University of St. Gallen

School of Management, Economics, Law, Social Sciences, International Affairs and Computer Science

(HSG)

Stochastic Methods in Finance

Project Report

Elio Froidevaux 23-619-232

Jan Dietrich 23-607-591

Leonardo Gonnelli 23-619-111

Stochastic Methods in Finance (6,252)

Dr. Enrico Giovanni De Giorgi

19 May 2025

List of Contents

List of Figures

List of Tables

1. Introduction

Task Description

For this assignment, we aim to price a European Asian call option with Microsoft Corporation (MFST) as the underlying asset using a multi-period binomial tree model. Additionally, we will use a normal distribution to validate the results. The option's initial time $t = 0$ corresponds to the 28th of April 2025, and maturity time is set to be 6 months later ($T = 0.5$ years).

Definitions

A European call option is a financial derivative that gives its holder the right, but not the obligation, to buy a specific underlying asset at a predefined price (called the strike price) on a specific future date. The payoff of the option is determined by the difference between the final price of the underlying asset and the strike price, or zero if that difference is not favorable, i.e., S < K, and therefore negative.

The payoff at maturity T is given by:

$$A_T = \max(S_t - K, 0)$$

Where:

- A_T is the payoff of the option
- S_t is the price of the asset at maturity T
- K is the strike price

A European Asian call option is a European call option whose payoff at maturity depends on the average of the underlying asset price observed over discrete time periods. The payoff at maturity T is given by:

$$A_T = \max\left(\frac{1}{1+n}\sum_{t=0}^{n} S_t - K, 0\right)$$

Motivation and Implications

The European Asian call option offers the advantage of generally having lower volatility compared to a European call option, since its payoff depends on the average price rather than the final price. It helps reduce the impact of extreme price movements, especially near the maturity date.

Since the payoff of the European Asian call option is based on the average value of the underlying asset's price, the calculation of the option price is significantly more complex: instead of simply considering the different final prices and their relative probabilities, we must account for every possible path because we need to know all the intermediate prices in order to compute the payoff. In this report, we assume the European Asian option is cash-settled (most common for Asian options) so that we can set the strike price $K = S_0$. Any positive payoff will be settled as a direct cash transfer, rather than acquiring the stock itself.

2. Dataset & Preprocessing

The first step in our data procurement was to download the necessary time series data for Microsoft's stock price from 2020 to the current date (7th May 2025). Since the Yahoo Finance API was no longer working, we pivoted to the official NASDAQ website. There, we were able to download the last 10 years of daily stock price data for Microsoft, consisting of the last closing price for any day (Close.Last), volume (Volume), opening price (Open), the daily high (High), and the daily low (Low).

As a next step, we examined the data, cleaned and prepared it for our analysis. First, we converted the Date column into a proper date object using as.Date(Date). Then, we cleaned the Close.Last column by removing dollar signs and commas with the gsub() function and converting the values to numeric format, since the original values were character strings unsuitable for calculations. Finally, we filtered the date range to only include the desired time period, from January 1st, 2020, to April 28th, 2025.

With the cleaned data, we conducted an initial examination of Microsoft's stock price development. We used the daily closing price to plot the chart below.

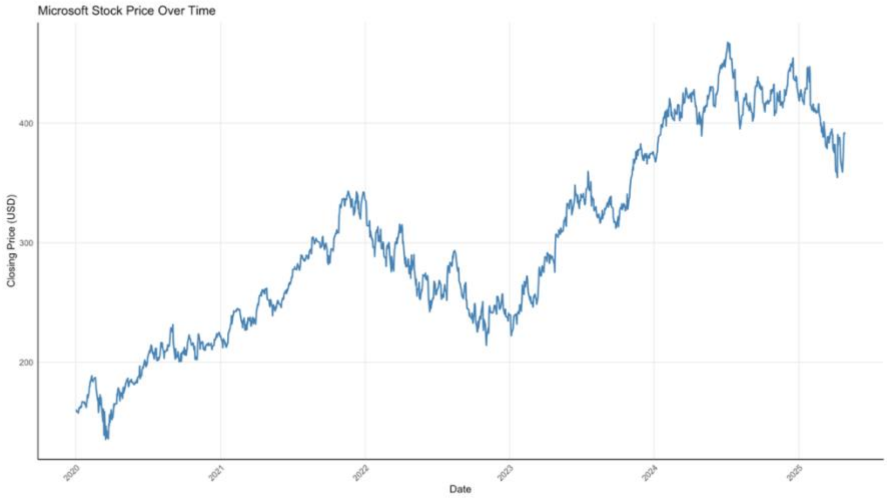

Figure 1 Stock price development of Microsoft Corporation

To analyze the daily performance of the stock, we calculated the log returns using the Close.Last prices. We did this by taking the natural logarithm of the ratio between the current day's closing price and the previous day's closing price. This transformation is standard in finance because it makes returns time-additive and symmetric for price increases and decreases. We implemented this using the mutate() function with the formula:

$$r_t = \log\left(\frac{P_t}{P_{t-1}}\right)$$

Where:

r_t is the log return at time t,

P_t is the closing price at time t,

P_{t-1} is the closing price at time t-1,

log() denotes the natural logarithm.

In a final step, we filtered out the missing value resulting from the first observation, which has no preceding value to compare to. After obtaining the daily log returns, we re-examined the data using a histogram.

We observed that the distribution was roughly bell-shaped and centered around zero, consistent with the expected behavior of log returns under normal market conditions. The majority of returns clustered closely around zero, indicating that most daily price changes were small.

However, there were a few more extreme values in both tails of the distribution, reflecting occasional larger market moves. This is a common feature in financial data, suggesting mild leptokurtosis, also known as “fat tails.” This suggests that while daily returns generally follow a normal distribution, real-world data often exhibit deviations such as asymmetry or extreme events. Overall, the histogram provided useful insight into the volatility and risk characteristics of the stock during the observed period.

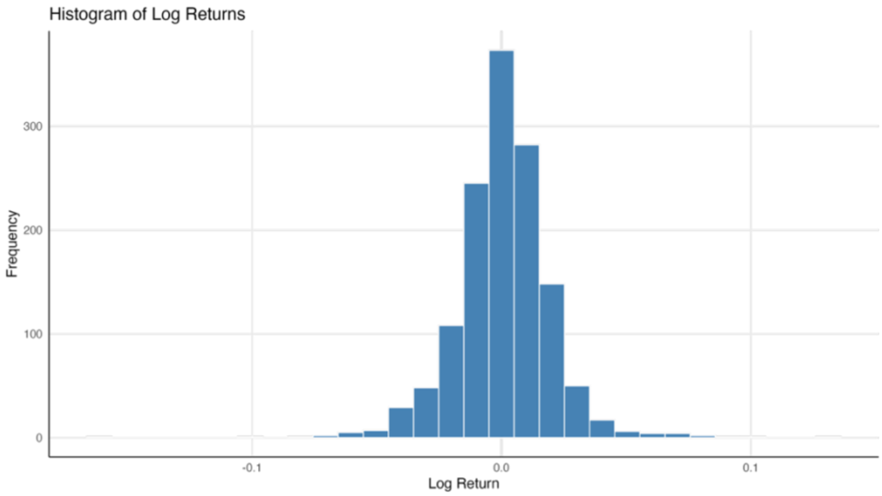

Figure 2 Daily log returns of Microsoft Corporation

3. Theoretical Background

A binomial tree is a method used in finance to model the possible future movements of an asset price over time. It provides a discrete framework for capturing uncertainty in markets and is especially useful for pricing options. At its core, the binomial tree assumes that at each step in time, the asset price can move up by a factor u or down by a factor d. This leads to a branching structure: from the initial price S_0, the tree expands step-by-step into a tree of possible future prices.

The process unfolds over a fixed number of time intervals n, covering a total maturity T. At each node, the model uses a risk-neutral probability q to determine the expected value of the option’s payoff in the next period. In the binomial model, the risk-neutral probability q is defined to ensure that the expected return of the asset is equal to the risk-free rate. It is given by the formula:

$$q = \frac{e^{-r\Delta t} - d}{u - d}$$

This expected return is then discounted at the risk-free rate and assigned to the current node. By repeating this procedure backward through the tree, we obtain the arbitrage-free price of the option or risk-neutral valuation. Under this measure, all assets are assumed to grow on average at the risk-free rate r, rather than at their real-world expected return.

This condition ensures that the current asset price equals the discounted expected price under the risk-neutral measure:

$$S_0 = e^{-r\Delta t} \cdot \left(qS_1^u + (1-q)S_1^d\right)$$

Note that we assumed interests are compounded continuously as it aligns with how asset returns and discounting are usually modeled in the market.

The Binomial Tree for European Asian Call Options

A European Asian call option is a type of financial derivative whose payoff depends on the average price of the underlying asset over a predetermined time horizon. Unlike a European option, which pays off based solely on the asset price at maturity. Therefore, a European Asian option reflects the price path, making it a path-dependent contract. For example, the payoff of a European Asian call option is given by:

$$A_T = \max(\bar{S} - K, 0) = \max\left(\frac{1}{1+n}\sum_{t=0}^{n} S_t - K,\ 0\right)$$

where $\bar{S} = \frac{1}{n+1}\sum_{t=0}^{n} S_t$ is the arithmetic mean of the underlying asset prices over the time steps t = 0, 1, ..., n, and K is the strike price.

For standard European options, we only need to consider the terminal asset prices at maturity. In contrast, for Asian options, we must account for the entire path of prices leading to maturity, since the average depends on all intermediate prices. This requires the tree to store not only the current asset price at each node but also the partial sum of prices along each path. As a result, the state space becomes more complex, and the tree no longer recombines fully; different paths can reach the same price at time t, but with different running averages.

Despite this added complexity, the risk-neutral valuation principle remains the cornerstone of the pricing method. Under the risk-neutral measure, all assets are assumed to grow at the risk-free rate r, and the fair price of the option is equal to the expected value of its discounted payoff. At each node in the binomial tree, the option value V_t is given by:

$$V_t = e^{-r\Delta t}\left(q \cdot X_{t+1}^{\mathrm{u}} + (1-q) \cdot X_{t+1}^{\mathrm{d}}\right),$$

where:

q is the risk-neutral probability of an up move,

$\Delta t = \frac{T}{n}$ is the time step size,

V_{t+1}^{u} and V_{t+1}^{d} are the option values at the next time step following an up or down move,

$e^{-r\Delta t}$ is the per-period discount factor under continuous compounding.

At maturity $t = T$, the option value at each terminal node is determined by:

$$A_T = \max\left(\frac{1}{n+1}\sum_{t=0}^{n} S_t - K, 0\right).$$

Starting from these terminal payoffs, one applies backward induction through the tree, using the formula above at each step to compute the arbitrage-free price X_0 at time zero.

4. Methodology

Parameter calibration

Maturity $T = 6$ months

Number of periods in the binomial tree $n = 25$

Risk-free rate per annum $r = 1\%$

Initial Microsoft stock price on 28.04.2025 $S_0 = \$391.16$

We assume the strike price is at the money, i.e. $K = \$391.16$

Up and down factors:

$$u = e^{\sigma\sqrt{\Delta t}},\ d = e^{\sigma\sqrt{\Delta t}}$$

Where $\Delta t = \frac{T}{n} =$ 0.02 years and σ is the annualized volatility of the stock return.

The stock movement probabilities are:

$$p(u) = \frac{1}{2},\ p(d) = \frac{1}{2}$$

We assume:

$$\sigma = \sigma_{daily} \cdot \sqrt{250}$$

We compute the annualized volatility σ as follows:

1. Compute the daily log returns:

$r_t = \log\left(\frac{S_{t+\Delta t}}{S_t}\right)$, where r_t is the daily log return at time t

2. Compute the daily volatility:

$\sigma_{daily} = sd(r_t)$

3. Annualize the volatility:

$\sigma = \sigma_{daily} \cdot \sqrt{250}$

Where 250 is the number of trading days in a year.

The following program was used to make these calculations:

Which executes the previous calculations and returns

```
# Load historical data
data = pd.read_csv("filtered_data.csv")
data["Close"] = data["Close.Last"].replace('[\$,]', '', regex=True).astype(float)
data["log_return"] = np.log(data["Close"] / data["Close"].shift(1))

# Compute volatility from historical returns
sigma_daily = data["log_return"].std(skipna=True)
sigma_annual = sigma_daily * np.sqrt(250)
print(sigma_annual)
```

Figure 3 Code snippet, calculation of log returns and volatility

Which executes the previous calculations and returns $\sigma = 0.30518697018420116$

Payoff per path computation

To be able to perform backward induction later to determine the fair price of the option, we need to know all the possible payoffs that the option can lead to and their associated probabilities. For a binomial tree with $n = 25$, there are $2^{25} = 33{,}554{,}432$ unique paths, with some of them leading to the same payoff. It is important to take them into account because several paths leading to a recurring payoff increase the weight of this particular payoff in the option price computation later.

Here is an example of up-down patterns that differ/match in the final payoff (all ending in the same end node):

$$A_{u-u-d-d} \neq A_{d-d-u-u}$$

but

$$A_{u-d-d-u} = A_{d-u-u-d}$$

To determine the fair price of the underlying asset and the corresponding payoff of the option in every possible case, we ran a program that computes both values for each unique path and stores the results in dedicated .csv files. We decided to organize the results in 26 separate .csv files, each containing the results for a specific number of j up-moves.

```python
for j in range(n + 1):
    up_combinations = list(itertools.combinations(range(1, n+1), j))

    average_prices = []
    payoffs = []

    for up_pos in up_combinations:
        path = [S0]
        for t in range(1, n+1):
            prev_price = path[-1]
            if t in up_pos:
                path.append(prev_price * u)
            else:
                path.append(prev_price * d)
        avg_price = np.mean(path)
        payoff = max(avg_price - K, 0)
        average_prices.append(avg_price)
        payoffs.append(payoff)

    df = pd.DataFrame({
        "Up_Moves": [j]*len(average_prices),
        "Average_Price": average_prices,
        "Payoff": payoffs
    })

    df.to_csv(f"j-{j}.csv", index=False)
```

Figure 4 Code snippet, path simulation

Risk-neutral probabilities

In order to later determine the fair price of the option with backward induction or normal approximation of the binomial distribution, we need to calculate the risk-neutral probabilities. These are theoretical probabilities based on the assumption that we are in a risk-neutral world — i.e., we assume that investors are indifferent to risk and do not demand additional return for taking on more risk.

Mathematical definition:

$$q = \frac{e^{r\Delta t} - d}{u - d}$$

Where q is the risk-neutral probabilities.

Applying the formula returns $q = 0.491528$

Note that this means that in the risk-neutral world, the up and down factors are

$$p(u) = q, p(d) = 1 - q$$

and that we will use these probabilities for the computation of the fair price of the option in the backward induction and the normal approximation processes.

Backward induction pricing process

We apply backward induction using a two-dimensional matrix $V = t \times j$, where each entry $[t][j]$ stores the expected value of the option at time t after considering all the paths resulted in the specific j up-moves. We start to fill the entries for $t = 25$ and we apply then the backward induction formula:

$$V(t, j) = e^{-r\Delta t}[q \cdot V(t + 1, j + 1) + (1 - q) \cdot V(t + 1, j)]$$

Which returns $V(0,0) = 21.3932$

Normal approximation method for pricing

To check if our backward induction pricing process is credible, we can implement a faster way to compute an **approximation** of the Asian option's value assuming that the average stock prices over time follows a normal distribution. Using the simulated data from the 2^{25} possible paths, we can compute the empirical mean μ and the standard deviation σ of the average prices across all the paths.

Once we have the empirical μ and σ, we can compute the expected payoff of the option using the following formula:

$$E\left[\max\left(\overline{S} - K, 0\right)\right] = (\mu - K) \cdot \Phi(d) + \sigma \cdot \phi(d), \text{with } d = \frac{\mu - K}{\sigma}$$

We now discount the expected payoff by the risk-free rate in order to get the approximation of the fair price of the option:

$$V_0 = e^{-rT} \cdot E\left[\max\left(\overline{S} - K, 0\right)\right]$$

Which returns \$22.3595.

5. Results

Risk-Neutral Probabilities

We applied risk-neutral valuation to compute the expected value of the option. The risk-neutral up-move probability q is defined as:

Using the parameters:

$r = 0.01$ (annual risk-free rate),

$\sigma = 0.212$ (annual volatility),

$T = 0.5$ (time to maturity),

$n = 25 \Rightarrow \Delta t = T/n = 0.02$,

we obtained:

$q \approx 0.4965$

This value was used in all subsequent computations to discount expected future payoffs under the risk-neutral measure.

Backward Induction and Option Price

We computed the option price at $t = 0$ using backward induction on the binomial tree. Terminal payoffs were precomputed and grouped by number of up moves j.

The option value at node (t,j) was computed using:

$$V(t,j) \;=\; e^{-r\Delta t}[q \cdot V(t+1, j+1) + (1-q) \cdot V(t+1, j)]$$

Starting from the expected terminal payoffs $V(25, j)$, we iterated backward to $t = 0$. The final arbitrage-free price of the Asian call option was:

$21.39

This reflects the fair value of the path-dependent payoff under risk-neutral pricing.

Robustness Check

We performed a sensitivity analysis by varying r and σ, while keeping other parameters fixed. The resulting option prices are shown below:

Parameter	Value	Option Price ($)
r	0.005	21.000253
r	0.010	21.393164
r	0.015	21.790383
r	0.020	22.191901
σ	0.150	22.149559
σ	0.200	21.393164
σ_{true}	0.212	21.235151
σ	0.250	20.771533
σ	0.300	20.219866

Table 1: Robustness check results: option price under varying r and σ

We observed that increasing the interest rate r led to a mild increase in the option price, as expected due to the effect of lower discounting. Interestingly, higher volatility σ led to a *decrease* in the option price — contrary to standard option pricing results. This behavior is explained by the averaging feature of Asian options, which smooths out extreme price movements and reduces the upside impact of volatility. Moreover, we observe that the option price changes are modest among changes in r and σ, which strengthens our pricing model's reliability.

Normal Approximation

To validate our binomial tree results, we applied the normal approximation to the distribution of average prices. Assuming $\bar{S} \sim N(\mu,\sigma)$, the expected payoff of the Asian call is approximated as:

Using empirical estimates from the simulation:

$\mu = E[\bar{S}]$,

$\sigma = \text{std}(\bar{S})$, • $K = 391.16$,

We obtained the following result:

22.36$

This approximation is close to the option price of \$21.39 computed with the binomial model, confirming that the normal approximation provides a reliable and computationally efficient method for pricing Asian options in this setup.

6. Conclusion

In this report, we implemented a multi-period binomial tree model to price a European Asian call option on Microsoft Corporation stock, using historical data and risk-neutral valuation principles. The path dependency involved in pricing the European Asian call option introduced an additional layer of complexity compared with a simple European call option.

We used historical Microsoft stock price data, which was obtained from the NASDAQ website. In a subsequent step, we cleaned and filtered the data to cover the period from January 1, 2020, to April 28, 2025, and calculated the daily log returns as an additional metric. A histogram analysis of these daily log returns confirmed a roughly normal distribution with slight leptokurtosis, highlighting the presence of sparse significant market movements.

Our pricing model yielded an option value of \$21.39 using the backward induction method. To cross-validate this result, we applied a normal approximation method using the simulated distribution of average prices, which gave a closely aligned value of \$22.36. This consistency between two independent methods demonstrates the robustness of our approach.

We also performed a sensitivity analysis on interest rates and volatility. As expected, increasing the interest rate slightly increased the option price due to increased discounting. However, we observed an inverse relationship between volatility and the option price — a counterintuitive result when compared to standard European options. In our case, higher volatility led to lower option prices. This outcome can be attributed to the averaging feature of Asian options, which dampens the effect of extreme movements: although volatility introduces more variation, it also brings in lower prices that reduce the average and therefore the expected payoff. This empirical observation highlights the unique behavior of path-dependent instruments and the limitations of applying classical option pricing intuitions to exotic derivatives.

Overall, our project demonstrates how theoretical models, when carefully implemented and validated with real data, can effectively price complex financial instruments. It also reinforces the importance of understanding the structural differences between standard and exotic options, especially when interpreting model results.