Pricing vanilla options using PDEs

M203 C++ project

Soughati Kenza, Saulis Lukas, Garriga Maxime, Collin Thibault

1 Introduction

The aim of this project was to build a pricer of vanilla call option from the PDE of the derivatives payoff. Key implementations include matrix inversion algorithm and Crank-Nicholson scheme resolution.

To this end, we built three classes, alongside a **main** file (*algo.cpp*) to centralize all operations:

- *bs.cpp* whose aim is to price vanilla calls and puts following the **Black-Scholes** formula with dividends and repo, used as a benchmark for our pricing through the PDE.
- *matrix.cpp* which creates and acts on a matrix object, mainly to perform an inversion following a **Gauss-Jordan** elimination in the finite difference resolution scheme later.
- *pricer.cpp* which encompasses the general finite difference pricing method, using a **Crank-Nicholson** scheme to build the final price grid, as well as an extraction method to isolate the price corresponding to the given spot.

2 Discussions and Decisions

In the construction of the algorithm, several parameters and conditions were selected by default, and all are easily and freely changeable by the user (*Prof.*)

Table 1: Option pricing parameters

Parameter	Value
spot	85
strike	100
rate	7%
divs	3%
repo	0%
matu	1 year
vol	29%

Table 2: Finite difference resolution parameters

Parameter	Value
multiplier	4
timeGridSize	75
spotGridSize	60

Table 3: Finite difference conditions

Parameter	Value
lowerBoundary	0
upperBoundary	$e^{(rate-divs-repo)(matu-t)}spot$
terminalCondition	$(spot - strike)^+$

3 Results

Running our algorithm with these exact parameters yields the following results:

Table 4: Resolution output

Parameter	Value
solvingTime	4.93 mins
Black-Scholes call option price	5.63718
Finite-Difference call option price	5.6321

We can easily see that the prices are close to match. Let us approach the problem with different choices of discretization grids.

Table 5: Parameters for a second resolution

Parameter	Value
multiplier	4
timeGridSize	7
spotGridSize	7

Table 6: Second resolution output

Parameter	Value
solvingTime	0.08 mins
Black-Scholes call option price	5.63718
Finite-Difference call option price	5.34224

Table 7: Parameters for a third resolution

Parameter	Value
multiplier	4
timeGridSize	100
spotGridSize	75

Table 8: Third resolution output

Parameter	Value
solvingTime	9.12 mins
Black-Scholes call option price	5.63718
Finite-Difference call option price	5.58424

We observe convergence as expected with a growing number of steps. The behaviour of our algorithm is consistent with theory.

4 Conclusion

We were able to successfully price a European call option following a finite difference scheme based on the PDE resolution. Moreover, the convergence towards a baseline theoretical price reinforces our conviction that the scheme works well.

5 Remarks

Pricing long-dated options requires a higher degree of discretization to ensure precision. This comes at the cost of an exponential increase in the computational complexity, although it makes little sense to price those options are they tend to be illiquid.