Pricing and Calibration in Local Volatility Models via Fast Quantization

Giorgia Callegaro *  Lucio Fiorin†  Martino Grasselli‡

September 12, 2014

Abstract

In this paper we propose the first calibration exercise based on quantization methods. Pricing and calibration are typically difficult tasks to accomplish: pricing should be fast and accurate, otherwise calibration cannot be performed efficiently. We apply in a local volatility context the recursive marginal quantization methodology to the pricing of vanilla and barrier options. A successful calibration of the Quadratic Normal Volatility model is performed in order to show the potentiality of the method in a concrete example, while a numerical exercise on barrier options shows that quantization overcomes Monte-Carlo methods.

Keywords: Quantization, local volatility, calibration, Normal quadratic volatility model.

1 Introduction

Quantization is a tool widely used in information theory, cluster analysis, pattern recognition, speech recognition, numerical integration, mathematical models in economics and, as in our case, approximation of stochastic processes. The birth of optimal quantization dates back to the 50’s, when the necessity to optimize signal transmission, by appropriate discretization procedures, arose. Quantization consists in approximating a signal admitting a continuum of possible values, by a signal that takes values in a discrete set. Vector quantization deals with signals that are finite dimensional, such as random variables, while functional quantization extends the concepts to the infinite-dimensional setting, as it is the case of stochastic processes. Quantization of random vectors can be considered as a discretization of the probability space by at most $N$ values, providing in some sense the best approximation to the original distribution. It is therefore crucial to optimize the geometric location of these points for a given distribution and to evaluate the resulting error. Some numerical procedures have been developed in order to get optimal quadratic quantization of the Gaussian (and even non Gaussian) distribution in high dimension, mostly based on stochastic optimization algorithms. Over the years many other application fields have been discovered, such as, in the 90’s, numerical integration. This opened the door, especially in France and in Germany, to new research perspectives in Numerical Probability and applications to Mathematical Finance. For a comprehensive introduction to optimal vector quantization and its applications, we refer to the recent paper of Pagès (2014) and references therein.

While theoretically sound and deeply investigated, optimal quantization suffers from the numerical burden that the algorithms involve, to the point that it seems hopeless to use classic optimal quantization in a calibration perspective (see for example the numerical results in Pagès and Printems (2005)). The main reason is related to the highly time-consuming procedure required by the determination of the optimal grid. However, very recently a promising type of quantization, called recursive marginal quantization, has been introduced by Pagès and Sagna (2014) and applied to the Euler scheme of a pseudo-CEV local volatility model in a pricing context. Following the lines of Pagès and Sagna (2014), in our paper we apply the recursive marginal quantization to a special local volatility model, namely the Quadratic Normal Volatility model, that has been investigated by Blachert (2001), Ingersoll (1997), Lipton (2002), Zühlsdorff (2002) and lately revisited by Andersen (2011) and Carr et al. (2013). We find stationary quantizers via a Newton-Raphson method, in order to price vanilla and exotic derivatives efficiently. In fact, the Newton-Raphson procedure, being deterministic, is very fast and it

*Department of Mathematics, University of Padova, via Trieste 63, 35121 Padova, Italy. Email: gcallega@math.unipd.it
†Department of Mathematics, University of Padova, via Trieste 63, 35121 Padova, Italy. Email: luciofiorin@gmail.com
‡Department of Mathematics, University of Padova (Italy) and Devinci Finance Lab, Pole Universitaire Léonard de Vinci, Paris La Défense (France) and Quanta Finanza S.r.l., via Cappuccina 40 Mestre-Venezia (Italy). Email: grassell@math.unipd.it.

Acknowledgments: We are especially indebted to Abass Sagna for fruitful discussions.
allows us to provide the first example of calibration based on quantization. The recursive marginal quantization is competitive even when closed form formulas for vanillas are available (as in the case of call and put prices for the Quadratic Normal volatility model). Finally, we show the flexibility and the efficiency of the recursive marginal quantization in the pricing of non-vanilla contracts, when compared to the classic Monte Carlo simulation. Our numerical algorithms have performed quite well (wrt to Monte Carlo), so that in this paper no speed-up procedure has been tested. As a consequence, this paper does not provide the fastest possible numerical method, but a procedure that is competitive enough if compared to Monte Carlo.

The paper is organized as follows: in Section 2 we give a quick application-oriented overview of the vector quantization methodology. Section 3 extends the vector quantization method to the class of Markov diffusion processes, leading to the recursive marginal quantization. Section 4 introduces the Quadratic Normal volatility model together with the well known results about closed form formulas for vanilla option prices. Moreover, we apply the recursive marginal quantization approach to the pricing of barrier options. Section 5 illustrates our numerical results, with particular emphasis to the calibration exercise on real data. Section 6 concludes and we recall some technical result in the Appendix.

2 Brief overview on vector quantization

We first provide some more technical details on vector quantization (see eg Graf and Luschgy (2000), Pagès and Printems (2005), Pagès et al. (2003), Pagès (2014)). Consider an \( \mathbb{R}^d \)-valued random variable \( X \) defined on a probability space \( (\Omega, \mathcal{A}, \mathbb{P}) \) with finite \( r \)-th moment and probability distribution \( \mathbb{P}_X \). As already mentioned, quantization can be considered as a discretization of the probability space by at most \( N \) values, providing in some sense the best approximation to the original distribution. Mathematically, quantizing \( X \) on a given grid \( \Gamma = \{x_1, \cdots, x_N \} \) consists then in projecting \( X \) on the grid \( \Gamma \) following the closest neighbor rule. An \( N \)-quantizer is a Borel function \( f_N : \mathbb{R}^d \rightarrow \Gamma \subset \mathbb{R}^d \) projecting \( X \) on \( \Gamma \). The induced mean \( L^r \)-error (for \( r > 0 \)) is called \( L^r \)-mean quantization error and is given by

\[
\|X - f_N(X)\|_r = \| \min_{1 \leq i \leq N} |X - x_i| \|_r
\]

where \( |X|_r := [\mathbb{E}(|X|^r)]^{1/r} \) is the usual norm in \( L^r \). The projection of \( X \) on \( \Gamma \), \( f_N(X) \), is called the quantization of \( X \) (in the sequel, we will alternatively use \( f_N(X) \) or \( \text{Proj}_{\Gamma}(X) \) to indicate the quantization of \( X \)). As a function of the grid \( \Gamma \), the \( L^r \)-mean quantization error is continuous and reaches a minimum over all the grids with size at most \( N \). A grid \( \Gamma^* \) minimizing the \( L^r \)-mean quantization error over all the grids with size at most \( N \) is called an \( L^r \)-optimal quantizer.

An optimal quantizer is then associated to an optimal grid of points \( \Gamma^* \) and to an optimal Borel partition of the space \( \mathbb{R}^d \), \( (C_i(\Gamma^*))_{1 \leq i \leq N} \), and vice versa, so that the quantizer is defined in this way:

\[
f_N(X) = \sum_{i=1}^{N} x_i \mathbb{1}_{C_i(\Gamma^*)}(X)
\]

where the above partition \( \{C_i(\Gamma^*)\}_{i=1, \cdots, N} \), with \( C_i(\Gamma^*) \subset \{ \xi \in \mathbb{R}^d : ||\xi - x_i|| = \min_{1 \leq j \leq N} ||\xi - x_j|| \} \), is called the Voronoi partition, or tessellation induced by \( \Gamma^* \). Moreover, the \( L^r \)-mean quantization error vanishes as the grid size \( N \rightarrow +\infty \) and the convergence rate has been computed in the celebrated Zador theorem (see Graf and Luschgy (2000)):

\[
\min_{\Gamma, |\Gamma| = N} \|X - \text{Proj}_{\Gamma}(X)\|_r = Q_r(\mathbb{P}_X)N^{-1/d} + o(N^{-1/d})
\]

where \( Q_r(\mathbb{P}_X) \) is a nonnegative constant (\( r = 2 \) of course will be of particular interest, with the corresponding quadratic optimal quantizer). From a numerical point of view, finding an optimal quantizer may be a very challenging task. This motivates the introduction of sub-optimal criteria, mostly because one is typically interested in quantizations which are close to \( X \) in distribution. We then introduce the notion of stationary quantizer.

**Definition 2.1.** An \( N \)-quantizer \( \Gamma^N = \{x_1, \cdots, x_N \} \) inducing the quantization \( f_N \) of \( X \) is said to be stationary if

\[
\mathbb{E}[X|f_N(X)] = f_N(X)
\]

In particular, if we introduce the distortion function associated with \( \Gamma^N \)

\[
D(\Gamma^N) := \sum_{i=1}^{N} \int_{C_i(\Gamma^N)} |z - x_i|^2 d\mathbb{P}_X(z)
\]

(1)
3 RECURSIVE MARGINAL QUANTIZATION

then it turns out that stationary quantizers are critical points of the distortion function (that is, a stationary quantizer $\Gamma^N$ satisfies $\nabla D(\Gamma^N) = 0$). How to numerically compute the quadratic optimal quantizers, or $L^r$-optimal (or stationary) quantizers in general, together with the associated weights and $L^r$-mean quantization errors is an important issue. Several algorithms are used in practice. In the one dimensional framework, the $L^r$-optimal quantizers are unique up to the grid size as soon as the density of $X$ is strictly log-concave. From a numerical point of view, stationary quantizers are interesting insofar their search is based on zero search recursive procedures like Newton’s algorithm that can be efficiently performed.

This is the essential on vector quantization we need in order to proceed. For a thorough treatment of this topic we refer to Graf and Luschgy [2000].

3 Recursive marginal quantization

In this section we consider the quantization of a continuous-time diffusive Markovian process $Y$, whose evolution is specified by the following SDE:

$$dY_t = b(t, Y_t)dt + a(t, Y_t)dW_t, \quad Y_0 = y_0 > 0$$

(2)

where $W$ is a standard Brownian motion and the functions $a(t, y), b(t, y)$ satisfy the usual conditions ensuring the existence of a strong solution to the SDE. In principle one should deal with the functional quantization approach, but the idea now is to discretize the process. Consider its Euler scheme and, following the approach presented in Pages and Sagna (2014), we discretize the process by exploiting the Markov property of $Y$ via vector quantization.

Having fixed a time horizon $T > 0$ and a time discretization grid $\{t_0, t_1, \ldots, t_M\}$, with $t_0 = 0$ and $t_M = T$ with constant step size $\Delta = t_k - t_{k-1}$, for every $k \geq 1$, the Euler scheme for the process $Y$ is given by

$$\hat{Y}_{t_k} = \hat{Y}_{t_{k-1}} + b(t_{k-1}, \hat{Y}_{t_{k-1}})\Delta + a(t_{k-1}, \hat{Y}_{t_{k-1}})\Delta W$$

$$\hat{Y}_{t_0} = \hat{Y}_0 = y_0$$

where $t_k = k\Delta$ and $\Delta W := W_{t_k} - W_{t_{k-1}}$ is a centered Normal random variable with variance $\Delta$, so that we have the following equality in distribution

$$\left(\hat{Y}_{t_k}|\hat{Y}_{t_{k-1}} = x\right) \sim N\left(m_{k-1}(x), \sigma_{k-1}^2(x)\right)$$

(3)

where

$$m_{k-1}(x) = x + b(t_{k-1}, x)\Delta$$

$$\sigma_{k-1}^2(x) = (a(t_{k-1}, x))^2\Delta$$

Our intention now is to use the vector quantization applied to every (one dimensional) random variable $\hat{Y}_{t_k}, k \geq 1$, since we know its marginal distribution conditional to $\hat{Y}_{t_{k-1}}$. This explains the term marginal of this quantization method. It can be seen in Pages et al. (2003) that the error made by quantizing the Euler scheme can be easily controlled, under some mild regularity assumptions on the process. The distortion function relative to $\hat{Y}_{t_{k+1}}$, denoted $D_{k+1}$ (recall (1)) reads

$$D_{k+1}(X) = \sum_{i=1}^N \int \mathbb{P}(\hat{Y}_{t_{k+1}} \in dy_{k+1}) (y_{k+1} - x)^2\mathbb{P}(\hat{Y}_{t_{k+1}} \in dy_{k+1})$$

(4)

where $N$ is the (fixed) size of the quantizer for every time step. The delicate point here is that, in order to quantize $\hat{Y}_{t_{k+1}}$ we have to apply the Newton-Raphson method without knowing its distribution. However, by using the conditional distribution in (3) we can rewrite the distortion function (4) in terms of $\hat{Y}_{t_k}$, obtaining then a recursive formula to compute the stationary quantizer. In fact, the distribution function of $\hat{Y}_{t_{k+1}}$ can be written as follows

$$\mathbb{P}(\hat{Y}_{t_{k+1}} \in dy_{k+1}) = \int \phi_{m_k(y_k), \sigma_k(y_k)}(y_{k+1})\mathbb{P}(\hat{Y}_{t_k} \in dy_k)dy_{k+1}$$

We also mention the optimal quantization website: http://www.quantize.maths-fi.com where one can download the optimized quadratic quantization grids of the $d$-dimensional Gaussian distributions $N(0, I_d)$, for $N = 1$ up to $10^4$ and for $d = 1, \ldots, 10$. Moreover, at the same link one can also find functional quantization grids of the standard Brownian motion over the interval $[0,1]$, of the Brownian bridge, as well as a detailed procedure to compute grids for the (normalized) Ornstein-Uhlenbeck process and its bridge.
where \( \phi_{\mu,\sigma} \) denotes the density function associated with a Normal random variable. With this result it is possible to compute the Hessian matrix of the distortion function. Note that we are interested in the quantization of the Euler scheme \( \hat{Y} \) that we denote by \( \hat{Y}_{t_k}, k \geq 0 \), so that we substitute \( \hat{Y}_{t_k} \) with \( \hat{Y}_{t_k} \) in (5). Due to the discrete nature of the quantizer, the integral in (4) becomes a finite sum, thus leading to extremely fast computations. In the sequel we will apply the recursive marginal quantization to a special local volatility model, namely the Quadratic Normal volatility model. We refer the interested reader to Pagès and Sagna (2014) for a complete background including the analysis of the errors generated by the recursive quantization method.

4 The Quadratic Normal volatility model

The class of Quadratic Normal volatility (QNV) models has drawn much attention in the financial industry due to its analytic tractability and flexibility. We will refer to the works of Blacher (2001), Ingersoll (1997), Lipton (2002), Andersen (2011).

A QNV model is associated to an asset \( Y \) evolving as follows

\[
dY_t = (e_1 Y_t^2 + e_2 Y_t + e_3) dW_t, \quad Y_0 = y_0 > 0
\]

for some \( e_1, e_2, e_3 \in \mathbb{R} \), where the Brownian motion \( W \) is taken under the risk neutral measure. This corresponds to the SDE (2) where \( b(y) = 0 \) (that is we consider the forward-price process) and \( a(t, y) = e_1 y^2 + e_2 y + e_3 \). Note that (4) includes, as special cases, the Brownian motion (for \( e_1 = e_2 = 0 \)), the geometric Brownian motion (for \( e_1 = e_3 = 0 \)) and the inverse of a three-dimensional Bessel process (for \( e_2 = e_3 = 0 \)) which leads to a strict local martingale (we refer to Andersen (2011) and Carr et al. (2013) for other technical properties of the model). Apart from technicalities, the intuition underlying or (5) is that mimicking a quadratic spot volatility gives some chances to get an implied volatility curve that is able to reproduce the smile and skew effects using a parsimonious number of parameters. This is more evident in the following parametrization (see eq (5) Andersen (2011)) that is more useful for practical implementations

\[
dY(t) = \sigma (qY(t) + 1 - q) y_0 + \frac{1}{2} s (Y(t) - y_0)^2 dW(t), \quad Y_0 = y_0 > 0
\]

Here \( \sigma > 0 \) is a proxy for the ATM volatility level, \( q \) is related to the implied volatility slope (that is \( q \) is the skew parameter) and \( s \) is a measure of the convexity of the quadratic volatility function (the vol-of-vol parameter).

4.1 Vanilla Options Pricing

The QNV model allows for closed form solutions for the prices of vanilla options (see Appendix A.2 taken from Andersen (2011)). The corresponding formulas depend on the roots of the polynomial in (5). Note that even if closed form formulas are available for vanillas, their implementation is time consuming and it requires some care, especially in the truncation of the trigonometric series. Moreover, a calibration procedure based on these formulas should allow for the possibility to switch from the first (real roots) to the second case (complex roots) without constraints. We will see in the calibration exercise that this is a real issue. On the contrary, in the recursive marginal quantization approach one never cares about this problem. Following the steps illustrated in the previous Section one easily computes the critical points of the distortion function together with its Hessian. In Appendix A.2 we present the formulas for the gradient, the Hessian matrix and the weights of the quantized random variable \( \hat{Y} \).

4.2 Pricing of Barrier Options

We focus now on barrier options and we follow the approach in Sagna (2010), where the author presents an algorithm based on optimal quantization, in order to approximate the price of any kind of barrier option. We consider up-and-out put options. Pricing formulas in the other cases are just a slight modifications of the ones that we are going to present here.

Given the Euler scheme \( \hat{Y} \) for the process \( Y \), the price of an up-and-out put option expiring at time \( T \), with strike \( K \) and up-and-out barrier \( L \) can be approximated by

\[
P_{\text{LO}} := e^{-rT} \mathbb{E} \left[ (K - \hat{Y}_T)^+ \mathbb{I}_{\{\sup_{t \in [0,T]} \hat{Y}_t \leq L\}} \right] = e^{-rT} \mathbb{E} \left[ (K - \hat{Y}_T)^+ \prod_{k=1}^M G_{\hat{Y}_{t_{k-1}}, \hat{Y}_k}(L) \right]
\]

where

\[
G_{x,y}(u) = \left( 1 - e^{-2M \max(x,y)} \right) \mathbb{I}_{\{u \geq \max(x,y)\}}
\]
5 NUMERICAL RESULTS

and where \( \sigma(\cdot) \) is the volatility function of \( Y \). The last equality in the above equation can be obtained via an application of the so-called “regular Brownian bridge method”, that is connected to the knowledge of the distribution of the minimum (or the maximum) of the continuous Euler scheme \( \hat{Y} \) relative to a process \( Y \) over a time interval \([0, T]\), given its values at the discrete time observation points \( 0 = t_0 < t_1 < \cdots < t_M = T \) (see, e.g., Glasserman (2003)).

Introducing the functions \( f(x) := (K - x)^+ \) and \( g_k(\tilde{Y}_{t_{k-1}}, \tilde{Y}_{t_k}) := G_{\tilde{Y}_{t_{k-1}}, \tilde{Y}_{t_k}}(L) \), \( k = 1, \ldots, M \), the expectation in Equation (7) reads

\[
\mathbb{E}\left( f(\tilde{Y}_{t_M}) \prod_{k=1}^{M} g_k(\tilde{Y}_{t_{k-1}}, \tilde{Y}_{t_k}) \right)
\]

Now, by applying recursively the tower property of conditional expectation we get

\[
\pi_M f := \mathbb{E}\left( f(\tilde{Y}_{t_M}) \prod_{k=1}^{M} g_k(\tilde{Y}_{t_{k-1}}, \tilde{Y}_{t_k}) \right) = \mathbb{E}\left( \mathbb{E}\left( f(\tilde{Y}_{t_M}) \prod_{k=1}^{M} g_k(\tilde{Y}_{t_{k-1}}, \tilde{Y}_{t_k}) | \mathcal{F}_{t_{M-1}} \right) \right)
\]

\[
= \mathbb{E}\left( \mathbb{E}\left( f(\tilde{Y}_{t_M}) g_M(\tilde{Y}_{t_{M-1}}, \tilde{Y}_{t_M}) | \mathcal{F}_{t_{M-1}} \right) \prod_{k=1}^{M-1} g_k(\tilde{Y}_{t_{k-1}}, \tilde{Y}_{t_k}) \right)
\]

\[
= \mathbb{E}\left( H_M(f(\tilde{Y}_{t_{M-1}})) \prod_{k=1}^{M-1} g_k(\tilde{Y}_{t_{k-1}}, \tilde{Y}_{t_k}) \right)
\]

(8)

where \( H_k, k = 1, \ldots, M \), is a family of bounded transition kernels defined on bounded measurable functions \( f \) by:

\[
H_k f(y) := \mathbb{E}\left[ f(\tilde{Y}_{t_k}) g_k(\tilde{Y}_{t_{k-1}}, \tilde{Y}_{t_k}) | \tilde{Y}_{t_{k-1}} = y \right]
\]

(9)

Furthermore, for every \( x \in E \), we have \( H_0 f(y) := \pi_0 f = \mathbb{E}\left[ f(\tilde{Y}_0) \right] \). It follows from (8) that

\[
\pi_k f = \pi_{k-1} H_k f, \quad k = 1, \ldots, M
\]

so that we finally obtain the recursive expression

\[
\pi_M = H_0 \circ H_1 \circ \cdots \circ H_M
\]

In order to apply the results in Sagna (2010), that is to compute recursively conditional expectations as in Equation (8), we need an approximation of the transition probability of \( \tilde{Y}_{t_{k+1}} \) given \( \tilde{Y}_{t_k} \). The idea now is to use the transition matrix of \( \tilde{Y}_{t_{k+1}} \) given \( \tilde{Y}_{t_k} \). The potentiality of the algorithm developed in Section 3 gives us an immediate way to compute the transition matrix. In fact, given the quantization grids \( x = (x_1, \ldots, x_N) \) of \( \tilde{Y}_{t_k} \) and \( y = (y_1, \ldots, y_N) \) of \( \tilde{Y}_{t_{k-1}} \), the computation of the approximated transition probabilities is straightforward, as well as the one of the kernels \( \hat{H}_k \)'s (the detailed formulas can be found in Appendix A.3). The approximated price follows then immediately:

\[
P^{LO} = e^{-rT} \mathbb{E}\left( (K - \hat{Y}_T)^+ \prod_{k=1}^{M} G_{\tilde{Y}_{t_{k-1}}, \tilde{Y}_{t_k}}(L) \right) \approx e^{-rT}(\hat{H}_0 \circ \hat{H}_1 \circ \cdots \circ \hat{H}_M)f
\]

5 Numerical results

In this section we provide the first example of competitive and efficient calibration of a quantization-based method to real data and we then apply our result to the pricing of vanilla and non-vanilla derivatives. Let us start by noticing that, having the stationary grid for each time step at hand, the pricing of a generic option becomes immediate. For example, the price at \( t = 0 \) of an European Vanilla Put option on \( Y \) with maturity \( T \) and strike \( K \), that we have \( N \)-quantized at \( t = T \) with an optimal grid \( y = (y_1, \ldots, y_N) \) and associated optimal quantizer \( \hat{Y}(T) \), is given by

\[
\mathbb{E}[(K - Y(T))^+] = \sum_{i=1}^{N} (K - y_i)^+ \mathbb{P}(\hat{Y}(T) = y_i)
\]

that can be immediately computed. Note also that \( C_i(y) = \left[ \frac{y_i + y_{i+1}}{2}, \frac{y_{i+1} + y_{i+2}}{2} \right] \) since we work in a one dimensional setting. While the dimension of the grid can vary at each step of the Euler scheme, we fix the dimension of every grid at 100, so that we will always use 100-dimensional quantizers.
5.1 Calibration on vanilla\textsuperscript{s}

We first test the goodness of the pricing via recursive marginal quantization. Here we use 9 different strikes, equally spaced from 80\% to 120\% of the initial value of the underlying, and 6 different maturities, from 2 months to 2 years. As an error measure for this test we consider the Res. Norm, that is the sum of the squared difference between the model implied volatilities and the ones generated by the closed form formulas of the previous Section. We use a 150-dimensional quantizer, and 8 time steps for every maturity. Computations are performed using \textit{Matlab} software on a CPU 2.4 GHz and 8 Gb memory computer. The inverse of the Hessian matrix is calculated using the LU-decomposition, since it is tridiagonal and symmetric. Each quantization of the process takes nearly 2.5 seconds.

The results in Table 1\textsuperscript{2} confirm the precision of prices generated by the quantization. Note that in the case of complex roots the two methods are almost equivalent, in fact by comparing the execution times it follows that pricing via quantization requires about 10 seconds, against 8 seconds to compute prices via the closed form formulas.

Let us now turn on real market data. Calibration is done via a standard non-linear least-squares optimizer that minimizes the total calibration error in terms of the difference between model and market implied volatilities \( \sum_n (\sigma_{n,\text{market}}^{\text{imp}} - \sigma_{n,\text{model}}^{\text{imp}})^2 \). Using a major provider, we take prices of European Vanilla Call-Put option on the Dax Index, as of 19 June 2014. Using the closed form formulas, it turns out that the implied volatility smile produced by the market is fitted better when the two roots are complex. As a consequence, quantization will lose only 2 seconds in average for each iteration of the calibration routine. What is more, closed form formulas do not perform well for short maturities, to the point that we are not able to present results of the calibration based on closed form formulas in this case, while we note that the flexibility of the quantization approach permits to overcome these difficulties. We therefore show the joint results of calibration via closed form formulas and via quantization only with long maturities (from 1.5 years up to 3 years), while with short maturities (from 2 months up to 5 months) we only display the calibration results for the recursive marginal quantization alone. The calibrated parameters are displayed in Table 2.

With the long maturities (resp. short maturities) the Res. Norm is calculated in Table 3 containing 4 maturities and 7 strikes (resp. 4 maturities and 5 strikes). By looking at Res. Norm, we see that overall the quality of the fit is not very good, but this is due to the particular model which is very parsimonious (only 3 parameters!). Nevertheless, we emphasize that despite the simplicity and the limits of the model, this represents the first successful calibration example based on quantization. Moreover, it is important to notice that the procedure here illustrated can be easily applied to any local volatility model for which the Euler discretization scheme is available.

5.2 Barrier Options

In order to test the goodness of this pure quantization method, we use the same data as in the previous subsection, focusing on short maturities. We fix the maturity \( T = \frac{1}{2} \) and the strike \( K = 100 \) and we consider a family of barriers \( L = y_0 + i \times 5 \), \( i = 1, \ldots, 5 \). All these barriers can be considered close to the initial price and in this framework we have a big price variation for small changes in the value of the barrier. We compare the price of up-and-out put options obtained via the proposed quantization method with the one obtained via a Monte Carlo simulation applied to the Euler scheme of the process.

The benchmark price is computed via a Monte Carlo algorithm with \( 10^3 \) discretization points in the time grid and \( 10^6 \) simulations. On the quantization side, we use 100-dimensional quantizers, while the Monte Carlo method is performed with \( 5 \times 10^5 \) simulations. In both cases, we set the time step equal to \( \frac{1}{360} \).

Prices can be found in Table 4 together with the computational time. With \( \Delta t \to 0 \), the Newton method used for the computation of the quantization grids is unstable, due to the bad conditioning of the Hessian matrix. In this case, as suggested in [Deuflhard (1974)], we use the Moore-Penrose pseudoinverse of the matrix, instead of the inverse.

The results show that the quantization method is a valid alternative to Monte Carlo when pricing these exotic options, with a computational time significantly smaller\textsuperscript{4}.

Finally, table 5 shows that, as the initial value of the underlying is no more close to the barrier, the quantization price becomes more and more stable and gives a good approximation of the real price, still being faster than Monte Carlo.

\textsuperscript{2}Of course both methods suffer from the numerical error introduced by the fact that the barrier is close to the initial value. However notice that with a reasonable time step of 1 observation per day, the quantization method is a very good alternative to Monte Carlo.
6 Conclusion

We have applied the recursive marginal quantization method to the local volatility model QNV as in Andersen (2011) in order to provide an alternative way to compute prices, without the numerical problems due to the real/complex nature of the roots. The procedure gives a fast way to price vanilla as well as barrier options, compared to Monte Carlo simulation. A successful calibration of the QNV model on real data shows the flexibility of the quantization method. Extensions of this work could also include less parsimonious local volatility models, since the speed of the algorithm does not depend on the number of parameters.

A Appendix

A.1 Closed form solution for the Call price

Formulas are taken from Andersen (2011). In the case of two distinct real roots \( u, l \in \mathbb{R}, u > l \), the price at time 0 of a call option is found to be given by

\[
p(0) = K_1 \Phi_{0.1}( - d_1^{(1)} ) - X_2 \Phi_{0.1}(d_2^{(2)}) - X_1 \Phi_{0.1}(d_1^{(1)}) + K_2 \Phi_{0.1}(d_2^{(2)})
\]

where \( K_i, X_i, d_\pm^{(i)} \) are given by

\[
K_1 = \frac{(K - u)(y_0 - l)}{u - l} \quad X_1 = \frac{(y_0 - u)(K - L)}{u - l} \\
K_2 = \frac{(K - l)(y_0 - l)}{u - l} \quad X_2 = \frac{(y_0 - u)(K - u)}{u - l}
\]

and

\[
d_\pm^{(i)} = \frac{\ln \left( \frac{X_i}{K_i} \right) \pm T}{\sqrt{T}}, \quad i = 1, 2
\]

and where \( \Phi_{0.1} \) denotes the standard Normal cumulative distribution. In the case of two complex roots, denoted by \( a + ib \), we have

\[
p(0) = \sqrt{bA(y_0)} e^{\frac{T}{2}} \left( 2 \frac{Z_U - Z_L}{Z_U - Z_L} \sum_{a=1}^{+\infty} e^{-\alpha_n T} \sin(-\alpha_n) \left( \tilde{K} I_n^{(c)} - I_n^{(s)} \right) \right) + \\
+ \sqrt{\frac{A(y_0)}{A(L)} (K - L)} \left( \frac{\sin(Z_U - Z_0)}{\sin(Z_U - Z_L)} - \frac{1}{(Z_U - Z_L)^2} \sum_{a=1}^{+\infty} n \pi \sin(-\alpha_n) e^{-(\alpha_n - \frac{\pi}{2}) T} \right)
\]

Moreover,

\[
Z_0 = \arctan \left( \frac{u - a}{b} \right) \quad Z_U = \arctan \left( \frac{u - a}{b} \right) \quad Z_L = \arctan \left( \frac{u - a}{b} \right) \n\]

\[
\tilde{K} = K - \frac{a}{b} \quad A(x) = b \left( 1 + \left( \frac{x - a}{b} \right)^2 \right) \\
\alpha_n = \frac{a^2 + x^2}{2(Z_U - Z_L)} \quad \alpha_n = \frac{a^2 + x^2}{2(Z_U - Z_L)}
\]

where \( U \) and \( L \) are the upper and lower bounds introduced to guarantee that the process is a martingale (a natural configuration is \( U = +\infty, L = 0 \)) and

\[
I_n^{(c)} = Z_U - Z_0 - \frac{\cos(\gamma_n)}{Z_U - Z_0 + a_n} - \frac{\cos(\gamma_n)}{Z_U - Z_0 - a_n} + \frac{2 \cos(\gamma_n) a_n}{a_n^2 - (Z_U - Z_0)^2} \\
I_n^{(s)} = Z_U - Z_0 - \frac{\sin(\gamma_n)}{Z_U - Z_0 + a_n} - \frac{\sin(\gamma_n)}{Z_U - Z_0 - a_n} + \frac{2 \sin(\gamma_n) a_n}{a_n^2 - (Z_U - Z_0)^2} \\
c_n^{(x)} = \left( 1 \pm \frac{\alpha_n}{Z_U - Z_L} \right) \left( \arctan(\tilde{K} - Z_0) \right) a_n + Z_0
\]
A.2 Formulas for the marginal quantization

Given the optimal quantization grid \( y = (y_1, \ldots, y_N) \) for \( \tilde{Y}_{tk} \)

\[
\frac{\partial D_{k+1}}{\partial x_j}(x) = 2 \sum_{i=1}^{N} \left\{ (x_j - m_k(y_i)) \left( \Phi_{0,1}(\tilde{x}_{k+1,i}^+(y_i)) - \Phi_{0,1}(\tilde{x}_{k+1,i}^-(y_i)) \right) + \sigma_k(y_i) \left( \phi_{0,1}(\tilde{x}_{k+1,i}^+(y_i)) - \phi_{0,1}(\tilde{x}_{k+1,i}^-(y_i)) \right) \right\} \mathbb{P}(\tilde{Y}_{tk} = y_i)
\]

where

\[
\tilde{x}_{k+1,i}^+(y_i) = \frac{x_j + x_{j+1} - 2m_k(y_i)}{2\sigma_k(y_i)}
\]

\[
\tilde{x}_{k+1,i}^-(y_i) = \frac{x_{j-1} + x_j - 2m_k(y_i)}{2\sigma_k(y_i)}
\]

For the Hessian matrix we get

\[
\frac{\partial^2 D_{k+1}}{\partial x_j \partial x_j}(x) = 2 \sum_{i=1}^{N} \left\{ \left( \Phi_{0,1}(\tilde{x}_{k+1,i}^+(y_i)) - \Phi_{0,1}(\tilde{x}_{k+1,i}^-(y_i)) \right) + \frac{1}{4\sigma_k(y_i)} \phi_{0,1}(\tilde{x}_{k+1,i}^+(y_i))(x_{j+1} - x_j) + \frac{1}{4\sigma_k(y_i)} \phi_{0,1}(\tilde{x}_{k+1,i}^-(y_i))(x_j - x_{j-1}) \right\} \mathbb{P}(\tilde{Y}_{tk} = y_i)
\]

\[
\frac{\partial^2 D_{k+1}}{\partial x_j \partial y_{j-1}}(x) = -\frac{1}{2} \sum_{i=1}^{N} \left\{ \phi_{0,1}(\tilde{x}_{k+1,i}^+(y_i)) \left( \frac{x_j - x_{j-1}}{\sigma_k(y_i)} \right) \right\} \mathbb{P}(\tilde{Y}_{tk} = y_i)
\]

\[
\frac{\partial^2 D_{k+1}}{\partial x_j \partial y_{j+1}}(x) = -\frac{1}{2} \sum_{i=1}^{N} \left\{ \phi_{0,1}(\tilde{x}_{k+1,i}^-(y_i)) \left( \frac{x_{j+1} - x_j}{\sigma_k(y_i)} \right) \right\} \mathbb{P}(\tilde{Y}_{tk} = y_i)
\]

Finally, the weights of the quantized process are given by the following approximation

\[
\mathbb{P}(\tilde{Y}_{tk+1} \in C_i(x)) = \sum_{j=1}^{N} \left[ \Phi_{0,1}\left( \tilde{x}_{k+1,i}^+(y_j) \right) - \Phi_{0,1}\left( \tilde{x}_{k+1,i}^-(y_j) \right) \right] \mathbb{P}(\tilde{Y}_{tk} = y_j)
\]

For practical implementation, \( x_0 = -\infty, x_{N+1} = +\infty \).

A.3 Formulas for the barrier pricing

Given the quantization grids \( x = (x_1, \ldots, x_N) \) of \( \tilde{Y}_{tk} \) and \( y = (y_1, \ldots, y_N) \) of \( \tilde{Y}_{tk-1} \), the approximated transition probabilities are given by:

\[
\tilde{p}_{k}^{ij} := \mathbb{P}(\tilde{Y}_{tk} \in C_j(x) | \tilde{Y}_{tk-1} \in C_i(y)) = \Phi_{0,1}\left( \tilde{x}_{k+1,i}^+(y_j) \right) - \Phi_{0,1}\left( \tilde{x}_{k+1,i}^-(y_j) \right)
\]

For the computation of the approximated transition kernels we have (the grid size is equal to \( N \)):

\[
\hat{H}_k = \sum_{j=1}^{N} q_k(x_{k-1}^j, x_k^j) \tilde{p}_{k}^{ij} \delta_{x_{k-1}^j}
\]

\( k = 1, \ldots, M \)

and

\[
\hat{H}_0 = \sum_{i=1}^{N} \mathbb{P}(\tilde{X}_0 = x_0^i) \delta_{x_0^i}
\]
References


Table 1: Results on the pricing via closed form formulas and quantization. In the case of two real roots the parameters are taken from [Andersen (2011): $\sigma = 0.2$; $q = 0.5$; $s = 0.1$ (in the (6) spedification). We then perturb the $s$ parameter in order to get two complex roots: $\sigma = 0.2$; $q = 0.5$; $s = 5$.

<table>
<thead>
<tr>
<th></th>
<th>Real roots</th>
<th>Complex roots</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comp. time</td>
<td>10.53919 sec</td>
<td>18.27272 sec</td>
</tr>
<tr>
<td>Res. Norm</td>
<td>$3.61711 \times 10^{-5}$</td>
<td>$5.13589 \times 10^{-4}$</td>
</tr>
</tbody>
</table>

Table 2: Calibrated parameters of the Quadratic Normal Volatility model.

<table>
<thead>
<tr>
<th></th>
<th>$\sigma$</th>
<th>$q$</th>
<th>$s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exact formulas / long maturities</td>
<td>0.16019</td>
<td>$-0.04380$</td>
<td>26.69999</td>
</tr>
<tr>
<td>Quantization / long maturities</td>
<td>0.17451</td>
<td>0.00005</td>
<td>7.62015</td>
</tr>
<tr>
<td>Quantization / short maturities</td>
<td>0.14536</td>
<td>$-4.67521$</td>
<td>16.7493</td>
</tr>
</tbody>
</table>

Table 3: Results on the calibration via closed form formulas and quantization.

<table>
<thead>
<tr>
<th></th>
<th>Closed form formulas</th>
<th>Quantization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comp. time</td>
<td>339.15013 sec</td>
<td>442.27048 sec</td>
</tr>
<tr>
<td>Res. Norm</td>
<td>$5.62922 \times 10^{-4}$</td>
<td>$4.26904 \times 10^{-4}$</td>
</tr>
<tr>
<td>Long maturities</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Short maturities</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Computational time</td>
<td>-</td>
<td>448.44574 sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.77111 $\times 10^{-4}$</td>
</tr>
</tbody>
</table>

Table 4: Results on the pricing of up-and-out put options via closed formulas and quantization. Barrier close to the initial price.

<table>
<thead>
<tr>
<th></th>
<th>Benchmark price</th>
<th>Quantization price</th>
<th>Monte Carlo price</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L = y_0 + 5$</td>
<td>22.39558</td>
<td>53.39679</td>
<td>55.16941</td>
</tr>
<tr>
<td>$L = y_0 + 10$</td>
<td>26.53438</td>
<td>57.01786</td>
<td>58.95591</td>
</tr>
<tr>
<td>$L = y_0 + 15$</td>
<td>30.76803</td>
<td>60.46311</td>
<td>62.63455</td>
</tr>
<tr>
<td>$L = y_0 + 20$</td>
<td>35.07570</td>
<td>64.02864</td>
<td>66.05857</td>
</tr>
<tr>
<td>$L = y_0 + 25$</td>
<td>39.61555</td>
<td>68.16988</td>
<td>69.56067</td>
</tr>
<tr>
<td>Computational time</td>
<td>-</td>
<td>14.61447 sec</td>
<td>52.25725 sec</td>
</tr>
</tbody>
</table>

Figure 1: Implied volatility squared errors for the calibration via quantization. Long maturities on the left, short maturities on the right.
Table 5: Results on the pricing of up-and-out put options via closed formulas and quantization. The barrier is in percentage of the initial price.

<table>
<thead>
<tr>
<th>Barrier (L)</th>
<th>Benchmark price</th>
<th>Quantization price</th>
<th>Monte Carlo price</th>
</tr>
</thead>
<tbody>
<tr>
<td>101.25</td>
<td>122.83447</td>
<td>141.06537</td>
<td>144.33757</td>
</tr>
<tr>
<td>102.5</td>
<td>205.96689</td>
<td>216.54927</td>
<td>220.69586</td>
</tr>
<tr>
<td>103.75</td>
<td>264.11461</td>
<td>268.67753</td>
<td>273.11058</td>
</tr>
<tr>
<td>105</td>
<td>299.71130</td>
<td>299.93827</td>
<td>304.81308</td>
</tr>
<tr>
<td>106.25</td>
<td>318.74848</td>
<td>316.32067</td>
<td>321.23110</td>
</tr>
<tr>
<td>Computational time</td>
<td>-</td>
<td>13.98614 sec</td>
<td>50.06883 sec</td>
</tr>
</tbody>
</table>