Add Greeks to binomial tree engines

Paul Giraud, Kouamé Yao & Loïc Turounet
(Dated: March 21, 2022)

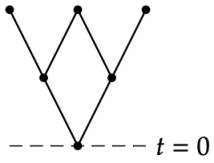
In this paper we will explore the QuantLib library by implementing it, but especially by looking at Greeks, how they are computed and how we can improve the current computation methods .[a]

I. Introduction

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

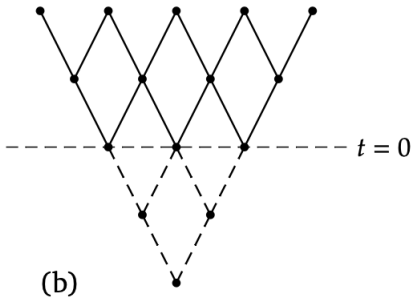
$$\Delta(u) = \frac{P(u+\delta u) - P(u)}{\delta u}, \Gamma(u) = \frac{\Delta u - \Delta(u-\delta u)}{\delta u} \tag{1}$$

where $P(u)$ is the price of the option as a function of the underlying value u and u is a small perturbation of such value; the current Delta and Gamma are $\Delta(u_0)$ and $\Gamma(u_0)$ where u_0 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 $t \neq 0$; 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 u_0, as shown in figure 2(b), so that we can obtain the option values for $u + \delta u$ and $u - \delta u$ 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 of 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 u for the points at the left and right side. Take this into account during the calculations. Hint: you don't need the part of the tree before $t = 0$ in figure 2(b).



(a)

FIG. 1: Tree with one node at t = 0



(b)

FIG. 2: Tree with three nodes at t = 0

II. Method

A. BinomialTree2

Hence, we have to change the size() method according to 2. Thus, the size no longer the position of the layer+1 but the position of the layer+3.

```
Size size(Size i) const {
    return i+3;
}
```

B. underlying price

According to the modification we have done, the number of node for a layer is no longer $index + 1$ with $index$ the index of the layer. With our changes it is now .Thus, we have an index shift of 1 with the old values that we used to calculate the values of our option. What used to be node 0 is now node 1, because we added one node below. In order not to change the way we calculate we must subtract 1 from our node indexes.

```
// new one
BigInteger j = 2*(BigInteger(index)-1) -
    BigInteger(i);
```

[a] Our code is available on https://github.com/paulgiraudIMT/IMT2022.

```
return this->x0_*std::exp(i*this->driftPerStep_
    + j*this->up_);
```

or :

```
// new one
return x0_ * std::pow(down_, Real(BigInteger(i)
    -(BigInteger(index)-1)))* std::pow(up_, Real
    (index)-1);
```

C. testing

Now we will want to test our modifications, for that we will have to change in *binomialengine* the rollback done and make a rollback at $t = 0$ by making sure that there are now 3 nodes. After doing this we will calculate our delta and gamma using the following formulas:

$$\Delta(u) = \frac{P(u_{up}) - P(u_{down})}{u_{up} - u_{down}} \tag{2}$$

$$\Gamma(u) = \frac{\Delta u_{up} - \Delta u_{down}}{u_{up} - u_{down}} \tag{3}$$

with :

$$\Delta u_{up} = \frac{P(u_{up}) - P(u_{middle})}{u_{up} - u_{middle}} \tag{4}$$

$$\Delta u_{down} = \frac{P(u_{middle}) - P(u_{down})}{u_{middle} - u_{down}} \tag{5}$$

We can now run our code with the changes and for example the it tree and check that our changes work correctly :

NPV	Delta	Gamma	Elapsed time
4.205	-0.846675	0.0721635	248 µs

III. Results and discussion

In the following table we find the values: Net Present Value, Delta, Gamma and Eclipsed Time for all the different tree types **before** our modifications.The calculations were done for an **American Option**, **Strike Price = 40** and **Spot price = 36** and for a maturity date **24 May 2022** (Today's date is 24 February 2022).

Tree	NPV	Delta	Gamma	Elapsed time
JarrowRudd	4.205	-0.849518	0.0720572	253 µs
CoxRossRubinstein	4.20472	-0.849587	0.0720719	173 µs
AddEQPBinTree	4.20559	-0.849272	0.0720707	166 µs
Trigeorgis	4.20473	-0.849586	0.0720719	163 µs
Tian	4.20488	-0.849297	0.0721481	331 µs
LeisenReimer	4.19732	-0.850354	0.0727937	331 µs
Joshi4	4.19732	-0.850355	0.0727937	416 µs

Then, our results **after** our modifications:

Tree	NPV	Delta	Gamma	Elapsed time
JarrowRudd	4.205	-0.846675	0.0721635	248 µs
CoxRossRubinstein	4.20472	-0.84679	0.0721567	141 µs
AddEQPBinTree	4.20559	-0.846429	0.0721755	143 µs
Trigeorgis	4.20473	-0.846789	0.0721567	141 µs
Tian	4.20488	-0.846834	0.0720787	342 µs
LeisenReimer	4.19732	-0.850244	0.0716231	342 µs
Joshi4	4.19732	-0.850244	0.0716231	338 µs

IV. Conclusion

To conclude, we can observe that the NPVs are identical and are therefore calculated in the right way. Furthermore, we have more accurate values for delta and gamma by using the given formulas 1. Moreover, we can see that globally the calculation times are shorter after our modifications because we have reduced our rollback in the *binomialengine* from 3 to only 1.

V. References

[1] THE BINOMIAL BLACK–SCHOLES MODEL AND THE GREEKS The Journal of Futures Markets, Vol. 22, No. 2, 143–153 (2002)