**Delaneau Clément**

**Fock-Lapp Sébastien**

**MIF3**

**Report**

**Goal :**

C++ Implementation of an options pricer.

**Program’s layout:**

* We created a superclass “Option” which contains all the attributes used for the pricing with the different methods.
* derived classes of “Option”: “EuropeanOption”, “AmericanOption”, “AsianOption”, “DigitalOption”, “BinLattice”
* Main: We created a menu which enable the user to choose the option type and then display the results for the different methods required.

1. **European Option**
2. Black-Sholes (TD5 exercise 1)

We computed the Black-Sholes formula thanks to the normal distribution.

1. CRR (TD5 exercise 2)

We calculated the option value at each node from the end of the tree to the first node which is our result.

1. CRR Closed Formula (TD6 exercise 1)

We used a closed-Formula to calculate our option value. However, we encountered a problem with the function “factorial” which doesn’t work with N>20. That’s why we changed the N parameter of the European options e1 and e2 to N2=20 to use this method.

We obtain good results.

1. CRR display of the binomial tree (TD6 exercise 3)

We used a vector of vector. The first dimension of the vector corresponds to the step n of the tree and the second dimension corresponds to the node i of the step n.

So, lattice[n][i] (from the class BinLattice) enable us to reach the node i at the step n.

To display binomial tree, we used a double boucle (one for n and one for i) to access all the actualized spots.

The method “espace” enable us to display correctly the tree thanks to the use of a decreasing affine function which lower the size of spaces at each step n.

We can only display binomial tree for a little N because of the console size.

That’s why we changed the N parameter of the European options e1 and e2 to N3=10 to use this method

1. Monte Carlo (TD7 exercise 1)

We used Box-Muller to generate the random paths for the underlying. We then calculated the payoff for each path. Finally, we made the average of the payoff values to determine the option price.

1. **Digital Option (TD6 exercise 2)**

This option has a fixed payoff (which is 1 or 0 in our example) and so an option price between 0 and 1.

1. **American Option (TD 8 exercises 1 & 2)**

Unlike the European Option which can be used only at the maturity, we can apply American Option at any time before the expiration date.

We computed CRR method and displayed binomial tree (TD8 exercise 1). We observed American call price was quite similar of European call whereas American Put differs slightly.

We also computed the last method (TD8 exercise 2) which approximates Black-Sholes with the binomial tree. For this we only changed the formula of “u” and “d”.

1. **Arithmetic Asian Option (TD7 exercise 2)**

An Asian Option differs from European and American options in the calculation of the Payoff which is the average of the underlying prices over a given period.

To price this, we just changed the calculation of the payoff in our Monte Carlo program for European Option.

For each path, the payoff is the average of the underlying prices. We computed then two functions “AsianPayoffCall()” and “AsianPayoffPut()” that we injected in the parameter “payoff\_sum” in our Call and Put methods.

**Notes:**

* We often used the formula “S\*pow(u(),2i-n)” instead of “S\*pow(u(),i)\*pow(d(),n-i)” to alleviate the code. The two formulas are equivalent as d=1/u.
* We computed the “Greeks” (in EuropeanOption.cpp) thanks to the distribution function of the normal centered reduced law and the probability density function of the normal centered reduced law