

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

## tr\_producttr

Input parameters:

- StepNumber  $N$

Output parameters:

- Price
- Delta1
- Delta2

Examples of that kind of two-dimensional trees maybe found in [1]. This is a 4-node tree with a state space which is a product of two Euler one-dimensional trees (cf [Routine tr.euler.c](#)), which is a binomial tree where the driving Brownian motion of the model is replaced by the symmetric random walk in a natural manner. In particular the weight of the two sons nodes are  $\frac{1}{2}$ . In the no correlation case the product tree gives equal weight  $\frac{1}{4}$  to the 4 nodes. Otherwise the probabilities are chosen so that the two projected trees in each direction remain Euler trees, and so that Kushner's local consistency condition is in force (cf [Convergence result for Tree methods in finance](#)), which grants convergence. The calculations are detailed [there](#).

/\*2D Price Array allocation\*/

/\*Up and Down factors\*/

Here  $u1 = e^{\left(r - \text{diviv}_1 - \frac{\sigma_1^2}{2}\right)h + \sigma_1\sqrt{h}}$ ,  $d1 = e^{\left(r - \text{diviv}_1 - \frac{\sigma_1^2}{2}\right)h - \sigma_1\sqrt{h}}$ ,  
 $u2 = e^{\left(r - \text{diviv}_2 - \frac{\sigma_2^2}{2}\right)h + \sigma_2\sqrt{h}}$ ,  $d2 = e^{\left(r - \text{diviv}_2 - \frac{\sigma_2^2}{2}\right)h - \sigma_2\sqrt{h}}$ , in each direction the grid is that of a standard Euler tree. Notice that in general this is not a flat tree (ie  $u1 * d1 = 1$  and  $u2 * d2 = 1$  does not hold).

/\*Risk-Neutral probabilities\*/  
 These are discounted as you can see.

/\*Terminal prices\*/

/\*Backward scheme\*/

We try to minimize the times operation, whence

$P[i][j] = p_{uu} * (P[i][j] + P[i+1][j+1]) + p_{ud} * (P[i+1][j] + P[i][j+1])$

Notice also that we need to recompute the values of the underlying  $(S_1, S_2)$  at each node in order to compute the intrinsic value-also recomputed at each node, in order to handle the american case. This tree is therefore much costly in computation time than a flat one.

/\*Deltas\*/

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

/\*First Time Step\*/

/\*Price\*/

/\*2D Price Array deallocation\*/

## References

- [1] H.KUSHNER P.G.DUPUIS. *Numerical Methods for Stochastic Control Problems in Continous Time*. Springer-Verlag, 1992. 1