Source | Model | Option
| Model_Option | Help on tr methods | Archived Tests

tr_babbs_put

Input parameters:

- StepNumber N

Output parameters:

- Price
- Delta

This routine is taken from [1]. It gives a solution to the problem of pricing Floating Lookback Options with a one-dimensional tree method. The idea is to work with the underlying S_t as a numeraire. Indeed, if $M_t = max(M_0, \sup_{0\leq u\leq t} S_t)$, the dynamic of the process $Y_t = \frac{M_t}{S_t}$ is Markov, and the dynamic of the the same process within the CRR Tree is also Markov given by

$$\begin{array}{rcl} Y_{n+1} & = & uY_n \text{with proba.} p^* \\ & & min(dY_n, 1) \text{with proba. } 1 - p^* \end{array}$$

where $p^* = p * u$ and p is the usual CRR risk-neutral discounted probability. Indeed,

$$E_n(f(M_{n+1}, S_{n+1})) = E_n(S_{n+1} f(Y_{n+1}, 1)) = S_n E_n(\frac{S_{n+1}}{S_n} f(Y_{n+1}, 1))$$

so that in numeraire S

$$E_n(f(M_{n+1}, S_{n+1})) = E_n(\frac{S_{n+1}}{S_n} f(Y_{n+1}, 1))$$

whence the value of the probability p*.

/*Price, intrinsic value arrays*/

/*Up and Down factors*/
Exactly those of CRR.

/*Critical Index*/
This is the value `eta0` of n such that $y_0 d^n \geq 1 \geq y_0 d^{n+1}$. The originality of this tree is that a node at level $y_0 d^n$ has a down son on the level 1. Moreover the two sons of a node at level 1 are u and 1, so that the tree may not be recombining. In fact, there are two trees, the usual one (before hitting the level 1) and the tree wich comes from the level 1 nodes. So, the algorithm is as follows:
(1) We compute the price along the second tree to get the prices at the level 1 nodes (they are placed in the `Boundary` array).
(2) We compute the price along the usual tree with barrier condition. We proceed as in the Derman-Kani algorithm (Routine tr_dermankani.c),except that there is not here the problem of the location of the barrier.

/*Risk-Neutral Probability */
Probability in the S-numeraire world.

/*First Stage:Computation of price value along the line spot=maximum*/
Note that if eta0=N the level 1 is not reached so this stage is not performes.

/*Intrinsic value initialization*/
We start the indexing from the level 1 (Index=0). Therefore we cannot make use of the usual backward scheme because the value at Index=0 would be recomputed whereas its old value is still needed. So we introduce an auxiliary array `Q`. The indexing of the `Boundary` array is the current time in the tree. The indexes below eta0 will not be used.

/*Backward Resolution*/
We store the value of `P[0]` at time $eta0+2$(this is time1 in this tree) in `oldvalue` in order to compute the delta in case s=maximum.

/*Second Stage..*/
From this point on, this is exactly as in Derman-Kani algorithm (Routine tr_dermankani.c), except that the handling of the critical node is replaced by: `P[npoints]=pd*P[npoints]`pu*Boundary[N+1-i]+;

/*Price*/

/*Delta*/

The y-derivative of `P[0]` is computed as a finite-difference approximation, at time h, between the value of the price at node (h, uy_0) and at node (h, dy_0) or 1. Then the value of the delta is obtained after the formula:

$$\frac{\partial f(\frac{M}{s})}{\partial s} = f(\frac{M}{s}) - \frac{M}{s}\frac{\partial f(y)}{\partial y}$$

/*Memory Desallocation*/

References

[1] S.BABBS. Binomial valuation of lookback options. *working paper,Midland Global Markets London*, 1992. 1