QuantLib Project 3

Add Greeks to binomial tree engines

Team : BRAHIMI Ali, GAUTIER Lucas, NADAT Hamzah, STUCKI Nicolas

Introduction :

Calculating Delta and Gamma has always been an issue, especially considering the importance of these two values. QuantLib's method currently implemented isn't the best and there are many ways to improve it. In order to achieve that, this project 3 focuses on two points which are the tree used for the underlying and the formulas of the Greeks.

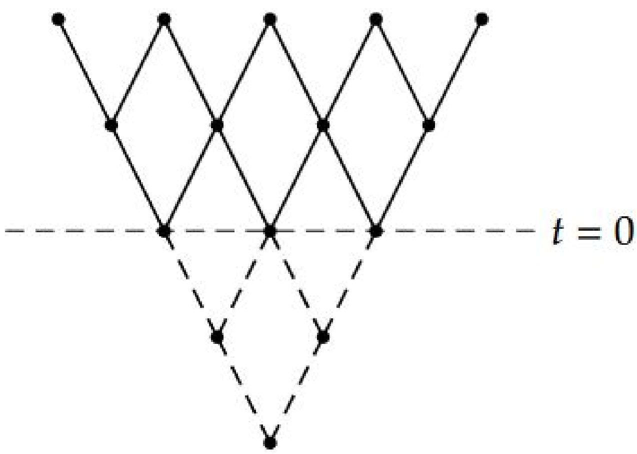
As it can be seen in QuantLib's code, points right after t=0 are used to calculate Greeks. Instead of that, modifying the tree with three points at the beginning will help to make a more honest approximation. Therefore, we will modify the BinomialTree class to have more nodes at the starting point and use them in BinomialEngine to improve the calculate() method. We used Taylor's work to get better formulas.

Binomial Tree :

At t = 0, we need 3 nodes. We need to modify BinomialTree template to have :

- size(-2) = 1
- size(-1) = 2
- size(0) = 3

Then size(i) = i + 3



We need that the result of underlying(0, 1) in the new tree be the same as the result of underlying(0,0) in the old tree. We modified underlying accordingly in each tree method.

<u>BinomialEngine :</u>

In order to calculate the greeks we rollback to t=0, thanks to the modification in BinomialTree we get 3 spot points and the associated option prices. Using equation (3), we calculate the gamma and then we use equation (1) to calculate the delta.

$$\Delta_u = \frac{P(u+d_1) - P(u)}{d_1} \cong P'(u) + \frac{d_1}{2} P''(u) \quad (1)$$

$$\Delta_d = \frac{P(u) - P(u-d_2)}{d_2} \cong P'(u) - \frac{d_2}{2} P''(u) \quad (2)$$

$$2 * \frac{\Delta_u - \Delta_d}{d1 + d_2} \cong P''(u) \quad (3)$$

<u>Tests :</u>

We've tested what we have implemented with various tree methods, volatility, risk free rate, time steps and dividend yield. The values of Delta and Gamma obtained were compared to Greeks calculated with Black-Scholes' model. An important point is that we've only used an European Call 1Y with fixed Strike (120) and Spot_0 (100). Before implementing our method, we assumed that the extra points would rise the computing time so we decided to time the code Before and After as it was advised.

The following table shows the results :

		Tests Results	Methods						
			Jarrow Rudd	Additive EQP	CRR	Trigeorgis	Joshi 4	Tian	Leisen Reimer
Set 1	Before	AD Delta	5.01E-05	3.70E-04	2.71E-06	3.61E-06	2.23E-04	9.89E-05	2.23E-04
		AD Gamma	5.61E-06	1.82E-06	3.77E-06	3.75E-06	1.28E-05	2.54E-06	1.28E-05
		Computing time (sec)	0.139	0.092	0.132	0.107	0.105	0.133	0.117
	After	AD Delta	3.93E-05	4.59E-04	6.38E-05	6.47E-05	4.04E-04	5.57E-05	4.04E-04
		AD Gamma	6.46E-06	1.39E-05	8.12E-06	8.13E-06	2.46E-06	7.92E-06	2.46E-06
		Computing time (sec)	0.131	0.106	0.093	0.089	0.121	0.105	0.098
Set 2	Before	AD Delta	4.25E-04	7.81E-04	1.38E-04	1.38E-04	6.80E-04	1.47E-04	6.80E-04
		AD Gamma	2.19E-05	2.91E-05	9.70E-06	9.69E-06	4.02E-05	1.05E-05	4.02E-05
		Computing time (sec)	0.014	0.027	0.026	0.024	0.025	0.013	0.016
	After	AD Delta	1.12E-04	2.42E-04	4.39E-04	4.39E-04	9.75E-04	2.34E-04	9.75E-04
		AD Gamma	5.16E-06	2.04E-06	1.67E-05	1.67E-05	2.83E-07	2.36E-05	2.79E-07
		Computing time (sec)	0.014	0.026	0.023	0.025	0.026	0.012	0.013

Table : Results

Note :
AD = Absolute Difference (comparison with Black-Scholes)
Green -> Better result
Set -> Parameters
Set 1 : (risk-free, dividend yield, time steps, volatility) = (8%, 4%, 800, 35%)
Set 2 : (risk-free, dividend yield, time steps, volatility) = (4%, 0%, 300, 25%)

Conclusion :

Overall the results are not great, even though for some tree methods we get better results. It may be because we have significant differences between d1 and d2 for multiplicative methods. Besides, comparing Set 1 and Set 2's results, we can observe that with the second set our method can be seen more positively.

An interesting observation is that the difference in computing time between Before and After isn't that big, but if it was used in a more realistic environment the little gaps would end up to be consequent. However, if we look at Set 1's results, we can see that After's computing takes more time. This is been more noticeable with a increase in the number of steps (time step). Indeed, we merely need to compare Set 1 and Set 2's computing times.

To conclude, our method allows us to have a better tree but the formulas still need improvement.