, the accurate re-pricing of plain vanilla instruments and of most liquid European options, together with the stable computation of the option sensitivity to model parameters and the availability of specific calibration procedures, make the LV modelling approach a popular choice. The LV models are also employed in practise to evaluate Asian options and other path-dependent options, although more sophisticated stochastic LV models are adopted when the path-dependency risk dominates the smile risk (see, for instance, Predota, 2005). We refer to (Ren et al., 2007) for more details on calibration and on pricing with LV models. The price of path-dependent derivative products is then computed either solving numerically a Partial Differential Equation or via MC methods. The Partial Differential Equation approach is computationally efficient but it requires the definition of a payoff specific pricing equation (see Wilmott et al., 1993, for an extensive survey on Partial Differential Equation approach in financial contexts). Moreover, some options with exotic payoffs and exercise rules are subtle to price even within the Black, Scholes, and Merton framework (Dewynne & Shaw, 2008; Siyanko, 2012). On the other hand, standard MC method suffers from some inefficiency – especially when pricing Out-of-The-Money (OTM) options – since a relevant number of sampled paths does not contribute to the option payoff. However, the MC approach is extremely flexible and several numerical techniques have been introduced to reduce the variance of the MC estimator (see Clewlow & Strickland, 1996; Glasserman, 2004). The backward MC algorithm pursues this task.

In this paper, we consider the FX market, where we can trade spot and forward contracts along with vanilla and exotic options. In particular, we model the EUR/USD rate using an LV dynamics. The calibration procedure is the one employed in Reghai et al. (2012) for the equity market and in Pallavicini (2016) for the FX market. Specifically, we calibrate the stochastic dynamics for the EUR/USD rate in order to reproduce the observed implied volatilities with a one basis point tolerance while the extrapolation to implied volatilities for maturities not quoted by the market is achieved by means of a
piecewise time-homogeneous LV model. In order to show the competitive performances of the backward MC algorithm, we compute the price of different kinds of options. We do not price basket options, but we only focus on derivatives written on a single underlying asset considering Asian calls, up-out barrier calls, and auto-callable options. We show that these instruments can be priced more effectively by simulating the discrete-time Markov Chain approximation of the diffusive dynamics from the maturity back to the initial date. In these cases, indeed, the backward MC algorithm leads to a significant reduction of the MC variance. We leave to a future work the extension of our analysis to stochastic LV models.

The rest of the paper is organized as follows. In Section 2, we introduce the key ideas of the backward MC algorithm on a multinomial tree. Section 3 presents the alternative schemes of implementation based on the RMQA and LTSA, and details the numerical investigations testing the performance of both approaches. Section 4 presents a piecewise time-homogeneous LV model for the FX market and reports the pricing performances of the backward MC algorithm, benchmarking them with different MC algorithms. Finally, in Section 5, we conclude and draw possible perspectives.

Notation We use the symbol “:=” for the definition of new quantities.

2 The backward Monte Carlo algorithm

First of all, let us introduce our working framework. We consider a probability space \((\Omega, \mathcal{F}, \mathbb{P})\), a given time horizon \(T > 0\) and a stochastic process \(X = (X_t)_{t \in [0,T]}\) describing the evolution of the asset price. We suppose that the market is complete, so that under the unique risk-neutral probability measure \(\mathbb{Q}\), \(X\) has a Markovian dynamics described by the following SDE:

\[
\begin{align*}
    dX_t &= b(t, X_t) \, dt + \sigma(t, X_t) \, dW_t, \\
    X_0 &= x_0 \in \mathbb{R},
\end{align*}
\]

where \((W_t)_{t \in [0,T]}\) is a standard one-dimensional \(\mathbb{Q}\)-Brownian motion, and \(b : [0, T] \times \mathbb{R} \to \mathbb{R}\) and \(\sigma : [0, T] \times \mathbb{R} \to \mathbb{R}_+\) are two measurable functions satisfying the usual conditions ensuring the existence and uniqueness of a (strong) solution to the SDE (2.1). Besides, we consider deterministic interest rates. Henceforth, we will always work under the risk-neutral probability measure \(\mathbb{Q}\), since we focus on the pricing of derivative securities written on \(X\). Specifically, we are interested in pricing financial derivative products whose payoff may depend on the whole path followed by the underlying asset, i.e. path-dependent options.

Let us now motivate the introduction of our novel pricing algorithm. Even in the classical Black, Scholes, and Merton (see Black & Scholes, 1973; Merton, 1973) framework, when pricing financial derivatives the actual analytical tractability is limited to plain vanilla call and put options and to few other cases (for instance, see the discussion in Hui et al., 2000; Vecer & Xu, 2004). Such circumstances motivate the quest for general and reliable pricing algorithms able to handle more complex contingent claims in more realistic stochastic market models. In this respect, the MC approach represents a natural candidate. Nevertheless, a general purpose implementation of the MC method is known
to suffer from a low rate of convergence. In particular, in order to increase its numerical
accuracy, it is either necessary to draw a large number of paths or to implement tailor-
made variance reduction techniques. Moreover, the standard MC estimator is strongly
inefficient when considering OTM options, since a relevant fraction of sampled paths
does not contribute to the payoff function. For these reasons, we present a novel MC
methodology which allows to effectively reduce the variance of the estimated price. To this
end, we proceed as follows. First, we introduce a discrete-time and discrete-space process
\( \hat{X} \) approximating the continuous time (and space) process \( X \) in equation (2.1). Then, we
propose an MC approach – the backward MC algorithm – to sample paths from \( \hat{X} \) and
to compute derivative prices.

In particular, we first split the time interval \([0, T]\) into \( n \) equally spaced subintervals
\([t_k, t_{k+1}]\), \( k \in \{0, \ldots, n-1\} \), with \( t_0 = 0 \) and \( t_n = T \) and we approximate the SDE in
equation (2.1) with an Euler–Maruyama scheme as follows:

\[
\begin{aligned}
\bar{X}_{t_{k+1}} &= \bar{X}_{t_k} + b(t_k, \bar{X}_{t_k}) \Delta t + \sigma(t_k, \bar{X}_{t_k}) \sqrt{\Delta t} \, Z_k, \\
\bar{X}_{t_0} &= X_0 = x_0,
\end{aligned}
\]

(2.2)

where \( (Z_k)_{0 \leq k \leq n-1} \) is a sequence of i.i.d. standard Normal random variables and \( \Delta t =
t_{k+1} - t_k = T/n \). Then, we assume that \( \forall k \in \{1, \ldots, n\} \) each random variable \( \bar{X}_{t_k} \) in
equation (2.2) can be approximated by a discrete random variable taking values in
\( \Gamma_k = \{\gamma^k_1, \ldots, \gamma^k_N\} \), whereas for \( t_0 \) we have \( \Gamma_0 = \gamma^0 = x_0 \). We denote by \( \hat{X}_{t_k} \) the discrete-
valued approximation of the random variable \( \bar{X}_{t_k} \). In this way, we constrain the discrete-
time Markov process \( (\hat{X}_{t_k})_{0 \leq k \leq n} \) to live on a multinomial tree. Notice that, by definition
\( |\Gamma_k| = N, \forall k \in \{1, \ldots, n\} \). Nevertheless, this is not the most general setting. For instance,
within the RMQA framework authors in Pagès & Sagna (2015) perform numerical
experiments letting the number of points in the space discretization grids vary over time.
However, they underline how the complexity in the time varying case becomes higher as
\( N \) increases, although the difference in the results is negligible.

In order to define our pricing algorithm, we need the transition probabilities from a
node at time \( t_k \) to a node at time \( t_{k+1} \), \( k \in \{0, \ldots, n-1\} \), so that in the next section we
provide a detailed description of two different approaches to consistently approximate
them. For the moment, we describe the backward MC algorithm assuming the knowledge
of both the multinomial tree \( (\Gamma_k)_{0 \leq k \leq n} \) and the transition probabilities.

As aforementioned, our final target is the computation at time \( t_0 \) of the fair price of a
path-dependent option with maturity \( T > 0 \). We denote by \( F \) its general discounted payoff
function. In particular, it is a function of a finite number of discrete observations. We are
not going to make precise \( F \) at this point, we only recall here that in this paper we will
focus on Asian options, up-and-out barrier options, and auto-callable options. According
to the arbitrage pricing theory (see, for instance, Björk, 2004), the price is given by the
conditional expectation of the discounted payoff under the risk-neutral measure \( Q \), given
the information available at time \( t_0 \). By means of the Euler–Maruyama discretization, we
can approximate the option price as follows:

\[
\mathbb{E}_{t_0}[F(x_0, \bar{X}_{t_1}, \ldots, \bar{X}_{t_n})] = \int_{\mathbb{R}^n} F(x_0, x_1, \ldots, x_n) \, p(x_0, x_1, \ldots, x_n) \, dx_1 \cdots dx_n,
\]
where \( p(x_0, x_1, \ldots, x_n) \) is the joint probability density function of \((\bar{X}_0, \bar{X}_{t_1}, \ldots, \bar{X}_{t_n})\). The previous expression can be further approximated exploiting the process \( \hat{X} \) and its discrete nature (recall that \( \Gamma_k = \{\gamma^k_1, \ldots, \gamma^k_{N} \} \)):

\[
\mathbb{E}_{t_0} \left[ F(x_0, \bar{X}_{t_1}, \ldots, \bar{X}_{t_n}) \right] \simeq \mathbb{E}_{t_0} \left[ F(x_0, \hat{X}_{t_1}, \ldots, \hat{X}_{t_n}) \right]
= \sum_{i_1=1}^{N} \cdots \sum_{i_n=1}^{N} F(x_0, \gamma^1_{i_1}, \ldots, \gamma^n_{i_n}) \mathbb{P}(x_0, \gamma^1_{i_1}, \ldots, \gamma^n_{i_n}),
\]

(2.3)

where

\[
\mathbb{P}(x_0, \gamma^1_{i_1}, \ldots, \gamma^n_{i_n}) \doteq \mathbb{P}(\hat{X}_{t_0} = x_0, \hat{X}_{t_1} = \gamma^1_{i_1}, \ldots, \hat{X}_{t_n} = \gamma^n_{i_n}).
\]

Exploiting the Markovian nature of \( \hat{X} \) and using Bayes’ theorem, we rewrite the right-hand side of equation (2.3) in the following, equivalent, way:

\[
\sum_{i_1=1}^{N} \cdots \sum_{i_n=1}^{N} F(x_0, \gamma^1_{i_1}, \ldots, \gamma^n_{i_n}) \mathbb{P}(\gamma^1_{i_1} | x_0) \cdots \mathbb{P}(\gamma^n_{i_n} | \gamma^{n-1}_{i_n}),
\]

(2.4)

where

\[
\mathbb{P}(\gamma^{k+1}_{i_{k+1}} | \gamma^k_{i_k}) \doteq \mathbb{P}(\hat{X}_{t_{k+1}} = \gamma^{k+1}_{i_{k+1}} | \hat{X}_{t_k} = \gamma^k_{i_k}),
\]

(2.5)

for all \( \gamma^k_{i_k} \in \Gamma_k \) and all \( k \in \{1, \ldots, n-1\} \).

In order to compute the expression in equation (2.4), a straightforward application of the standard MC theory would require the simulation of \( N_{MC} \) paths all originating from \( x_0 \) at time \( t_0 = 0 \). The same aforementioned arguments about the lack of efficiency of the MC estimator for the case of a continuum of state-spaces still hold for the discrete case. However, forcing each random variable \( \bar{X}_{t_k}, 1 \leq k \leq n \), to take at most \( N \) values leads in general to a reduction of the variance of the MC estimator.

For each \( t_k \in \{t_1, \ldots, t_{n-1}\} \), we denote by \( \Pi^{k+1,k} \) the \((N \times N)\)-dimensional matrix whose elements are the transition probabilities:

\[
\Pi^{k+1,k}_{i,j} \doteq \mathbb{P}(\gamma^{k+1}_{j} | \gamma^k_{i}), \quad \gamma^k_{i} \in \Gamma_k, \quad \gamma^{k+1}_{j} \in \Gamma_{k+1}, \quad \text{and} \quad i, j \in \{1, \ldots, N\}
\]

and analogously we define

\[
\Pi^{l,k+1}_{i,m} \doteq \mathbb{P}(\gamma^{l}_{m} | \gamma^{k+1}_{i}), \quad \gamma^{l}_{m} \in \Gamma_l, \quad \gamma^{k+1}_{i} \in \Gamma_{k+1}, \quad \text{and} \quad l, m \in \{1, \ldots, N\}.
\]

The key idea behind the backward MC algorithm is to express \( \Pi^{k+1,k}_{i,j} \) as a function of \( \Pi^{k,k+1}_{i,j} \) by applying Bayes’ theorem:

\[
\Pi^{k+1,k}_{i,j} = \frac{\Pi^{k,k+1}_{i,j} \mathbb{P}^k_{i,j}}{\mathbb{P}_j^{k+1}},
\]

(2.6)

where, \( \mathbb{P}_j^k \doteq \mathbb{P}(\hat{X}_{t_k} = \gamma^k_{i} | \hat{X}_{t_0} = x_0) \), for \( i = 1, \ldots, N, \quad k = 1, \ldots, n \).

Iteratively, we recover all the backward transition probabilities. These allow us to go through the multinomial tree in a backward way, from each of the terminal points \( \gamma^N_{i} \) to
consistently, we obtain the following proposition, containing the core of our pricing algorithm:

**Proposition 2.1** The price of a path-dependent option with discounted payoff $F$, $E_0\left[F(x_0, \hat{X}_{t_1}, \ldots, \hat{X}_{t_n})\right]$, can be approximated by

$$
E_0\left[F(x_0, \hat{X}_{t_1}, \ldots, \hat{X}_{t_n})\right] = \sum_{i_1=1}^{N} P_{n_{i_1}} \sum_{i_1=1}^{N} \cdots \sum_{i_{n-1}=1}^{N} \left( \prod_{k=0}^{n-1} \Pi_{k+1,i_k} \right) F(x_0, \gamma_{i_1}^1, \ldots, \gamma_{i_n}^n)
$$

where $\mathcal{F}(x_0, \gamma_{i_n}^n)$ is the expectation of the payoff function $F$ with respect to all paths starting at $x_0$ and terminating at $\gamma_{i_n}^n$.

The expectation $\mathcal{F}(x_0, \gamma_{i_n}^n)$ can be computed sampling $N_{MC}^{i_k}$ MC paths from the conditional law of $(\hat{X}_{t_1}, \ldots, \hat{X}_{t_{n-1}})$ given $x_0$ and $\gamma_{i_n}^n$, thus obtaining, at the same time, the error $\sigma_{i_n}$ associated to the MC estimator. By virtue of the Central Limit Theorem, the errors scale with the square root of $N_{MC}^{i_k}$, so that the smaller $N_{MC}^{i_k}$ is, the smaller the error. In particular, if we indicate with $\tilde{T}_n = \{\gamma_1^n, \ldots, \gamma_N^n\}$, with $N^+ \leq N$, those points of $T_n$ for which the payoff $F$ is different from zero, we can estimate the boundary values corresponding to the 95% confidence interval for the derivative price as

$$
\pm 1.96 \sqrt{\frac{1}{N^+} \sum_{i_k=1}^{N^+} \left( \Pi_{i_k} \sigma_{i_k} \right)^2}.
$$

In the previous equation, $\mathcal{F}(x_0, \gamma_{i_n}^n)$ corresponds to the MC estimator of $\mathcal{F}(x_0, \gamma_{i_n}^n)$. It is worth noticing that the error in equation (2.7) does not take into account the effect of the finiteness of $\tilde{T}_n$ and of the time discretization. The sensitivity of the price to the finite size $N$, of the grids and to the choice of the boundary conditions are investigated in Appendices C and D, respectively. We have also carried out numerical experiments – not reported in this paper for the sake of brevity but available under request – to quantify the impact of the time discretization. The numerical results presented in this paper are computed with time and price grids such that the error due to the finite granularity is negligible with respect to the statistical error due to the finiteness of the MC sample.

A sizeable variance reduction results from having split the $n$ sums in equation (2.4) into the external summation over the points of the deterministic grid $\tilde{T}_n$ and the evaluation of an expectation of the payoff with fixed initial and terminal points. This procedure
corresponds to the variance reduction technique known as stratified sampling MC (see Glasserman, 2004). In particular, in Glasserman (2004), the author proves analytically that the variance of the MC estimator without stratification is always greater than or equal to that of the stratified one. Besides, he points out that stratified sampling involves consideration of two issues: (i) the choice of the points in $\Gamma_n$ and the allocation $\mathcal{N}^n_{\text{MC}}$, $i_n \in \{1, \ldots, N\}$, (ii) the generation of samples from $\hat{X}$ conditional on $\hat{X}_{t_n} \in \Gamma_n$ and on $\hat{X}_{t_0} = \gamma_0$. Our procedure resolves both these points. Precisely, once selected $\mathcal{T}_n$, the backward MC algorithm allows us to choose the number of paths from all the points in $\mathcal{T}_n$, independently on the value of $P^n_{i_n}$.

At this point, two are the main ingredients needed in order to compute the quantities in equation (2.7): (i) the transition probabilities, (ii) the fast backward simulation of the process $\hat{X}$. For both purposes, we introduce ad-hoc numerical procedures. As regards, the former point, we analyse and extend two approaches already present in the literature. The first one is based on the concept of optimal state-partitioning of a random variable (called stratification in Barraquand & Martineau (1995) and quantization in Bally & Pagès (2003)) and employs the RMQA (see Callegaro et al., 2015; Pagès & Sagna, 2015). The second approach provides a recipe to compute in an effective way the transition probability matrix between any two arbitrary dates for a piecewise time-homogeneous process (see Albanese, 2007; Reghai et al., 2012). More details on these two methods will be given in Section 3.

For what concerns the backward simulation, we employ the Alias method introduced in Kronmal & Peterson (1979). More specifically, for every $k$ from $n - 1$ to 1, the (backward) simulation of $\hat{X}_{t_k}$ conditional on $\{\hat{X}_{t_{k+1}} = \gamma_{j}^{k+1}\}$ is equivalent to sampling at each time $t_{k+1}$ from a discrete non-uniform distribution with support $\Gamma_k$ and probability mass function equal to the $j$th row of $\Pi_{k+1}^{k+1,k}$. Given the discrete distribution, a naïve simulation scheme consists in drawing a uniform random number from the interval $[0, 1]$ and recursively search over the cumulative sum of the discrete probabilities. However, in this case, the corresponding computational time grows linearly with the number $N$ of states. The Alias method, instead, reduces this numerical complexity to $O(1)$ by cleverly pre-computing a table – the Alias table – of size $N$. We base our implementation on this method, which enables a large reduction of the MC computation time. A more detailed description of the Alias method can be found at www.keithschwarz.com.

3 Recovering the transition probabilities

We present the two approaches used for the approximation of the transition probabilities of a discrete-time Markov Chain. The RMQA is described and extended in Section 3.1. In particular, we first provide a brief overview on optimal quantization of a random variable, then we propose an alternative implementation of the RMQA. The LTSA is presented in Section 3.2, where we also provide a brief introduction on Markov processes and on Markov generators.
3.1 A quantization-based algorithm

The reader who is familiar with quantization can skip the following subsection.

3.1.1 Optimal quantization

We present here the concept of optimal quantization of a random variable by emphasizing its practical features, without providing all the mathematical details behind it. A more extensive discussion can be found in Graf & Luschgy (2000), Pagès et al. (2004), Pagès & Printems (2005), Pagès (2015). Let \( \bar{X} \) be a one-dimensional continuous random variable defined on a probability space \((\Omega, \mathcal{F}, \mathbb{P})\) and \( \mathbb{P}_X \) the measure induced by it. The quantization of \( \bar{X} \) consists in approximating it by a one-dimensional discrete random variable \( \bar{b} \bar{X} \). In particular, this approximation is defined by means of a quantization function \( q_N \) of \( \bar{X} \), that is to say \( \bar{b} \bar{X} = q_N(\bar{X}) \), defined in such a way that \( \bar{b} \bar{X} \) takes finitely many values in \( \mathbb{N}^+ \). The finite set of values for \( \bar{b} \bar{X} \), denoted by \( \Gamma = \{\gamma_1, \ldots, \gamma_N\} \), is the quantizer of \( \bar{X} \), while the image of the function \( q_N \) is the related quantization. The components of \( \Gamma \) can be used as generator points of a Voronoi tessellation \( \{C_i(\Gamma)\}_{i=1}^{\infty} \). In particular, one sets up the following tessellation with respect to the absolute value in \( \mathbb{R} \):

\[
C_i(\Gamma) \subset \{\gamma \in \mathbb{R} : |\gamma - \gamma_i| = \min_{1 \leq j \leq N} |\gamma - \gamma_j|\},
\]

and the associated quantization function \( q_N \) is defined as follows:

\[
q_N(\bar{X}) = \sum_{i=1}^{N} \gamma_i \mathbb{1}_{C_i(\Gamma)}(\bar{X}).
\]

Notice that in our setting, we are going to quantize the random variables \( (\bar{X}_i)_{0 \leq k \leq n} \) introduced in equation (2.2).

Such a construction rigorously defines probabilistic setting for the random variable \( \bar{X} \), by exploiting the probability measure induced by the continuous random variable \( \bar{X} \). The approximation of \( \bar{X} \) through \( \bar{b} \bar{X} \) induces an error, whose \( L^2 \) version – called \( L^2 \)-mean quantization error – is defined as

\[
\|\bar{X} - q_N(\bar{X})\|_2 = \sqrt{\mathbb{E} \left[ \min_{1 \leq i \leq N} |\bar{X} - \gamma_i|^2 \right]}. \tag{3.1}
\]

The expected value in equation (3.1) is computed with respect to the probability measure which characterizes the random variable \( \bar{X} \). The purpose of the optimal quantization theory is finding a quantizer indicated by \( \Gamma^* \), which minimizes the error in equation (3.1) over all possible quantizers with size at most \( N \). We remind that in one dimension the uniqueness of the optimal \( N \) quantizer is guaranteed if the distribution of \( \bar{X} \) is absolutely continuous with a log-concave density function (see, for instance, Pagès, 2015). From the theory (see Graf & Luschgy, 2000, among others), we know that the mean quantization error vanishes as the grid size \( N \) tends to infinity. Besides, its rate of convergence is ruled by Zador theorem. However, computationally, finding explicitly \( \Gamma^* \) can be a challenging
task. This has motivated the introduction of sub-optimal criteria linked to the notion of stationary quantizer (see Pagès, 2015):

**Definition 3.1** A quantizer $\Gamma = \{\gamma_1, \ldots, \gamma_N\}$ inducing the quantization $q_N$ of the random variable $\bar{X}$ is said to be stationary if

$$\mathbb{E} [\bar{X} | q_N(\bar{X})] = q_N(\bar{X}).$$  \hspace{1cm} (3.2)

**Remark 3.2** An optimal quantizer is stationary, the vice-versa does not hold true in general (see Pagès, 2015).

In order to compute optimal (or sub-optimal) quantizers, one first introduces a notion of distance between a random variable $\bar{X}$ and a quantizer $\Gamma$ as follows:

$$d(\bar{X}, \Gamma) = \min_{1 \leq i \leq N} |\bar{X} - \gamma_i|,$$

then one considers the so called distortion function

$$D(\Gamma) = \mathbb{E} [d(\bar{X}, \Gamma)^2] = \mathbb{E} \left[ \min_{1 \leq i \leq N} |\bar{X} - \gamma_i|^2 \right] = \sum_{i=1}^{N} \int_{C_{\Gamma}} |\xi - \gamma_i|^2 \, d\mathbb{P}_{\bar{X}}(\xi).$$ \hspace{1cm} (3.3)

It can be shown (see Pagès, 2015), that the distortion function is continuously differentiable as a function of $\Gamma$. In particular, it turns out that stationary quantizers are critical points of the distortion function, that is, a stationary quantizer $\Gamma$ is such that $\nabla D(\Gamma) = 0$. Several numerical approaches have been proposed in order to find stationary quantizers (for a review, see Pagès et al., 2004). These approaches can be essentially divided into two categories: gradient-based methods and fixed-point methods. The former class includes the Newton–Raphson algorithm, whereas the second category includes the Lloyd I algorithm. More specifically, the Newton–Raphson algorithm requires the computation of the gradient, $\nabla D(\Gamma)$, and of the Hessian matrix, $\nabla^2 D(\Gamma)$, of the distortion function. The Lloyd I algorithm, on the other hand, does not require the computation of the gradient and Hessian and it consists in a fixed-point algorithm based on the stationary equation (3.2).

### 3.1.2 The recursive marginal quantization algorithm

The RMQA is a recursive algorithm, which has been recently introduced in Pagès & Sagna (2015). It consists in quantizing the stochastic process $X$ in equation (2.1) by working on the (marginal) random variables $\bar{X}_t$, all $k \in \{1, \ldots, n\}$ in (2.2). The key idea behind the RMQA is that the discrete-time Markov process $\bar{X} = (\bar{X}_t)_{0 \leq k \leq n}$ in equation (2.2) is completely characterized by the initial distribution of $\bar{X}_0$ and by the transition probability densities. We indicate by $\hat{\bar{X}}_{t_k}$ the quantization of the random variable $\bar{X}_{t_k}$ and by $\hat{D}(\Gamma_k)$ the associated distortion function.

**Remark 3.3** The process $\hat{\bar{X}} = (\hat{\bar{X}}_{t_k})_{0 \leq k \leq n}$ is not, in general, a discrete-time Markov Chain. Nevertheless, it is known (see, for instance, Pagès et al., 2004) that there exists a
discrete-time Markov Chain $\hat{X}^c = (\hat{X}_t^c)_{0 \leq k \leq n}$, with initial distribution and transition probabilities equal to those of $\hat{X}$. Hence, throughout the rest of the paper, when we will write “discrete-time Markov Chain” within the Recursive Marginal Quantization framework we will refer, by tacit agreement, to the process $\hat{X}^c$.

Here, we give a quick drawing of the RMQA. First of all, one introduces the Euler operator associated to the Euler scheme in equation (2.2):

$$E_k(x, At; Z) = x + b(t_k, x)At + \sigma(t_k, x)\sqrt{At} Z,$$

where $Z \sim N(0, 1)$, so that, from (2.2), $\bar{X}_{t_k+1} = E_k(\bar{X}_{t_k}, At; Z_k)$.

**Lemma 3.4** Conditionally on the event $\{\bar{X}_{t_k} = x\}$, the random variable $\bar{X}_{t_k+1}$ is a Gaussian random variable with mean $m_k(x) = x + b(t_k, x)At$ and standard deviation $v_k(x) = \sqrt{At} \sigma(t_k, x)$, all $k = 1, \ldots, n - 1$.

**Proof** It follows immediately from the equality $\bar{X}_{t_k+1} = E_k(\bar{X}_{t_k}, At; Z_k)$, given that $Z_k$ is a standard Normal random variable.

At this point, one writes down the following crucial equalities:

$$\bar{D}(\Gamma_{k+1}) = \mathbb{E} \left[ d(\bar{X}_{t_{k+1}}, \Gamma_{k+1})^2 \right] = \mathbb{E} \left[ \mathbb{E} \left[ d(\bar{X}_{t_{k+1}}, \Gamma_{k+1})^2 | \bar{X}_{t_k} \right] \right] = \mathbb{E} \left[ d(E_k(\bar{X}_{t_k}, At; Z_k), \Gamma_{k+1})^2 \right],$$

where $(Z_k)_{0 \leq k \leq n}$ is a sequence of i.i.d. one-dimensional standard Normal random variables.

As said, stationary quantizers are zeros of the gradient of the distortion function. By definition, the distortion function $\bar{D}(\Gamma_{k+1})$ depends on the distribution of $\bar{X}_{t_{k+1}}$, which is, in general, unknown. Nevertheless, thanks to Lemma 3.4, the distortion in equation (3.4) can be computed explicitly. Equation (3.4) is the essence of the RMQA. More precisely, one starts setting the quantization of $\bar{X}_0$ to $x_0$, namely $q_N(\bar{X}_0) = x_0$. Then, one approximates $\bar{X}_t$ with $\bar{X}_t = E_0(x_0, At; Z_1)$ and the distortion function associated to $\bar{X}_t$ with that associated to $\bar{X}_t$, namely $\bar{D}(\Gamma_1) \approx \bar{D}(\Gamma_1) = \mathbb{E} \left[ d(E_0(x_0, At; Z_1), \Gamma_1)^2 \right]$. Then, one looks for a stationary quantizer $\Gamma_1$ by searching for a zero of the gradient of the distortion function, using either Newton–Raphson or Lloyd I method. The procedure is applied iteratively at each time step $t_k$, $1 \leq k \leq n$, leading to the following sequence of stationary (marginal) quantizers:

$$\hat{X}_0 = \bar{X}_0, \quad \hat{X}_{t_k} = q_N(\bar{X}_{t_k}) \quad \text{and} \quad \bar{X}_{t_{k+1}} = E_k(\hat{X}_{t_k}, At; Z_{k+1})$$(Z_k)_{1 \leq k \leq n} i.i.d. Normal random variables independent from $\bar{X}_{t_0}$. 


In Pagès & Sagna (2015), the authors give an estimation of the (quadratic) error bound 
\[ \| \bar{X}_{t_k} - \tilde{X}_{t_k} \|_2, \] 
for fixed \( k = 1, \ldots, n \).

At this point, the approximated transition probabilities (termed companion parameters in Pagès & Sagna, 2015) are obtained instantaneously given the quantization grids and Lemma 3.4. In particular,

\[
\Pi_{i,j}^{k+1} = \mathbb{P}(i_{i,j}^{k+1} = j_{i,j} | i_{i,j}^{k} = i) \approx \mathbb{P}(\tilde{X}_{t_{k+1}} \in C_{j} | \tilde{X}_{t_k} \in C_i) \cdot (3.5)
\]

In Appendix A, we provide the explicit expressions of the distortion function \( D(\Gamma_{k+1}) \) and of the approximated transition probabilities \( \mathbb{P}(\tilde{X}_{t_{k+1}} \in C_{j} | \tilde{X}_{t_k} \in C_i) \).

In order to compute numerically the sequence of stationary quantizers \( (\Gamma_k)_{1 \leq k \leq n} \) in Pagès & Sagna (2015) and Callegaro et al. (2015), authors employ the Newton–Raphson algorithm. However, as pointed out in Callegaro et al. (2015), it may become unstable when \( \Delta t \to 0 \) due to the ill-condition number of the Hessian matrix \( \nabla^2 D(\Gamma) \). An alternative approach is based on fixed-point algorithms, such as the Lloyd I method, even though such method converges to the optimal solution with a smaller rate of convergence (see Kieffer, 1982, for a discussion). For these reasons and as original contribution, we combine it with a particular acceleration scheme, called Anderson acceleration.

3.1.3 The Anderson accelerated procedure

The acceleration scheme, called Anderson acceleration, was originally discussed in Anderson (1965), and outlined in Walker & Ni (2011) together with some practical considerations for implementations. For completeness, in Appendix B, we give some details on how the Lloyd I method works when employed in the recursive marginal quantization setting.

Now, we discuss the major differences between a general fixed-point algorithm – and its associated fixed-point iterations – and the same fixed-point method coupled with the Anderson acceleration. We outline the practical features without giving all the technical details concerning the numerical implementation of the accelerated scheme (please refer to Walker & Ni, 2011, for an extensive discussion on this issue). A general fixed-point problem – also known as Picard problem – and its associated fixed-point iteration are defined as follows:

\[ \text{Fixed-point problem : Given } g : \mathbb{R}^N \to \mathbb{R}^N, \text{ find } \Gamma \in \mathbb{R}^N \text{ s.t. } \Gamma = g(\Gamma). \]

\[ \text{Algorithm (Fixed Point Iteration) } \]

\[ \text{Given } \Gamma^0, \]

\[ \text{for } l \geq 0, l \in \mathbb{N} \]

\[ \text{set } \Gamma^{l+1} = g(\Gamma^l). \]

(3.6)
The same problem coupled with the Anderson acceleration scheme is modified as follows:

**Algorithm (Anderson acceleration)**

Given \( \Gamma^0 \) and \( m \geq 1, m \in \mathbb{N} \),

set \( \Gamma^1 = g(\Gamma^0) \),

for \( l \geq 1, l \in \mathbb{N} \)

set \( m_l = \min(m, l) \)

set \( F_l = (f_{l-m_l}, \ldots, f_l) \), where \( f_i = g(\Gamma^i) - \Gamma^i \)

(3.7)

\[ \text{determine } \mathcal{A}^{(l)} = (\mathcal{A}_0^{(l)}, \ldots, \mathcal{A}_{m_l}^{(l)})^T \text{ that solves} \]

\[ \min_{\mathcal{A}^{(l)} \in \mathbb{R}^{m_l+1}} \| F_l \mathcal{A}^{(l)} \|_2 \text{ s.t. } \sum_{i=0}^{m_l} \mathcal{A}_i^{(l)} = 1 \]

set \( \Gamma^{l+1} = \sum_{i=0}^{m_l} \mathcal{A}_i^{(l)} g(\Gamma^{l-m_l+i}) \).

The Anderson acceleration algorithm stores (at most) \( m \) user-specified previous function evaluations and computes the new iterate as a linear combination of those evaluations with coefficients minimizing the Euclidean norm of the weighted residuals. In particular, with respect to the general fixed-point iteration, Anderson acceleration exploits more information in order to find the new iterate.

In equation (3.7), Anderson acceleration algorithm allows to monitor the conditioning of the least squares problem. In particular, we follow the strategy used in Walker & Ni (2011) where the constrained least squares problem is first transformed in an unconstrained one, then solved using a QR decomposition. The usage of the QR decomposition to solve the unconstrained least square problem represents a good balance of accuracy and efficiency. Indeed, if we name \( \mathcal{F}_l \) the least squares problem matrix, it is obtained from its predecessor \( \mathcal{F}_{l-1} \) by adding a new column on the right. The QR decomposition of \( \mathcal{F}_l \) can be efficiently attained from that of \( \mathcal{F}_{l-1} \) in \( O(m_l N) \) arithmetic operations using standard QR factor-updating techniques (see Golub & Van Loan, 2012). The Anderson acceleration scheme speeds up the linear rate of convergence of the general fixed-point problem without increasing its computational complexity. More importantly, it does not suffer the extreme sensitivity of the Newton–Raphson method to the choice of the initial point (grid). We refer to the numerical experiments in Appendix C for an illustration of both the improvement of the Anderson acceleration with respect to the convergence speed of the fixed point iteration and of the over-performance of Lloyd I method with respect to the stability of the Newton–Raphson algorithm. Appendix C is by no means intended to be exhaustive, since it illustrates the performance of the Anderson acceleration algorithm in some examples.

### 3.2 The large time step algorithm

The LTSA is employed to recover the transition probability matrix associated to a time and space discretization of an LV model. We start here by recalling some known results
about Markov processes, that will be used in what follows. We work under the following assumption:

**Assumption 3.5** The asset price process $X$ follows the dynamics in equation (2.1), where the drift and diffusion coefficients $b$ and $\sigma$ are piecewise-constant functions of time.

Let us consider the Markov process $X$ in equation (2.1) and let us denote by $p(t', \gamma'|t, \gamma)$, with $0 \leq t < t' \leq T$ and $\gamma, \gamma' \in \mathbb{R}$, the transition probability density from state $\gamma$ at time $t$ to state $\gamma'$ at time $t'$. Under some non-stringent assumptions, it is known that $p$, as a function of the backward variables $t$ and $\gamma$, satisfies the backward Kolmogorov equation (see Karatzas & Shreve, 1998; Kijima, 1997):

$$
\frac{\partial p(t', \gamma'|t, \gamma)}{\partial t} + (Lp)(t', \gamma'|t, \gamma) = 0 \quad \text{for} \quad (t, \gamma) \in (0, t') \times \mathbb{R},
$$

(3.8)

where $\delta$ is the Dirac delta and $L$ is the infinitesimal operator associated with the SDE (2.1), namely a second-order differential operator acting on functions $f : \mathbb{R}_+ \times \mathbb{R} \to \mathbb{R}$ belonging to the class $C^{1,2}$ and defined as follows:

$$
(Lf)(t, \gamma) = b(t, \gamma) \frac{\partial f}{\partial \gamma}(t, \gamma) + \frac{1}{2} \sigma^2(t, \gamma) \frac{\partial^2 f}{\partial \gamma^2}(t, \gamma).
$$

(3.9)

The solution to equation (3.8) can be formally written as

$$
p(t', \gamma'|t, \gamma) = e^{(t'-t)L} p(t, \gamma).
$$

(3.10)

The LTSA consists in approximating the transition probabilities relative to a discrete-time finite-state Markov chain approximation of $X$ using equation (3.10). We report now a simple example to clarify how the LTSA works.

**Example 3.6** Consider for example the case when $b$ and $\sigma$ in equation (2.1) are defined as

$$
b(t, X_t) = b_1(X_t) \mathbb{1}_{[0, T_1]}(t) + b_2(X_t) \mathbb{1}_{[T_1, T_2]}(t),
$$

$$
\sigma(t, X_t) = \sigma_1(X_t) \mathbb{1}_{[0, T_1]}(t) + \sigma_2(X_t) \mathbb{1}_{[T_1, T_2]}(t),
$$

where $T_1$ and $T_2 = T$ are two target maturities and $b_1, b_2, \sigma_1, \sigma_2$ suitable functions. The transition probabilities in this case are explicitly given. In particular, if we denote by $L^1_T$ and $L^2_T$ the infinitesimal Markov generators of the Markov chain approximation of $X$ in $[0, T_1]$ and $[T_1, T_2]$, respectively, the transition probabilities between any two arbitrary dates $t$ and $t'$ are given by

$$
e^{(t'-t)L^1_T} \quad \text{for} \quad 0 \leq t < t' \leq T_1,
$$

$$
e^{(T_1-t)L^1_T}e^{(t'-T_1)L^2_T} \quad \text{for} \quad 0 \leq t \leq T_1 \leq t' \leq T_2,
$$

$$
e^{(t'-t)L^2_T} \quad \text{for} \quad T_1 \leq t \leq t' \leq T_2.$$
In real market situations, the above assumption on \( b \) and on \( \sigma \) is not at all restrictive, as we are going to see in Section 4.

Let us now give more details on the algorithm. First of all, once a time discretization grid \( \{u_0, u_1, \ldots, u_m\} \) has been chosen (think for example to the calibration pillars or to the expiry dates of the calibration dates of vanilla options), we need to obtain the space discretization grids \( \Gamma_k, 0 \leq k \leq m \). As a major difference with respect to the RMQA, the LTSA grids do not stem from the minimization of any distortion function. In particular, they can be defined with a large degree of arbitrariness (see, for instance, Appendix D) as follows:

\[
\Gamma_0 \equiv x_0 \quad \text{and} \quad \Gamma_k \equiv \Gamma = \{\gamma_1, \ldots, \gamma_N\}, \quad k = 1, \ldots, m,
\]

with the only restriction that the grid \( \Gamma_k \equiv \Gamma \) for all \( k = 1, \ldots, m \). Actually, to ensure the applicability of the LTSA (see also Remark 3.7), it is essential to keep \( \Gamma_k \) constant within each time interval where the underlying process is time homogeneous. This apparent disadvantage of the LTSA is indeed a crucial positive feature when numerical simulations have to be implemented on Graphic Processing Units to exploit fast exponentiation.

Then, the method consists in discretizing, opportunely, the Markov generator \( L \) and in calculating, in an effective and accurate way, the transition probabilities. As regards, the discretization, authors in Albanese & Mijatović (2007) give a recipe to construct the discrete counterpart of \( L \) – denoted by \( L_\Gamma \) – so that the Markov chain approximation of \( X \) converges to the continuous limit process in a weak or distributional sense (see Kushner & Dupuis, 2001). In particular, \( L_\Gamma \) corresponds to the discretization of equation (3.9) through an explicit Euler finite difference approximation of the derivatives (see Mitchell & Griffiths, 1980). In Appendix D, we provide more details on the discretization of \( L \).

Once \( L_\Gamma \) is constructed, one writes a (matrix) Kolmogorov equation for the transition probability matrix. In particular, using operator theory (Albanese, 2007), the transition probability matrix between any two arbitrary dates \( u_k \) and \( u_{k'} \) with \( 0 \leq u_k < u_{k'} \leq T \) can be expressed as a matrix exponential.

**Remark 3.7** The piecewise time-homogeneous feature of the process \( X \) plays a crucial role as regards the computational burden required to compute the transition probability matrix. Indeed, in case of time-dependent drift and volatility coefficient, it can no longer be expressed, in a straightforward way, as the exponential of the (time-dependent) Markov generator \( L_\Gamma \) (see, for instance, Blanes et al., 2009).

The LTSA is computationally convenient with respect to the RMQA when pricing path-dependent derivatives whose payoff specification requires the observation of the asset price on a pre-specified set of dates, for example, \( \{u_0, u_1, \ldots, u_m\} \). Indeed, in this case, we first calculate off-line the \( m \) transition matrices as in equation (3.6), then we price the derivative products via MC with coarse-grained resolution. In Figure 1, we plot an example of a possible path corresponding to the case \( m = 3 \). This major difference between RMQA and LTSA becomes more evident looking at Figures 2 and 3, where we plot, respectively, an MC simulation connecting the initial point \( x_0 \) with a random final
Figure 1. Example of one Monte Carlo path sampled with the LTSA with \( u_1 = t_1, u_2 = t_3, \) and \( u_3 = t_5. \)

Figure 2. Example of one MC path sampled with RMQA over a time-grid computed with six time buckets.
point \( \hat{X}_{t_5} \), and a direct jump to date simulation to random points \( \hat{X}_{tk} \) with \( k = 1,\ldots,5 \), respectively.

We underline that, both the RMQA and the LTSA enable the computation of the price of vanilla options by means of a straightforward scalar product. Indeed, the price of a vanilla call option with strike \( K \) and maturity \( T = t_m \) can be computed as follows:

\[
\sum_{i=1}^{N} \mathbb{P}(r_i^n|x_0)(r_i^n - K)^+.
\]  

\[ (3.11) \]

3.2.1 LTSA implementation: More technical details

In order to compute effectively and accurately a matrix exponential and so recover the transition probability for the LTSA, we use the so called Scaling and Squaring method along with Padé approximation. In particular, we implement the version of the method proposed by Higham (2005) because it outperforms, both in efficiency and in accuracy, previous implementations proposed in Sidje (1998) and Ward (1977). We now give a brief drawing of the algorithm implemented in Higham (2005), outlining its practical features but without giving all the mathematical details behind it. For a more extensive analysis on the method, we refer to the original paper. Besides, we refer to Baker & Graves-Morris (1996) for an extensive description of Padé approximation.

The Scaling and Squaring algorithm exploits the following trivial property of the
exponential function:

\[ e^{\tilde{\mathcal{L}}_r} = \left( e^{\tilde{\mathcal{L}}_\Gamma} \right)^\beta, \]

(3.12)

where \((t_k - t_k)\Gamma = \tilde{\mathcal{L}}_r\), together with the fact that \(e^{\tilde{\mathcal{L}}_r}\) is well approximated by a Padé approximation near the origin, that is for small \(\|\tilde{\mathcal{L}}_r\|\), where \(\|\cdot\|\) is any subordinate matrix norm. In particular, Padé approximation estimates \(e^{\tilde{\mathcal{L}}_r}\) with the ratio between two (matrix) polynomials of degree 13. The mathematical elegance of the Padé approximation is enhanced by the fact that the two approximating polynomial are known explicitly. The main hint of the Scaling and Squaring method is to choose \(\beta\) in equation (3.12) as an integer power of 2, \(\beta = 2^n\) say, so that \(e^{\tilde{\mathcal{L}}_r/\beta}\) has norm of order 1, to approximate \(e^{\tilde{\mathcal{L}}_r/\beta}\) by a Padé approximation, and then to compute \(e^{\tilde{\mathcal{L}}_r}\) by repeating squaring \(n\) times. In particular, we define \(\delta t = (t_k - t_k)/2^n\). We use Padé approximation because of the usage of the explicit Euler Scheme for the discretization of \(\mathcal{L}\) (see Appendix D). Indeed, one needs to impose the so called Courant condition for the matrix \(\delta t \Gamma\). The Courant condition requires that \(\|\delta \Gamma\|_\infty < 1\). This translates into the following stringent condition for \(\delta t\):

\[ \delta t < \frac{1}{2} \frac{(\Delta \gamma)^2}{\sigma(\gamma)}, \forall 1 \leq i \leq N. \]

The usage of the Padé approximation permits to relax the last constraint. In particular, the implementation in Higham (2005) allows \(\|\delta \Gamma\|_\infty\) to be much larger.

**Remark 3.8** The Backward MC algorithm can be applied also to price financial derivatives in stochastic volatility models. We do not deal with this aspect in this paper, leaving it for future research. Indeed, the computational effort necessary to recover grids and transition probabilities increases substantially and to effectively compute matrix–matrix multiplications we need to switch to an alternative technology based on Graphic Processing Units.

### 4 Financial applications

In this final section, we present and discuss how results achieved in the previous Sections 2 and 3 can be applied to finance, and in particular to option pricing in the FX market, where spot and forward contracts, along with vanilla and exotic options are traded (see Reiswich & Uwe, 2012, for a broad overview on the FX market). In particular, we consider the following types of path-dependent options: (i) Asian calls, (ii) up-and-out barrier calls, (iii) automatic callable (or auto-callable). We choose two different models for the underlying EUR/USD FX rate: an LV model as a benchmark and the CEV (see Cox, 1975), coming from the academic literature.

#### 4.1 Model, payoff specifications, and variance reduction techniques

Let us first introduce some notations relative to the LV model and to the CEV model. Recall that we have assumed in Section 2 deterministic interest rates. Moreover, we
indicate by $X_t$ the spot price at time $t$ of one EUR expressed in USD and by $X_t(T)$ the corresponding forward price at time $t$ for delivery at time $T$.

We introduce the so-called normalized spot exchange rate $x^{LV}_t \equiv X_t/X_0(t)$, all $t \in [0, T]$ (where the superscript “LV” clearly stands for LV), and we suppose that the process $x^{LV} = (x^{LV}_t)_{t \in [0, T]}$ follows the SDE

$$\begin{cases} dx^{LV}_t = x^{LV}_t \eta(t, x^{LV}_t) \, dW_t, \\ x^{LV}_0 = 1. \end{cases}$$

Hence, in this LV model, $x^{LV}$ corresponds to the underlying process $X$ introduced in Section 2, and besides making reference to equation (2.1) we have $b(t, x^{LV}_t) = 0$ and $\sigma(t, x^{LV}_t) = \eta(t, x^{LV}_t) x^{LV}_t$, where $\eta : [0, T] \times \mathbb{R} \to \mathbb{R}_+$ corresponds to the LV function. Specifically, it is a cubic monotone spline for fixed $t \in [0, T]$ (see Fritsch & Carlson, 1980, for an overview on interpolation technique) with flat extrapolation. The set of points to be interpolated is determined numerically during the calibration procedure. Precisely, we use the calibration procedure proposed in Reghai et al. (2012) and refined in Pallavicini (2016). This procedure is particularly robust. Indeed, the resulting LV surface is ensured to be a smooth function of the spot. In particular, this procedure leads to a piecewise time-homogeneous dynamics for the process $x^{LV}$.

The data set used in the calibration is available upon requests.

As a second example, we consider the CEV, i.e., we assume that the asset price process $X$ follows a CEV dynamics

$$\begin{cases} dX_t = r X_t \, dt + \sigma X_t^\alpha \, dW_t, \\ X_0 = x_0 \in \mathbb{R}_+, \end{cases} \quad (4.1)$$

where $r \equiv r_d - r_f \in \mathbb{R}$ is the risk-free interest rate with $r_d$ the domestic interest rate and $r_f$ the foreign interest rate, $\sigma \in \mathbb{R}_+$ is the volatility, and $\alpha \in [1/2, 1)$ is a constant parameter. Even though coefficients in equation (4.1) do not satisfy the usual Lipschitz conditions assumed in Section 2, it is well known that in our setting the SDE (4.1) admits a unique strong solution. Moreover, Jeanblanc et al. (2009) (see Lemma 6.4.4.1) show that – for the above choice of the parameter alpha – the solution is non-negative, in the sense that zero is an absorbing state. As far as the discretization scheme is concerned, in order to prevent $X_{t_{k+1}}$ to take negative values, the usual Euler–Maruyama recipe has to be slightly adjusted. Labbé et al. (2011) review various possible modifications discussed in the literature. For the general CEV model under consideration here, a non-negative definite Euler–Maruyama scheme and its weak and strong convergence has been studied in Bossy & Diop (2007) and Berkaoui et al. (2008), respectively.

Then, given a time discretization grid $\{0 = t_0, t_1, \ldots, t_n = T\}$ on $[0, T]$ as in Section 2 and making reference to the Euler–Maruyama scheme in equation (2.2) we consider the unidimensional payoff specifications below. In particular, we compute the price at time $t_0 = 0$. 
(i) **Asian calls.** The discounted payoff function of a discretely monitored Asian call option is

\[ F_A(\tilde{X}_0, \ldots, \tilde{X}_n) = e^{-r(T_n-t_0)} \max \left( \frac{1}{n+1} \sum_{i=0}^{n} \tilde{X}_{t_i} - K, 0 \right), \]

where \( K \) is the strike price and \( T > 0 \) the maturity. As benchmark price for an Asian call, we consider the price computed with control variates (see, for instance, Glasserman, 2004, Chapter 4). Specifically, under the risk-neutral measure, the discounted asset price is a martingale and we can employ it as the control variate variable. The comparison among the different results is postponed to the next section.

(ii) **Up-and-out barrier calls.** We consider barrier options of European style. The discounted payoff at maturity \( T > 0 \) of an up-and-out barrier call is given by

\[ e^{-rT} \max(X_T - K, 0)1_{\{\tau > T\}}, \tag{4.2} \]

where \( K \) is the strike price, \( \tau = \inf \{ t \geq 0 : X_t \geq B \} \) and \( B \) is the upper barrier.

The simplest method to price such an option is to consider the discrete version of the continuous time discounted payoff in equation (4.2)

\[ F_B(\tilde{X}_{t_0}, \ldots, \tilde{X}_{t_n}) = e^{-r(t_n-t_0)} \max(\tilde{X}_{t_n} - K, 0) \prod_{k=0}^{n} 1_{\{\tilde{X}_k < B\}}. \]

Proceeding this way, however, the price of the option is overestimated because \( X \) can cross the barrier at some time \( t \) between two grid points \( k \Delta t \) and \( (k+1) \Delta t, 0 \leq k \leq n-1 \), while being never above the barrier at the observation dates \( t_0, t_1, \ldots, t_n \).

In Gobet (2000) (see, also Glasserman, 2004), the author proposes a strategy to obtain a better approximation of the price of the option in equation (4.2) when employing MC simulation. It relies on the observation that the probability of the Brownian motion breaching the barrier is computable from the knowledge of the volatility and of the value of the spot \( X \) at the two end points of the observation interval. Namely, given the points \( \tilde{X}_{t_k} \) and \( \tilde{X}_{t_{k+1}}, 0 \leq k \leq n-1 \), the probability that \( X \) has reached the barrier \( B \) from below in \((t_k, t_{k+1})\) is given by

\[ \tilde{p}_k = 1 - \exp \left[ -\frac{2}{\sigma_k \Delta t}(B - \tilde{X}_{t_k})(B - \tilde{X}_{t_{k+1}}) \right], \]

where \( \sigma_k \) is the diffusive coefficient of the underlying asset price in \((t_k, t_{k+1})\) (assuming \( B \) is greater than both \( \tilde{X}_{t_k} \) and \( \tilde{X}_{t_{k+1}} \)). So, at each \( t_k = k \Delta t, 1 \leq k \leq n \), and for all the MC paths \( \ell, 1 \leq \ell \leq N_{MC} \), after verifying that \( \tilde{X}_\ell(t_k) < B \), we investigate whether the simulated path has reached the barrier \( B \) over the interval \((t_{k-1}, t_k)\). In particular, we draw a random variable from a Bernoulli distribution with parameter \( (1 - \tilde{p}_{k-1}) \): If the outcome is favourable, then the barrier has been reached over this interval and the price associated to the path is zero.

The variance of the MC estimator can be further reduced by applying an importance sampling technique. Given the value of \( \tilde{X}_{t_{k-1}}, \) each \( \tilde{X}_{t_k} \) is distributed as a Gaussian random variable with mean \( \tilde{\mu}_k = \tilde{X}_{t_{k-1}} + b(t_{k-1}, \tilde{X}_{t_{k-1}}) \Delta t \) and standard
deviation $\bar{\sigma}_{t_k} = \sigma(t_{k-1}, \bar{X}_{t_{k-1}}) \sqrt{\Delta t}$. Then, one modifies this density so that values of $\bar{X}_{t_k}$ greater than $B$ are not sampled. The procedure to draw a Gaussian below $B$ is to sample a uniform random variable $U$ and set $\bar{X}_{t_k} = F_{\bar{X}_{t_k}}^{-1}(\theta U)$, with $\theta = \mathbb{P}(\bar{X}_{t_k} < B)$. The final value is multiplied by $\theta$. This algorithm is implemented in Joshi & Leung (2011) to price continuous barrier options in a jump-diffusion model.

As for the Asian option case, we postpone the comparison among the various techniques to Section 4.2.

(iii) **Automatic callable (or auto-callable)** The discounted payoff of an auto-callable option (they were first issued in the U.S. by BNP Paribas in August 2003 as cited for instance in Deng et al. (2011)) with unitary notional is given by

\[
\begin{align*}
&\left\{ 
\begin{array}{ll}
\frac{e^{-r(t_i - t_0)} Q_i}{X_0^i} & \text{if } \bar{X}_{t_{j}} < X_0^b \leq \bar{X}_{t_i} \quad \text{for all } j < i, \\
\frac{e^{-r(t_i - t_0)} X_0^i}{X_0^i} & \text{if } \bar{X}_{t_i} < X_0^b \quad \text{for all } i = 1, \ldots, m,
\end{array}
\right.
\end{align*}
\]

where $\{t_1, \ldots, t_m\}$ is a set of pre-fixed call dates, $b > X_0$ is a pre-fixed barrier level, and $\{Q_1, \ldots, Q_m\}$ is a set of pre-fixed coupons. The set of call dates $\{t_1, \ldots, t_m\}$ does not coincide with the set of times of the Euler scheme discretization $\{t_0, \ldots, t_n\}$. In particular, the latter has finer time resolution grid.

We show in Section 4.2 that all previous payoffs can be priced efficiently by using our novel algorithm, i.e., by reverting the MC paths and simulating them from maturity back to the initial date.

### 4.2 Numerical results and discussion

Let us introduce some terminology that we will use in the summary tables of our numerical results. In particular, we will denote by (i) **Euler Scheme** the prices obtained via a standard MC procedure on the process $\bar{X}$, (ii) **Euler Scheme CV** the prices obtained via a MC procedure on the process $\bar{X}$ when using the variance reduction technique control variates, (iii) **Euler Scheme IS** the prices obtained via a MC procedure on the process $\bar{X}$ when using the variance reduction technique importance sampling, (iv) **Forward RMQA** the prices obtained via a forward MC procedure on $\hat{X}$ from the starting date to the maturity, when grids and transition probabilities are computed through the RMQA, (v) **Forward LTSA** the prices obtained via a forward MC procedure on $\hat{X}$ from the starting date to the maturity, when grids and transition probabilities are computed through the LTSA, (vi) **Backward RMQA** the prices computed through the backward MC algorithm on $\hat{X}$ when grids and transition probabilities are computed through the RMQA, (vii) **Backward LTSA** the prices computed through the backward MC algorithm on $\hat{X}$ when grids and transition probabilities are computed through the LTSA, (viii) **Benchmark** the price computed either as an Euler Scheme price (in case of Asian and up-and-out barrier call options) or as a Forward LTSA price (in case of auto-callable option) with an intensive MC simulation in order to reduce the pricing error below the last significant digit. Besides, in brackets, we will report the numerical error corresponding to one standard deviation.

Let us now stress some aspects related to the implementation of the backward MC algorithm along with the procedures described in Sections 3.1 and 3.2.
In order to have a meaningful comparison between Euler Scheme-, Forward-, and Backward-type prices, for each of the $N^+$ points $\tilde{\gamma}_n^i \in \tilde{\Gamma}_n$, we generate $N_{MC}^b$ random paths in such a way that $N_{MC}^b \times N^+ = N_{MC}$, where $N_{MC}$ indicates the number of simulations employed to compute either Euler Scheme- or Forward-type prices. Besides, $\tilde{\Gamma}_n = \{ \gamma_n^i : K \leq \gamma_n^i \leq B \}$ when pricing up-and-out call barrier options and $\tilde{\Gamma}_n = \Gamma_n$ when pricing In-The-Money (ITM), At-The-Money (ATM), Out-The-Money Asian call options, and auto-callable options.

Concerning the granularity of the state-space, we fix it in such a way that the error on vanilla call option prices, computed as

$$|\sigma^{nkt} - \sigma^{alg}|$$

is less than or equal to 5 bps (recall that 1 bp = $10^{-4}$). In equation (4.3), $\sigma^{nkt}$ is the market implied volatility, whereas $\sigma^{alg}$ is the volatility implicitly obtained by using equation (3.11). In particular, we set the cardinality of the quantizers $\Gamma_k$, $1 \leq k \leq n$, to 100. We refer to Appendices C and D for precise considerations on this aspect.

The stopping criteria for the RMQA corresponds to $\|\Gamma_{k+1}^l - \Gamma_k^l\| \leq 10^{-5}$, $1 \leq k \leq n$, where $\Gamma_k^l$ is the quantizer computed by the algorithm at time $t_k \in \{t_1, \ldots, t_n\}$ at the $l$th iteration. Moreover, in the Backward MC algorithm case, for each point in $\tilde{\Gamma}_n$, we generate $N_{MC}^b = N_{MC} / |N^+| = 10^4 / |\tilde{\Gamma}_n|$ random paths. Let us now come to the discussion of the numerical results.

In Tables 1 and 2, we report up-and-out barrier call option prices for both LV and CEV, as well as their relative pricing errors. In order to test the performances of our algorithm, we price ITM, ATM, and OTM options. In particular, for both models the initial spot price is $\tilde{X}_0 = 1.36$. This value corresponds to the value of the EUR/USD exchange rate as of date 23-June-2014. Besides, for both dynamics, the value of the pair strike-barrier, $(K, B)$, is set to $(1.35, 1.39)$, $(1.36, 1.39)$, and to $(1.37, 1.39)$ for ITM, ATM, and OTM up-and-out call barrier options, respectively. The maturity $T$ is 6 months in the case of CEV model. In the case of LV model, the maturity is $T = 1$ month, $T = 6$ months or $T = 12$ months. The number of Euler steps is $n = 9$, $n = 51$, and $n = 100$ for $T = 1$ month, $T = 6$ months, or $T = 12$ months, respectively. For the CEV model, we fix $\alpha = 0.5$ and $r = r_d - r_f = 0.32\%$ (the latter corresponds to the value of the 6 months domestic interest rates implied by the forward USD curve at pricing date). As regards, the parameter $\sigma$ we vary it from $\sigma = 5\%$ to $\sigma = 20\%$ with steps $\Delta \sigma = 5\%$.

In Tables 1 and 2, we compare the Euler Scheme MC price and error with those of the other methods for the LV and CEV dynamics, respectively. For both models, the Backward MC algorithm exhibits the best performances. To better visualize this fact, Figures 4 and 5 report the ratio between the Euler Scheme MC error and that of the other techniques, henceforth Error ratio, for the LV and the CEV, respectively. Some observations arise. The two methods employed to recover grids and transition probabilities lead to similar results. Actually, in the case of CEV model with $\sigma = 20\%$, the algorithm performs better when grids and probabilities are recovered through the LTSA. Figure 5 suggests that the gain in efficiency is more evident if we increase the value of the parameter $\sigma$. Intuitively, this happens because the probability for the price paths to hit the barrier $B$ over the life of the option increases with $\sigma$. Moreover, for a fixed value of $\sigma$, the gain in efficiency
Table 1. Numerical values for Euler Scheme, Euler Scheme I.S., Forward RMQA, Forward LTSA, Backward RMQA, Backward LTSA, Benchmark prices for an up-and-out barrier call option, and LV dynamics. Errors (in brackets) correspond to one standard deviation. The initial spot price is $\bar{X}_0 = 1.36$, whereas the pair strike-barrier is set to $(1.35, 1.39)$, $(1.36, 1.39)$, $(1.37, 1.39)$ for ITM, ATM, and OTM options, respectively. The maturity $T$ is 1 month, 6 months, or 12 months.

| Algorithm               | ITM          | ATM          | OTM          |
|-------------------------|--------------|--------------|--------------|
|                         | $T = 1$ month|              |              |
| Euler Scheme            | 5.156E-3 (7.3E-5) | 2.316E-3 (4.5E-5) | 6.90E-4 (2.1E-5) |
| Euler Scheme I.S.       | 5.089E-3 (6.4E-5) | 2.338E-3 (4.1E-5) | 6.95E-4 (1.9E-5) |
| Forward RMQA            | 4.950E-3 (6.4E-5) | 2.073E-3 (3.8E-5) | 5.74E-4 (1.6E-5) |
| Forward LTSA            | 4.929E-3 (6.4E-5) | 2.081E-3 (3.8E-5) | 5.82E-4 (1.6E-5) |
| Backward RMQA           | 5.159E-3 (2.7E-5) | 2.291E-3 (1.4E-5) | 6.78E-4 (5E-6)   |
| Backward LTSA           | 5.118E-3 (2.7E-5) | 2.252E-3 (1.4E-5) | 6.65E-4 (5E-6)   |
| Benchmark               | 5.150E-3      | 2.308E-3      | 6.94E-4       |
|                         | $T = 6$ months|              |              |
| Euler Scheme            | 1.089E-3 (3.7E-5) | 4.47E-4 (1.9E-5) | 1.37E-4 (9E-6)  |
| Euler Scheme I.S.       | 1.082E-3 (3.2E-5) | 4.40E-4 (1.7E-5) | 1.30E-4 (8E-6)  |
| Forward RMQA            | 9.92E-4 (3.45-5) | 4.11E-4 (1.9E-5) | 1.24E-4 (3E-6)  |
| Forward LTSA            | 1.046E-3 (1.7E-5) | 4.59E-4 (9E-6)  | 1.38E-4 (8E-6)  |
| Backward RMQA           | 1.053E-3 (1.7E-5) | 4.54E-4 (9E-6)  | 1.38E-4 (7E-6)  |
| Backward LTSA           | 1.069E-3      | 4.56E-4       | 1.41E-4       |
| Benchmark               |              |              |              |
|                         | $T = 12$ months|             |              |
| Euler Scheme            | 3.01E-4 (2.0E-5) | 1.28E-4 (1.1E-5) | 3.84E-5 (5.2E-6) |
| Euler Scheme I.S.       | 2.70E-4 (1.6E-5) | 1.37E-4 (1.0E-5) | 3.09E-5 (3.8E-6) |
| Forward RMQA            | 2.91E-4 (1.9E-5) | 1.20E-4 (1.05-5) | 3.05E-5 (3.5E-6) |
| Forward LTSA            | 2.81E-4 (1.8E-5) | 1.17E-4 (1.0E-5) | 3.27E-5 (5.4E-6) |
| Backward RMQA           | 2.884E-4 (8E-6) | 1.17E-4 (4E-6)  | 3.64E-5 (1.3E-6) |
| Backward LTSA           | 2.91E-4 (8E-6)  | 1.16E-4 (4E-6)  | 3.47E-5 (1.3E-6) |
| Benchmark               | 2.769E-4      | 1.18E-4       | 3.72E-5       |

is more evident when pricing OTM options. This happens because for OTM options a relevant number of forward paths do not contribute to the payoff. In order to increase the pricing accuracy of the Euler Scheme MC, it would be necessary to force paths to sample the region in which the payoff is different from zero, namely between the strike $K$ and the barrier $B$. Remind that in the Euler Scheme price, we have already taken into account the possibility that the price path hits the barrier over a time interval $(t_{k-1}, t_k)$, $1 \leq k \leq n$. 


Table 2. Numerical values for Euler Scheme, Euler Scheme I.S., Forward RMQA, Forward LTSA, Backward RMQA, Backward LTSA, Benchmark prices for an up-and-out barrier call options, and CEV dynamics. Errors (in brackets) correspond to one standard deviation. The initial spot price is $\bar{X}_0 = 1.36$, whereas the pair strike-barrier is set to (1.35, 1.39), (1.36, 1.39), (1.37, 1.39) for ITM, ATM, and OTM options, respectively. The maturity $T$ is 6 months. $\sigma$ varies from 5% to 20% with step of 5%.

| Algorithm           | ITM      | ATM      | OTM      |
|---------------------|----------|----------|----------|
| CEV model           |          |          |          |
| $\sigma = 5\%$      |          |          |          |
| Euler Scheme        | 2.491E-3 (6.8E-5) | 1.102E-3 (3.9E-5) | 3.34E-4 (1.8E-5) |
| Euler Scheme I.S.   | 2.457E-3 (6.0E-5) | 1.087E-3 (3.4E-5) | 3.52E-4 (1.6E-5) |
| Forward RMQA        | 2.417E-3 (6.7E-5) | 1.088E-3 (3.9E-5) | 3.40E-4 (1.8E-5) |
| Forward LTSA        | 2.680E-3 (7.1E-5) | 1.109E-3 (4.0E-5) | 3.32E-4 (1.9E-5) |
| Backward RMQA       | 2.426E-3 (3.4E-5) | 1.087E-3 (1.7E-5) | 3.54E-4 (7E-6)   |
| Backward LTSA       | 2.525E-3 (3.3E-5) | 1.114E-3 (1.7E-5) | 3.50E-4 (7E-6)   |
| Benchmark           | 2.509E-3 | 1.117E-3 | 3.46E-4  |
| $\sigma = 10\%$    |          |          |          |
| Euler Scheme        | 4.06E-4 (2.8E-5) | 1.71E-4 (1.5E-5) | 5.63E-5 (7.4E-6) |
| Euler Scheme I.S.   | 4.22E-4 (2.4E-5) | 1.57E-4 (1.2E-5) | 5.82E-5 (6.3E-6) |
| Forward RMQA        | 5.07E-4 (3.25-5) | 1.90E-04 (1.7E-5) | 5.26E-5 (7.3E-6) |
| Forward LTSA        | 4.25E-4 (2.9E-5) | 1.86E-04 (1.7E-5) | 5.28E-5 (7.2E-6) |
| Backward RMQA       | 4.16E-4 (1.2E-5) | 1.86E-4 (6E-6) | 5.24E-5 (2.2E-6) |
| Backward LTSA       | 4.13E-4 (1.1E-5) | 1.71E-4 (5E-6) | 5.02E-5 (2.1E-6) |
| Benchmark           | 4.08E-4 | 1.73E-4 | 5.12E-5  |
| $\sigma = 15\%$    |          |          |          |
| Euler Scheme        | 1.64E-4 (1.8E-5) | 5.26E-5 (8.8E-6) | 1.75E-5 (4.2E-6) |
| Euler Scheme I.S.   | 1.36E-4 (1.2E-5) | 5.81E-5 (7.1E-6) | 2.08E-5 (3.4E-6) |
| Forward RMQA        | 1.44E-4 (1.7E-5) | 5.48E-5 (9.25-5) | 1.96E-5 (4.7E-6) |
| Forward LTSA        | 1.35E-4 (1.6E-5) | 5.28E-5 (9.0E-5) | 1.78E-5 (4.2E-6) |
| Backward RMQA       | 1.24E-4 (5E-6) | 5.22E-5 (2.7E-6) | 1.69E-5 (1.1E-6) |
| Backward LTSA       | 1.28E-4 (5E-6) | 5.77E-5 (2.7E-6) | 1.93E-5 (1.1E-6) |
| Benchmark           | 1.31E-4 | 5.54E-5 | 1.65E-5  |
| $\sigma = 20\%$    |          |          |          |
| Euler Scheme        | 6.20E-5 (1.2E-5) | 2.38E-5 (5.8E-6) | 5.24E-6 (2.2E-06) |
| Euler Scheme I.S.   | 5.49E-5 (8E-6) | 3.14E-5 (5.4E-6) | 6.22E-6 (1.8E-06) |
| Forward RMQA        | 6.71E-5 (1.2E-5) | 2.34E-5 (5.9E-6) | 5.93E-6 (2.3E-06) |
| Forward LTSA        | 5.73E-5 (9E-6) | 2.45E-5 (4.4E-6) | 4.27E-6 (1.9E-06) |
| Backward RMQA       | 5.75E-5 (3E-6) | 2.22E-5 (1.4E-6) | 6.50E-6 (6E-07) |
| Backward LTSA       | 5.61E-5 (3E-6) | 2.06E-5 (1.2E-6) | 6.28E-6 (4E-07) |
| Benchmark           | 5.66E-5 | 2.38E-5 | 7.10E-6  |
Figure 4. Plot of the Error ratio as a function of the maturity $T$ for LV model when pricing up-and-out barrier call option. The initial spot price is $\bar{X}_0 = 1.36$, whereas the pair strike-barrier is set to $(1.35, 1.39), (1.36, 1.39), (1.37, 1.39)$ for ITM, ATM, and OTM options, respectively.

In Tables 3 and 4, we compare performances of the Euler Scheme MC with those of the Backward MC when pricing Asian call options, for both LV and CEV model. We set $\bar{X}_0 = 1.36$. As done for up-and-out barrier calls, we test the efficiency of our novel algorithm when pricing ITM ($K = 1.35$), ATM ($K = 1.36$), and OTM ($K = 1.37$) options. The maturity $T$ is 6 months in the case of CEV model. In the case of LV model, the maturity is $T = 1$ month, $T = 6$ months, or $T = 12$ months. The number of Euler steps is $n = 9$, $n = 51$, and $n = 100$ for $T = 1$ month, $T = 6$ months, or $T = 12$ months, respectively. Also in this case, for CEV model, we fix the value of the risk-free rate $r = r_d - r_f = 0.32\%$ and of $\alpha = 0.5$, and we vary the value of $\sigma$ from 5% to 20% with steps $\Delta \sigma = 5\%$. Results for the LV dynamics are reported in Table 3, whereas Table 4 reports the results for the CEV. Table 3 suggests that the strategy of reverting the MC paths and simulating them from maturity back to starting date is an effective alternative to the pure Euler MC or to Euler MC combined with a control variates variance reduction technique. In this case, the improvement in efficiency derives from the fact that with Backward MC we decide the number of paths to sample from each of the final points in $\Gamma_n$, sampling efficiently also those regions that are infrequently explored.
A backward Monte Carlo approach to exotic option pricing

σ = 5 %
σ = 10 %
σ = 15 %
σ = 20 %

Barrier Options, ITM, CEV model

Barrier Options, ATM, CEV model

Barrier Options, OTM, CEV model

Figure 5. Plot of the Error ratio as a function of the parameter σ for CEV model when pricing up-and-out barrier call option. The initial spot price is \( \bar{X}_0 = 1.36 \), whereas the pair strike-barrier is set to \((1.35, 1.39), (1.36, 1.39), (1.37, 1.39)\) for ITM, ATM, and OTM options, respectively.

by the price process because of its diffusive behaviour. Besides, we note that grids and transition probabilities recovered through the RMQA perform better than LTSA. This is because the grids in the RMQA are obtained as local minima of the distortion function, i.e., they well approximate the law of the stochastic process at any discretization date. Figures 6 and 7 support our conclusions, especially when pricing OTM options. The Error ratio is almost constant across the value of σ for a fixed scenario (ITM, ATM, or OTM).

In Table 5, we report prices of auto-callable options for LV dynamics. In this case, we compare only the Backward LTSA with the Forward LTSA. The Euler Scheme MC is ineffective for payoffs specifications which depend on the observation of the underlying on a pre-specified set of dates (such as auto-callable and European). The multinomial tree and the transition probability matrices are recovered by means of the LTSA. In order to compare the two MC methodologies, we fix a set of call-dates \( \{t_1, \ldots, t_4\} \) and a set of pre-fixed coupon \( \{Q_1, \ldots, Q_4\} \) and we vary the value of the barrier b. Precisely, \( \{t_1, \ldots, t_4\} = \{1, 3, 6, 12\} \) months, \( \{Q_1, \ldots, Q_4\} = \{5\%, 10\%, 15\%, 20\%\} \) with unitary notional, and \( b \in \{\bar{X}_0, 1.05\bar{X}_0, 1.1\bar{X}_0\} \). As usual, at pricing date, the EUR/USD exchange rate is \( \bar{X}_0 = 1.36 \). Results in Table 5 show that the improvement on pricing
Table 3. Numerical values for Euler Scheme, Euler Scheme CV, Forward RMQA, Forward LTSA, Backward RMQA, Backward LTSA, Benchmark prices for an Asian option, and LV dynamics. Errors (in brackets) correspond to one standard deviation. The initial spot price is $\bar{X}_0 = 1.36$, whereas the value of the strike is set to 1.35, 1.36, and 1.37 for ITM, ATM, and OTM options, respectively. The maturity $T$ is 1 month, 6 months, or 12 months. The Euler step is $\Delta t = 0.01$ ($n = 9, n = 51$, and $n = 100$ when $T$ is 1 month, 6 months, or 12 months, respectively).

| Algorithm          | ITM     | ATM     | OTM     |
|--------------------|---------|---------|---------|
| Euler Scheme       | 8.669E-3 (8.0E-5) | 3.920E-3 (5.9E-5) | 1.321E-3 (3.1E-5) |
| Euler Scheme C.V.  | 8.669E-3 (4.1E-5) | 3.988E-3 (3.7E-5) | 1.379E-3 (2.9E-5) |
| Forward RMQA       | 8.860E-3 (8.2E-5) | 4.062E-3 (6.1E-5) | 1.433E-3 (3.7E-5) |
| Forward LTSA       | 8.773E-3 (8.2E-5) | 3.988E-3 (6.0E-5) | 1.430E-3 (3.7E-5) |
| Backward RMQA      | 8.725E-3 (4.2E-5) | 3.948E-3 (3.1E-5) | 1.362E-3 (1.7E-5) |
| Backward LTSA      | 8.690E-3 (5.8E-5) | 3.957E-3 (4.3E-5) | 1.373E-3 (2.3E-5) |
| Benchmark          | 8.750E-3 | 3.978E-3 | 1.374E-3 |

$T = 1$ month

| Algorithm          | ITM     | ATM     | OTM     |
|--------------------|---------|---------|---------|
| Euler Scheme       | 1.3656E-2 (1.64E-4) | 9.408E-3 (1.41E-4) | 6.149E-3 (1.17E-4) |
| Euler Scheme C.V.  | 1.3980E-2 (1.06E-4) | 9.707E-3 (0.98E-4) | 6.402E-3 (0.88E-4) |
| Forward RMQA       | 1.3698E-2 (1.64E-4) | 9.446E-3 (1.40E-4) | 6.197E-3 (1.16E-4) |
| Forward LTSA       | 1.3535E-2 (1.61E-4) | 9.228E-3 (1.37E-4) | 5.792E-3 (1.12E-4) |
| Backward RMQA      | 1.3536E-2 (1.09E-4) | 9.305E-3 (0.92E-4) | 6.068E-3 (0.74E-4) |
| Backward LTSA      | 1.3567E-2 (1.43E-4) | 9.321E-3 (1.23E-4) | 5.852E-3 (0.93E-4) |
| Benchmark          | 1.3724E-2 | 9.4663E-3 | 6.196E-3 |

$T = 6$ months

| Algorithm          | ITM     | ATM     | OTM     |
|--------------------|---------|---------|---------|
| Euler Scheme       | 1.8160E-2 (2.33E-4) | 1.4019E-2 (2.10E-4) | 1.0548E-2 (1.86E-4) |
| Euler Scheme C.V.  | 1.7840E-2 (1.59E-4) | 1.3662E-2 (1.50E-4) | 1.0203E-2 (1.39E-4) |
| Forward RMQA       | 1.8085E-2 (2.27E-4) | 1.3912E-2 (2.04E-4) | 1.0413E-2 (1.79E-4) |
| Forward LTSA       | 1.7709E-2 (2.28E-4) | 1.3527E-2 (2.06E-4) | 1.0084E-2 (1.83E-4) |
| Backward RMQA      | 1.8025E-2 (1.66E-4) | 1.3862E-2 (1.48E-4) | 1.0341E-2 (1.29E-4) |
| Backward LTSA      | 1.7898E-2 (2.10E-4) | 1.3685E-2 (1.87E-4) | 1.0183E-2 (1.62E-4) |
| Benchmark          | 1.8111E-2 | 1.3939E-2 | 1.0457E-2 |

$T = 12$ months

Accuracy due to the backward simulation of the price paths widens with the increase of the value of the barrier $b$ (in real market usually one has $b > \bar{X}_0$). The ratio between the pricing error of the Forward MC and that of the Backward MC is $\approx 0.8$, $\approx 2$ and $\approx 5.5$ for $b = \bar{X}_0$, $b = 1.05\bar{X}_0$, and $b = 1.1\bar{X}_0$, respectively. Intuitively, this happens because an increase in the value of the barrier $b$ makes the early exercise of the option less
A backward Monte Carlo approach to exotic option pricing

Table 4. Numerical values for Euler Scheme, Euler Scheme CV, Forward RMQA, Forward LTSA, Backward RMQA, Backward LTSA, Benchmark prices for an Asian call option, and CEV dynamics. Errors (in brackets) correspond to one standard deviation. The initial spot price is \( \bar{X}_0 = 1.36 \), whereas the strike is 1.35, 1.36, and 1.37 for ITM, ATM, and OTM options, respectively. The maturity \( T \) is 6 months. The Euler time step is \( \Delta = 0.01 \) (\( n = 51 \)). \( \sigma \) varies from 5% to 20% with step of 5%.

| Algorithm        | ITM                      | ATM                      | OTM                      |
|------------------|--------------------------|--------------------------|--------------------------|
| Asian call       | CEV model                |                          |                          |
| \( \sigma = 5\% \) |                          |                          |                          |
| Euler Scheme     | 1.5942E-2 (1.75E-4)      | 9.962E-3 (1.43E-4)       | 5.651E-3 (1.09E-4)       |
| Euler Scheme C.V.| 1.6284E-2 (1.04E-4)      | 1.0245E-2 (9.5E-5)       | 5.859E-3 (8.0E-5)        |
| Forward RMQA     | 1.5194E-2 (1.74E-4)      | 9.969E-3 (1.42E-4)       | 5.649E-3 (1.07E-4)       |
| Forward LTSA     | 1.6202E-2 (1.77E-4)      | 1.0021E-2 (1.44E-4)      | 5.727E-3 (1.10E-4)       |
| Backward RMQA    | 1.5844E-2 (1.05E-4)      | 9.867E-3 (6.5E-5)        | 5.542E-3 (6.5E-5)        |
| Backward LTSA    | 1.5941E-2 (1.53E-4)      | 1.0021E-2 (9.3E-5)       | 5.649E-3 (9.3E-5)        |
| Benchmark        | 1.5964E-2                | 9.985e-3                 | 5.662E-3                 |
| \( \sigma = 10\% \) |                          |                          |                          |
| Euler Scheme     | 2.4825E-2 (3.17E-4)      | 1.9362E-2 (2.84E-4)      | 1.474E-2 (2.50E-4)       |
| Euler Scheme C.V.| 2.5449E-2 (1.99E-4)      | 1.9929E-2 (1.89E-4)      | 1.5245E-2 (1.75E-4)      |
| Forward RMQA     | 2.4634E-2 (3.12E-4)      | 1.9202E-2 (2.79E-4)      | 1.4611E-2 (2.44E-4)      |
| Forward LTSA     | 2.4972E-2 (3.17E-4)      | 1.9487E-2 (2.84E-4)      | 1.4847E-2 (2.50E-4)      |
| Backward RMQA    | 2.4588E-2 (1.92E-4)      | 1.9135E-2 (1.71E-4)      | 1.4506E-2 (1.49E-4)      |
| Backward LTSA    | 2.5015E-2 (2.50E-4)      | 1.9490E-2 (2.23E-4)      | 1.4808E-2 (1.94E-4)      |
| Benchmark        | 2.4871E-2                | 1.9441E-2                | 1.4789E-2                |
| \( \sigma = 15\% \) |                          |                          |                          |
| Euler Scheme     | 3.4044E-2 (4.60E-4)      | 2.8765E-2 (4.27E-4)      | 2.4047E-2 (3.93E-4)      |
| Euler Scheme C.V.| 3.4938E-2 (2.93E-4)      | 2.9617E-2 (2.83E-4)      | 2.4848E-2 (2.70E-4)      |
| Forward RMQA     | 3.4071E-2 (4.57E-4)      | 2.8818E-2 (4.23E-4)      | 2.4113E-2 (3.89E-4)      |
| Forward LTSA     | 3.4378E-2 (4.62E-4)      | 2.9075E-2 (4.29E-4)      | 2.4334E-2 (3.95E-4)      |
| Backward RMQA    | 3.3832E-2 (2.78E-4)      | 2.8548E-2 (2.57E-4)      | 2.3820E-2 (2.36E-4)      |
| Backward LTSA    | 3.4027E-2 (3.51E-4)      | 2.8680E-2 (3.24E-4)      | 2.3921E-2 (2.95E-4)      |
| Benchmark        | 3.411E-2                 | 2.883E-2                 | 2.4130E-2                |
| \( \sigma = 20\% \) |                          |                          |                          |
| Euler Scheme     | 4.3345E-2 (6.05E-4)      | 3.8166E-2 (5.72E-4)      | 3.3414E-2 (5.38E-4)      |
| Euler Scheme C.V.| 4.4514E-2 (3.88E-4)      | 3.9306E-2 (3.77E-4)      | 3.4506E-2 (3.58E-4)      |
| Forward RMQA     | 4.3372E-2 (5.97E-4)      | 3.8191E-2 (5.63E-4)      | 3.3428E-2 (5.29E-4)      |
| Forward LTSA     | 4.3422E-2 (6.01E-4)      | 3.8246E-2 (5.67E-4)      | 3.3479E-2 (5.34E-4)      |
| Backward RMQA    | 4.3013E-2 (3.62E-4)      | 3.7836E-2 (3.41E-4)      | 3.3086E-2 (3.20E-4)      |
| Backward LTSA    | 4.3505E-2 (4.55E-4)      | 3.8291E-2 (4.28E-4)      | 3.3502E-2 (3.99E-4)      |
| Benchmark        | 4.3426E-2                | 3.8265E-2                | 3.3524E-2                |
Figure 6. Plot of the Error ratio as a function of the maturity $T$ for LV model when pricing Asian call option. The initial spot price is $\bar{X}_0 = 1.36$, whereas the strike is set to 1.35, 1.36, and 1.37 for ITM, ATM, and OTM options, respectively.

...probable, and a larger number of paths will reach the final domain of integration. Then, our methodology allows to efficiently sample paths drawing from regions that would be infrequently explored by the forward Euler algorithm.

5 Conclusions

In this paper, we present a novel approach – termed backward MC – to the MC simulation of continuous time diffusion processes. We exploit recent advances in the quantization of diffusion processes to approximate the continuous process with a discrete-time Markov Chain defined on a finite grid of points. Specifically, we consider the RMQA and as a first contribution we investigate a fixed-point scheme – termed Lloyd I method with Anderson acceleration – to compute the optimal grid in a robust way. As a complementary approach, we consider the grid associated with the explicit scheme approximation of the Markov generator of a piecewise constant volatility process. The latter approach – termed LTSA – turns out to be competitive in pricing payoff spe-
Figure 7. Plot of the Error ratio as a function of the parameter $\sigma$ for CEV model when pricing Asian call options. The initial spot price is $\bar{X}_0 = 1.36$, whereas the strike is set to 1.35, 1.36, and 1.37 for ITM, ATM, and OTM options, respectively.

Table 5. Numerical values for Forward, Backward, and Benchmark prices for an auto-callable option in the LV. Errors (in brackets) correspond to one standard deviation. The initial spot price is $\bar{X}_0 = 1.36$. The value of the barrier $b$ is equal to $\bar{X}_0$, $1.05\bar{X}_0$, and $1.1\bar{X}_0$ for the three experiments considered. The call-dates are fixed to $\{1, 3, 6, 12\}$ months and the pre-fixed coupons are fixed to $\{5\%, 10\%, 15\%, 20\%\}$ of a unitary notional

| Auto-callable options | Barrier $b$ | $b = \bar{X}_0$ | $b = 1.05\bar{X}_0$ | $b = 1.1\bar{X}_0$ |
|----------------------|------------|-----------------|---------------------|-------------------|
| **Algorithm**        |            |                 |                     |                   |
| Forward LTSA         | 0.04107    | 0.01902         | 0.00447             |
| Backward LTSA        | 0.04099    | 0.01856         | 0.00377             |
| Benchmark            | 0.04099    | 0.01820         | 0.00357             |
cifications which require the observation of the price process over a finite number of pre-specified dates. Both methods – quantization and the explicit scheme – provide us with the marginal and transition probabilities associated with the points of the approximating grid. Sampling from the discrete grid backward – from the terminal point to the spot value of the process – we design a simple but effective mechanism to draw MC path and achieve a sizeable reduction of the variance associated with MC estimators. Our conclusion is extensively supported by the numerical results presented in the final section.

Acknowledgements

All authors are grateful to the anonymous Referees for fruitful comments, and thank participants of the Junior Female Researchers in Probability 2015 in Berlin, of the XVII Workshop on Quantitative Finance 2016 in Pisa, and of the SIMAI2016 in Milan for fruitful comments.

References

ALBANESE, C. (2007) Operator methods, Abelian processes and dynamic conditioning. URL: http://ssrn.com/abstract=1018490, 29 March 2017.

ALBANESE, C., LO, H. & MIJATOVIĆ, A. (2009) Spectral methods for volatility derivatives. Quantitative Finance 9(6), 663–692.

ALBANESE, C. & MIJATOVIĆ, A. (2007) Convergence rates for diffusions on continuous-time lattices. URL: http://ssrn.com/abstract=1018609, 29 March 2017.

ANDERSON, D. G. (1965) Iterative procedures for nonlinear integral equations. J. ACM (JACM) 12(4), 547–560.

ANDREASEN, J. & HUIE, B. (2011) Volatility interpolation. Risk Mag. 24(3), 76.

BAKER, G. A. & GRAVES-MORRIS, P. R. (1996) Padé Approximants. Cambridge University Press, New York.

BALLY, V. & PAGÈS, G. (2003) A quantization algorithm for solving multidimensional discrete-time optimal stopping problems. Bernoulli 9(6), 1003–1049.

BARRAUD, J. & MARTINEAU, D. (1995) Numerical valuation of high dimensional multivariate American securities. J. Financ. Quant. Anal. 30(3), 383–405.

BERKAOUI, A., BOSSY, M. & DIOP, A. (2008) Euler scheme for SDEs with non-Lipschitz diffusion coefficient: Strong convergence. ESAIM: Probability and Statistics 12, 1–11.

BÖRCK, T. (2009) Arbitrage Theory in Continuous Time. Oxford University Press, Oxford.

BLACK, F. & SCHOLES, M. (1973) The pricing of options and corporate liabilities. J. Political Economy 81(3), 637–654.

BLANES, S., CASAS, F., OTEO, J. A. & ROS, J. (2009) The Magnus expansion and some of its applications. Phys. Rep. 470(5–6), 151–238.

BORMETTI, G., MONTAGNA, G., MORENI, N. & NICROSINI, O. (2006) Pricing exotic options in a path integral approach. Quant. Finance 6(1), 55–66.

BOSSY, M. & DIOP, A. (2007) An efficient discretization scheme for one dimensional SDEs with a diffusion coefficient function of the form $|x|^a$, $a$ in $[1/2,1)$. Res. Rep. INRIA RR-5396, 44.

BOUCHAUD, J.-P. & POTTERS, M. (2003) Theory of Financial Risk and Derivative Pricing: From Statistical Physics to Risk Management. Cambridge University Press, Cambridge.

CALLEGARO, G., FIORIN, L. & GRASSELLI, M. (2015) Quantized calibration in local volatility. Risk Mag. 28(4), 62–67.
A backward Monte Carlo approach to exotic option pricing

CARMONA, R., DEL MORAL, P., Hu, P. & OUDJANE, N. (2012) Numerical Methods in Finance. Springer Proceedings in Mathematics, Vol. 12.

CLEWLOW, L. & STRICKLAND, C. (1996) Implementing Derivative Models (Wiley Series in Financial Engineering). John Wiley & Sons, New York.

COLEMAN, T. F., LI, Y. & VERMA, A. (1999) Reconstructing the unknown local volatility function. J. Comput. Finance 2(3), 77–100.

COX, J. (1975) Notes on Option Pricing I: Constant Elasticity of Variance Diffusions. Unpublished note, Stanford University, Graduate School of Business.

DENG, G., MALLETT, J. & McCANN, C. (2011) Modeling autocallable structured products. J. Derivatives Hedge Funds 17(4), 326–340.

DERMAN, E., KANI, I. & ZOU, J. Z. (1996) The local volatility surface: Unlocking the information in index option prices. Financ. Anal. J. 52(4), 25–36.

DEWYNNE, J. N. & SHAW, W. T. (2008) Differential equations and asymptotic solutions for arithmetic Asian options: Black–Scholes formulae for Asian rate calls. Eur. J. Appl. Math. 19(4), 353–391.

DUPIRE, B. (1994) Pricing with a smile. Risk Mag. 7(1), 18–20.

FRITSCH, F. N. & CARLSON, R. E. (1980) Monotone piecewise cubic interpolation. SIAM J. Numer. Anal. 17(2), 238–246.

GATHERAL, J. (2011) The Volatility Surface: A Practitioner’s Guide. John Wiley & Sons, Hoboken, New Jersey.

GLASSERMAN, P. (2004) Monte Carlo methods in Financial Engineering. Applications of Mathematics (New York), Vol. 53, Stochastic Modeling and Applied Probability. Springer-Verlag, New York.

GOBET, E. (2000) Weak approximation of killed diffusion using Euler schemes. Stoch. Process. Appl. 87(2), 167–197.

GOLUB, G. H. & VAN LOAN C. F. (2012) Matrix Computations. JHU Press, USA.

GRAF, S. & LUSCHGY, H. (2000) Foundations of Quantization for Probability Distributions. Lectures Notes in Mathematics, Vol. 1730, Springer-Verlag, Berlin.

HIGHAM, N. J. (2005) The scaling and squaring method for the matrix exponential revisited. SIAM J. Matrix Anal. Appl. 26(4), 1179–1193.

HUI, C. H., LO, C.-F. & YUEN, P. H. (2000) Comment on ‘Pricing double barrier options using Laplace transforms’ by Antoon Pelsser. Finance Stoch. 4(1), 105–107.

HULL, J. (2006) Options, Futures, and Other Derivatives, 9th ed., Pearson Education, India.

JEANBLANC, M., YOR, M. & CHESNEY, M. (2009) Mathematical Methods for Financial Markets. Springer-Verlag, Berlin.

JOSHI, M. S. & LEUNG, T. S. (2011) Using Monte Carlo simulation and importance sampling to rapidly obtain jump-diffusion prices of continuous barrier options. J. Comput. Finance 10(4), 93–105.

KAHALÉ, N. (2004) An arbitrage-free interpolation of volatilities. Risk Mag. 17(5), 102–106.

KARATZAS, I. & SHREVE, S. (1998) Brownian Motion and Stochastic Calculus. Graduate Texts in Mathematics, Vol. 113. Springer-Verlag, New York.

KIEFFER, J. B. (1982) Exponential rate of convergence for Lloyd’s method I. IEEE Trans. Inform. Theory 28(2), 205–210.

KIJIMA, M. (1997) Markov Processes for Stochastic Modelling. Stochastic Modeling Series. Chapman & Hall, London.

KRONMAL, R. A. & PETERSON JR, A. V. (1979) On the Alias method for generating random variables from a discrete distribution. Am. Stat. 33(4), 214–218.

KUSHNER, H. & DUPUIS, P. G. (2001) Numerical Methods for Stochastic Control Problems in Continuous Time. Applications of Mathematics (New York), 2nd ed., Vol. 24, Stochastic Modeling and Applied Probability, Springer-Verlag, New York.
Appendix A The distortion function and companion parameters

We suppose to have access to the quantizer $\Gamma_k$ of $X_{tk}$ and to the related Voronoi tessellations \{ $C_i(\Gamma_k)$ \} $i = 1, \ldots, N$. We derive an explicit expression for the distortion function
\[ \tilde{D}(\Gamma_{k+1}) \text{ as follows:} \]
\[
\tilde{D}(\Gamma_{k+1}) = \mathbb{E} \left[ d(\xi_k(\widehat{X}_{t_k}, \Delta t; \tilde{Z}_{t_{k+1}}), \Gamma_{k+1})^2 \right]
\]
\[
= \sum_{i=1}^{N} \mathbb{E} \left[ d(\xi_k(\gamma_{t_k}^i, \Delta t; Z_{t_{k+1}}), \Gamma_{k+1})^2 \right] \mathbb{P}(\widehat{X}_{t_k} \in C_i(\Gamma_k))
\]
\[
= \sum_{i=1}^{N} \sum_{j=1}^{N} \left( m_k(\gamma_{t_k}^i) - \gamma_{t_k+1}^i \right)^2 (\Phi(\gamma_{k+1,j}^i(\gamma_{t_k}^i)) - \Phi(\gamma_{k+1,j}^i(\gamma_{t_k}^i))) \mathbb{P}(\widehat{X}_{t_k} \in C_i(\Gamma_k))
\]
\[-2 \sum_{i=1}^{N} \sum_{j=1}^{N} \left( m_k(\gamma_{t_k}^i) - \gamma_{t_k+1}^i \right) v_k(\gamma_{t_k}^i) (\Phi(\gamma_{k+1,j}^i(\gamma_{t_k}^i)) - \Phi(\gamma_{k+1,j}^i(\gamma_{t_k}^i))) \mathbb{P}(\widehat{X}_{t_k} \in C_i(\Gamma_k))
\]
\[+ \sum_{i=1}^{N} \sum_{j=1}^{N} v_k(\gamma_{t_k}^i)^2 (\gamma_{k+1,j}^i(\gamma_{t_k}^i)) \Phi(\gamma_{k+1,j}^i(\gamma_{t_k}^i)) \mathbb{P}(\widehat{X}_{t_k} \in C_i(\Gamma_k))
\]
\[+ \sum_{i=1}^{N} \sum_{j=1}^{N} v_k(\gamma_{t_k}^i)^2 (\Phi(\gamma_{k+1,j}^i(\gamma_{t_k}^i)) - \Phi(\gamma_{k+1,j}^i(\gamma_{t_k}^i))) \mathbb{P}(\widehat{X}_{t_k} \in C_i(\Gamma_k)), \quad (A1) \]

where \( \Phi \) and \( \varphi \) indicate the cumulative distribution function and the probability density function of a standard Normal random variable, respectively. To simplify notation, in equation (A1), we set for all \( k \in \{0, \ldots, n-1\} \) and for all \( j \in \{1, \ldots, N\} \)

\[ \gamma_{k+1,j}^i(\gamma) = \frac{\gamma_{t_k}^{k+1} - m_k(\gamma)}{v_k(\gamma)} \quad \text{and} \quad \gamma_{k+1,j}^i(\gamma) = \frac{\gamma_{t_k}^{k+1} - m_k(\gamma)}{v_k(\gamma)} \]

where

\[ \gamma_{t_k}^{k+1} = \frac{\gamma_{t_k}^{k+1} + \gamma_{t_k+1}^{k+1}}{2}, \quad \gamma_{t_k}^{k+1} = \frac{\gamma_{t_k}^{k+1} + \gamma_{t_k+1}^{k+1}}{2}, \quad \gamma_{t_k}^{k+1} = -\infty, \quad \text{and} \quad \gamma_{t_k}^{k+1} = +\infty. \]

The so-called companion parameters \( \{\mathbb{P}(\widehat{X}_{t_k} \in C_i(\Gamma_k))\}_{i=1}^{N} \) and \( \{\mathbb{P}(\widehat{X}_{t_k} \in C_j(\Gamma_k)|\widehat{X}_{t_{k-1}} \in C_j(\Gamma_k))\}_{j=1}^{N} \) are computed in a recursive way as follows:

\[ \mathbb{P}(\widehat{X}_{t_k} \in C_i(\Gamma_k)) = \sum_{j=1}^{N} (\Phi(\gamma_{k,j}^i(\gamma_{t_k}^i)) - \Phi(\gamma_{k,j}^i(\gamma_{t_k}^i))) \mathbb{P}(\widehat{X}_{t_{k-1}} \in C_j(\Gamma_{k-1})), \]

\[ \mathbb{P}(\widehat{X}_{t_k} \in C_j(\Gamma_k)|\widehat{X}_{t_{k-1}} \in C_j(\Gamma_{k-1})) = \Phi(\gamma_{k,j}^i(\gamma_{t_k}^i)) - \Phi(\gamma_{k,j}^i(\gamma_{t_k}^i)). \]

**Appendix B Lloyd I method within the RMQA**

We present a brief review of the Lloyd I method within the Recursive Marginal Quantization framework. Let us fix \( t_k \in \{t_1, \ldots, t_n\} \) and suppose we have access to the quantizer \( \Gamma_k \) of \( X_{t_k} \) and to the associated Voronoi tessellations \( \{C_i(\Gamma_k)\}_{i=1}^{N} \). We want to quantize \( \widetilde{X}_{t_{k+1}} = \xi_k(\widehat{X}_{t_k}, \Delta t; Z_{t_{k+1}}) \) by means of a quantizer \( \Gamma_{k+1} = \{\gamma_{t_k+1}^1, \ldots, \gamma_{t_k+1}^N\} \) of cardinality \( N \). One starts with an initial guess \( \Gamma_{k+1}^0 \) and then one sets recursively a sequence \( (\Gamma_{k+1}^l)_{l \in \mathbb{N}} \) such that

\[ \gamma_{t_k+1}^{k+1,l+1} = \mathbb{E} \left[ \frac{\widetilde{X}_{t_{k+1}}}{\widetilde{X}_{t_{k+1}} \in C_j(\Gamma_{k+1}^l)} \right], \quad (B1) \]
where \( l \) indicates the running iteration number. One can easily check that previous equation implies that

\[
q_{N}^{l+1}(\tilde{X}_{t_{k+1}}) = \mathbb{E} \left[ \tilde{X}_{t_{k+1}} | q_{N}^{l}(\tilde{X}_{t_{k+1}}) \right] = \left( \mathbb{E} \left[ \tilde{X}_{t_{k+1}} | \tilde{X}_{t_{k+1}} \in C_{i}(\Gamma^{l+1}_{k+1}) \right] \right)_{1 \leq i \leq N},
\]

where \( q_{N}^{l} \) is the quantization associated with \( \Gamma^{l+1}_{k+1} \). It has been proven (see Carmona et al., 2012; Pagès et al., 2004) that \( \{\|\tilde{X}_{t_{k+1}} - q_{N}^{l}(\tilde{X}_{t_{k+1}})\|_{2}, l \in \mathbb{N}^{+}\} \) is a non-increasing sequence and that \( q_{N}^{l}(\tilde{X}_{t_{k+1}}) \) converges towards some random variable taking \( N \) values as \( l \) tends to infinity. From equation (B 1) and exploiting the idea of RMQA, we have

\[
\gamma_{k+1}^{j+1} = \mathbb{E} \left[ \tilde{X}_{t_{k+1}} | \tilde{X}_{t_{k+1}} \in C_{j}(\Gamma^{l}_{k+1}) \right]
\]

\[
= \mathbb{E} \left[ \tilde{X}_{t_{k+1}} \mathbb{I} \{ \tilde{X}_{t_{k+1}} \in C_{j}(\Gamma^{l}_{k+1}) \} \right] / \mathbb{P}(\tilde{X}_{t_{k+1}} \in C_{j}(\Gamma^{l}_{k+1}))
\]

\[
= \mathbb{E} \left[ \mathbb{I} \{ \tilde{X}_{t_{k+1}} \in C_{j}(\Gamma^{l}_{k+1}) \} \tilde{X}_{t_{k+1}} \right]
\]

\[
= \sum_{i=1}^{N} \mathbb{E} \left[ \mathbb{E}_{k}^{(y_{i}^{j})}(\Delta t; Z_{t_{k+1}}) \mathbb{I} \{ \mathbb{E}_{k}^{(y_{i}^{j})}(\Delta t; Z_{t_{k+1}}) \in C_{j}(\Gamma^{l}_{k+1}) \} \right] \mathbb{P}(\tilde{X}_{t_{k}} \in C_{j}(\Gamma_{k}))
\]

\[
= \sum_{i=1}^{N} \mathbb{P}(\mathbb{E}_{k}^{(y_{i}^{j})}(\Delta t; Z_{t_{k+1}}) \in C_{j}(\Gamma^{l}_{k+1}) \mathbb{P}(\tilde{X}_{t_{k}} \in C_{j}(\Gamma_{k}))
\]

The last term in previous equation is equivalent to the stationary condition in equation (3.2) for the quantization \( q_{N}(\tilde{X}_{t_{k+1}}) \). Then, the stationary condition is equivalent to a fixed point relation for the quantizer.

### Appendix C Robustness checks

We test the convergence of Lloyd I method with and without Anderson acceleration on the quantization of a standard Normal random variable initialized from a distorted quantizer (at www.quantize.maths-fi.com a database providing quadratic optimal quantizers of the standard univariate Gaussian distribution from level \( N = 1 \) to \( N = 1000 \) is available). We indicate by \( \Gamma_{N(0,1)}^{*} \) the optimal quantizer of a standard Normal random variable and we distort it through the multiplication by a constant \( c \), that is to say \( c \times \Gamma_{N(0,1)}^{*} \). Then, we monitor the convergence of both algorithms to \( \Gamma_{N(0,1)}^{*} \) starting from \( c \times \Gamma_{N(0,1)}^{*} \). The error at iteration \( l \) is defined as \( \|I_{1}^{N(0,1)} - \Gamma_{N(0,1)}^{l}\|_{2} \), with \( I_{1}^{N(0,1)} \) the quantizer found by the algorithms at the \( l \)th iteration and \( \| \cdot \|_{2} \) the Euclidean norm in \( \mathbb{R}^{N} \). The stopping criteria is set to \( \|I_{1}^{N(0,1)} - \Gamma_{N(0,1)}^{l}\|_{2} \leq 10^{-7} \), the level of the quantizer to \( N = 10 \), and the constant \( c \) to 1.01. The results of our investigation are summarized in Figure C 1. We can graphically assess the rate of convergence for both algorithms. We recall that a sequence \( (\Gamma^{l})_{l \in \mathbb{N}} \) converging to a \( \Gamma^{*} \neq \Gamma^{l} \) for all \( l \) is said to converge to \( \Gamma^{*} \) with order \( \alpha \) and asymptotic error constant \( \lambda \) if there exist positive constants \( \alpha \) and \( \lambda \) such that

\[
\lim_{l \to \infty} \frac{\|I_{1}^{N(0,1)} - \Gamma^{*}\|_{2}}{\|I_{1}^{l} - \Gamma^{*}\|_{2}^{2}} = \lambda.
\]
In case of Lloyd I method without acceleration, the convergence is, as expected, linear. For Lloyd I method with acceleration, the rate is not well defined, but Figure C 1 shows the impressive improvement in the convergence towards the known optimal quantizer. Figure C 2 supports the same conclusion in terms of the number of iterations necessary to reach the stopping criterion.

Then, we investigate numerically the sensitivity of Lloyd I method with Anderson acceleration and Newton–Raphson algorithm to the initial guess as a function of the distortion $c$ applied to the optimal quantizer $\Gamma_{\mathcal{N}(0,1)}$. The results of our investigation are
Figure C.3. Quantization of a standard Normal random variable: Comparison between Lloyd I method with Anderson acceleration and Newton–Raphson algorithm.

summarized in Figure C.3. The four panels correspond to different levels of distortion \( c = \{1.1, 1.2, 1.25, 1.35\} \). As before, we set \( N = 10 \) whereas on the \( y \) axis we report the residual at iteration \( l \), \( \|\Gamma_{l+1} - \Gamma_l\|_2 \). For low levels of distortion, Newton–Raphson method converges to the optimal solution more quickly than Lloyd I method. This result confirms the theoretical behaviour due to the quadratic rate of convergence of the Newton–Raphson algorithm. However, when the initial guess is quite far from the solution – as it is for the cases of 25% and 35% distortion – the algorithm may spend many cycles far away from the optimal grid.

**Remark C.1** Notice that it is also possible to design a hybrid algorithm which starts as Lloyd with acceleration and after some iterations, when it is convenient, it switches to Newton–Raphson.

Finally, we examine the convergence of Lloyd I and Newton–Raphson algorithms when considering the following Euler–Maruyama discrete scheme:

\[
\begin{align*}
\hat{X}_{t_{k+1}} &= \hat{X}_{t_k} + r\hat{X}_{t_k} \Delta t + \sigma \hat{X}_{t_k} \sqrt{\Delta t} \ Z_t, \\
X_0 &= x_0,
\end{align*}
\]
with \( r, \sigma, \) and \( x_0 \) strictly positive real constants, and \( \Delta t = t_{k+1} - t_k \) for all \( k = 1, \ldots, n - 1 \). To enlighten the greater sensitivity of the Newton–Raphson method to the grid initialization in comparison with the Lloyd I with Anderson acceleration, it is sufficient to stop at \( n = 2 \) with \( \Delta t = 0.01 \). We set the level of the quantizers \( \Gamma_1 \) and \( \Gamma_2 \) equal to \( N = 30 \) and \( x_0 = 1 \). By definition, the random variable \( \tilde{X}_{t_1} \sim \mathcal{N}(m_0(x_0), v_0(x_0)) \) where \( m_0(x_0) = x_0 + r x_0 \Delta t \) and \( v_0(x_0) = \sigma x_0 \). In order to compute the quantizer for \( \tilde{X}_{t_1} \), we initialize the algorithms at time \( t_1 \) to \( m_0(x_0) + v_0(x_0) \Gamma_{\mathcal{N}(0,1)}^{*} \), with \( \Gamma_{\mathcal{N}(0,1)}^{*} \) the optimal quantizer of a standard Normal random variable. Once we have obtained the optimal quantizer \( \Gamma_1^{*} = \{ \gamma_1^{*1}, \ldots, \gamma_1^{*30} \} \), we set the initialization of the quantizer \( \Gamma_{\text{Init}}^{1} = \{ \gamma_1^{1}, \ldots, \gamma_1^{30} \} \) at time \( t_2 \) using one of the following alternatives:

(i) The optimal quantizer at the previous step

\[
\Gamma_{\text{Init}}^{1} = \Gamma_1^{*}.
\]

(ii) The Euler operator

\[
\bar{\gamma}_2^i = m_1(\gamma_1^i) + v_1(\gamma_1^i) \Gamma_{\mathcal{N}(0,1)}^{*d},
\]

for \( i = 1, \ldots, 30 \).

(iii) The mid-point between Euler operator and the optimal quantizer at the previous step

\[
\gamma_2^i = 0.5 \bar{\gamma}_2^i + 0.5 (m_1(\gamma_1^i) + v_1(\gamma_1^i) \Gamma_{\mathcal{N}(0,1)}^{*d}),
\]

for \( i = 1, \ldots, 30 \).

(iv) The expected value

\[
\gamma_2^i = m_1(\gamma_1^i),
\]

for \( i = 1, \ldots, 30 \).

The left panel of Figure C4 shows the four different initial grids which correspond to above specifications. In the same panel, on the right side, we also plot the optimal quantizer \( \Gamma_2^{*} \) to which both Lloyd I with Anderson acceleration and Newton–Raphson methods should converge. The right panel report the quantization error – defined as \( \sqrt{D_2(\Gamma_2^{*})} \) – as a function of the iteration number \( l \). We stop the algorithm when the residual falls below \( 10^{-5} \). The numerical investigation shows that the Newton–Raphson method converges to the optimal grid faster than the Lloyd I method, with the only exception represented by the case \( \Gamma_{\text{Init}}^{1} \) equal to the mid-point. However, when initialized with the Euler operator or the mid-point Newton–Raphson algorithm fails to converge due to the bad condition number of the Hessian matrix (corresponding lines are not reported on the Figure). This result is in line with the findings in Pagès & Printems (2003) where the authors stress that the Newton–Raphson method may fail even for symmetric initial vectors since the anomalous behaviour of some components of the Voronoi tessellation.

In light of above explorations, we finally conclude that the approach based on a fixed-point algorithm such as the Lloyd I method with Anderson acceleration is much more robust than a Newton–Raphson approach – which in the present application relies on the computation of the Hessian of the matrix.

Finally, we investigate the impact of the state space finite granularity on option prices. In the numerical experiments, we employ the CEV model – see equation (4.1) – with
Figure C 4. Quantization of the Euler–Maruyama scheme associated to a Geometric Brownian motion. Left panel: four different initial grids $\Gamma_{\text{init}}^2$ and optimal grid $\Gamma^*_2$ on the left and right sides, respectively. Right panel: quantization error as a function of the iteration number.

$\bar{t} - \bar{r} = 0.32\%$, $\alpha = 0.5$, and $X_0 = \bar{X}_0 = 1.36$. Besides $\Delta t = 0.01$. The error on vanilla call option prices has been computed according to equation (4.3) as a function of $N$ for different values of $\sigma$ and different exercise conditions (ITM, ATM, OTM). The four panels of Figure C 5 report the numerical results. When the level of the volatility parameter $\sigma$ grows the error increases consistently. As expected, the mispricing is more noticeable for ITM than for OTM options. $N$ equal to 100 guarantees – at least empirically – the reduction of the absolute error below 5 bps. Accordingly, in numerical applications in the main text, we fix $N = 100$.

Appendix D Construction of the Markov generator $L_{\Gamma}$ and sensitivity analysis

We define $\Delta \gamma_i \equiv \gamma_{i+1} - \gamma_i$, $1 \leq i \leq N - 1$. The finite difference approximation of the first and second partial derivative in equation (3.9) is defined as

$$\frac{\partial u}{\partial \gamma}(\gamma, t) \approx \frac{u(\gamma_{i+1}, t) - u(\gamma_i - \Delta \gamma, t)}{2\Delta \gamma},$$

$$\frac{\partial^2 u}{\partial \gamma^2}(\gamma, t) \approx \frac{u(\gamma_{i+1}, t) - 2u(\gamma_i, t) + u(\gamma_i - \Delta \gamma, t)}{(\Delta \gamma)^2},$$

for all $t \in [0, T]$. The Markov generator $L_{\Gamma}$ is the $N \times N$ matrix defined as

$$L_{\Gamma} \doteq \begin{pmatrix}
  d_1 & u_1 & 0 & 0 & \cdots & 0 & 0 \\
  l_2 & d_2 & u_2 & 0 & \cdots & 0 & 0 \\
  0 & \ddots & \ddots & \ddots & \cdots & 0 & 0 \\
  0 & 0 & l_i & d_i & u_i & 0 & 0 \\
  0 & 0 & 0 & \ddots & \ddots & \ddots & 0 \\
  0 & 0 & 0 & 0 & l_{N-1} & d_{N-1} & u_{N-1} \\
  0 & 0 & 0 & 0 & 0 & l_N & d_N
\end{pmatrix},$$
A backward Monte Carlo approach to exotic option pricing

Sensitivity Analysis, $\Delta t = 0.01$, CEV model

- ITM $\sigma = 5\%$
- ATM $\sigma = 5\%$
- OTM $\sigma = 5\%$

Sensitivity Analysis, $\Delta t = 0.01$, CEV model

- ITM $\sigma = 10\%$
- ATM $\sigma = 10\%$
- OTM $\sigma = 10\%$

Sensitivity Analysis, $\Delta t = 0.01$, CEV model

- ITM $\sigma = 15\%$
- ATM $\sigma = 15\%$
- OTM $\sigma = 15\%$

Sensitivity Analysis, $\Delta t = 0.01$, CEV model

- ITM $\sigma = 20\%$
- ATM $\sigma = 20\%$
- OTM $\sigma = 20\%$

Figure C 5. Study of the influence of the granularity $N$ of the state space on the pricing of call vanilla options when employing the RMQA, CEV model. Top left: $\sigma = 5\%$, Top right: $\sigma = 10\%$, Bottom left: $\sigma = 15\%$, Bottom right: $\sigma = 20\%$.

where the coefficients $l_i$, $d_i$, and $u_i$ are given by

$$l_i = -\frac{b(\gamma_i)}{2\Delta\gamma} + \frac{1}{2} \frac{\sigma(\gamma_i)^2}{(\Delta\gamma)^2},$$

$$d_i = -\frac{\sigma(\gamma_i)^2}{(\Delta\gamma)^2},$$

$$u_i = +\frac{b(\gamma_i)}{2\Delta\gamma} + \frac{1}{2} \frac{\sigma(\gamma_i)^2}{(\Delta\gamma)^2},$$

for all $1 \leq i \leq N$. The coefficients of the first and last row are chosen so that the Markov chain is reflected at the boundaries of the state domain. The choice of the boundary conditions should have a negligible effect provided that the range of the state domain is sufficiently large. To quantify what sufficiently large means in practical situation, we consider again the CEV model corresponding to equation (4.1) with $r_d - r_f = 0.32\%$, $\alpha = 0.5$, and $X_0 = \bar{X}_0 = 1.36$. Besides $\Delta t = 0.01$. The contribution of the drift to the dynamics is negligible, and to choose the grid boundary values we proceed as follows. For a given value of $\sigma$, we first take symmetric intervals around the starting point $X_0$. 
Figure D 1. Sensitivity analysis of the influence of the boundary conditions on the pricing of Call vanilla options for the LTSA and CEV dynamics. From left to right, from top to bottom: Contour plot, \texttt{contour}(x, y, Z) where \(x\) is the vector containing the variable \(N\), \(y\) is the vector containing the coefficient \(\beta\), and \(Z\) is the matrix of errors in basis points. The white regions correspond to errors larger than 19 basis points. First row: \(\sigma = 5\%\), Second row: \(\sigma = 10\%\), Third row: \(\sigma = 15\%\), Fourth row: \(\sigma = 20\%\).
i.e., $X_0 \pm \beta X_0$, for different values of the coefficient $\beta$. Then, for a fixed $\beta$, we vary the granularity $N$ of the state space and we compute the error on vanilla prices according to equation (4.3). We display a contour plot $\text{CONTOUR}(x, y, Z)$, where $x$ is the vector of $N$ values, $y$ contains the values of the coefficient $\beta$, and $Z$ is the matrix of errors in basis points. The panels of Figure D report the sensitivity analysis for different values of $\sigma$, and different exercise conditions (ITM, ATM, OTM). Accordingly, in numerical applications, we fix the cardinality of the quantizer to 100 and $\beta$ to 0.15, 0.25, 0.35, and 0.45 for $\sigma = 5\%$, $\sigma = 10\%$, $\sigma = 15\%$, and $\sigma = 20\%$, respectively.