6808-Project

1 Modeling volatility

We choose Yahoo Finance as our data source and set **MSFT** as our underlying stock. The date ranges from 2021-01-01 to 2024-09-30, the data are shown in the picture (the last two columns are added by us, original data only contain the first 7 columns)

	symbol	open	high	low	close	volume	adj_close	y	GARCH_volatility
date									
2021-01-05	MSFT	217.259995	218.520004	215.699997	217.899994	23823000	211.047668	0.015306	0.258540
2021-01-06	MSFT	212.169998	216.490005	211.940002	212.250000	35930700	205.575394	-0.417045	0.254274
2021-01-07	MSFT	214.039993	219.339996	213.710007	218.289993	27694500	211.425415	0.445432	0.261943
2021-01-08	MSFT	218.679993	220.580002	217.029999	219.619995	22956200	212.713577	0.096427	0.269000
2021-01-11	MSFT	218.470001	218.910004	216.729996	217.490005	23031300	210.650558	-0.154711	0.264916
...	...	...	...	...	...	...	...	...	...
2024-10-10	MSFT	415.230011	417.350006	413.149994	415.839996	13848400	415.839996	-0.061722	0.194835
2024-10-11	MSFT	416.140015	417.130005	413.250000	416.320007	14144900	416.320007	0.018314	0.192464
2024-10-14	MSFT	417.769989	424.040009	417.519989	419.140015	16653100	419.140015	0.107166	0.189646
2024-10-15	MSFT	422.179993	422.480011	415.260010	418.739990	18900200	418.739990	-0.015158	0.187623
2024-10-16	MSFT	415.170013	416.359985	410.480011	416.119995	15508900	416.119995	-0.099637	0.184998

952 rows × 9 columns

1.1 Geometric Brownian Motion

1.1.1 Statistic Estimation

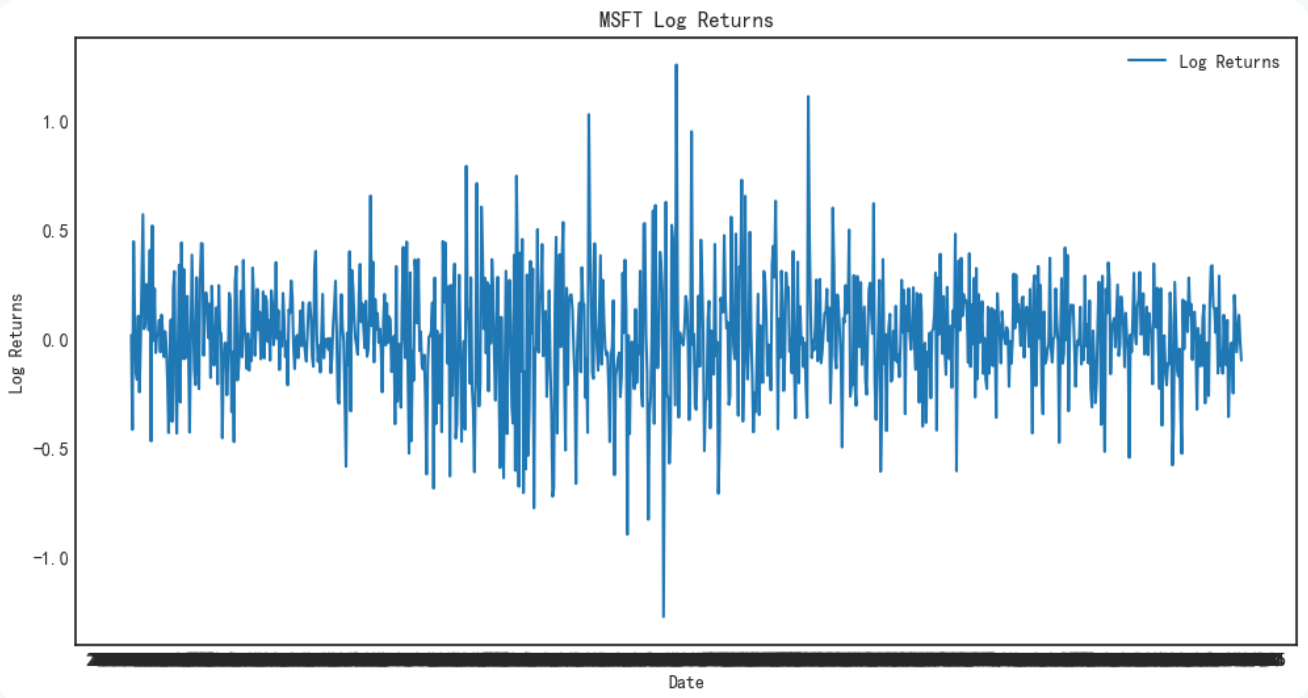
If the stock price follows **Geometric Brownian Motion**, then

$$
\begin{aligned}
d\ln(S) &= 0\times dt + \frac{1}{S}dS - \frac{1}{2}\frac{1}{S^2}(dS)^2 \\
&= (\mu dt + \sigma dB) - \frac{1}{2}(\mu dt + \sigma dB)^2 \\
&= \mu dt + \sigma dB - \frac{1}{2}\sigma^2 dt \\
&= \left(\mu - \frac{1}{2}\sigma^2\right)dt + \sigma dB
\end{aligned}
$$

hence

$$\begin{aligned}
\Delta \ln S &= \left(\mu - \frac{1}{2}\sigma^2\right)\Delta t + \sigma \Delta B \\
\ln S(t_i) - \ln S(t_{i-1}) &= \left(\mu - \frac{1}{2}\sigma^2\right)\Delta t + \sigma(B(t_i) - B(t_{i-1})) \\
\frac{\ln S(t_i) - \ln S(t_{i-1})}{\sqrt{\Delta t}} &= \left(\mu - \frac{1}{2}\sigma^2\right)\sqrt{\Delta t} + \sigma \frac{B(t_i) - B(t_{i-1})}{\sqrt{\Delta t}}
\end{aligned}$$

here we let $\Delta t = 1 \text{ day } = 1/252 \text{ year}$, let $y_i = \frac{\ln S(t_i) - \ln S(t_{i-1})}{\sqrt{\Delta t}}$, the plot of log-return is



and the estimator of$\sigma$is

$$\hat{\sigma}^2 = \frac{1}{N-1}\left(\sum_{i=1}^{N} y_i^2 - N\bar{y}^2\right)$$

Hence we can compute each y_i and then get $\hat{\sigma}^2$, the result is

$$\hat{\sigma} = 0.2625$$

1.1.2 Maximum Likelihood Estimation

We can also use maximum likelihood estimation to estiamte σ, since for each y_i, we have

$$y_i \sim N\left(\left(\mu - \frac{1}{2}\sigma^2\right)\sqrt{\Delta t}, \sigma^2\right)$$

Then the likelihood function is

$$L(\mu, \sigma) = \Pi \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{\frac{\left(y_i - (\mu - \frac{1}{2}\sigma^2)\sqrt{\Delta t}\right)^2}{2\sigma^2}\right\}$$

$$l(\mu, \sigma) = \ln L(\mu, \sigma) = -\frac{n}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum\left(y_i - \left(\mu - \frac{1}{2}\sigma^2\right)\sqrt{\Delta t}\right)^2$$

Hence we can maximize $l(\mu, \sigma)$ to find $\hat{\sigma}$, the result is also

$$\hat{\sigma} = 0.2625$$

1.2 GARCH Model

We use GARCH(1, 1) model to fit y_i, to get our conditional variance, we use `arch.arch_model` to fit our data, the result is show as follows

Constant Mean - GARCH Model Results

Dep. Variable:	y	R-squared:	0.000
Mean Model:	Constant Mean	Adj. R-squared:	0.000
Vol Model:	GARCH	Log-Likelihood:	-25.0136
Distribution:	Normal	AIC:	58.0273
Method:	Maximum Likelihood	BIC:	77.4615
		No. Observations:	952
Date:	Sat, Nov 02 2024	Df Residuals:	951
Time:	09:21:40	Df Model:	1

Mean Model

	coef	std err	t	P>|t|	95.0% Conf. Int.
mu	0.0153	7.434e-03	2.064	3.900e-02	[7.745e-04,2.991e-02]

Volatility Model

	coef	std err	t	P>|t|	95.0% Conf. Int.
omega	3.0420e-04	2.702e-04	1.126	0.260	[-2.253e-04,8.337e-04]
alpha[1]	0.0324	8.732e-03	3.715	2.031e-04	[1.533e-02,4.956e-02]
beta[1]	0.9627	1.033e-02	93.208	0.000	[0.942, 0.983]

for model

$$\sigma_{i+1}^2 = a + by_i^2 + c\sigma_i^2$$

the result is

$$a = 0.0003 \qquad b = 0.0324 \qquad c = 0.9627$$

The bule line is $\textit{GARCH Volatility}$ and green line is the close price of **MSFT** stock. The red line is the constant volatility we get in 1.1

1.3 Volatility Smile

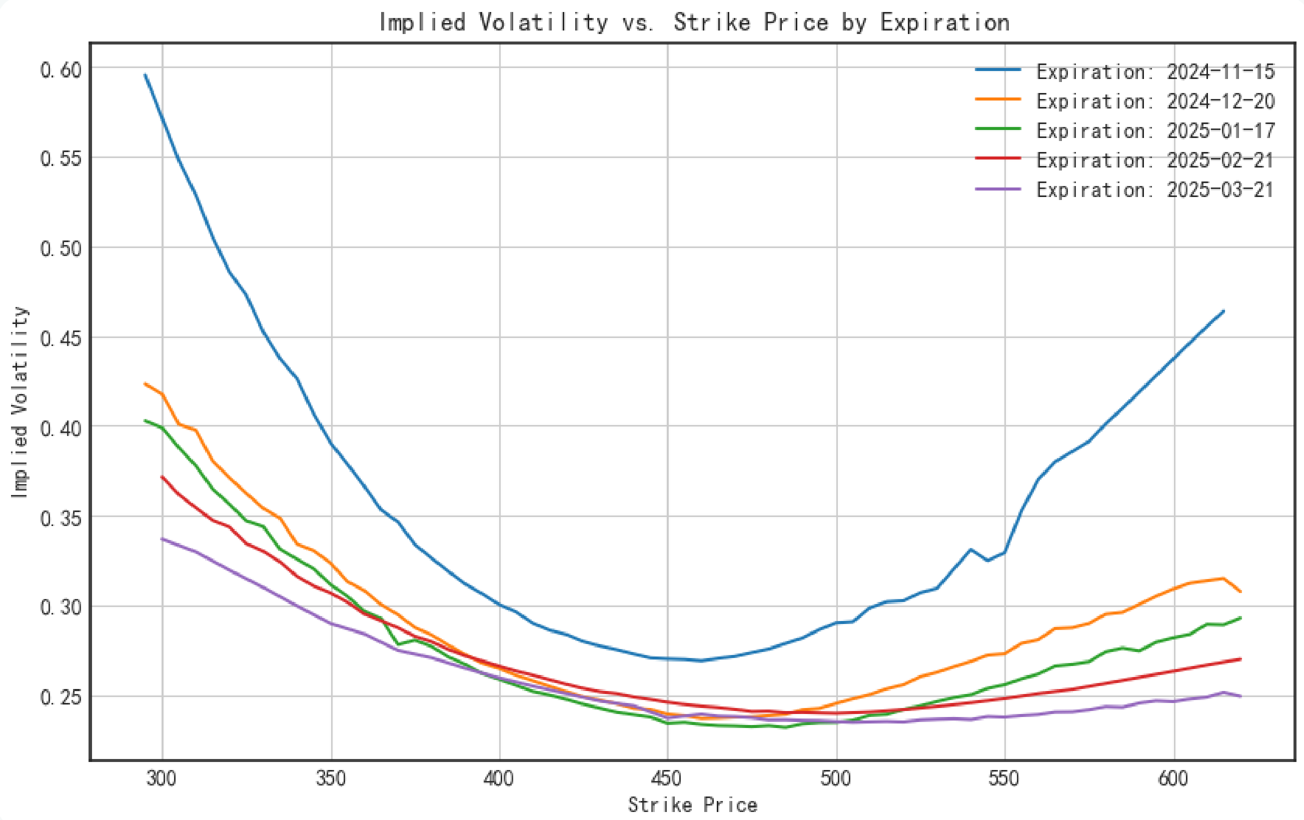
The European Call option data are collected from **Bloomberg** on 16th Oct.

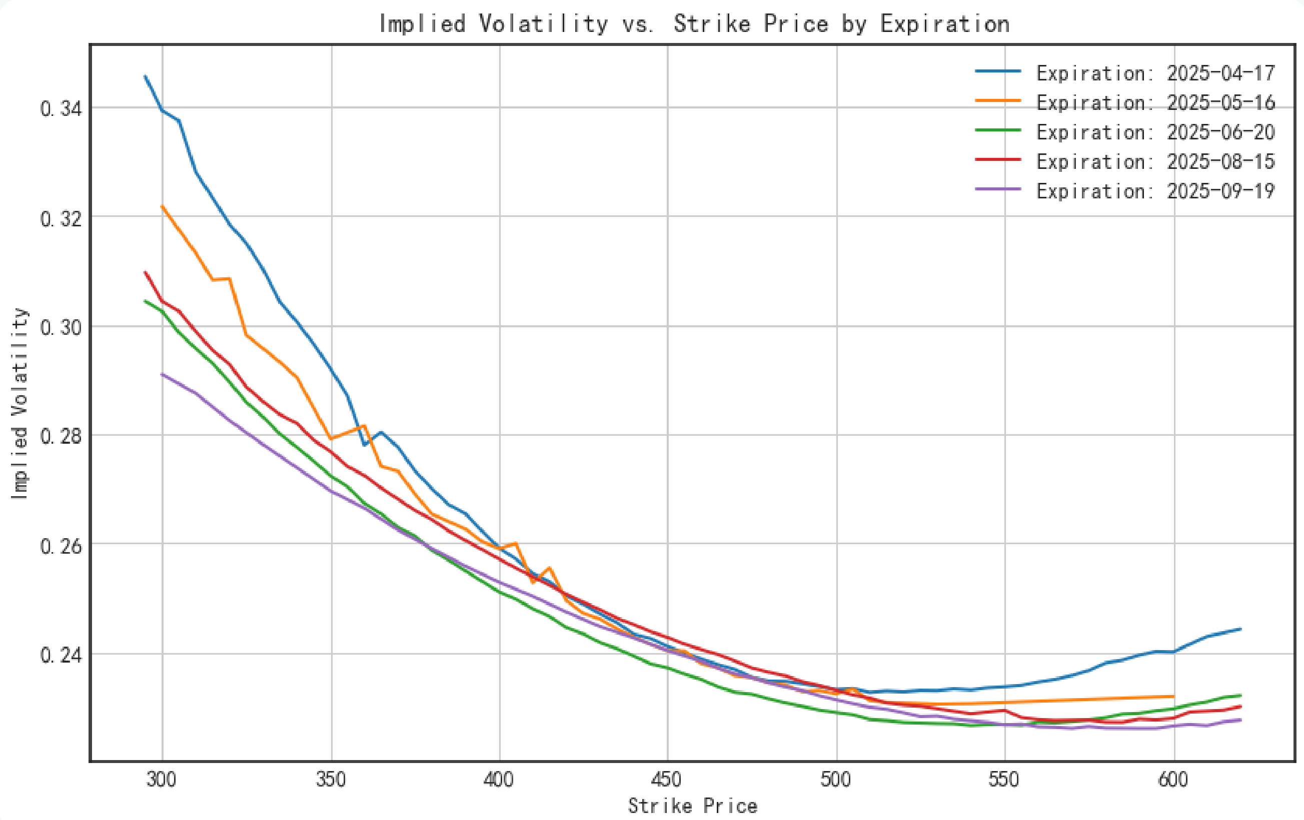
The close price of **MSFT** is $416.12

since there are some outliers in option data, so we pick some of them, which are

1. Maturity ranges from 2024-11-15 to 2025-09-19, thus the time to maturity(TTM) is from 0.0822 to 0.9260
2. Strike price is 0.7 to 1.5 times the current price, which is about 300 to 630
3. Implied Volatility is between 0.01 to 0.6
4. The trading price of the option must greater than 0.01 (since some options are never be traded so we exclude them)

Hence the volatility smile is shown as follows, we can find that the smaller the TTM, the more obvious the smile.

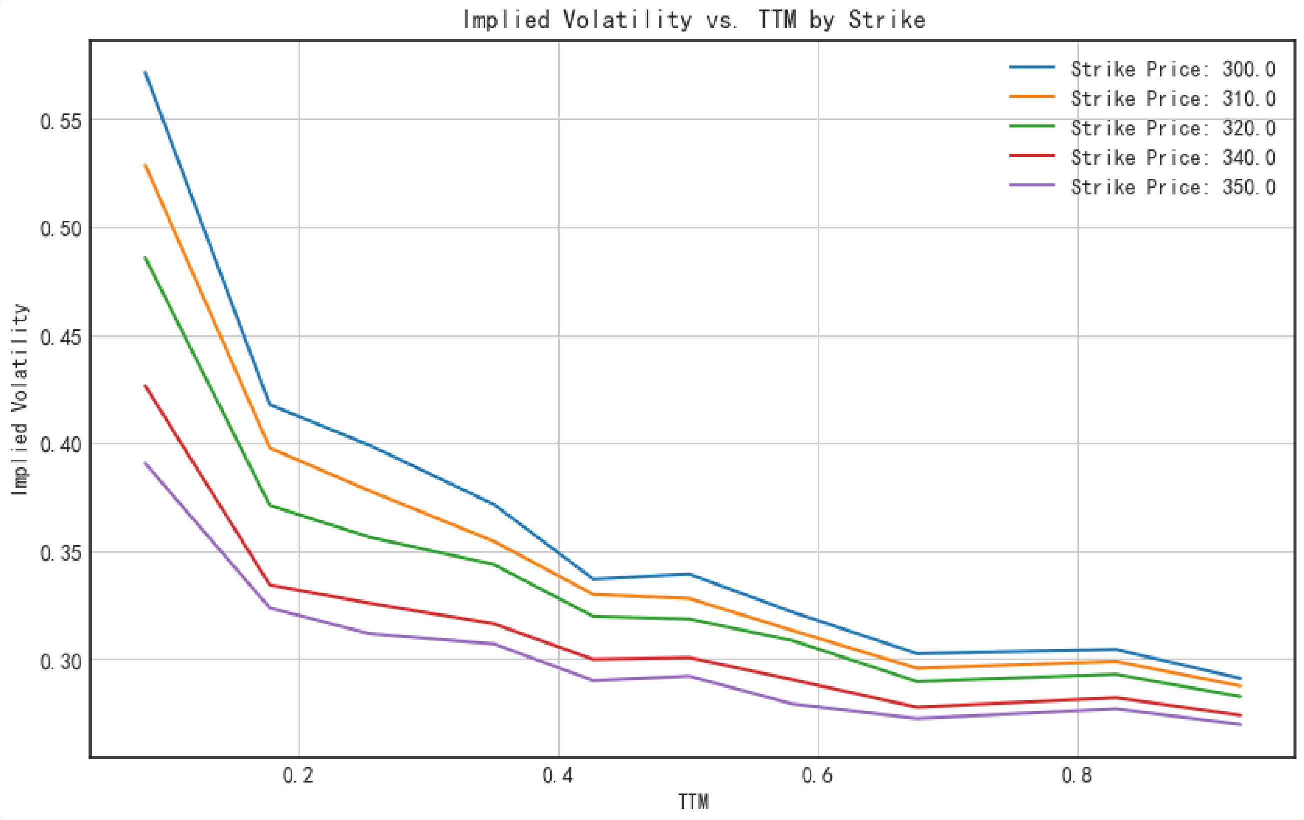




1.4 Term Structure

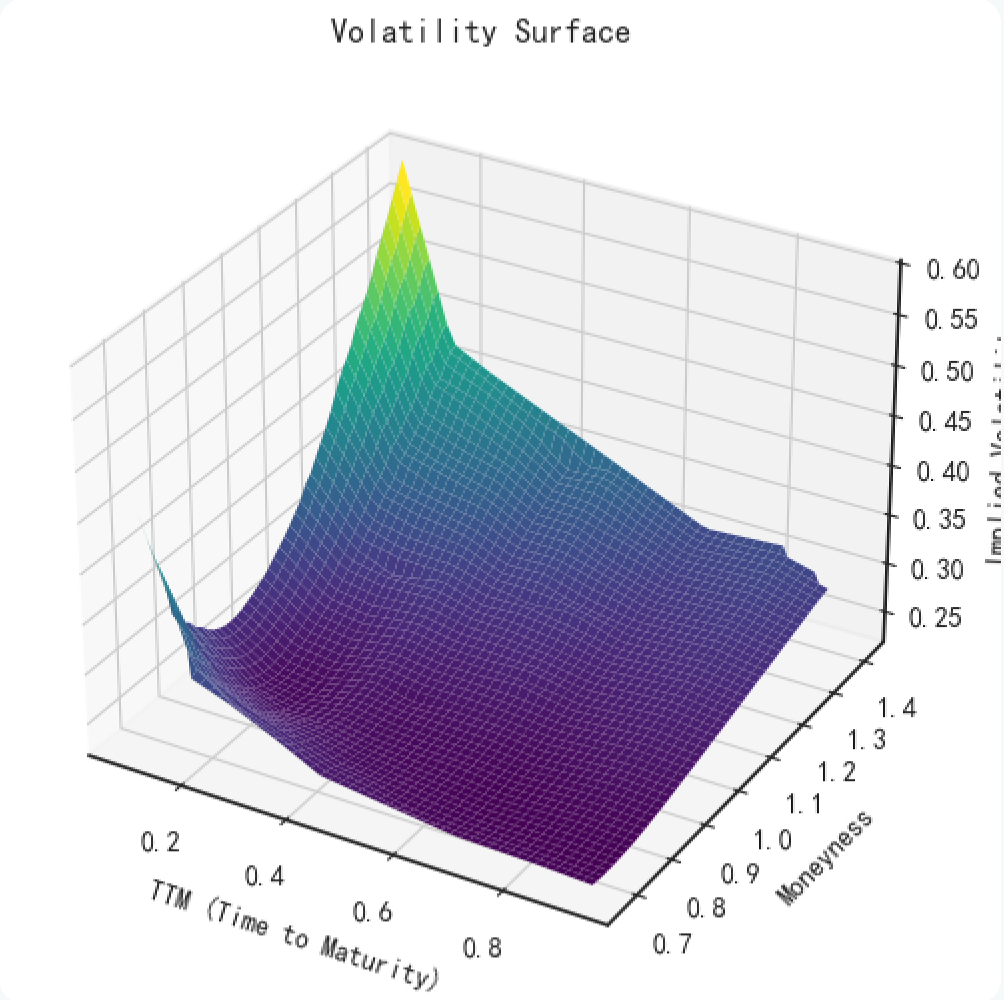
Since there are many strike prices for each option with a certain TTM, So we selected the 15 strike prices with the highest frequency.

The term structure is shown in the following 3 pictures. We find that as TTM increase, the volatilit shows a decrease trend



1.5 Volatility Surface

Based on the data we choose, the surface is shown as follows



2 Vanilla European Option Pricing

In this question, we explore four methods to calculate the price of a European call option:

1. Monte Carlo Simulation
2. Numerical PDE
3. Binomial Model
4. Black-Scholes Formula

The option's strike price and TTM is chosen by criteria in 1.3.

2.1 Monte Carlo Simulation

We utilize the GARCH(1,1) model obtained in Question 1(2) to substitute the fixed volatility, and combined with the GBM model, we can calculate the price of the call option:

Let

$$x_i = \left(\exp\left\{\ln S(0) + \left(r - \frac{1}{2}\sigma_i^2\right)T + \sigma_i\sqrt{T}z\right\} - K\right)^+$$

by simulating this process for 10000 times,$e^{-rT}\bar{x}$is an estimate of the option price.

At each time, we simulate a y_i based on our σ_i and

$$y_i = \frac{\ln S(t_i) - \ln S(t_{i-1}) - \left(\mu - \frac{1}{2}\sigma_i^2\right)\Delta t}{\sqrt{\Delta t}} \sim N(0, \sigma_i^2 \Delta t)$$

2.2 Numerical PDE

Numerical PDE is based one the equation below：

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0$$

where:

- V is the option price as a function of the stock price S and time t,
- σ is the volatility of the stock,
- r is the risk-free interest rate.

Here we use implicit scheme to solve the PDE, since

$$\frac{V_i^{j+1} - V_i^j}{\Delta t} + \frac{1}{2}\sigma^2 S_i^2 \frac{V_{i+1}^j - 2V_i^j + V_{i-1}^j}{(\Delta S)^2} + rS_i \frac{V_{i+1}^j - V_{i-1}^j}{2\Delta S} - rV_i^j = 0$$

we define

$$\alpha_i = \frac{\Delta t}{4}\left(\sigma^2 S_i^2/(\Delta S)^2 - rS_i/\Delta S\right)$$
$$\beta_i = -\frac{\Delta t}{2}\left(\sigma^2 S_i^2/(\Delta S)^2 + r\right)$$
$$\gamma_i = \frac{\Delta t}{4}\left(\sigma^2 S_i^2/(\Delta S)^2 + rS_i/\Delta S\right)$$

then we get

$$\alpha_i V_{i-1}^{j+1} + (1-\beta_i)V_i^{j+1} + \gamma_i V_{i+1}^{j+1} = V_i^j$$

at each time step j, we need to solve a linear system

$$AV^{j+1} = d$$

where A is tri-diagonal matrix contains α, β, γ, and d is the option price of last time step, for such question, we can use Thomas Algorithm to solve it

2.3 Binomial Model

The Binomial Model approximates the price evolution of the underlying asset by creating a binomial tree with N time steps. At each step, the price can either go up or down by a factor u or d.

Assume an N-period recombining binomial model:

- At maturity, we have $N+1$ nodes.
- Denote i as the number of **up**s, then $i = 0, 1, 2, \ldots, N$ and $N-i$ is the number of **down**s; each possible value of i represents one node at T.
- The probability to reach i is a binomial probability:

$$\binom{N}{i} p^i (1-p)^{N-i} = \frac{N!}{i!(N-i)!} p^i (1-p)^{N-i}$$

- The value of a European call:

$$V(T) = \sum_{i=0}^{N} \frac{N!}{i!(N-i)!} p^i (1-p)^{N-i} \max(u^i d^{N-i} S - K, 0)$$

To calculate the value of these parameters, we apply two methods below:

- **Cox-Ross-Rubinstein**

$$u = e^{\sigma\sqrt{\Delta t}}, \quad d = \frac{1}{u}, \quad p = \frac{e^{r\Delta t} - d}{u - d}$$

- **Jarrow-Rudd**

$$u, d = e^{(r - \frac{1}{2}\sigma^2)\Delta t \pm \sigma\sqrt{\Delta t}}, \quad p = \frac{1}{2}$$

2.4 Black-Scholes Formula

The *B-S* formula is

$$C_E(S,t) = SN(d_1) - Ke^{-r(T-t)} N(d_2)$$

where

$$d_1 = \frac{\ln(S/K) + (r + \frac{\sigma^2}{2})(T-t)}{\sigma\sqrt{T-t}}$$

and

$$d_2 = d_1 - \sigma\sqrt{T-t} = \frac{\ln(S/K) + (r - \frac{\sigma^2}{2})(T-t)}{\sigma\sqrt{T-t}}$$

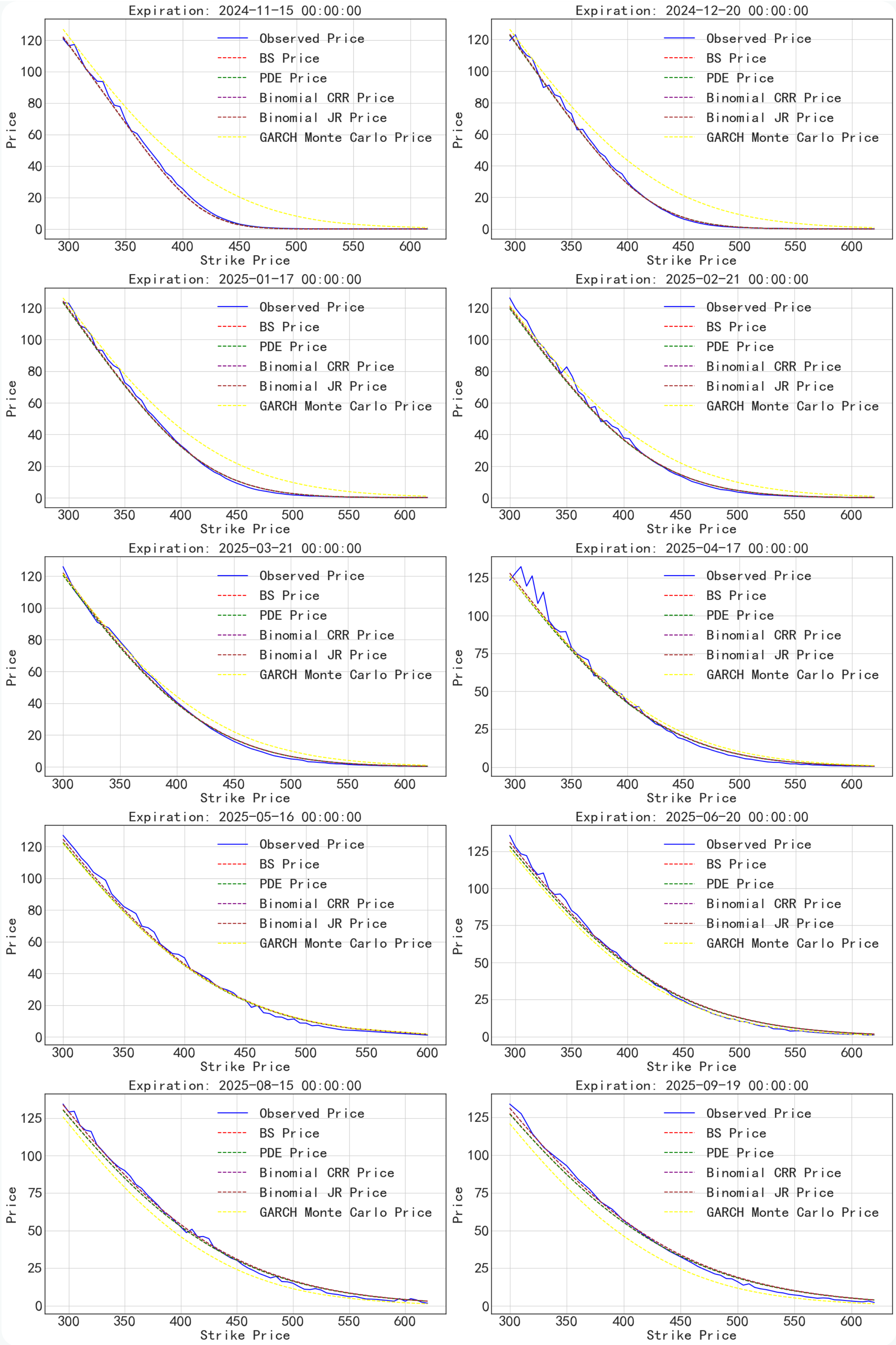
By applying this formula, we are able to calculate the option price directly

2.5 Compare and Comment

After calculating the option price with four different method, we try to plot the observed option price along with the price calculated from different methonds.

As we can observe in the line graph, although the volatility of the obseved price cannot be simulated by different models.

All the methods of calculation option price fits well with the observed price except GARCH Monte Carlo method


Expiration: 2024-11-15 00:00:00
Expiration: 2024-12-20 00:00:00
Expiration: 2025-01-17 00:00:00
Expiration: 2025-02-21 00:00:00
Expiration: 2025-03-21 00:00:00
Expiration: 2025-04-17 00:00:00
Expiration: 2025-05-16 00:00:00
Expiration: 2025-06-20 00:00:00
Expiration: 2025-08-15 00:00:00
Expiration: 2025-09-19 00:00:00
Observed Price
BS Price
PDE Price
Binomial CRR Price
Binomial JR Price
GARCH Monte Carlo Price
Price
Strike Price

2.5.1 Black-Scholes (BS) Price

- **Efficiency**: The Black-Scholes model is an analytical solution, so it computes option prices very quickly with just one calculation.
- **Accuracy**: For market environments that theoretically meet BS assumptions， the BS model is usually very accurate. However, it does not account for volatility changes, which can lead to errors when market volatility is unstable.
- **Sources of Error**: The primary source of error is the assumption of constant volatility, which fails to capture the volatility smile and other market realities.
- **Improvement Suggestions**: Using implied volatility or models that account for volatility changes (e.g., the Heston model) can help reduce errors.

2.5.2 Numerical PDE Pricing

- **Efficiency**: The PDE method requires iterative calculations over a grid, making it more computationally intensive.
- **Accuracy**: PDE methods are relatively accurate for valuing European options and can adapt to changing volatility, making it more accurate in scenarios where BS assumptions are not fully met.
- **Sources of Error**: Grid step size selection affects accuracy; larger steps may lead to numerical instability, while smaller steps increase computational cost.

- **Improvement Suggestions**: Increasing grid density or optimizing the step size can improve the accuracy of PDE pricing but will also increase computation time.

2.5.3 Binomial Model (Binomial CRR and Binomial JR)

- **Efficiency**: The binomial model is slower than the BS model but faster than PDE. Increasing the number of time steps improves accuracy but also increases computational cost.
- **Accuracy**: The binomial model simulates the price path over discrete time steps and can accurately estimate option prices in environments with different volatilities and volatility smiles. The performance of CRR and JR models may vary slightly based on their assumptions.
- **Sources of Error**: The number of time steps limits model accuracy. Fewer steps lead to a discontinuous price distribution, which affects accuracy.
- **Improvement Suggestions**: Increasing the number of time steps enhances model accuracy but significantly raises computation time, especially for options with long expiration times.

2.5.4 Monte Carlo Simulation

- **Efficiency**: Monte Carlo methods are the slowest because they require numerous simulations.
- **Accuracy**: Monte Carlo simulation can flexibly use a GARCH volatility model to capture dynamic volatility. However, your plot shows that the

MC results are not always accurate, which may be due to a low volatility cap setting or insufficient sample size.

- **Sources of Error**: Errors arise from the number of samples, the choice of volatility model, and the volatility cap setting. When using GARCH volatility, simulated price distributions may not align perfectly with actual distributions.
- **Improvement Suggestions**: Increasing the number of simulations can reduce random error, and adjusting the volatility cap may help better capture actual market volatility. Additionally, using variance reduction techniques such as control variates can enhance MC accuracy.

2.5.5 Summary

For a balance of accuracy and efficiency, **PDE and Binomial models** are preferred. The **Black-Scholes model** is fast but limited in volatile market environments. **Monte Carlo simulation** offers flexibility but requires longer computation times and careful adjustments to volatility settings.

3 Variance Reduction in Monte Carlo

3.1 Crude Monte Carlo

We used the Geometric Brownian Motion from Question 1(1) to simulate 10,000 stock price paths. Then, we calculate the average stock price for each path and subsequently determine the Asian Call Option payoff for each path. The estimated price of the option is obtained by averaging all the discounted payoffs. The pricing formula for the Asian Call Option is:

$$\text{Payoff} = \max\left(\frac{1}{M}\sum_{i=1}^{M} S(t_i) - K, 0\right)$$

where $S(t_i)$ is the price of the underlying asset at time t_i, M is the number of time intervals in the observation period, and K is the strike price of the option.

3.2 Control Variate

We used the **Control Variate Method** and **Antithetic Variate Method** to reduce the variance in estimating option prices.

The Control Variate Method calculates two option payoffs: one based on the arithmetic average price and the other based on the geometric average price. Then, we compute the regression coefficient between them using linear regression. Next, we compute the price of the geometric average Asian call option using the formula:

$$\phi = \exp((\mu - r_f)T)S_0\Phi(d_1) - \exp(-r_f T)K\Phi(d_2)$$

where (The expectation price of geometric average Asian call option is found from this website)

$$\mu = \frac{(n+1)}{2n}(r_f - \frac{\sigma^2_{\text{origin}}}{2}) + \frac{\sigma^2}{2}$$

$$\sigma = \sqrt{\frac{(n+1)(2n+1)}{6n^2}\sigma^2_{\text{origin}}}$$

$$d_1 = \frac{\ln(S_0/K) + (\mu - \frac{\sigma^2}{2})T}{\sigma\sqrt{T}}$$

$$d_2 = d_1 - \sigma\sqrt{T}$$

The estimated option price is:

$$\text{Price} = x + \beta(\phi - y)$$

3.3 Antithetic Variate

Antithetic Variate Method calculates two option payoffs: one based on the forward simulated paths x and the other based on the antithetic (reverse) simulated paths y. Then, it discounts the average of these two payoffs to the present value, using the formula:

$$\text{Price} = \exp(-r_f T)\frac{x+y}{2}$$

Hence when we simulate the price paths, we need simulate a antithetic price path as well

3.4 Results

We draw **pdf plots** to show how much efficiency is gained. According to the plot, the means obtained by the three methods are very close, specifically 35.77, 36.26, and 36.33.

Both the Control Variate and Antithetic Variate methods reduced the variance, but the efficiency of the Control Variate method is significantly higher than that of the Antithetic Variate method.