Numerical methods and volatility models for valuing cliquet options

Ludovic Goudenège

Décembre 2011

Premia 14

1 Model

Let S represent the price of the underlying asset. Its path is modelled by the following stochastic differential equation

$$\frac{\mathrm{d}S_t}{S_t} = \xi \mathrm{d}t + \sigma \mathrm{d}Z_t, \tag{1}$$

where ξ is the drift rate, σ is the volatility and $\mathrm{d}Z_t$ is the increment of a standard Gauss-Wiener process. Let $V(S,t)$ be the value of a European contract that depends on the underlying asset value S at time t with maturity T. Then

$$V_\tau = \frac{\sigma(\Gamma,S,t)^2 S_t^2}{2} \partial_{SS} V + r S_t \partial_S V - rV, \tag{2}$$

where $\tau = T - t$, $\Gamma = \partial_{SS}V$ and r is the risk free rate of interest. We have allowed the volatility to be a function of $\Gamma = \partial_{SS}V$, as well as the underlying asset price S and time t. In an uncertain volatility model, it is assumed that

$$\sigma_{\min} \leq \sigma \leq \sigma_{\max}. \tag{3}$$

The worst case value for an investor with a long position in the option is determined from the solution to equation 2 with $\sigma(\Gamma)$ given by

$$\sigma(\Gamma) = \begin{cases} \sigma_{\max} & \text{if } \Gamma < 0, \\ \sigma_{\min} & \text{if } \Gamma > 0. \end{cases} \tag{4}$$

Conversely, the best case value corresponds to

$$\sigma(\Gamma) = \begin{cases} \sigma_{\max} & \text{if } \Gamma > 0, \\ \sigma_{\min} & \text{if } \Gamma < 0. \end{cases} \tag{5}$$

A cliquet option offers a combination of floors and caps on returns on the underlying asset. Let $S(t_i)$ be the value of the underlying asset at time t_i. There are a total of N_{obs} observation times over the life of the contract. Define the return during the period $[t_{i-1}, t_i]$ to be

$$R_i = \frac{S(t_i) - S(t_{i-1})}{S(t_{i-1})}. \tag{6}$$

The payoff of a cliquet is

$$\text{Payoff} = \text{Notional} \times \max\left(F_g, \min\left(C_g, \sum_{i=1}^{N_{obs}} \max\left(F_l, \min\left(C_l, R_i\right)\right)\right)\right), \tag{7}$$

where C_l, F_l are local caps and floors placed on the individual returns, and C_g, F_g are a global cap and floor.

2 New variables

We introduce two new state variables: P corresponding to the asset price at the previous observation

$$P(t) = S(t_k), \qquad \text{for all } k \geq 1, \text{ for } t \in (t_k, t_{k+1}), \tag{8}$$

and Z, such that for $k \geq 1$, for $t \in (t_k, t_{k+1})$

$$Z(t) = \sum_{i=1}^{k} \max\left(F_l, \min\left(C_l, R_i\right)\right), \tag{9}$$

with $Z(t < t_1) = 0$. Consequently, the payoff at time $t = T$ becomes

$$\text{Payoff} = \text{Notional} \times \max\left(F_g, \min\left(C_g, Z\right)\right). \tag{10}$$

If t_k^-, t_k^+ are the times the instant before and after the k^{th} observation, then, no arbitrage considerations lead to the following jump conditions:

$$R = \frac{S(t_k) - P(t_k^-)}{P(t_k^-)} \tag{11}$$

$$R^* = \max(F_l, \min(C_l, R)) \tag{12}$$

$$Z(t_k^+) = Z(t_k^-) + R^* \tag{13}$$

$$P(t_k^+) = S(t_k) \tag{14}$$

$$V(S, t_k^-, P(t_k^-), Z(t_k^-) = V(S, t_k^+, P(t_k^+), Z(t_k^+), \tag{15}$$

Since the stochastic differential equation is independent of the new state variables (P, Z), we can discretize the state variables as

$$\{P_1, \ldots, P_j, \ldots, P_{j_{\max}}\} \text{ and } \{Z_1, \ldots, Z_k, \ldots, Z_{k_{\max}}\} \tag{16}$$

and solve a one dimensional PDE for each discrete value of (P_j, Z_k) between two observation dates. To move the solution across an observation date, we use the jump conditions. Notice that the jump conditions are undefined if $P = 0$ (which is in fact unattainable in general case). Therefore, it is important to discretize P such that $P_1 > 0$. We use points with an hyperbolic repartition between $S_{\min}$ and $S_{\max}$ (respectively the minimal and maximal value of the space domain of the underlying). These points are centered on S_0, the initial value of the underlying. The exact formula is

$$P_j = S_0 + \left(\operatorname{arctanh}\left(-0.9 + \frac{(j-1)(0.9+L)}{j_{\max}-1}\right) - \operatorname{arctanh}(-0.45 + 0.5L))\right) \times \frac{(S_{\min} - S_0)}{\operatorname{arctanh}(-0.9)} \tag{17}$$

where

$$L = \tanh\left(\frac{S_{\max} - S_0}{S_{\min} - S_0} \times \operatorname{arctanh}(-0.9)\right) \lessapprox 1 \tag{18}$$

such that

$$\begin{aligned} P_1 &= S_{\min} - (\operatorname{arctanh}(-0.45 + 0.5L))) \times \frac{(S_{\min} - S_0)}{\operatorname{arctanh}(-0.9)} \approx S_{\min}, \\ P_{(j_{\max}+1)/2} &= S_0, \\ P_{j_{\max}} &= S_{\max} - \left(\frac{\operatorname{arctanh}(-0.45 + 0.5L)}{\operatorname{arctanh}(-0.9)}\right) \times (S_{\min} - S_0) \approx S_{\max}. \end{aligned}$$

As discussed by Forsyth and al. (2002), it is generally necessary to carry out an interpolation operation to approximate the jump conditions at observation dates. If we assume that

$$\sigma = \sigma(S, t, P) = \sigma(\rho S, t, \rho P) \qquad \text{for all } \rho > 0, \tag{19}$$

then V is homogeneous of degree zero in (S, P)

$$V(S, t; P, Z) = V(\rho S, t; \rho P, Z) \quad \text{for all } \rho > 0, \tag{20}$$

which implies that we need only solve for one reference value $P = P^*$. This effectively reduces the dimensionality of the problem from three to two. This assumption seems somewhat peculiar, but has a modelling rationale explained in [1]. In this case, we speak about similarity reduction.

With regard to the mesh for the Z variable, there are no particular noteworthy issues, so we use a linear spacing between $\min(F_l, 0) \times N_{obs}$ and $C_l \times N_{obs}$ with $k_{\max}$ values. However, some issues arise in the construction of the P grid (for cases where no similarity reduction is available). Suppose that we use an S grid $(S_1, \ldots, S_{i_{\max}})$ and a P grid $(P_1, \ldots, P_{j_{\max}})$. In a Cartesian product $S \times P$ grid - with the same node spacing in the S and P directions - no interpolation is required during the application of the state variable updating rule $P_{t^+} = S$. The major advantage of this *repeated grid* is that no interpolation error is introduced at each observation date. But in Windcliff et al. (2001), it is shown that this type of grid results in poor convergence. Normally, we choose a fine node spacing near the initial asset price $S = S^*$, since this is the region of most interest. However, since the nodes $P = S$ for all values of S are required during the application of the jump condition, these values may have poor accuracy in areas where the S node spacing is large. it is therfore desirable to have a fine node spacing in the S direction for all nodes near the diagonal of the (S, P) grid. We have a prototype grid constructed in 17 such that S^* is in the discretization, so we use the following algorithm to construct the *scaled grid*:

```
Set $j_{\max} = i_{\max}$.
For $j = 1, \ldots, j_{\max}$
  Set $S_{i,j} = P_i P_j / S^*$, for all $i = 1, \ldots, i_{\max}$
```

Since there is an i^* such that $P_{i^*} = S^*$, then $S_{i^*,j} = P_j$. In other words, for each line of constant P_j, there is a node on the diagonal $S_{i,j} = P_j$. In the *scaled grid*, interpolation is required to satisfy the state variable updating rule. An obvious method is to linearly interpolate along the S axis and then along the P axis, which we refer to as the *xy* interpolation in the following. We use the following algorithm:

```
For the update between $S$ and $P$ (for all possible time $t$ and variable $Z$).
Find the index $s$ in the $S$ grid such that $S_s \leq S \leq S_{s+1}$.
Find the index $p$ in the $P$ grid such that $P_p \leq P \leq P_{p+1}$.
Define the variables:
$V^{S,P_p} = V(S_s, P_p) + \frac{V(S_{s+1}, P_p) - V(S_s, P_p)}{S_{s+1} - S_s}(S - S_s)$.
$V^{S,P_{p+1}} = V(S_s, P_{p+1}) + \frac{V(S_{s+1}, P_{p+1}) - V(S_s, P_{p+1})}{S_{s+1} - S_s}(S - S_s)$.
Create the interpolated value:
$V(S, P) = V^{S,P_p} + \frac{V^{S,P_{p+1}} - V^{S,P_p}}{P_{p+1} - P_p}(P - P_p)$.
```

An other method is given by the *diagonal* interpolation and the following algorithm:

```
For the update between $S$ and $P$ (for all possible time $t$ and variable $Z$).
Find the index $p$ in the $P$ grid such that $P_p \leq S \leq P_{p+1}$.
Create the interpolated value:
$V(S, P = S) = V(S = P_p, P_p) + \frac{V(P_{p+1}, P_{p+1}) - V(P_p, P_p)}{P_{p+1} - P_p}(S - P_p)$.
```

For each one dimensional partial differential equation, we must specified the boundary conditions on the domain $[S_{\min}, S_{\max}]$. At $S_{\min}$ and $S_{\max}$ we specified $\partial_{SS} V = 0$. In order to avoid non-desired effects, we may select a value of $S_{\min}$ on a coarse grid, and then reduce $S_{\min}$ as the grid is refined, so as to ensure the correct limiting behavior. This is ensure by the formula 17.

3 Boundary conditions and effect of grid interpolation

If the scaled one dimensional grids are constructed, then there will be situations where $S_{i,j} > P_{j_{\max}}$ or $S_{i,j} < P_1$. In these cases, our computational domain does not have sufficient data to allow interpolation of the state variable updating rule $P(t^+) = S$. If this happens, we assume that this data can be approximated by assuming that the similarity reduction is locally valid. This means that

$$\begin{aligned} V(S_{i,j}, t; P = S_{i,j}, Z) &\sim V(P_{j_{\max}}, t; P_{j_{\max}}, Z), \qquad && S_{i,j} > P_{j_{\max}} \\ &\sim V(P_1, t; P_1, Z), && S_{i,j} < P_1. \end{aligned}$$

In [1], the authors have studied the effect of grid interpolation. A series of refined grids was constructed where on each refinement the timestep size was halved, new nodes were inserted between each coarse grid node, and a new node was inserted in the S grid in $(0, S_1)$. The results in table **??** indicate that a *repeated grid* results in very poor convergence even though there is no interpolation error in the (S, P) directions when applying the jump conditions. Clearly, the use of the *scaled grid* satisfying conditions described before is very effective. Hence, we will use only this method.

Nodes	Timesteps	Scaled Grid	Scaled Grid	Repeated Grids
$(i_{\max}, j_{\max}, k_{\max})$		(diagonal interpolation)	(xy interpolation)	
$35 \times 35 \times 13$	40	.167847	.169728	.148230
$70 \times 70 \times 25$	80	.167229	.167837	.159672
$140 \times 140 \times 49$	160	.167046	.167211	.164720

Table 1: Value of a cliquet option given in [1].

4 Complete algorithm

- First we create a function in order to find the index i^* in the S grid such that, for a given point S_*, $S_{i^*} \leq S^* \leq S_{i^*+1}$. This function is extended to the P and Z grid.
- We define the *scaled grid* for S and P values and a uniform grid for Z values.
- We define the terminal value at time N_{obs}. It is $\max(F_g, \min(C_g, Z))$.
- We start a loop on observation dates (denoted as t below).

- We start a loop on the Z values (denoted Z_* below).
    - We start a loop on the P values (denoted P_* below).
        - We start a loop on the S values (denoted S_* below).
            - We define the variables R and R^* needed for the jump condition given in 11.
            - We find the indexes s, p and z such that $P_s \leq S_* \leq P_{s+1}$, $P_p \leq P_* \leq P_{p+1}$ and $Z_z \leq Z_* + R^* \leq Z_{z+1}$.
            - We use the diagonal interpolation to compute the $V(S_*, t; P_*, Z_z)$ and $V(S_*, t; P_*, Z_{z+1})$.
            - We do a linear interpolation between $V(S_*, t; P_*, Z_z)$ and $V(S_*, t; P_*, Z_{z+1})$ to compute $V(S_*, t; P_*, Z_*)$.
            - If $s = 0$ or $s = j_{\max}$, then we are on the boundary, and we use the extrapolation method according to a similarity reduction.
        - End of loop on the S values.
    - End of loop on the P values.
- End of loop on the Z values.
- We start a loop on the Z values (denoted Z_* below).

  - We start a loop on the P values (denoted P_* below).
    - We solve the partial differential equation for each P_* and Z_* values.
    - We use the boundary conditions at S_1 and $S_{\max}$.
  - End of loop on the P values.
- End of loop on the Z values.

- End of loop on the observation dates.
- Finally, we do an interpolation to find the value of the cliquet option at point $Z = 0$. So we find the indexes s, p and z such that $P_s \leq S_0 \leq P_{s+1}$, $P_p \leq S_0 \leq P_{p+1}$ and $Z_z \leq 0 \leq Z_{z+1}$. s and p are generally well defined (it is $(j_{\max} + 1)/2$, see 17). But the indexes z could be undefined, especially if the minimal value $Z_1 > 0$. In this case, we fix $z = 1$ and we do a linear interpolation with the values $V(s, 0; s, Z_1)$ and $V(s, 0; s, Z_2)$.

5 Conlusion

The discretely observed cliquet valuation problem reduces to solving a set of one dimensional partial differential equations embedded in a two or three dimensional space. These one dimensional problems exchange information through jump conditions at each sampling date. The type of grid used and interpolation method employed for enforcement of the jump conditions at observation dates has a very large impact on the convergence of the solution. Hence, we will use only the scaled grid and the diagonal interpolation, coupled with an extrapolation method for determining missing data at the extremes of the grid.

References

[1] H.A. Windcliff, P.A. Forsyth, and K.R. Vetzal. Numerical methods and volatility models for valuing cliquet options. *Applied Mathematical Finance*, 13, 2006. 3, 4