Project 3: Add Greeks to binomial tree engines

31/03/2022

Ben Fradj Alaeddine

Benslimane Hazim

Morilhat Germain

1 - The problem **2**

2- Solution **2**

2-1 Augment the size of the columns by 2 2

2-2 Adapting the index argument to the extra nodes 3

2-3 Functional Principle 3

2-3-1 EqualProbabilitiesBinomialTree 3

2-3-2 The different tree architecture 4

1 - The problem

The Delta and Gamma of an option can be calculated numerically as

$$\Delta(u) = \frac{P(u+\delta u) - P(u)}{\delta u}, \ \Gamma(u) = \frac{\Delta(u+\delta u) - \Delta(u)}{\delta u}$$

where P (u) is the price of the option as a function of the underlying value u and δu is a small perturbation of such value; the current Delta and Gamma are Δ(u0) and Γ(u0) where u0 is the current value (that is, at t = 0) of the underlying.

On a binomial tree as currently implemented, there's a single point at t = 0 as shown in figure 1(a). This makes it impossible to calculate quickly and accurately the Delta and Gamma of an option; we can either re-run the calculation with a perturbed value of the underlying asset (which is slow, since we need three full calculations for Delta and Gamma) or we can use points at the first time step after t = 0 to obtain option values for a perturbed underlying (which is not accurate, since the option value at those points would be taken at t 6 = 0; this is what the BinomialVanillaEngine class currently does).

The problem can be solved by having three points for t = 0 with the center point at u0, as shown in figure 1(b), so that we can obtain the option values for u + δu and u − δu to use in the formulas for Delta and Gamma. Modify the BinomialTree class accordingly, and modify the BinomialVanillaEngine class so that it calculates Delta and Gamma correctly. Do you expect the extra points at each step to have a noticeable effect on the performance of the engine? Why? Verify your intuition by timing the code before and after your changes.

Note: the tree is built based on log u instead of u, so you'll have two different δu for the points at the left and right side. Take this into account during the calculations.

2- Solution

2-1 Augment the size of the columns by 2

The BinomialTree is a base class for all types of trees. In order to augment the size of the tree we modified the size(i) function. At column i the number of nodes is no longer i+1 but i+3, the two additional nodes correspond to the price perturbations required to compute delta and gamma at time step i.

2-2 Adapting the index argument to the extra nodes

The computation of the underlying prices at node (i,index) depends on the values of both i and index where i is the column or the time step number and index is the number of the nodes located at the ith time step. For a given i,in the original configuration, the index goes from 0 to i+1. With the new configuration it goes now from 0 to i+2. Hence, after increasing the size of the tree by two nodes, the real index needed for the underlying calculation was shifted by 1. To get back the right underlying price, as it was given by the original structure, we had to shift back index value by 1 to get the original tree structure. Through all the derived binomial tree classes the underlying(i,index) function was modified to account for the aforementioned issue. We created a new_index=index–1 which substitutes the old index in calculating the underlying price.

2-3 Functional Principle

To see how it works, we will take a look at the ***EqualProbabilitiesTree*** exemple.

2-3-1 EqualProbabilitiesBinomialTree

First we will check the first step. If we take in consideration the different changes that we made, we get $u_i^{index} = x_0 e^{i.dx + newIndex.up}$.

We obtain with i = 0 :

$$u_0 = u_0^1 = x_0 \,;\, u_0^+ = u_0^2 = x_0 e^{2up\_} \,;\, u_0^- = u_0^0 = x_0 e^{-2up\_}$$

With this , we get :

$$\Delta = \frac{P(u_0^+) - P(u_0^-)}{u_0^+ - u_0^-} \,,\, \Gamma(u) = \frac{\Delta^+ - \Delta^-}{\frac{u_0^+ - u_0^-}{2}}$$

With :

$$\Delta^+ = \frac{P(u_0^+) - P(u_0)}{u_0^+ - u_0} \,,\, \Delta^- = \frac{P(u_0) - P(u_0^-)}{u_0 - u_0^-}$$

To test our changes, we took $u = 36$ for the new architecture on JarrowRudd tree, which is a derivative tree of the EqualProbabilitiesTree, we got a $\Delta = -0.84566$ which is almost equal to the analytical $\Delta = -0.85$ obtained via the pricing engine that we runned three times with three different initializations of the value u . The NPV= 4.205 was not affected by the new architecture.

2-3-2 The different tree architecture

For an American Option with $u = 36$, strike = 40 and a date of maturity 3 months after, we get:

Tree	Original NPV	Current NPV	Original Δ	Current Δ	Original Γ	Current Γ	Elapsed Time
JarrowRude	4.205	4.205	-0.849518	-0.84566	0.0720572	0.072163	0.000599
CoxRossRubinstein	4.20472	4.20472	-0.849587	-0.84679	0.0720719	0.072156	0.000546
AdditiveEQPBinominal	4.20559	4.20559	-0.849272	-0.84642	0.0720707	0.072175	0.000518
Trigeorgis	4.20473	4.20473	-0.849586	-0.84678	0.0720719	0.072156	0.00052
Tian	4.20488	4.20488	-0.849297	-0.84683	0.0721481	0.072078	0.003956
LeisenReimer	4.19732	4.19732	-0.850354	-0.85024	0.0727937	0.071623	0.003481
Joshi4	4.19732	4.19732	-0.850355	-0.85024	0.0727937	0.071623	0.003983

Although the NPV remained unchanged, we get a satisfying approximation of the original greeks, we also notice that the execution time was reduced significantly because we compute Δ and Γ at the same time by executing the rollback function only once.