Weighted Monte Carlo: Preserving the Martingale Condition

Alberto Elices & Eduard Giménez

Model Validation Group
Risk Division
Grupo Santander

Quant Congress Europe 2006
London, Oct 11th and 12th, 2006
Outline of the presentation

- Introduction.
- Formulation of the Weighted Monte Carlo (WMC) method.
- Calibration algorithm.
- Interpretation of calibration results.
- Pitfalls: preserving martingale condition.
- Case Study.
Conceptually, there are two types of pricing models:

- Those which make assumptions for the underlying process and calibrate the parameters of the model to the market.
- Those which make no explicit assumptions for the underlying process but by construction replicate the market (WMC belongs to this group).

The conceptual idea of the WMC is very simple:

- An initial set of paths is generated.
- For a regular Monte Carlo, all these paths will have the same probability.
- The WMC modifies the probability of each path to replicate the prices of the options of the smile at different maturities.
Finding the WMC solution:
- To calibrate the prices of smile options, there are as many degrees of freedom as number of paths: infinite solutions.
- The solution is found by solving an optimization problem so that the posterior probability distribution is as close as possible to the prior in a certain sense.

Results of WMC:
- When several maturities are calibrated, the behaviour of the optimization algorithm might not be intuitive.
- There is not a clear control over the resulting underlying process after calibration.
- It has been found that the martingale condition might not be preserved.
Formulation of the Weighted Monte Carlo Method

- **Pricing an european option** \( \Pi_g \) using Monte Carlo involves:
  - The simulation of \( \nu \) underlying paths according to the hypothesis considered.
  - The calculation of the discounted payoffs at maturity \( g_i \)
  - The calculation of the price according to following equation:

\[
\Pi_g = \frac{1}{\nu} \sum_{i=1}^{\nu} g_i
\]

- This price will not necessarily match the market. Therefore, WMC provides a set of probabilities \( p_i \) close to \( q_i = 1/\nu \) in a certain sense.

\[
\Pi_g = \sum_{i=1}^{\nu} g_i p_i
\]
The “distance” between two distributions \( p \) and \( q \) is defined using the relative entropy:

\[
D(p / q) = \sum_{i=1}^{\nu} p_i \ln \left( \frac{p_i}{q_i} \right)
\]

When the probabilities of the prior distribution are equal, the former equation reduces to:

\[
D(p / q) = \ln \nu + \sum_{i=1}^{\nu} p_i \ln p_i
\]
The calibration problem consists of finding a set of $p_i$ so that the following equations are satisfied:

$$\sum_{i=1}^{v} p_i g_{i,j} = C_j , \quad j = 1 \cdots N$$

Where

- $C_j$ are the prices of traded options.
- $g_{ij}$ is the payoff of each option $C_j$ for each path $w_i$. 
The following equation shows the formulation of the optimisation problem to get the posterior probabilities:

\[
\min_{\mathbf{p}} D(\mathbf{p} / \mathbf{q}) = \min_{\mathbf{p}} \left( \ln \nu + \sum_{i=1}^{\nu} p_i \ln p_i \right)
\]

Subject to \( \sum_{i=1}^{\nu} p_i g_{i,j} = C_j \), \( j = 1 \ldots N \)
This optimisation problem can be solved by lagrange multipliers with the following min-max problem:

\[
\min_{\lambda} \left\{ \max_{p} \left\{ -D(p / q) + \sum_{j=1}^{N} \lambda_j \left( \sum_{i=1}^{v} p_i g_{i,j} - C_j \right) \right\} \right\}
\]

Where:
- \(\lambda_j\): lagrange multipliers.
The solution of the max problem is given by the probabilities $p_i$ as a function of $\lambda_i$.

$$p_i = \frac{1}{Z(\lambda)} \exp \left( \sum_{j=1}^{N} g_{ij} \lambda_j \right) \text{ with } Z(\lambda) = \sum_{i=1}^{\nu} \exp \left( \sum_{j=1}^{N} g_{ij} \lambda_j \right)$$

The formulation of the min problem after replacing $p_i$ in the min-max equation is the following:

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

- To find the minimum, a second order Taylor approximation of the objective function is used:

\[
W(\lambda) \approx W(\hat{\lambda}) + \nabla_\lambda W(\hat{\lambda})(\lambda - \hat{\lambda}) + \frac{1}{2} (\lambda - \hat{\lambda})^T J (\lambda - \hat{\lambda})
\]

- The resulting quadratic programming problem has analytical solution:

\[
\nabla_\lambda W(\hat{\lambda}) + J(\lambda - \hat{\lambda}) = 0
\]

\[
\lambda = \hat{\lambda} - J^{-1} \nabla_\lambda W(\hat{\lambda})
\]
The gradient shows that in the optimum (grad=0), the market prices are replicated:

\[ \nabla_\lambda W(\hat{\lambda}) = \begin{pmatrix} E^p(g_1) - C_1 \\
\vdots \\
E^p(g_N) - C_N \end{pmatrix} \quad \lambda - \hat{\lambda} = \begin{pmatrix} \lambda_1 - \hat{\lambda}_1 \\
\vdots \\
\lambda_N - \hat{\lambda}_N \end{pmatrix} \]
The Jacobian turns out to be the covariance matrix with respect to the probability $p$ of the securities $g_{ij}$.

This matrix is always semi-positive definite. Therefore, the problem is convex and has an optimal.

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

$$= \sum_{j=1}^{v} g_{aj} g_{bj} p_j - \left( \sum_{j=1}^{v} g_{aj} p_j \right) \left( \sum_{j=1}^{v} g_{bj} p_j \right)$$
The steps of the algorithm could be summarised as follows:

1. Simulate \( \nu \) paths at different maturities with regular Monte Carlo.
2. Calculate the prices \( C_k \) of the calibration set.
3. Calculate the discounted payoffs \( g_{ij} \) of all these instruments for each of the simulated paths.
4. Fix the initial value for the lagrange multipliers \( \lambda_i \) equal to zero (all paths are equally weighted).
5. Calculate the gradient vector and jacobian matrix.
6. Calculate the optimum for the second order approximation solving the quadratic programming problem.
7. Go to step 5 until the gradient is close enough to zero.
Calibration algorithm

Practical tips to get the algorithm working:

- Very out-of-the-money options should be removed from the calibration set (covariance matrix would be ill-conditioned).
- The algorithm does not converge when there are arbitrage opportunities due to illiquid mis-priced products. These products should be removed.
- A step shortening factor should be introduced ($\alpha < 1$):

$$\lambda = \hat{\lambda} - \alpha J^{-1} \nabla_{\lambda} W(\hat{\lambda})$$

- This factor is doubled at each step up to 1. When the condition number of the jacobian increases by a factor of 10, $\alpha$ is divided by 5.
Exact fitting may cause convergence problems due to illiquid products which should be manually removed.

Instead of calibrating exactly, it might be better to minimize the sum of weighted least squares:

\[
\chi^2_\omega = \frac{1}{2} \sum_{j=1}^{N} \frac{1}{\omega_j} \left( E^p(g_j) - C_j \right)^2
\]

The resulting optimization problem is the following:

\[
\min_p \left[ D(q/p) + \chi^2_\omega \right]
\]
Calibration algorithm

- This minimization is equivalent to the following:

\[
\min_{\lambda} H(\lambda) \quad \text{where} \quad H(\lambda) = W(\lambda) + \frac{1}{2} \sum_{j=1}^{N} \omega_j \lambda_j^2
\]

- There are minor changes in the gradient and jacobian (this is much better conditioned):

\[
\frac{\partial H(\lambda)}{\partial \lambda_a} = E^p(g_a) - C_a + \omega_a \lambda_a
\]

\[
J_{ab} = \frac{\partial H(\lambda)}{\partial \lambda_a \partial \lambda_b} = \text{Cov}^p(g_a, g_b) + \omega_a 1_{\{a=b\}}
\]
## Interpretation of calibration results

### IBEX index 6y, S=10007, r = 2.95%, q = 3%

<table>
<thead>
<tr>
<th>KM</th>
<th>0.08y</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>6505</td>
<td>35.86</td>
<td>30.38</td>
<td>25.21</td>
<td>24.04</td>
<td>22.91</td>
<td>22.33</td>
<td>22.82</td>
<td>23.37</td>
<td>23.80</td>
<td>24.55</td>
</tr>
<tr>
<td>7505</td>
<td>27.86</td>
<td>24.38</td>
<td>21.31</td>
<td>20.74</td>
<td>20.01</td>
<td>20.33</td>
<td>20.77</td>
<td>21.37</td>
<td>22.00</td>
<td>23.45</td>
</tr>
<tr>
<td>8506</td>
<td>19.86</td>
<td>18.38</td>
<td>17.51</td>
<td>17.44</td>
<td>17.31</td>
<td>18.33</td>
<td>18.77</td>
<td>19.47</td>
<td>20.30</td>
<td>22.35</td>
</tr>
<tr>
<td>9006</td>
<td>16.16</td>
<td>15.63</td>
<td>15.61</td>
<td>15.79</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>10007</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>10507</td>
<td>10.02</td>
<td>10.31</td>
<td>11.13</td>
<td>11.79</td>
<td>12.46</td>
<td>14.68</td>
<td>15.17</td>
<td>16.17</td>
<td>17.35</td>
<td>20.30</td>
</tr>
<tr>
<td>11508</td>
<td>13.00</td>
<td>9.98</td>
<td>10.50</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>12008</td>
<td>15.50</td>
<td>10.08</td>
<td>10.40</td>
<td>10.07</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>
<tr>
<td>12509</td>
<td>18.50</td>
<td>10.48</td>
<td>10.40</td>
<td>9.87</td>
<td>10.31</td>
<td>12.18</td>
<td>12.87</td>
<td>13.77</td>
<td>14.85</td>
<td>18.80</td>
</tr>
</tbody>
</table>
Interpretation of calibration. Case 0.
Interpretation of calibration. Case 1: $K = 1*S$, $P_{CALL} = 0.1495 + 0.0005$
Interpretation of calibration. Case 2: $K = 1*S$, $P_{\text{CALL}} = 0.1495-0.0007$
Interpretation of calibration. Case 3: $K = 0.75 \times S$, $P_{PUT} = 0.0742 + 0.0003$
Interpretation of calibration. Case 4: \( K = 0.75*S, \ P_{PUT} = 0.0742-0.0006 \)
Interpretation of calibration. Case 5: \( K = 0.65S \), \( P_{\text{PUT}} = 0.0525 + 0.0012 \)
Interpretation of calibration. Case 6: $K = 1.3*S$, $P_{\text{CALL}} = 0.0485-0.0007$
Interpretation of calibration. Case 7: $K = [0.85 \ 1 \ 1.15]*S$; $P[+3 \ +5 \ -5]$ pips
Pitfalls: preserving martingale condition

- All possible processes for valuation of exotics must satisfy at least two conditions:
  - Reproduce the vanilla prices of the volatility surface.
  - Be a martingale.

- The process that must be a martingale is the following:

\[ S_t e^{-(r_t - q_t) t} \sim \text{Martingale} \]

- The martingale condition:

\[ E \left[ S_{t_2} e^{-(r_{t_2} - q_{t_2}) t_2} / I_{t_1} \right] = S_{t_1} e^{-(r_1 - q_1) t_1} \]
Pitfalls: preserving martingale condition

- To study the martingale condition for a Monte Carlo in between two maturities, windows should be considered:

\[
E\left[ S_{t_2} e^{-(r_2-q_2)t_2} / S_{t_1} \in W \text{ and } I_{t_1} \right] = E\left[ S_{t_1} e^{-(r_1-q_1)t_1} / S_{t_1} \in W \right]
\]
Pitfalls: preserving martingale condition

Martingale condition of Regular and Weighted Monte Carlo

![Graph](image)

Martingale condition

In between $t_1$ and $t_2$:

$$
\sum_{\{j/S_{j,t_1} \in W\}} \left( S_{j,t_2} \frac{F(S_{t_1})}{F(S_{t_2})} - S_{j,t_1} \right) \frac{p_j}{\sum_{\{h/S_{h,t_1} \in W\}} p_h} = 0
$$

Overweighted path

Without Margingale Constrains
To solve this problem, synthetic options are added to force the martingale condition:

- The price of the synthetic option is zero.
- The payoff for each path is the following:

\[
h_j = \left( S_{j,t_2} \frac{F(S_{t_1})}{F(S_{t_2})} - S_{j,t_1} \right) 1_{\{S_{j,t_1} \in W\}} \quad j = 1 \ldots v
\]
Case Study.

- A 6y geometric cliquet option on IBEX index:
  - $S = 10007$, $r = 2.95\%$, $q = 3\%$ valued on Jul 21st 2005.
  - $t_i = \text{Nov } 2^{nd}, 2005 \text{ to } 2010 \text{ and Oct } 25^{th}, 2011.$
  - $C = 1.1.$

\[
\Pi = \max \left( 0, \prod_{j=2}^{7} \left[ \min \left( \frac{S_{t_j}}{S_{t_{j-1}}}, C \right) \right] - 1 \right)
\]

- 20K Regular Monte Carlo ATMF paths where simulated.
- The smile and forward was calibrated at $t_i$ with errors less than a tenth of a basis point.
Case Study.

- The resulting prices are the following:

<table>
<thead>
<tr>
<th></th>
<th>ATMF</th>
<th>SML $t_1$ to $t_7$</th>
<th>SML $t_1$ to $t_7$ Mtgl</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.0332</td>
<td>0.0387</td>
<td>0.0547</td>
</tr>
</tbody>
</table>

![Graph showing Martingale condition of Regular and Weighted Monte Carlo](image)

Martingale condition of Regular and Weighted Monte Carlo

- Per unit window level
- Per unit martingale mismatch

Without Margingale Constrains

With Margingale Constrains

![Graph showing Martingale condition of Regular and Weighted Monte Carlo](image)

Martingale condition of Regular and Weighted Monte Carlo

- Per unit window level
- Per unit martingale mismatch

With Margingale Constrains
Case Study.

- Forcing the martingale condition reduces the probabilities of zigzagging paths which do not contribute to the payoff (they are capped).

Non-paying paths

Distribution of non-paying paths at $t_7$

Cumulative probability

Underlying in per unit of spot

Paying paths

Distribution of paying paths at $t_7$

Cumulative probability

Underlying in per unit of spot
Conclusions

- When WMC is calibrated to several maturities, the martingale condition might not be preserved.
  - This problem is solved by adding additional constraints.

- A new robust, fast and easy to implement calibration algorithm is proposed.
  - It can handle a considerable number of constraints (215 in the case study).

- This improved method has been applied to a well-known geometric cliquet option.
  - The price is 160bp higher in line with trader and market experience.