Advanced Derivatives Coursework
ALL QUESTIONS

M.Sc. in Financial Engineering
Master Level M2 - Academic Year 2024-25
Professor: Dr. D. O'Kane

Deadline:

9AM Monday 23rd December 2024

Read the following instructions carefully

- This coursework forms 30% of your overall course grade.
- Groups must be of 3-4 persons.
- Email me a ZIP file to dominic.okane@edhec.edu with the title ADVANCED_DERIVATIVES_COURSEWORK_XXXXX_YYYYY_ZZZZZ where the XXXXX, YYYYY and ZZZZZ are the student ID's of the students in the group.
- The ZIP file should contain ONE notebooks, If you want to work on separate notebooks and then merge them at the end then here is an example of how to do this in python `https://stackoverflow.com/questions/33957418/merging-two-noteboooks-into-one-in-jupyter-ipython`.
- The notebook should contains all of the answers to questions 1-8 in this coursework. Make sure it is clearly divided into each question. Make it very clear to understand and ensure that you comment on what you did and why.

1. This question is about Black-Scholes-Merton Hedging.

(a) Install financepy and for speed issues, in the following, call directly into the model library rather than go via the EquityOptions class. Here is some example code:

```
v0 = bs_value(S[0], T, K, r, q, sigma, OptionTypes.EUROPEAN_CALL.value)
delta[0] = bs_delta(S[0], T, K, r, q, sigma, OptionTypes.EUROPEAN_CALL.value)
```

This will be about 60 times faster than going via pure Python code.

(b) In Python in a notebook, write a function called **OptionSim** that simulates the delta hedging of a European call option from trade date until expiry. It should use the function in (a) for calculating the option price and delta. The function inputs must include the option strike K, spot price of the stock S, risk-free rate r, the stock price drift μ (we do not assume that the stock grows at r), volatility σ and years to expiry T. The other input must be the hedging frequency per year N.

The dynamics of the stock price should be assumed to be lognormal with a drift μ (hedging does not have to set equal $\mu = r$ as the true stock price evolution is not risk-neutral) and a volatility σ_{Real}. The output of your function should be a tuple that has 6 elements:

- The terminal stock price $S(T)$ in the simulation path
- The call option payoff
- The number of shares held in the hedging portfolio
- The cash balance in the hedging portfolio
- The total value of the shares and cash held
- The replicating error which is the difference between the total value of the hedging portfolio and the option payoff.

Make sure your code is clear and well-commented with good variable names.

(c) Write another function that calls the previous OptionSim function and which can then be used to calculate the hedging error over 10,000 different paths. You must provide a clear and easy-to-understand listing of your code in the answer.

(d) Consider a call option with $S(0) = 100$, $K = 100$, $r = 5\%$, $T = 1.0$ and $\sigma = 20\%$. Assume here that $\mu = 5\%$. For this option, make a scatterplot of the hedging error (y-axis) versus the terminal stock price (x-axis) for $N = 12$ (monthly), $N = 52$ (weekly) and $N = 252$ (daily). Use different symbols or colours to distinguish the points.

(e) For each value of N also calculate the mean and variance of this option hedging error over 10,000 different paths. You can use this to generate the answers to the remaining parts of this question. Present this in a simple table format.

(f) For the same call option, calculate the mean absolute error value and the variance of the hedging error for $\mu = 2.5\%, 5.0\%, 7.5\%, 10\%$ by sampling $10,000$ hedging paths using $N = 52$. Show the results in a table. What does this tell you ? Does the value of the drift change the hedging by a little or a lot ?

2. This question is about Transaction Costs.

(a) Starting with the code from Question 1, amend your code to have a round trip bid-ask transaction cost of $\phi\%$ for each share purchase and sale (you buy at a price of $S \cdot (1 + \phi/2)$ and sell at a price of $S \cdot (1 - \phi/2)$). The transaction costs of each trade will be taken out of the cash account.

(b) Calculate the mean hedging error value for an annual heding frequency of (i) $N = 12, 52, 250$ with $\phi = 0.5\%, 1.0\%, 2.0\%$ by sampling across $P = 1,000$ hedging paths. In your simulations set the stock drift $\mu = r$. Show the results in a table. Explain your results.

(c) Leland (Journal of Finance, 1985)[1] showed that the cost of hedging taking into account hedging costs is equivalent to adjusting the volatility as follows

$$\sigma^{Adjusted} = \sigma \left(1 + \sqrt{\frac{2}{\pi}} \frac{\phi}{\sigma} \sqrt{N} \right)^{1/2} \tag{1}$$

where ϕ is the round-trip transaction cost as a percentage of the stock price, σ is the stock volatility and N is the annual hedging frequency. Compare the price increase you obtained with the prices using this adjusted volatility in Black-Scholes. Is Leland's model accurate ?

[1] `https://pages.stern.nyu.edu/~lpederse/courses/LAP/papers/Derivatives/Leland85.pdf`.

3. This question is about determining the implied density of the terminal stock price from the volatility skew.

(a) Suppose that we have managed to fit the 1-year volatility smile of the equity option market using a function

$$\sigma(x) = ax^2 + bx + c$$

where x is the "moneyness" ($x = K/S(0)$) and the initial stock price $S(0) = 100$ and $a = 0.025$, $b = -0.225$ and $c = 0.50$. Build a Python function that extracts the market-implied distribution of $S(T)$ at a 1-year horizon from $\sigma(x)$ using the Breedon-Litzenberger formula we derived in class. You should use FinancePy to calculate the option prices.

(b) A digital call option pays \$1 if $S(T) > K$ and zero otherwise. Using the probability density function implied by this volatility smile, calculate a table of prices of the 1-year European digital call option with strikes at $K = 60, 80, 100, 120, 140$. Assume that $r = 5\%$ and $q = 0.0\%$.

(c) Price the same set of digital call options using the Black-Scholes pricing formula for digital call options in the lecture notes or use FinancePy.

(d) Explain why the results of (b) and (c) do not agree with each other and explain which prices you think are more correct.

(e) Calculate the value of a put option with strike \$100 which only pays out if the stock falls below \$60 at expiry (this is a European and not a path dependent option). Explain carefully and clearly how this was done.

4. This question is about modelling the volatility skew.

(a) We wish to capture the volatility skew to price equity options using the CEV model with dynamics

$$\frac{dS(t)}{S(t)} = (r - q)dt + \sigma(S(t)/S(0))^{\beta} dW_t$$

Write a Monte Carlo simulation to price a T-year call or put option in Python using this model. Include your code listing in the pdf file.

In your simulation you will need to step through time and I suggest you use $dt = 0.02$. Use Numpy and Numba to make it as fast as possible. You should also use antithetic variables to improve the Monte Carlo convergence. Note that we recover Black Scholes if we set if $\beta = 0$. This is a useful test to check your code.

(b) Using the CEV model price a 6-month call option where $S(0) = 100$, $r = 5\%$ and $\sigma = 20\%$ for $K = 80$ to $K = 120$ in steps of 5 dollars. Do this for $\beta = -0.5, -0.25, 0.0, 0.25, 0.5$. Use $P = 10,000$ paths to ensure reasonable accuracy. This may take some time. Use numba if you can to speed up the code.

(c) Using your FinancePy implied volatility function (you need to create an EquityVanillaOption object which has an implied volatility function), convert the prices calculated in (c) to implied volatilities to show the model-implied skew. Plot the skew as a function of the option moneyness for each of the values of β. Discuss your results.

(d) For each value of K and β calculate the corresponding option delta in the CEV model by bumping the input stock price by 0.1 and recalculating the option price. Make sure you use the same random number seed when doing the calculations as you will need to use the same sequence of random numbers for S and $S + 0.1$. Compare the deltas calculated to the Black-Scholes model delta and discuss the intuition behind your results.

5. This question is about Variance Swaps.

(a) In this question you simulate a lognormal stock price process and simulate the dynamic hedging of a 1 year Variance swap

(b) Assume daily hedging for a process with $S(0) = 100$, a volatility $\sigma = 0.30$ and a growth rate of $\mu = 0.05$. Assume 252 business days a year.

(c) Calculate the net proceeds of the dynamic hedging strategy of investing $2/T$ dollars per day in the stock.

(d) Take into account the cost and payoff of the log contract.

(e) Simulate 10,000 paths of the stock price and use them to compute a histogram of the final P&L of a variance swap that has been hedged dynamically.

6. This question is about Swap and Swaption Pricing.

(a) Using Python, write a function to reproduce as closely as possible the Excel swap pricing grid sheet that we saw in class and use it to generate the swap curve discount factors from a swap curve with semi-annual coupons using swap maturities of 1, 2, 3, 5, 7 and 10 years. The function should take in the swap rates as inputs and output the discount factors.

(b) Using a value date for the swap curve of December 7 2024 and use the following swap curve: 1Y 2.3% 2Y 2.6% 3Y 2.86% 5Y 3.03% 7Y 3.18% 10Y 3.25% to build the curve. Extract the discount factors from the curve and demonstrate that all of the swaps reprice to zero upfront.

(c) Calculate the forward rate for a 1x7 year swap with semi-annual fixed leg payments. Show the value for $Z(T_1), Z(T_2)$ and the swap PV01.

(d) Using Black's model, value a 1x7 Payer and a 1x7 Receiver swaption with a strike 2.15%. Assume a swaption volatility of 20%. Explain the prices.

(e) Calculate the change in value for the payer and receiver swaption with respect to a 1bp change in the 1Y, 3Y, 5Y, 7Y and 10Y swaptions. Explain how these swaptions would be hedged ? Show the results in a table. Do not do any further calculations.

7. This question is about CDS Valuation and Risk.

(a) In Python, write a function to replicate the Excel CDS valuation spreadsheet - specifically the lower part with the dates from 04-Feb-11 to 20 Mar 2016 and all of the associated columns - I showed you in class in a dataframe. The function will take the hazard rates in as inputs and also the recovery rate. You should also make it possibe to change the value date so that a new schecule of dates is used. Use Financepy's schedule or pandas's date functions to generate the payment dates out 5 years to the next IMM date. The output should be the term structure of CDS spreads. Ensure that you can reproduce all of the outputs.

(b) Set today (value date) to be the 8th December 2022 - the code should automatically generate the dates out to 20 December 2027 - and build a CDS curve with the following points: 1Y - 65bp, 2Y - 69 bp, 3Y - 75bp, 4Y - 80bp, 5Y - 83bp. Use a recovery rate assumption of 40%. What values do you get for the term structure of hazard rates and the survival probabilities. You will either need to adjust these by hand or use a solver to find the hazard rates that minimise the sum of the squared differences between the model and the market.

(c) Determine the market spread of a 3.5 Year CDS that matures on the 20 June 2026. Explain how it is determined.

(d) Calculate the value of an existing long protection CDS contract traded with a contractual spread of 115bp with a maturity of 20 June 2026 and a notional of $10m.

(e) Calculate the value of the contract for $R = 0\%, 10\%, 20\%, 30\%$. Are the changes significant or not ? Can you explain why ?

(f) Calculate the change in the CDS value to a 1bp increase in each of the 1Y, 2Y, 3Y, 4Y and 5Y CDS market rates. Show the results in a table. Explain what you find and how a dealer would hedge this 3.5 year CDS trade. Do not do any further calculations.

8. This question is about using Bloomberg to value derivatives. I am happy for you to select a derivative product - it must be something we have studied in this course. You don't get bonus points for complexity so perhaps avoid Bermudan swaptions! **However, it should be more than a call or a put option.**

(a) Find the product on BBG and understand its terms and pricing inputs. This can include the types of discount curves and the choice of models. Use Bloomberg help pages to get any information you may need.

(b) Choose the contract details of the one you wish to analyse. Set them out clearly in your notebook. Please make them reasonable and realistic - e.g. at the money, less than 5-years to maturity. State them clearly.

(c) Using BBG, value the contract using current market conditions. Using screenshots, record all of the inputs (curves, rates, prices, model) and valuation and also the risk measures given by BBG. Paste these screenshots into your notebook.

(d) Repeat valuation using FinancePy. There are example notebooks in the github project that you can use to begin. Also try to reproduce the risk measures shown. Try to explain any differences.