FINM 32000 Final Exam

University of Chicago

March 16, 2022

Open book, open notes, open computers. No collaboration. Cite all sources.

Problem 1

Today is March 14, 2022. A trader known as Gigantic Rebirth has entered into a \$10 million bet with Do Kwon (co-founder of the Terra protocol), that the price of the LUNA coin in 1 year will be lower than its price today.

(a) Write Python code to calculate the implied volatility of this transaction.

(b) Is Gigantic Rebirth long or short vol here? (Is the vega of his bet positive or negative?)

Assumptions:

Ignore the risk from the deposits (\$20 million) being held in escrow by a "trusted" third party, Cobie, who has threatened to steal the prize fund.

Ignore the distinction between USDC and USDT (one party paid in USDC, and the other paid in USDT; both are stablecoins pegged to the US Dollar). Pretend that both are just USD (\$).

Model this transaction from Gigantic Rebirth's perspective, as paying \$10 million today and receiving either \$0 or \$20 million in 1 year, depending on whether the LUNA price in 1 year is, respectively, above (payout \$0), or below (payout \$20M) the LUNA price today.

Assume that LUNA price follows geometric Brownian motion starting at \$88 today. Assume that the risk-free interest rate on USD is 3%. Assume that LUNA pays zero dividend (ignoring the fact that LUNA can be lent or staked, generating, in effect, a positive dividend or repo rate). As always, assume frictionless markets and no-arbitrage unless otherwise directed.

You are to write Python code to find the volatility parameter in the geometric Brownian motion, such that the payout (\$0 or \$20M) has value \$10M today. See the `ipynb` file.

Hint: The implied volatility function from your homework cannot be directly used here, because it assumes a plain vanilla call option, which this bet is not.

Problem 1, TLDR version

Let S follow geometric Brownian motion with $S_0 = 88$. The interest rate is 0.03. No dividends.

A contract which pays 1 dollar at time 1 if $S_1 < 88$, or 0 dollars otherwise, has time-0 price 0.5.

(a) Write Python code to calculate the volatility of the Geometric Brownian motion.

(b) Is this contract long or short vol, meaning: is its vega positive or negative?

Hint: The implied volatility function from your homework cannot be directly used here, because it assumes a plain vanilla call option, which this contract is not.

Problem 2

You have a deck of 3 cards labeled with two negative numbers and one positive number:

$$-25, -10, +20$$

randomly shuffled with all permutations of the 3 cards equally likely.

At each time $n = 1, 2, 3$ you reveal the nth card, without replacement. For every card that you reveal, you receive the amount of dollars shown on the card.

So, for example, if you choose to see the first card, and it turns out to be -25, then your total profit at time 1 will be -25 dollars. Then, if you choose to see a second card, and it turns out to be $+20$, then your total profit at time 2 will be -5 dollars. If you stop playing at that time, then your final total profit is -5 dollars.

You may stop playing the game at any stopping time that you choose, taking values in $\{0, 1, 2, 3\}$. Interest rates are zero. Build a tree to answer the following questions.

(a) What is the (time-0, before any cards are revealed) expectation of your final total profit, using the strategy that optimizes expected final total profit?

(b) Suppose that after playing one turn, the first card is revealed to be -10. In that case, what is the (time-1, after the first card is revealed) expectation of your optimized final total profit? In that case, is it optimal to stop at time 1?

Problem 3

Assume zero interest rates. In (a,b), you may, but are not required to, use Python. In any case, please show your calculations.

(a) Suppose that a non-dividend-paying stock has dynamics

$$\mathrm{d}S_t = \sigma(t) S_t \mathrm{d}W_t$$

where W is Brownian motion under risk-neutral probabilities, and where the time-dependent but *non-random* instantaneous or local volatility function $\sigma : [0, 0.25] \to \mathbb{R}$ is a step function,

$$\sigma(t) = \begin{cases} 0.6 & \text{for } t \in [0, 0.20] \\ 0.43 & \text{for } t \in (0.20, 0.25] \end{cases}$$

Calculate the implied volatility of call options expiring at time $T = 0.25$. Your final answer should be a number.

(b) For some fixed S_0 (which you don't need to know) and some call option with strike K (which you don't need to know) and expiration $T = 0.25$, let C denote the time-0 Black-Scholes call price as a function of volatility. Let Vega denote the first derivative of C and let Volga denote the second derivative of C.

Assume that $C(0.6) = 1.42$ and $\text{Vega}(0.6) = 9.43$ and $\text{Volga}(0.6) = 29.3$.

Using a second-order Taylor expansion of C (including zeroth-order, first-order, and second-order terms), approximate the time-0 price of the K-strike T-expiry call option, on a stock following the dynamics in (a). Your final answer should be a number.

(c) This question is separate from (a,b). The assumptions of (a,b) do not apply here.

Assuming zero interest rates, consider a call option with $K = S_0$ (so the call is at-the-money, which was not assumed in (a,b)), and general expiration $T > 0$ (so we do not assume $T = 0.25$ in this part).

Let C denote the time-0 Black-Scholes call price as a function of volatility, and let Vega denote the first derivative of C.

Find $C(0)$ and $\text{Vega}(0)$, which may depend on S_0 and/or T. Use them to find a first-order (including zeroth and first order terms) Taylor approximation of the call price $C(\sigma)$ for small σ. Your final answer will be a formula.

Hint: You may use the fact that, for $\sigma \geqslant 0$,

$$C(\sigma) = S_0(2N(\sigma\sqrt{T}/2) - 1)$$

where the function N is the standard normal CDF.

Problem 4

Let S be the price of a non-dividend paying stock which follows Geometric Brownian motion with volatility 70% and $S_0 = 10$. The interest rate is 0.02.

Using Monte Carlo, find the time 0 price of a contract which pays at time 1

$$\begin{cases} (S_1 - 11)^+ & \text{if } S_1 > 14 \text{ or } S_{0.5} > 12 \\ (S_1 - 10)^+ & \text{otherwise} \end{cases}$$

where $S_{0.5}$ denotes S at time 0.5.

Each simulated payoff should use no more than 2 pseudo-random normals. If you wish to use conditional Monte Carlo (which is not required), then only 1 pseudo-random normal is needed per simulation. Ordinary Monte Carlo using 2 pseudo-random normals per simulation is fine.

Do not use numerical integration (quadrature).

Complete the code in the `ipynb` file, and report a price and standard error (estimated standard deviation of your estimate).