Pricing Models: Assignment #1

N.Borovykh, E. van Raaij

Natalia.Borovykh@Rabobank.com
Erik.van.Raaij@Rabobank.com

TU Delft — February 16, 2024

Introduction

In this document the first assignment is explained high level. It also provides some clarity on the deliverables that have to be handed in. It's expected that you make the assignments in pairs of two using the *TUD01 EuropeanOption.zip* provided to you.

Info: The zip file consist of different folders:

- Code
- Documentation
- Input
- Output
- Solutions

The 'scenario' that generate the results of the assignment should be provided in the Solutions. The Code folder contains several files with functions that might be re-used. To have readable code these functions are not coded 'inline' but taken out. It's advised to scan first the folders and files to see what is available and to get a feel for the structure.

Info: Please use Anaconda distribution 2023.07 or higher which can be downloaded from

- https://www.anaconda.com/products/individual

The assignment will be checked either in Spyder or Visual Studio Code.

Deliverable 01: Please add your name and email address to the *Students.csv* file in the Input folder.

1 European Option

In the first assignment European options are priced assuming lognormal dynamics of the underlying. First, the analytical distribution is compared to the one simulated with Monte Carlo. Second, the price of an analytically calculated option is compared to the price of the same option calculated using Monte Carlo. Last, but not least, the prices of puts and calls have to be expressed in volatilities.

1.1 Lognormal distribution

Info: The details of the lognormal underlying are the following:

- spot price = 443
- volatility = 20%
- risk free rate = 6%
- expiry = 2 (year)

Simulate the values of the underlying at expiry with Monte Carlo (MC) and create an empirical distribution using the .hist function of matplotlib.pyplot package. As you've to reuse the path generator this should be coded in *paths.py*. The lognormal PDF should be added to the *distribution.py*. The MC simulation should be run with 2000 (2k) and 20k paths.

Deliverable 02: Create one figure with two subplots where the empirical distribution is compared to (the same) analytical distribution. Save the figure to the Output folder. In the first subplot use 2k paths. In the second subplot use 20k paths. The code to generate the figure should be put in *Solution11.py*

1.2 Call option prices

Info: See section 1.1. In addition,

- strike price = 443
- number of paths = [5k, 10k, 50k, 100k, 150k, 200k, 400k, 600k, 800k, 1000k]

Calculate the value of the European option analytically and several times with MC using the given number of paths. Therefore add an analytical pricer and a MC pricer to *payoffs.py*. The code should be generic such that put prices can be calculated as well. The MC pricer should return both the price and its standard error. Keep track how long it takes to generate the MC paths and the MC price for each number of paths.

Deliverable 03: Create one figure with two subplots. Show in the first plot the analytical price and the MC prices. In addition plot the standard error for each MC run. Show in the second subplot how long each MC run took. Save the figure to the Output folder. The data used to create the figures should be saved in CSV format to the output folder as well. The code to generate the figure and the output data should be put in *Solution12.py*

1.3 Implied Volatilities

Info: In the Input folder *OptionPrices.csv* is stored. This file contains:

- out of the money (OTM) European put and call prices
- spot values
- strike prices
- expiries
- risk free rates
- CPs (1 = call, -1 = put)

In addition a threshold is needed, which is set to 0.0001.

Calculate for each option/strike its corresponding implied volatility (IV). The functionality to calculate the IV should be added to *bsfunctions.py*. The function should also keep track of the number iterations it needs to find the IV given the above threshold.

Deliverable 04: Create one figure with two subplots. Show in the first plot both the prices and the volatilities. Plot in the second subplot per option/strike the number of iterations it took to converge. Save the figure to the Output folder. The data used to create the figures should be saved in CSV format to the Output folder as well. The code to generate the figure and the output data should be put in *Solution13.py*