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

tr_ritchken_downout

Input parameters:

- StepNumber N

Output parameters:

- Price
- Delta

This is taken from [1]. It is another answer to the issue of Barrier option pricing within a tree, like Derman-Kani algorithm (cf. Routine tr_dermankani.c). The algorithm is dicussed there: the idea is to take advantage of the degree of freedom of a trinomial tree to set a level of the tree exactly at the barrier. It works well except if the barrier is too close to the underlying $S0$: indeed the first level of the tree below $S0$ is $S0 * e^{-\lambda\sigma\sqrt{h}}$ with the constraint that λ should be greater or equal than 1. This leads to huge values of N if one tries to fit to $S0 * e^{-\lambda\sigma\sqrt{h}} = L$. This excessively close case is not handled by the algorithm at fixed N .
The convergence follows from Kushner's theorem there. The convergence rate seems to be an open question yet.

/*return values: 0-ok 1-unable to allocate memory 2-barrier l to close to s*/

/*Price, intrinsic value arrays*/
Since this is a flat tree we store the intrinsic values in an array `iv`

/*Number of down moves just before breaching the barrier*/
Here `eta0` is the integer such that $S0 * e^{-eta0\sigma\sqrt{h}} \geq L \geq S0 * e^{-(eta0+1)\sigma\sqrt{h}}$

/*The barrier is too close to S0-the algorithm fails*/
This in case `eta0=0`

/*Adjustment of lambda to set a level of the tree at the barrier*/ /*In case the step number is not sufficient, then take the default parameter*/
Indeed it may occur that the Barrier is too low. In such a case we keep the default value of λ given in parameter. This value is set in the function `MET(Init)`, cf Source.

/*Adjusted up and down moves*/
/*Discounted Ritchken Probabilities*/
The up and down factors and probabilities of the Kamrad-Rithcken algorithm. They are computed there.

/*Intrisic value initialization and terminal values*/
Alike a standard flat trinomial tree. We start the indexing from above, therefore the index of the critical level at the barrier is `N+eta0`

/*Backward Resolution*/
Two parts: first the barrier is active, then (backward) the tree is strictly above the barrier.
/*First part-the barrier is active*/
`npoints` stands for the index of the barrier from the upper node of the tree at each time step.
By contract the value of the option at this level is the rebate whence `P[npoints]=rebate`

/*Second part-the barrier is strictly below the tree*/
/*npoints stands now for the number of points to be computed at the current time*/
Here things go like a standard trinomial tree, cf for instance Routine tr_kamradritchken.c.

/*Delta*/

/*First time step*/

/*Price*/

*ptprice=P[0];

/*Memory Desallocation*/

References

[1] P.RITCHKEN. On pricing barrier options. *Journal Of Derivatives*, pages 19–28, Winter 95 1995. 1