FE 621 homework 1

Kaiwen Fan

Part 1 Data gathering component

Two primary tickers were selected this project, SPY (S&P 500 ETF) and TSLA (Tesla, Inc.).

To ensure the theoretical model aligned with the current economic reality, I integrated two external benchmarks: Risk-Free Rate (r) and Market Volatility Index (VIX).

The data was downloaded using python yfinance API, which interfaces with Yahoo Finance to provide equity prices and option chains. The process was executed across two consecutive trading days [Day 1(02/11/26) and Day 2(02/12/26)] to facilitate the predictive analysis required in later sections. The final cleaned datasets were saved to DATA_FE621_DAY1.csv and DATA_FE621_DAY2.csv.

Description of Symbols and Market Instruments

SPY (S&P 500 ETF Trust):

The SPY is an Exchange-Traded Fund (ETF) that tracks the S&P 500 Index. An ETF is a type of investment fund and exchange-traded product that is traded on stock exchanges, much like individual stocks. The purpose of SPY is to provide investors with exposure to the 500 largest cap-weighted companies in the U.S. across all sectors. It is one of the most liquid financial instruments in the world, making its option chain ideal for Black-Scholes analysis due to narrow bid-ask spreads and high volume.

^VIX (CBOE Volatility Index):

The VIX, often called the "Fear Gauge," measures the market's expectation of 30-day volatility implied by S&P 500 index options. It is not a stock you can buy directly, but an index. Its purpose is to provide a benchmark for market sentiment; a high VIX suggests high uncertainty and expensive options premiums, while a low VIX suggests market stability and cheaper premiums.

Option Symbols:

I utilized standard OCC (Options Clearing Corporation) symbols. For example, a symbol like TSLA260220C00200000 is broken down as follows:

TSLA: The underlying ticker.

260220: The expiration date (Year: 2026, Month: 02, Day: 20).

C: The option type (Call).

00200000: The strike price (200.00).

Recorded Market Parameters

To maintain mathematical consistency in our Black-Scholes model, the following parameters were recorded at the exact moment of data retrieval:

Underlying Prices (S): At the time of download on Day 1 (Feb 11), the spot price of **SPY** was **$680.12** and **TSLA** was **$416.32**. On Day 2 (Feb 12), the prices slightly appreciated to **$681.27** and **$417.07**, respectively.

Risk-Free Interest Rate (r): We utilized the **Federal Funds (Effective) Rate** for mid-February 2026, which stood at **3.64%**. This was converted to the decimal **0.0364** for all Black-Scholes calculations.

The VIX Context: On Day 1, the **VIX** was at **21.15**, suggesting a moderate level of market anxiety. By Day 2, it dropped slightly to **20.82**. This slight decrease in broad market volatility explains why some of the "Actual" option prices on Day 2 might have been lower than our Day 1 predictions—the "volatility premium" was shrinking.

Parameter	Day 1 (02/11/26)	Day 2 (02/12/26)
TSLA Spot Price	**$416.32**	**$417.07**
SPY Spot Price	**$680.12**	**$681.27**
VIX Index (^VIX)	**21.15**	**20.82**
Risk-Free Rate (r)	**3.64%**	**3.64%**

Time to Maturity (T):

For the purposes of the Black-Scholes model, the time to maturity must be expressed as a fraction of a year. We calculated this by taking the number of calendar days between the data download date and the expiration date of the options (February 20, 2026), divided by 365.

- For Day 1 (Feb 11): $T = 9 / 365 = 0.0247$
- For Day 2 (Feb 12): $T = 8 / 365 = 0.0219$

Part 2

2.1 Implementation of the Black-Scholes-Merton (BSM) Model

The Black-Scholes-Merton framework was used in this section. The engine was designed to handle European-style options on SPY and TSLA. The core logic calculates the theoretical price by modeling the underlying stock as a geometric Brownian motion.

Key Technical Decisions:

Numerical Stability: I implemented the cumulative distribution function (CDF) using scipy.stats.norm, ensuring high-precision tails for deep Out-of-the-Money (OTM) options.

Input Sensitivity: The model accounts for dividends and uses the annualized risk-free rate sourced in Part 1 to maintain theoretical consistency.

2.2 Implied Volatility (IV) and Root-Finding Algorithms

Finding the volatility (sigma) that equates the theoretical price to the observed market mid-price. Since the BSM formula cannot be algebraically inverted for sigma, I implemented and compared two numerical root-finding methods:

- Newton-Raphson Method: Leveraged the analytical Vega (the partial derivative of price with respect to volatility) to achieve rapid, quadratic convergence.
- Bisection Method: Used as a robust "fail-safe." While slower, the Bisection method guaranteed a solution within the bounds of [0.0001, 5.0], preventing the mathematical "divergence" that Newton-Raphson occasionally experienced with illiquid, low-Vega strikes.

2.3 The Volatility Smile and Skew

Interpretation of Results:

This is a critical finding as it highlights the primary limitation of the Black-Scholes model: the assumption of constant volatility. The market prices OTM Puts at a higher IV than At-the-Money (ATM) options, reflecting a higher premium for "tail risk" or market crashes.

2.4 Greek Sensitivities: Analytical vs. Numerical

To validate our risk management metrics, I calculated Delta, Gamma, and Vega using two distinct approaches:

Analytical Greeks: Derived via exact partial differential equations (PDEs).

Numerical Greeks: Calculated using the Finite Difference Method by shocking the spot price by a small Delta S = 0.01.

Finding: The two methods yielded nearly identical results (typically matching to 5+ decimal places). This convergence serves as a formal verification of the engine's mathematical accuracy.

2.5 Inter-temporal Prediction (Question 12)

The final test of the engine involved using Day 1's calculated Implied Volatility to predict Day 2's market prices.

Observation: For highly liquid, near-the-money strikes, the prediction error was remarkably low (<1%).

Error Analysis: Significant errors were observed in strikes where the underlying spot price moved significantly or where the broad market VIX shifted overnight. This demonstrates that while IV is a "persistent" variable, it must be dynamically updated to account for changing market sentiment.

Part 3: Automated Market Maker (AMM) Mechanics

3.1 Economic Theory and Derivation

Ianalyzed the mechanics of a Constant Product Market Maker (xy=k). We derived the relationship between the price of the underlying assets and the inventory levels held by the Liquidity Provider (LP).

Key Insight: the price P is the ratio of the reserves (y/x).

The "Arb" Gap: An arbitrageur will only trade when the market price deviates from the AMM price by more than the fee rate gamma.

3.2 Expected Revenue and Volatility (sigma)

Using numerical integration (Trapezoidal Rule), the Expected Fee Revenue (E[R]) for an LP.

Findings for 3(b): Using a volatility (sigma) of 0.2 and a fee rate (gamma) of 0.003 (30 bps), an expected daily revenue of approximately 0.000012 per unit of liquidity.

The Volatility Link: Our results confirmed that revenue is a function of the variance of the price process. Higher volatility creates more "Arb Gaps," which increases the frequency of trades and, consequently, the total fees collected.

3.3 Optimal Fee Rate (gamma^*)

Plotted the Optimal Fee Rate against varying levels of market volatility.

Interpretation: The plot shows an upward-sloping curve, meaning gamma^* increases as sigma increases. This is because higher volatility leads to higher Impermanent Loss (or Loss-Versus-Rebalancing). To compensate the LP for this increased risk, the "equilibrium" fee must be higher.

This provides a clear mathematical justification for why high-volatility pairs in DeFi (like new altcoins) require higher fee tiers than stable pairs.

Part 4: Numerical Integration - The Bonus

4.1 Methodology: Composite 2D Simpson's Rule

For the bonus section, we implemented a high-order numerical integration scheme to approximate double integrals over a rectangular domain $R = [0, 1] \times [0, 3]$. Composite 2D Simpson's Rule, which applies a weighted average of function values at the corners, midpoints, and center of each sub-grid.

4.2 Accuracy and Convergence

Tested the algorithm on two functions to verify its precision:

$f_1(x, y) = xy$: The numerical result was exactly 2.25. Because Simpson's Rule is designed to be exact for polynomials of low degree, the error was effectively zero ($< 10^{-16}$), confirming the logic of our grid weights.

$f_2(x, y) = e^{x+y}$: This transcendental function provided a test for convergence. As we decreased the step sizes (dx, dy), the numerical result converged to the analytical value of 32.79.

4.3 Results Table

The following table illustrates the error reduction as the grid density increased, proving the robustness of our implementation:

dx, dy	f1 Error	f2 Error
(0.5, 1.5)	0.00e+00	1.70e+00
(0.25, 1.0)	0.00e+00	7.24e-01
(0.1, 0.5)	8.88e-16	1.77e-01
(0.05, 0.25)	4.44e-16	4.44e-02