QFGB8960 Advanced C++ for Finance
Homework 8

Spring 2025

Problem 1 (20 points) Correlation and Independence

The random variable Z is distributed according to the standard normal distribution $\mathcal{N}(0,1)$. Compute the correlation between Z and Z^2. Are the two variables independent? Discuss the result.

Problem 2 (40 points) Volatility Term Structure in MC

a) Modify the `BsMcPricer` class to take as input a smart pointer to a `VolatilityTermStructure`, a class that you coded in homework 6.

b) Modify the Python function `qf.euroBSMC` to accept either a numerical value or an object handle for the volatility term structure. When the input is numerical value, a constant volatility term structure should be created on the fly and passed to the `BsMcPricer` class.

c) Create a Python notebook. Construct of the “USD” yield curve and the “VOLTS” volatility term structure as named objects. For the yield curve use

Maturity	SpotRate
1/12	1.00%
1/4	2.00%
1/2	3.00%
3/4	3.50%
1	4.00%
2	4.75%
3	5.25%
5	6.00%

For the volatility term structure use

Maturity	SpotRate
1/12	25%
1/4	26%
1/2	27%
3/4	28%
1	30%
2	35%
3	40%

Price using MC pricing an ATM one year put, using the yield curve and volatility handles from above. Set the spot and the strike to 100 and the div. yield to 2%. Use the Euler path generator and 200,000 paths. Price the put in two ways:

1) with the volatility term structure and

2) with constant vol of 30% (the one year value). Assert that the results are the same.

Compare the MC results with the exact answer from the Black-Scholes formula (`qf.euroBS` function).

Problem 3 (40 points) Digital Call Convergence Graph

a) Create a class `DigitalCallPut` modeling the European digital (cash or nothing) payoff in the file `qflib/products/digitalcallput.hpp`.

b) Implement a Python callable function `qf.digiBSMC` that has the same interface as the existing `qf.euroBSMC`.

c) On the same Python notebook, create a yield curve with 4% flat spot rate. Set spot to 100, expiration to one year, dividend yield to 0 and volatility to 30%.
Create a convergence graph for two digital calls, one with strike 100 (ATM) and one with strike 120 (OTM), by running `qf.digiBSMC` with npaths = 100, 200, 400, 800, 1600, 3200,...204800.

Plot the price and the standard error of the digital call as a function of the number of paths. Use a logarithmic scale for the number of paths axis. Compare the MC price against the analytical answer from `qf.digiBS`.

Create a separate convergence graph for each strike.