

[Source](#) | [Model](#) | [Option](#)  
[Model\\_Option](#) | [Help on tr methods](#) | [Archived Tests](#)

## tr\_kamradritchken

Input parameters:

- StepNumber  $N$

Output parameters:

- Price
- Delta1
- Delta2

This is taken from [1]. It is a 5-node tree which is a two-dimensional version of the 3-node Kamrad-Ritchken (KR) tree. The space state is a product of the one-dimensional KR tree, with the same stretch parameter  $\lambda$  in the two directions (for some Barrier Options contracts it may be interesting to take different  $\lambda$ 's. The corresponding calculation of the risk-neutral probabilities should be done yet). This is a flat tree with  $(2N + 1)^2$  possible values of the underlying  $(S_1, S_2)$  throughout the option's life.

/\*Memory Allocation\*/

/\*Up and Down factors\*/

Here  $u1 = e^{\lambda\sigma_1\sqrt{h}}$ ,  $d1 = e^{-\lambda\sigma_1\sqrt{h}}$ ,  $u2 = e^{\lambda\sigma_2\sqrt{h}}$ ,  $d2 = e^{-\lambda\sigma_2\sqrt{h}}$ : in each direction the grid is that of a standard KR tree.

/\*Risk-Neutral Probabilities\*/

These are computed by matching the two first moment conditions with the same trick as in the one-dimensional KR tree (cf. [Routine tr\\_kamradritchken.c](#)) : the second moment condition is replaced by the equality of the second *moment* of the conditional random walk in the tree with the variance of the continuous limit logarithm of the Black-Scholes diffusion: the variances still match at order  $o(h)$  so that convergence follows from Kushner's theorem (cf [Convergence result for Tree methods in finance](#)) whereas the calculations are simpler.

/\*Terminal Values\*/

Since this is a flat tree we store the intrinsic values in an array as explained in [Routine tr\\_coxrossrubinstein.c](#).

/\*Backward Cycle\*/

Notice that the indexing of the price array  $P$  is relative to the lower-left corner of the state space values at a fixed time whereas the intrinsic value array indexing  $iv$  is absolute. This accounts for the shift  $k$  in the index in

$P[i][j] = \text{MAX}(iv[k+i][k+j], P[i][j]);$

/\*Deltas\*/

We call a function which computes the two deltas in a finite-difference manner in [bs2d\\_std2d.h](#).

/\*First Time Step\*/

/\*Price\*/

/\*Memory desallocation\*/

## References

- [1] B.KAMRAD P.RITCHKEN. Multinomial approximating models for options with k state variables. *Management Science*, 37:1640–1652, 1991.