Fix oscillations in binomial tree engines Quantlib

SAY Ahoussi Armand   Sami BENABIDALLAH   Mohamed KAFFEL
Julien BEAUVOIS

February 2017

1 Introduction

Options prices calculates on trees with different time steps oscillate between two values depending on whether the number of time steps is even or odd. Among others, Chung and Shackleton argue that this can be helped by settings the option values at the penultimate nodes (i.e., those at the last time before maturity) to the analytic Black- Scholes values for a European option. This is acceptable because on the tree there's no exercise between the penultimate nodes and the last nodes (i.e., those at maturity). Our goal is to implement a solution that will fix those oscillations through the time steps so the BinomialVanillaEngine class will be modify to implement this pricing scheme.The Cox Ross Rubinstein tree will be used for test and validation during the implementation .

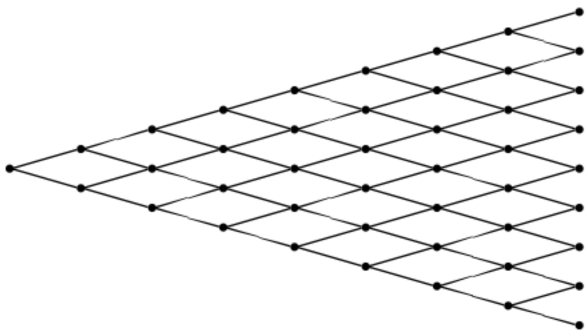

Figure 1: Quantlib Lattice data structure

2 Problem overview

In this section we are going to implement a simple solution that shows the option price that oscillate. To achieve our goal we provide the class diagram that explains the implementation. Note that some upper class (Observer , Observerable , StochasticProcess ....) are not shown to simplify the diagram and have a clear vision.

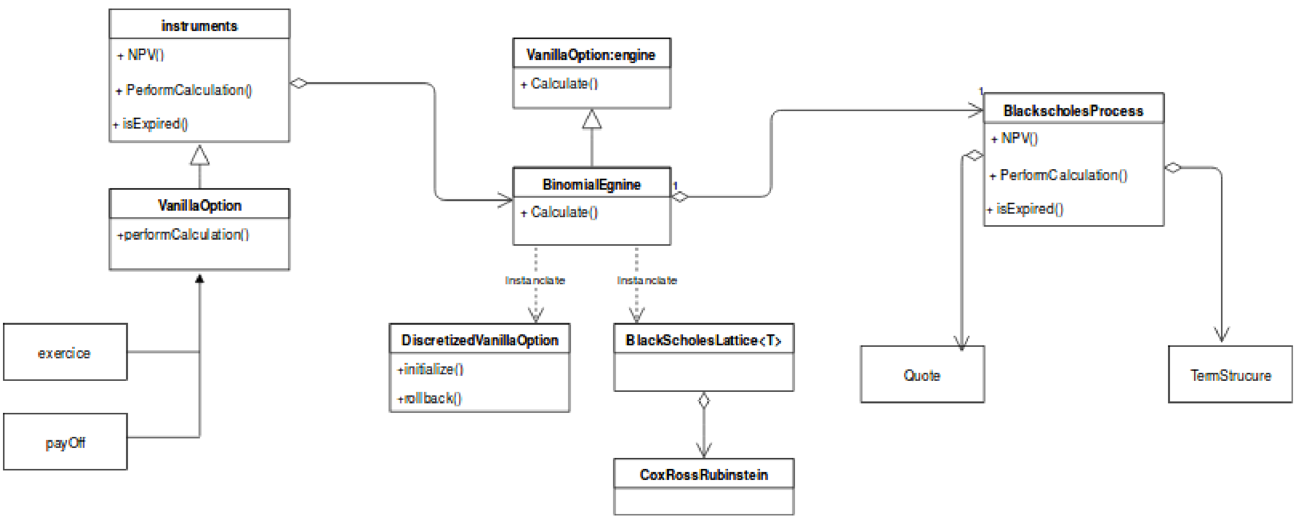

Figure 2: Implementation structure

The result of the option price with the different time steps is represented on the graphic below. For the simulation we've used as parameters the strike $K = 110$, the underlying price $S_0 = 100$, settlement date (01,March, 2017) the maturity date (27, February, 2018) the risk free Rate $r = 0.03$ the volatility $v = 0.2$ the dividend yield rate $q = 0$.We have used the Cox Ross Rubinstein tree for the implementation. The result of the implementation is analyzed between 20 to 300 time steps show that the price oscillates around the mean price 3.319.

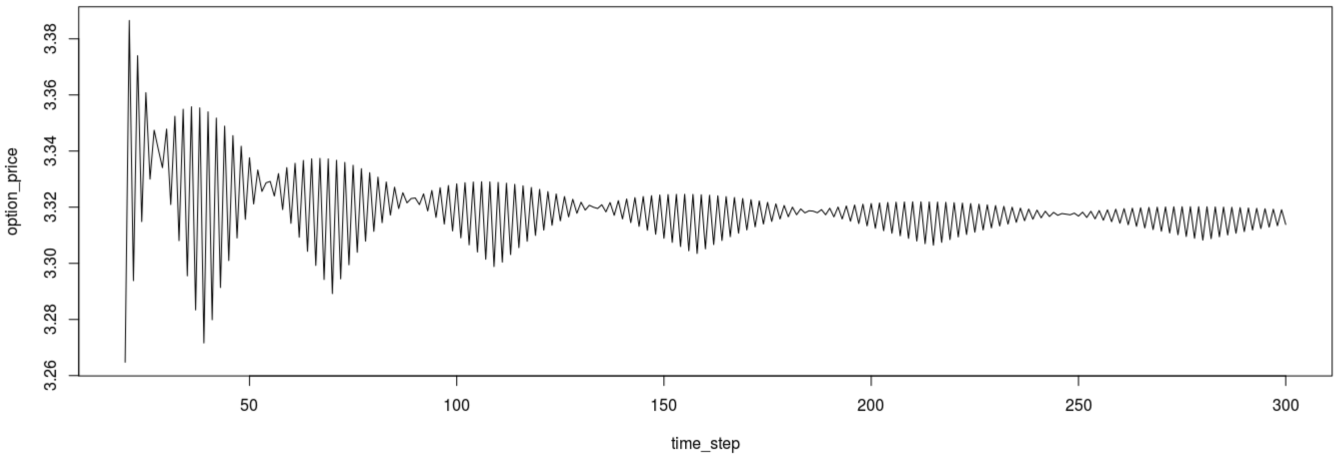

Figure 3: option price per time step

Now we have the proof that the price oscillate between 2 values we are going to implement a solution that fix those oscillations.

3 Optimized solution

In this section we will set up the optimized solution that fix the oscillations in the BinomialEngine class .To achieve this Goal we have redefine a new lattice class OptimizedLattice that will fix the issue.As suggested previously in the introduction the penultimate nodes will be replaced with the blackscholes Option price. The implementation diagram is below.

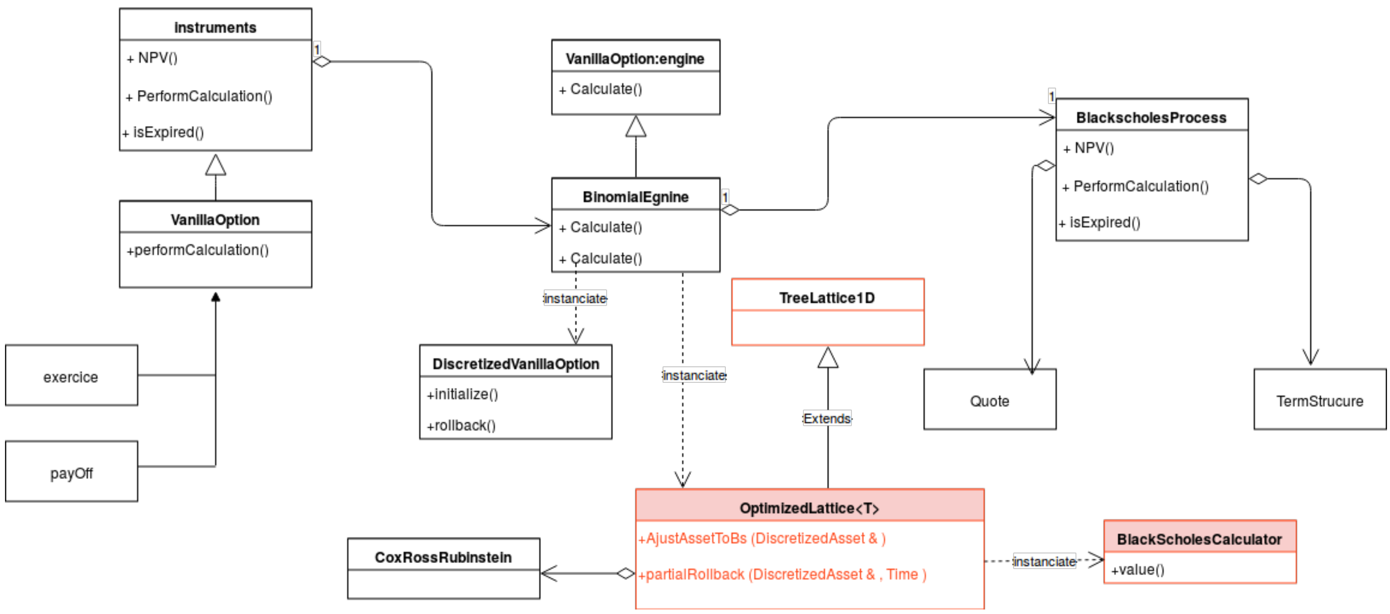

Figure 4: Optimized structure

The new class OptimizedLattice have one major function $AjustAssetToBs$ that modify the pricing scheme through the inherited method $partialRollback$. the $AjustAssetToBs$ function check if the asset time is greater than the time one step back before last time and perform the a *stepback* and ajust the penultimate nodes to the Black-Scholes analytic values and then ajust the asset time.

The results of the simulation are plotted on the graphic below but we have changed the maturity time to one month and the other parameters remain the same from the first simulation

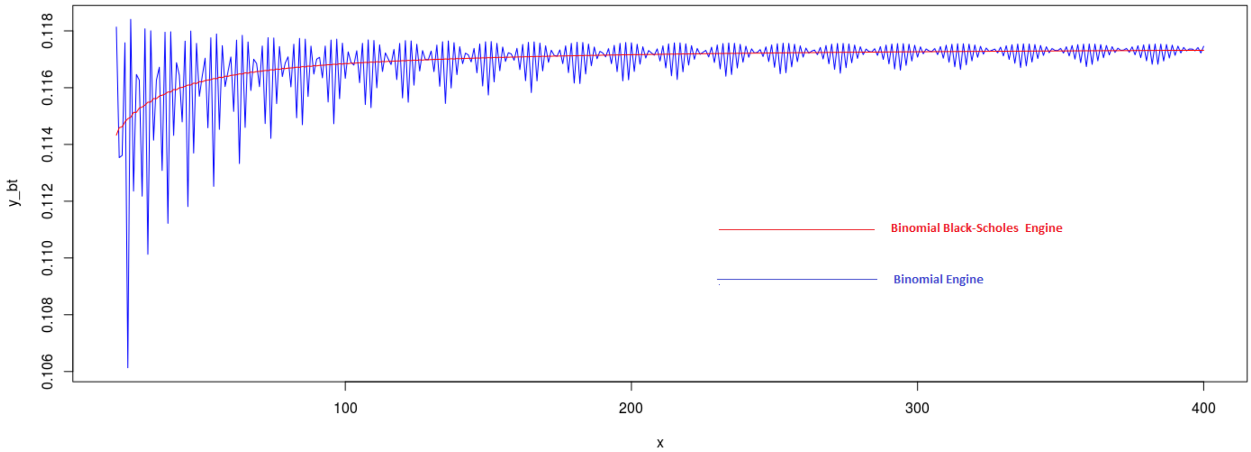

Figure 5: Comparaison between Binomial tree and the Binomial Black Scholes

You can see in red the Binomial Black Scholes Engine result that have a better performance that the binopmial Engine note that the convergence of the the price depends on the maturity. With a long maturity time you may have a convergence of your price after a lot of time steps.

4 Conclusion

This is an example of using the Quantlib library to solve option pricing issue.The performed simulations 1 and 2 showed the price that oscillate with the number of time steps provided but fixed with the implemented solution.For reminder the data has been analysed with R for a graphical overview of the problem. All commits and review during the project phases can be found on https://github.com/lballabio/IMT2017/pull/5.