Weighted Monte Carlo

Most pricing models assume an asset behaviour and calibrate its parameters to fit the market. Weighted Monte Carlo is able to calibrate the market without making specific assumptions about the asset behaviour. When only vanilla products are considered, the asset behaviour may allow for arbitrage. Alberto Elices and Eduard Giménez show that this is indeed the case and introduce a synthetic option to overcome the problem.

Taking into account the smile effect when pricing exotic derivatives is an issue of major concern for both traders and practitioners. This is because the impact of this effect on price and hedging can be very significant. Under an efficient market hypothesis, skew and smile in the vanilla option market can be explained either by the fact that volatility is not constant or because the asset price process might have jumps.

There have been many proposals for asset price processes to capture the above-mentioned effects. Merton (1976) deduced the risk-neutral partial-integral-differential equation that a derivative should follow under the assumption of a jump diffusion process with constant volatility. He also found the analytical solution for a call option. Dupire (1994) proposed a model based on the assumption that the volatility of the asset is a deterministic function of time and price. This model has no general analytical solution for call options.

Heston (1993) found the analytical solution for a call option assuming that the volatility follows the continuous version of a Garch process. Other proposals for stochastic volatility models can be found in Hull & White (1987), Stein (1991) and Hagan et al (2002). Other authors have focused on providing no-arbitrage pricing models that might not have an explicit expression of the asset price process but are able to replicate a broad set of traded options. The implied tree of Derman & Kani (1994) and the weighted Monte Carlo of Avellaneda et al (2001) belong to this category. Another interesting contribution is given by Britten-Jones (2000), in which different models, all replicating the option prices of the smile, value the same options differently.

The conceptual idea of weighted Monte Carlo is very simple. An initial set of paths is generated. These paths can incorporate any desired feature of the underlying process, such as mean-reverting stochastic volatility, different regimes of volatility, volatility correlated with the underlying process or even jumps. For regular Monte Carlo, all these paths will have the same probability. Weighted Monte Carlo modifies the probability of each path to replicate the prices of the options at different maturities. As the number of degrees of freedom is huge (as many as the number of paths), it is clear that to fit a few option prices, an infinite number of solutions must exist in the absence of arbitrage. From all those, the solution chosen is such that the posterior distribution is as close as possible to the prior in a certain sense. This solution is found by solving an optimisation problem under as many constraints as the number of option prices to be fitted.

The main drawback of this method is that when several maturities are simulated and the smile effect is steep, the probability distribution at each maturity after calibration may be significantly different from the original. The method has no control over the resulting price process after calibration if no additional action is taken. In particular, the martingale property may not be satisfied. The main contribution of this article overcomes this problem by introducing additional constraints that force the conditional martingale condition to be satisfied. This is the only property that all processes must share for consistent pricing. Another contribution of this article is a robust, fast and easy-to-implement solution algorithm for the optimisation problem. This algorithm allows the handling of a considerable number of constraints.

The following section reviews the calibration method and presents a robust, fast and easy-to-implement algorithm to find the solution. We then present the formulation of the additional constraints to force the conditional martingale property of the underlying process. Then we give an example of a geometric cliquet option that is considerably undervalued when the martingale condition is not enforced (about 160 basis points). This conclusion is very much in agreement with trader experience and market pricing. We then conclude.

Review of the method

Formulation of the problem. Pricing a European option $\Pi$ using Monte Carlo involves the simulation of $n$ underlying paths according to the hypothesis considered, the calculation of the discounted payouts at maturity ($g_i$) and the calculation of the price according to equation (1):

$$\Pi = \frac{1}{v} \sum_{i=1}^{v} g_i$$  \hspace{1cm} (1)

The prices of the call and put options at different strikes and maturities will in general not be reproduced from the paths generated. Weighted Monte Carlo produces a set of probabilities $p_i$ close to $q_i = 1/v$ in a certain sense providing a new price for $\Pi$ as shown in equation (2):

$$\Pi = \sum_{i=1}^{v} g_i p_i$$  \hspace{1cm} (2)

The ‘distance’ between two distributions $p$ and $q$ is defined using the

$$d(p, q) = \sum_{i=1}^{v} |p_i - q_i|$$
Relative entropy (althoughStrictly speaking the relative entropy is not a distance) as defined in equation (3):

$$D(p/q) = \sum_{i=1}^{N} p_i \ln \left( \frac{p_i}{q_i} \right)$$  (3)

When the probabilities of the prior distribution are equal, equation (3) reduces to equation (4):

$$D(p/q) = \ln v + \sum_{i=1}^{N} p_i \ln p_i$$  (4)

Equation (5) presents a matrix of discounted payouts for the $N$ securities to which the model will be calibrated. Each column is a security and the rows are the discounted payouts for each path. This set of instruments includes the payouts of the smile options (calls and puts) for different maturities, the forward prices at each maturity (the simulation of the paths at each maturity) and the additional constraints presented in the following section:

$$\begin{bmatrix}
S_{1,1} & S_{1,2} & \cdots & S_{1,N} \\
S_{2,1} & S_{2,2} & \cdots & S_{2,N} \\
\vdots & \vdots & \ddots & \vdots \\
S_{v,1} & S_{v,2} & \cdots & S_{v,N}
\end{bmatrix} = (g_1, g_2, \ldots, g_v)$$  (5)

The market prices of these securities are $C_j$, to $C_v$. The whole purpose of the calibration is to find a set of $p_i$ so that equation (6) is satisfied:

$$\sum_{i=1}^{N} p_i S_{i,j} = C_j, \quad j = 1, \ldots, N$$  (6)

Equation (7) shows the formulation of the optimisation problem to get the posterior probabilities $p_i$:

$$\min_{\lambda} D(p/q) = \min_{\lambda} \left[ \ln v + \sum_{i=1}^{N} p_i \ln p_i \right]$$  (7)

subject to $\sum_{i=1}^{N} p_i S_{i,j} = C_j$, $j = 1, \ldots, N$

This optimisation problem can be solved using Lagrange multipliers (see Bertsekas, 1999) as the min-max problem of equation (8) (with the same notation as Avellaneda et al, 2001):

$$\min_{\lambda} \left[ \max_{\lambda} \left[ -D(p/q) + \sum_{j=1}^{N} (Z(\lambda) - C_j) \right] \right]$$  (8)

The solution of the maximisation problem has been extensively studied by Cover & Thomas (1991). For a given set of $\lambda$, the optimal probability is given by equation (9). The normalisation constant $Z$ ensures the probabilities add up to one:

$$p_i = \frac{1}{Z(\lambda)} \exp \left( \sum_{j=1}^{N} g_j \lambda_j \right)$$  (9)

If these probabilities are replaced in equation (8), the resulting optimisation problem is given by equation (10):

$$\min_{\lambda} W(\lambda) \quad \text{where} \quad W(\lambda) = \ln Z(\lambda) - \sum_{j=1}^{N} (C_j)$$  (10)

The box ‘Derivation of the optimisation problem’ provides the detail of the calculations.

At any minimum, the gradient must equal zero. Equation (11) details the calculation of the gradient and shows that the expected values of the discounted payouts with respect to the posterior probability distribution $p_i$ equal the desired option prices $C_j$:

$$\frac{\partial W(\lambda)}{\partial \lambda_k} = \frac{1}{Z(\lambda)} \frac{\partial Z(\lambda)}{\partial \lambda_k} - C_k = 0$$  (11)

$$= \frac{1}{Z(\lambda)} \sum_{i=1}^{N} g_i \exp \left( \sum_{j=1}^{N} g_j \lambda_j \right) - C_k = E_p(g_i) - C_k = 0$$  (11)

It is shown in Avellaneda et al (2001) that when market prices are consistent, this minimum exists and it is unique.

**Solution algorithm.** This section presents the algorithm to find $\lambda$, such that $W(\lambda)$ is minimised. To achieve a robust, fast and easy-to-implement algorithm, an iterative method based on the second-order Taylor approximation of $W(\lambda)$ around an initial point $\lambda$ is used. Equation (12) shows this second-order approximation:

$$W(\lambda) = W(\lambda_0) + V_\lambda W(\lambda_0) (\lambda - \lambda_0) + \frac{1}{2} (\lambda - \lambda_0) ^ T J (\lambda - \lambda_0)$$  (12)

The matrix $J$ is the Jacobian (second derivatives with respect to $\lambda$) and it turns out to be the covariance matrix with respect to the probability $p$ of the securities $g_i$ to calibrate (see box, ‘Derivation of the Jacobian’, on the next page). Equation (13) shows a general element $J_{ij}$ of the Jacobian and equation (14) shows the gradient and the vector ($\lambda - \lambda$):

$$J_{ab} = \frac{\partial^2 W(\lambda)}{\partial \lambda_a \partial \lambda_b} = \text{Cov}_p(g_a, g_b)$$  (13)

$$= \sum_{j=1}^{N} g_a g_b P_j - \left( \sum_{i=1}^{v} g_a P_i \right) \left( \sum_{i=1}^{v} g_b P_i \right)$$  (13)
Derivation of the Jacobian

The Jacobian is the matrix of second derivatives. The following equation shows the element at the row \( u \) and column \( v \) of this matrix:

\[
J_{w_v} = \frac{\partial W(\lambda)}{\partial g_{v}} = \frac{\partial}{\partial g_{v}} \left( \mathbf{E}[g_j] - C_v \right) = \frac{\partial E^p(\lambda)}{\partial g_{v}} = \mathbf{C} \mathbf{v}
\]

The derivative of \( W(\lambda) \) with respect to \( \lambda \) is given by equation (11) and the following equation presents the derivative of the expected value of a general payout vector \( h \):

\[
\frac{\partial E^p(h)}{\partial \lambda} = \frac{1}{Z(\lambda)} \sum_{i=1}^{N} \left( \sum_{j=1}^{N} g_{ij} \right) h_i
\]

As the Jacobian is a covariance matrix, it is semi-positive definite and there is always a minimum (and there is always a minimum (\( W(\lambda) \)) increases with local variations of \( \lambda \) around \( \lambda^* \)). This condition will hold for any point where we do not consider the vector \( g_{ij} \) of discounted payouts will have zeros in most entries. This means that the covariance of this product with itself or any other products will be very low and the Jacobian matrix will be badly conditioned (it will have a row and column of almost zero entries). These products should be removed from the calculation. The products considered in the smile are out-of-the-money calls for strikes over the forward and out-of-the-money puts for strikes below the forward. This allows a symmetric removal of out-of-the-money products (for very low maturities, only a small window of strike options with strikes around the forward will be calibrated). In addition, when there are arbitrage opportunities in the market (for example, a call and a put with different strikes and same price), the algorithm does not converge. The only way to solve this problem is to remove usually illiquid products that may be mis-priced.

The partial solution of step six can differ significantly from the solution of the previous step. This is an undesirable effect as the second-order approximation of the previous step may no longer be valid. Therefore, a standard under-relaxed Newton update is introduced by a step-shortening factor \( \alpha \) and the partial solution (16) is replaced by equation (17). This factor is set initially to 0.01 and is multiplied by two after each iteration until the shortening factor \( \alpha \) is equal to 1:

\[
\lambda = \lambda - \alpha^{-1} \nabla \lambda W(\lambda^*)
\]

It can be seen that the condition number of the Jacobian can increase significantly when the shortening factor is too high. Therefore, the step is divided by five when the condition number increases by a factor greater than 10 from one iteration to the following.

**Formulation of the problem using weighted least-squares.** The formulation above calibrates exactly to the desired products. When products are mis-priced due to low liquidity or wide bid-ask spreads, the algorithm may not converge well and these products should be removed. Therefore, it may be better not to fit the products exactly but to minimise the sum of weighted least-squares of equation (18):

\[
\chi^2 = \frac{1}{2} \sum_{j=1}^{N} \sum_{i=1}^{N} \alpha_j \left( E^p(\lambda) - C_j \right)^2
\]

where \( \alpha_j \) are positive weights. Now, what is minimised is equation (19) without constraints:

\[
\min_p \left[ D(q^p) + \chi^2 \right]
\]

Minimising (19) with respect to \( p \) is equivalent to minimising (20) with respect to \( \lambda^* \). The proof of this can be found in Avellaneda et al. (2001) and will be omitted here. If equation (20) is compared with equation (10), the difference between exact and weighted least-
least-squares fitting is the term after \(W(\lambda)\) in (20):
\[
\min_k \left[ H(\lambda) \right] \text{ where } H(\lambda) = W(\lambda) + \frac{1}{2} \sum_{j=1}^{N} \omega_j \lambda_j^2 \tag{20}
\]

Equations (21) and (22) show the minor changes of gradient and Jacobian for weighted least-squares fitting:
\[
\frac{\partial H(\lambda)}{\partial x_a} = Ep(\omega_a) - C_a + \omega_a \lambda_a \tag{21}
\]
\[
J_{ab} = \frac{\partial H(\lambda)}{\partial x_a} \frac{\partial H(\lambda)}{\partial x_b} = Cov(\omega_a, \omega_b) + \omega_a I_{a=b} \tag{22}
\]

The modified Jacobian is the covariance matrix plus a diagonal matrix with the weights of each product. Now it is much better conditioned, as there is no row or column with zero entries. The solution algorithm is the same as in the previous section.

Forcing the martingale condition

The resulting price process will be feasible for pricing exotic derivatives if it satisfies two conditions: it reproduces the prices of traded derivatives, and it is a martingale. The calibration fulfills the first condition but the second will not in general be satisfied. In particular, given the continuously compounded risk-free rate \(r_t\) and dividend rate \(q_t\) from present time to time \(t\), equation (23) shows the process that must be a martingale:
\[
S_t e^{(r_t-q_t) t} \sim \text{Martingale} \tag{23}
\]

Equation (24) shows the martingale condition between two maturities \((t_1\) and \(t_2\)), where \(S_t\) represents the information up to time \(t_1\):
\[
E \left[ \frac{S_t e^{(r_t-q_t) t_1}}{I_{t_1}} \right] = S_t e^{(r_t-q_t) t_1} \tag{24}
\]

To verify this condition within the Monte Carlo environment, it is necessary to consider the paths that go through a window \(W\) at \(t_1\), as shown in figure 1.

If small enough windows are considered, equation (24) can be understood as equation (25). Equation (26) shows this condition applied to Monte Carlo paths. Only the paths that go through window \(W\) are considered. \(S_{t_1}^{j}\) and \(S_{t_2}^{j}\) are the prices of path \(j\) at times \(t_1\) and \(t_2\), and \(F(t)\) is the forward price of the underlying:
\[
E \left[ \frac{S_{t_j} e^{(r_t-q_t) t_1}}{I_{t_1}} \right] = E \left[ \frac{S_{t_j} e^{(r_t-q_t) t_1}}{S_{t_1} e^{(r_t-q_t) W}} \right] = \frac{S_{t_j} e^{(r_t-q_t) W}}{S_{t_1} e^{(r_t-q_t) W}} \tag{25}
\]
\[
\sum \left\{ \frac{P_j}{S_{t_1} e^{(r_t-q_t) W}} \right\} \left( \frac{F(S_{t_j})}{F(S_{t_1})} - S_{t_1}^{j} \right) \sum \left\{ \frac{P_j}{S_{t_1} e^{(r_t-q_t) W}} \right\} P_k = 0 \tag{26}
\]

It is therefore possible to create a synthetic security with a zero price and discounted payouts as shown in equation (27):
\[
h_j = \sum \left\{ \frac{P_j}{S_{t_1} e^{(r_t-q_t) W}} \right\} \left( \frac{F(S_{t_j})}{F(S_{t_1})} - S_{t_1}^{j} \right) I_{S_{t_j} \notin \omega W} \tag{27}
\]

The narrower the windows are, the more precise the martingale condition is enforced. However, narrower windows imply more simulations and constraints, increasing significantly the computation time. The trade-off chosen is to use the windows defined by the

### A. Volatility surface

<table>
<thead>
<tr>
<th>Rate</th>
<th>0.68y</th>
<th>0.25y</th>
<th>0.50y</th>
<th>0.75y</th>
<th>1y</th>
<th>2y</th>
<th>3y</th>
<th>4y</th>
<th>5y</th>
<th>10y</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.065</td>
<td>35.86</td>
<td>30.38</td>
<td>25.21</td>
<td>24.04</td>
<td>23.91</td>
<td>23.33</td>
<td>23.03</td>
<td>23.67</td>
<td>23.88</td>
<td>24.55</td>
</tr>
<tr>
<td>8.506</td>
<td>19.86</td>
<td>18.38</td>
<td>17.51</td>
<td>17.44</td>
<td>17.51</td>
<td>18.33</td>
<td>18.77</td>
<td>19.47</td>
<td>20.30</td>
<td>22.35</td>
</tr>
<tr>
<td>9.006</td>
<td>16.16</td>
<td>15.63</td>
<td>15.61</td>
<td>15.76</td>
<td>15.91</td>
<td>17.33</td>
<td>17.77</td>
<td>18.57</td>
<td>19.50</td>
<td>21.80</td>
</tr>
<tr>
<td>10.007</td>
<td>10.42</td>
<td>11.37</td>
<td>12.21</td>
<td>12.89</td>
<td>13.53</td>
<td>15.53</td>
<td>15.97</td>
<td>16.97</td>
<td>18.05</td>
<td>20.80</td>
</tr>
<tr>
<td>11.508</td>
<td>13.00</td>
<td>9.98</td>
<td>10.90</td>
<td>10.47</td>
<td>11.21</td>
<td>13.28</td>
<td>13.87</td>
<td>14.77</td>
<td>16.05</td>
<td>19.45</td>
</tr>
<tr>
<td>12.008</td>
<td>15.00</td>
<td>10.98</td>
<td>10.40</td>
<td>10.40</td>
<td>10.71</td>
<td>12.68</td>
<td>13.27</td>
<td>14.17</td>
<td>15.45</td>
<td>19.10</td>
</tr>
</tbody>
</table>
strikes of the options of the smile. Additional windows up and down the highest and lowest strike of the smile are also considered with similar widths. These additional windows go up to two standard deviations up and down from the current spot level. These standard deviations are considered for the last maturity with at-the-money-forward (ATMF) volatility.

The case study
The weighted least-squares calibration method has been applied to a geometric cliquet option with payoff (28), where

\[
\Pi = \max \left\{ \sum_{j=2}^{7} \min \left( \frac{S_j}{S_{j-1}}, C \right) - 1 \right\} \tag{28}
\]

The underlying is the Spanish Ibex index with spot price \( S_0 = 699.0 \). The risk-free and dividend rates are respectively 2.95% and 3% (they have been set constant for simplicity), the least-squares weights for all constraints are \( 10^{-7} \) and the volatility surface is presented in table A.

The paths were generated with regular Monte Carlo with ATMF deterministic volatility (the volatility level is interpolated in table A using the forward at each maturity). The smile has been fitted to the seven maturities of the cliquet option and the martingale condition has been imposed on windows starting on 0.35 times the spot up to 2.25 at each pair of consecutive maturities. The width is the distance between consecutive strikes of matrix in table A. Windows outside this matrix use the last upper and lower window width. From 346 constraints \((15 \times 7 - 105)\) smile options, \( 39 \times 6 = 234 \) martingale conditions and seven forwards), 131 were removed (the five upper and four lower options of the smile at the first maturity, the upper smile option of the second maturity and the rest were out-of-the-money martingale conditions). All option prices were fitted with a precision better than \( 10^{-5} \) (errors less than a tenth of a basis point). The total number of paths is 20,000. The algorithm converges after 14 iterations.

Table B compares three prices. The first corresponds to the regular Monte Carlo with ATMF volatility (all paths have the same probability). The last two prices correspond to the weighted Monte Carlo fitting the smile at every maturity without imposing the martingale condition and imposing it. All prices are calculated with the same paths, only changing the path probabilities according to the calibration. Enforcing the martingale condition makes a big difference: the option is 160 basis points more expensive. This is in agreement with trader experience and market pricing. Figure 2 presents the distribution of the underlying at \( t_4 \) and shows how the distribution is skewed to lower values (the left queue is considerably bigger than the right one).

To explain the difference in price, consider figure 3 with the martingale condition mismatches with martingale constraints (right plot) and without them (left plot). They correspond to the period \( t_4 \) to \( t_5 \) (the rest of the periods look very similar). The horizontal axis has the levels of the martingale windows and the vertical axis shows equation (26) both in per unit of the spot. The dotted line corresponds to the regular Monte Carlo and the solid line to the weighted Monte Carlo. Both plots show that the regular Monte Carlo reasonably satisfies the martingale condition (the mismatches are below 2%). The left plot shows that the weighted Monte Carlo without martingale constraints does not satisfy the condition. The underlying process
is a super-martingale at $t_f$ for levels between 0.7 to 1.2 of the spot and a sub-martingale outside them. The right plot clearly satisfies the condition because it is enforced.

At first sight it seems reasonable to think that the option should be more expensive for the left plot of figure 3, as more likely underlying values around the spot would have a higher expected value (higher return). To prove that this perception is false, figure 4 shows the cumulative probability distributions of non-paying (left) and paying (right) paths at $t_f$, when the martingale condition is and is not enforced. A paying path is a path for which equation (28) is different from zero.

These distributions are not conditional and therefore about 70% of the paths do not contribute to the payout whereas about 30% do. The right plot of figure 4 shows that all the paying paths of the weighted Monte Carlo where the martingale condition is enforced (solid line) have higher probabilities than when it is not (dotted line). That is why the price is significantly higher. The left plot of figure 4 shows that the paths that finish below 1.2 times the spot have the same probabilities with and without martingale condition. These paths correspond to decreasing paths that do not contribute to the payout. The paths that finish above 1.2 times the spot significantly change their probabilities with and without the martingale condition. Figure 3 (left) shows that paths whose spot is between 0.7 and 1.2 are more likely to go up and above 1.2 are more likely to go down. Therefore, it is clear from the martingale mismatches of figure 3 (left) that zigzagging paths are over-weighted (otherwise the plot would be flat). These zigzagging paths do not contribute significantly to the payout (very negative returns decrease the payout and very positive returns do not compensate as they are capped). Therefore, the price of the option reduces. When the martingale condition is enforced, these zigzagging paths reduce their probabilities as they are responsible for the distortion of the martingale condition and allow other paying paths to increase their probabilities, increasing the price of the option.

**Conclusion**

When the weighted Monte Carlo is calibrated to several maturities with steep smiles, the resulting underlying process after calibration may not be a martingale. For certain types of path-dependent options, this may have a big impact in price.

A simple solution to overcome this problem is presented. It consists of adding additional constraints to the problem. A new robust, fast and easy-to-implement calibration algorithm is proposed. It can cope with a significant number of constraints.

The improved method is applied to a well-known geometric cliquet option that has a big impact on price and is also in line with trader and market experience.

Alberto Elices and Eduard Giménez are senior quant team members in the model validation group of the risk department at Grupo Santander in Madrid. They would like to thank Roland Ordovás and other members of the bank for helpful comments. Email: aelices@gruposantander.com, edgimenez@gruposantander.com

---

**References**


---

**Risk.net**