The Binomial Model for US Stock Option Pricing: A Dynamic Programming Approach

Nathaniel Jonathan Rusli - 13523013[1,2]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13523013@std.stei.itb.ac.id, [2]omgitsnathaniels@gmail.com

***Abstract*—This paper presents an algorithmic implementation of the Binomial Option Pricing Model (BOPM) using a bottom-up dynamic programming approach to efficiently value US stock options. Unlike closed-form solutions such as Black-Scholes, which are limited to European-style options, the binomial model offers greater flexibility by supporting early exercise features, making it applicable to both European- and American-style contracts. This study focuses on translating the model's recursive logic into a computationally efficient structure. Key input variables, including spot price, historical volatility, dividend yield, and risk-free rate, are derived from real market data. The model is applied to high-volume US equities, with results showing theoretical consistency: for non-dividend-paying stocks, American call options exhibit no optimal early exercise opportunity and therefore match their European counterparts in value. A sensitivity analysis on the discretization parameter *N* reveals convergence behavior in option prices as the number of time steps increases. These findings validate the stability and reliability of the binomial method under varying model resolutions. Overall, the study demonstrates how dynamic programming reduces the model's time complexity from exponential to polynomial, making BOPM a practical, scalable, and robust tool for quantitative finance and algorithmic analysis.**

***Keywords*—Binomial model, Option pricing, Dynamic programming, US stock options, Financial engineering**

I. Introduction

In modern finance, financial derivatives such as stock options are indispensable tools for risk management and investment strategies. The accurate valuation of these instruments is critical for market participants to hedge exposure and speculate on future asset movements. The groundbreaking Black-Scholes model provides a foundational continuous-time framework for this purpose [1].

However, its practical application is constrained by rigid assumptions, e.g., constant volatility and dividend payouts, and its difficulty in pricing American-style options which can be exercised prior to expiration. As an alternative, the Binomial Option Pricing Model (BOPM) offers a more intuitive and flexible discrete-time approach, capable of overcoming these limitations by modeling asset price evolution as a lattice of possible future outcomes [2].

The computational core of the BOPM involves a backward induction process, starting from the option's known value at expiration. A naive recursive implementation of this process is highly inefficient due to redundant calculations of the same states, leading to an exponential time complexity. This makes Dynamic Programming (DP) the ideal strategy, as it systematically solves each subproblem once and stores the result, transforming the problem into one with polynomial-time complexity.

This paper presents a comprehensive analysis of the Binomial Option Pricing Model from an algorithmic standpoint. The primary objectives are to develop an efficient implementation of Binomial Option Pricing Model using a bottom-up dynamic programming approach and to evaluate its performance in pricing both European- and American-style call options. By focusing on numerical stability, computational complexity, and early exercise behavior, the study highlights dynamic programming as a powerful algorithmic paradigm for solving core problems in quantitative finance, bridging the gap between financial theory and practical, high-performance computation.

II. Theoretical Foundation

A. Financial Derivatives and Stock Options

A derivative is a financial contract that derives its value from an underlying asset, asset group, or benchmark. As agreements between two or more parties, derivatives are traded either on centralized exchanges or in private over-the-counter (OTC) markets, with their prices fluctuating in relation to the value of their underlying assets.

Utilized for risk management, speculation, or leveraging a position, derivatives are primarily classified into two categories. "Lock" products, including futures, forwards, and swaps, legally bind all parties to the contract's terms for its entire duration. Conversely, "option" products provide the holder the right, but not the obligation, to buy or sell the underlying asset at a specified price [3].

A stock option is one form of option product with a stock as its underlying asset. In the equity market, options enable traders to express bullish or bearish expectations on a stock's performance without being obligated to directly buy or sell the stock itself. Option holders may choose not to exercise their contract if it is out-of-the-money, meaning the market price of the stock makes the option unprofitable (e.g., a call option with a strike price higher than the current market price), allowing it to expire worthless. Alternatively, if the option is in-the-money, where exercising would result in a gain (e.g., a put option with a strike price above the market price), traders may sell the contract prior to expiration date to capture its remaining value [4].

Stock options consist of several fundamental parameters that are critical for traders and investors to comprehend [5]:

1) Option Styles: There are two different styles of options when the contract can be exercised:

- *a) European-style option:* Less common in US markets, can only be exercised on the expiration date.
- *b) American-style option:* More flexible and common in US markets, can be exercised anytime between the purchase and expiration date.

2) Expiration Date: This is the date on which the option contract expires.

3) Strike Price: This is the fixed price at which the underlying stock can be bought (in the case of a call option) or sold (in the case of a put option).

4) Contract Size: This defines the number of shares covered by a single option contract. In US equity markets, the contract size is typically 100 shares, meaning each option gives rights over 100 shares of the underlying stock.

5) Premium: The premium is the price paid by the option buyer to the option seller for acquiring the contract, typically reflecting the option's intrinsic value (if applicable) and its time value.

B. Option Pricing Models

Option pricing models are mathematical frameworks that utilize a set of market variables to determine the theoretical fair value of an option. This theoretical value represents an estimate of what the option should be worth based on all available and relevant inputs, e.g., underlying price and strike price, time to expiration, interest rates, and volatility. Having an accurate estimate of the fair value of an option enables finance professionals to adjust their trading strategies and portfolios. As such, option pricing models serve as powerful tools involved in option trading and risk management practices [6, 7].

Before discussing different option pricing models, it is critical to comprehend the concept of risk-neutral probability,a foundational principle in modern quantitative finance and widely used in derivative pricing. A risk-neutral probability is a theoretical probability measure under which the expected return of all assets equals the risk-free-rate (a theoretical rate of return on an investment with zero risk, typically proxied by short-term US Treasury yield). This concept relies on two fundamental assumptions:

1) The current value of an asset equals the expected value of its future payoffs, discounted at the risk-free rate.

2) The market does not permit arbitrage opportunities.

Under this measure, the price dynamics of assets are adjusted so that investors are considered indifferent to risk when computing expected payoffs. Risk-neutral probability allows option prices to be computed as the present value of expected payoffs under a transformed, arbitrage-free world. Option pricing can be approached through various methodologies [7]:

1) Binomial Model: This model is based on the assumption of fully efficient markets. Under this assumption, the model allows for option valuation at discrete time intervals by constructing a price tree in which the underlying asset can either increase or decrease in value during each time step.

Given the potential future prices of the underlying asset and the strike price of the option, the model calculates the corresponding option payoffs in each scenario. These payoffs are then discounted back to the present using a risk-free rate, resulting in an estimate of the option's current theoretical value.

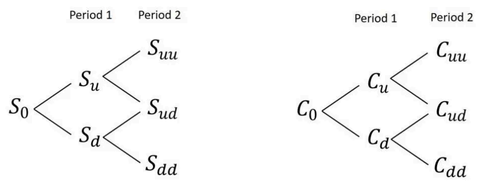

Fig. 1. Two-period binomial tree visualization
[Source: https://corporatefinanceinstitute.com/resources/derivatives/option-pricing-models/, Accessed: Jun. 21, 2025.]

2) Black-Scholes Model: This model was discovered in 1973 by economists Fischer Black and Myron Scholes, whose work received the Nobel Memorial Prize in Economic Sciences in 1997. The Black-Scholes model was developed primarily for pricing European options on stocks which operates under a set of assumptions regarding the distribution of the stock price and the economic environment.

Assumptions about the stock price distribution include:

- *a)* Continuously compounded returns on the stock are normally distributed and independent over time.
- *b)* The volatility of continuously compounded returns is known and constant
- *c)* Future dividends are known.

Assumptions about the economic environment include:

- *a)* The risk-free rate is known and constant.
- *b)* There are no transaction costs or taxes.
- *c)* It is possible to short-sell with no cost and to borrow at the risk-free rate.

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

Fig. 2. Black-scholes equation
[Source: https://en.wikipedia.org/wiki/Black%E2%80%93Scholes_model, Accessed: Jun. 21, 2025.]

3) Monte Carlo Simulation: This model simulates a large number of possible future stock price paths using stochastic processes, and then estimates the option's value by averaging the discounted payoffs across all simulations. It is particularly useful for pricing complex or path-dependent options. However, it is computationally intensive and less suitable for American-style options due to the challenge of modeling early exercise decisions.

Among the three widely recognized option pricing models, this study focuses on the binomial model. The Black-Scholes model, while elegant and analytically tractable, relies on rigid assumptions and is restricted to European-style options which are less common in US markets. Monte Carlo simulation, on the other hand, is highly flexible and well suited for pricing complex, path-dependent derivatives, yet its computational intensity and limitations in modeling early exercise make it less appropriate for valuing American-style options.

The binomial model serves as a compelling middle ground, i.e. mathematically structured yet practically adaptable. By modeling asset price movements in discrete time steps, it enables the construction of the recombining tree that effectively captures a range of market dynamics. Critically, it perfectly supports the valuation of American-style options by incorporating the possibility of early exercise at each node. In addition, its recursive nature aligns naturally with dynamic programming strategies and principles, making it particularly well-suited for an algorithmic implementation, which is the focus of this paper.

C. Binomial Model for Option Pricing

The binomial model is a discrete-time framework for calculating the fair value of financial options by modeling possible movements in the underlying asset's price over successive time steps. Originally introduced by Cox, Ross, and Rubinstein in 1979, the model constructs a recombining binomial tree that represents the potential evolution of the underlying asset over time. At each node in the tree, the asset price may either increase by an up factor (u) or decrease by a down factor (d), thus creating a lattice of possible future prices. This structure allows for flexibility in handling early exercise features and non-linear payoff structures, making it especially useful for pricing both European-style options and American-style options [8].

Before constructing the binomial tree, several key inputs or variables must be defined to parameterize the model accurately. These inputs govern how the option behaves under the assumptions of the model and directly influence the valuation outcome:

1) Spot Price (S_0): The current market price of the underlying asset. It serves as the root of the binomial price tree.

2) Strike Price (K): The fixed price at which the option can be exercised. It defines the payoff structure at each node.

3) Time to Maturity (T): The total time remaining until the option expires, typically expressed in years. It determines the number of discrete steps N used in the tree:

$$\Delta t = \frac{T}{N} \tag{1}$$

4) Volatility (σ): The annualized standard deviation of the underlying asset's continuously compounded returns. It measures the uncertainty in the asset price. This input is critical in calculating the up and down factors

$$u = e^{\sigma\sqrt{\Delta t}},\ d = \frac{1}{u} = e^{-\sigma\sqrt{\Delta t}} \tag{2}$$

5) Risk-Free Rate (r): The annualized return of a riskless investment, often proxied by short-term US Treasury yield. It is utilized to discount figure option payoffs:

$$e^{-r\Delta t} \tag{3}$$

6) Dividend Yield (q): The continuous yield paid by the underlying asset. If dividend is not applicable, calculations incorporating dividend yield may use the value $q = 0$.

7) Risk-Neutral Probability (p): Under the risk-neutral measure, the expected payoff of the option is computed using adjusted probabilities. The probability of an upward movement is:

$$p = \frac{e^{(r-q)\Delta t} - d}{u - d} \tag{4}$$

With the probability of downward movement:

$$1 - p \tag{5}$$

Once these parameters are established, the model proceeds with constructing the binomial price tree and computing the option value through backward induction. At each node (i, j) of the tree, the asset price is given by:

$$S_{i,j} = S_0 \,.\, u^j \,.\, d^{i-j} \tag{6}$$

where i denotes the time step, and j is the number of upward movements. This expression defines the structure of the tree and serves as the basis for calculating the terminal payoffs at maturity.

Let S_0 be the current spot price of the underlying asset, and let the option mature at time T, divided into N discrete steps. The time step Δt is given by Equation (1). The model uses the assumption of constant volatility σ and risk-free rate r. Under the Cox-Ross-Rubinstein (CRR) formulation, the up factor (u) and down factor (d) are given by Equation (2). To remain arbitrage-free, the model employs risk-neutral valuation, in which the expected value of the option payoff is discounted at the value of r. The corresponding risk-neutral probability p of an upward price movement is given by Equation (4).

Once the tree of possible asset prices has been completely constructed, the option value is determined by the backward induction process. For each terminal node $j \in \{0, 1, ..., N\}$, the call and put option payoff (applicable for both European-style option and American-style option) are respectively computed as follows:

$$C_{N,j} = max(S_{N,j} - K,\ 0) \tag{7}$$

$$P_{N,j} = max(K - S_{N,j},\ 0) \tag{8}$$

Then, working backwards from maturity to the present, the value of each exercise opportunities at each node (i, j) is obtained by discounting the expected value at the next step:

$$V_{i,j} = e^{-r\Delta t}.\,(q.\,V_{i+1,j+1} + (1 - q).\,V_{i+1,j}) \tag{9}$$

For American-style options, the model incorporates early exercise opportunities at each node by taking the maximum of the continuation value and the immediate payoff:

$$V_{i,j} = max(Payoff_{i,j},\ e^{-r\Delta t}(q.\,V_{i+1,j+1} + (1 - q).\,V_{i+1,j})) \tag{10}$$

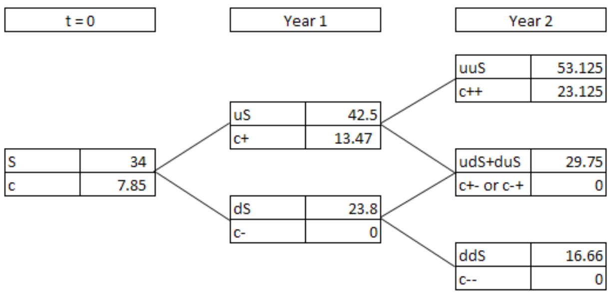

Fig. 3. Example of binomial tree visualization
[Source: https://xplaind.com/552187/binomial-options-pricing-model, Accessed: Jun. 21, 2025.]

D. Dynamic Programming Strategy

Dynamic programming (DP) is an algorithm designed to optimize recursive approaches by eliminating redundant computations of identical subproblems. Its fundamental principle lies in storing the results of previously solved subproblems to ensure each is computed only once. Typically, a dynamic programming solution begins with a recursive formulation where the problem exhibits overlapping subproblems, meaning the same function is invoked multiple times with the same input parameters. To enhance computational efficiency, the outcomes of these recursive calls are cached and reused. There are two primary strategies of dynamic programming [10]:

1) Top-Down Approach (Memoization): This approach keeps the solution recursive and adds a memoization to avoid repeated calls of the same subproblems. Before executing a recursive call, the algorithm first checks the memoization table to determine whether the subproblem has already been solved. If not, the recursive computation proceeds, and upon completion, the result is stored in the table for future reuse.

2) Bottom-Up Approach (Tabulation): This approach begins with the smallest subproblems and incrementally builds up to the final solution. Rather than using recursion, it employs an iterative formulation to eliminate function call overhead. A dynamic programming table is initialized, where base cases are filled explicitly. Subsequently, the remaining entries are computed using the recursive relationship, applied directly on the table entries without making any actual recursive calls.

The Binomial Option Pricing Model naturally aligns with the core principles of dynamic programming:

1) Optimal Substructure: The value of an option at each node depends only on the values at its two immediate successor nodes.

2) Overlapping Subproblems: Intermediate nodes can be reached via multiple paths, and a naive recursive implementation would redundantly recompute subtrees multiple times.

The backward induction method employed in the binomial model is a direct application of the bottom-up dynamic programming strategy. It begins from the terminal payoffs at maturity and iteratively computes values backward to the root, efficiently building the final solution from simpler subproblems.

By using dynamic programming, the computational efficiency is significantly enhanced. It reduces the time complexity of option valuation from exponential $O(2^N)$ in naive recursion to polynomial $O(N^2)$ with the binomial model. This makes multi-step pricing models computationally feasible and scalable for practical financial applications.

E. US Stock Options and Market Context

The United States (US) hosts the world's most active and liquid equity options market, where stock options are among the most widely traded financial instruments. This activity is heavily concentrated in a select group of high-profile companies. For instance, as of June 23, 2025, the most actively traded stock options were dominated by technology and growth stocks, with Tesla (TSLA) and Nvidia (NVDA) leading with daily volumes exceeding one million contracts each. Other highly traded options on this day included Advanced Micro Devices (AMD), Hims & Hers Health (HIMS), and Apple (AAPL). While the exact daily rankings may fluctuate, options on mega-cap stocks such as Amazon (AMZN) and Meta Platforms (META) also consistently rank among the most traded, underscoring their central role in the strategies of both institutional and retail traders [11].

Given their high trading volume, US stock options provide a relevant and representative case study for evaluating option pricing models. While the majority of exchange-listed equity options in the US follow the American-style format, the Binomial Option Pricing Model offers the flexibility to handle both European-style (exercise at maturity) and American-style (early exercise allowed) options. As such, the model is well-suited for a broad range of equity derivatives. This study adopts an experimental design that accommodates both option types, enabling a comprehensive and realistic application of the model while emphasizing the importance of numerical accuracy in real-world financial computations.

III. Implementation

A. Programming Language and Libraries

For the implementation of binomial option pricing model, Python is used as the programming language, leveraging its extensive ecosystem of supporting libraries. These libraries streamline the process of data collection, processing, analysis, and visualization:

1) yfinance: Used to fetch historical stock data, including price and dividend yields, directly from Yahoo Finance.

2) pandas: Used for data manipulation and exporting variables to CSV files.

3) numpy: Provides efficient numerical operations for building the binomial price tree and performing backward induction.

4) matplotlib: Enables visualization of option pricing results and insights.

B. Data Collection and Variables Processing

To calibrate the binomial model, we require several key inputs or variables derived from historical market data, e.g., the spot price (S_0), risk-free rate (r), and dividend yield (q).

```
TICKERS         = ["TSLA", "AAPL"]
HISTORY_DAYS    = 178
RISK_FREE_RATE  = 0.04273
```

The equity tickers analyzed in this study include TSLA (Tesla) and AAPL (Apple), representing high-volume US stock options. A 178-day historical window, spanning from June 24, 2025 to the selected expiration date of December 27, 2025, is used to estimate the recent volatility of the underlying assets. The risk-free interest rate is fixed at 4.273%, corresponding to the U.S. 3-month Treasury yield as of June 24, 2025.

The spot price (S_0) is defined as the most recent closing price of the asset over the selected window:

```
spot_price = float(close_prices.iloc[-1])
```

The annualized historical volatility (σ) is computed from the standard deviation of log returns, scaled by the square root of 252 trading days:

```
log_returns = np.log(close_prices /
close_prices.shift(1)).dropna()
volatility  = float(log_returns.std() *
np.sqrt(252))
```

The dividend yield (q) is estimated based on the sum of dividends paid over the past 12 months, normalized by the current spot price:

```
recent_div = dividends[dividends.index >
(END_DATE - timedelta(days=365))]
annual_div = recent_div.sum()
dividend_yield = annual_div / spot_price
```

The computed values for each ticker are stored in a structured CSV format for later use in the pricing model. This modular approach ensures that the core variables required for binomial option pricing are efficiently extracted and persisted for subsequent modeling. The following shows an example of the exported variables for TSLA, including the spot price, historical volatility, risk-free rate, and dividend yield inside a CSV file:

```
ticker,spot_price,volatility,risk_free_rate,di
vidend_yield
TSLA,348.67999267578125,0.778321788296038,0.04
273,0.0
```

C. Binomial Option Pricing Model

The binomial pricing model is implemented using a bottom-up dynamic programming approach to efficiently compute the values of both European and American call options. The model is parameterized using market-relevant values for Tesla (TSLA) as of June 24, 2025:

```
SPOT_PRICE      = 351.70
STRIKE_PRICE    = 400
VOLATILITY      = 0.7784
RISK_FREE_RATE  = 0.04273
T               = 175 / 365
N               = 100
DIVIDEND_YIELD  = 0.0
```

To begin, the tree parameters including time increment, up and down movement factors, and risk-neutral probability are computed using Equation (1), (2), and (4) respectively:

```
dt = T / N
u = np.exp(volatility * np.sqrt(dt))
d = 1 / u
p = (np.exp((risk_free_rate - dividend_yield)
* dt) - d) / (u - d)
```

The binomial tree is generated in forward fashion using Equation (6). Only the terminal layer is explicitly computed, as backward induction is performed in-place:

```
for j in range(N + 1):
    price_tree[N, j] = spot_price * (u ** j) *
    (d ** (N - j))
```

At maturity ($t = T$), the call option payoff is calculated using Equation (7):

```
for j in range(N + 1):
    european[N, j] = max(price_tree[N, j] -
    strike_price, 0)
    american[N, j] = european[N, j]
```

This expression is used to initialize both the European-style and American-style option value arrays in the model. It is important to note that this study focuses exclusively on call options, and does not explore put options or their payoff structures. This decision is made to maintain a consistent analytical scope, particularly in examining early exercise behavior and dynamic programming efficiency within the binomial framework.

Utilizing backward induction, the option price is recursively computed from the terminal nodes back to the root by discounting the expected future value. European-style options are priced using Equation (9), whereas American-style options are computed using Equation (10), which additionally considers the possibility of early exercise at each node:

```
for i in range(N - 1, -1, -1):
    for j in range(i + 1):
        # European
        cont_val = np.exp(-risk_free_rate *
        dt) * (p * european[i + 1, j + 1] + (1
        - p) * european[i + 1, j])

        european[i, j] = cont_val

        # American
        cont_val_am = np.exp(-risk_free_rate *
        dt) * (p * american[i + 1, j + 1] + (1
        - p) * american[i + 1, j])

        early_ex       = max(price_tree[i, j]-
        strike_price, 0)

        american[i, j] = max(early_ex,
        cont_val_am)
```

As previously mentioned, the binomial model is capable of handling both European-style and American-style call options within the same tree structure. The key distinction lies in the exercise rules: European options can only be exercised at maturity, while American options allow early exercise at any point before expiration. This study demonstrates the model's flexibility by evaluating both option styles, illustrating how the binomial method accommodates different contract specifications under a unified computational approach.

D. Number of Discrete Time Steps Sensitivity Test

Among the parameters required by the binomial model, the number of discrete time steps (N) is unique in that it does not rely on historical data or market-derived inputs. Instead, N is a user-defined parameter that governs the resolution of the binomial tree and the granularity of price evolution across the time horizon. In this study, N is treated as a tunable hyperparameter, and its impact on the computed option prices is empirically evaluated.

Using fixed inputs for the AAPL stock (spot price, volatility, risk-free rate, dividend yield, and maturity), we vary N across multiple values, specifically N = 10, N = 50, N = 100, N = 250, N = 500, N = 1000, N = 2500, and N = 5000, to observe how the discretization level affects call option prices. This experimentation helps validate convergence behavior of the model and assess how coarse or fine binomial grids influence pricing accuracy and early exercise evaluation.

IV. Results and Discussion

A. Binomial Model Valuation

To evaluate the performance of the binomial option pricing model, we simulate a call option on Tesla (TSLA) using the input parameters shown in Table I. These values represent realistic market conditions as of June 23, 2025 and are used to build a 100-step (N = 100) binomial tree.

TABLE I
Tesla binomial option pricing model inputs

Variables[a]	*Value*[b]
Spot Price (S_0)	\$351.70
Strike Price (K)	\$400.00
Volatility (σ)	0.7783
Risk-Free Rate (r)	0.04273
Time to Maturity (T)	178/365
Discrete Time Steps (N)	100
Dividend Yield (q)	0.0

The results of the model for both European and American call options are summarized in Table II.

TABLE II

BINOMIAL MODEL VALUATION RESULT

Call Option Styles[a]	*Valuated Premium*[b]
European-style option	\$61.1437
American-style option	\$61.1437

As seen in Table II, both the European and American call options yield the same valuation of \$61.1437. This is significantly higher than the observed market premium of \$45.30 as of June 23, 2025. Given that the spot price (\$351.70) is below the strike price (\$400.00), the option is currently out-of-the-money, which likely explains the lower market premium.

Despite the American call option theoretically offering early exercise flexibility, its value is identical to the European counterpart. This phenomenon is expected in the absence of dividends and is further illustrated by the early exercise map shown in Figure 4.

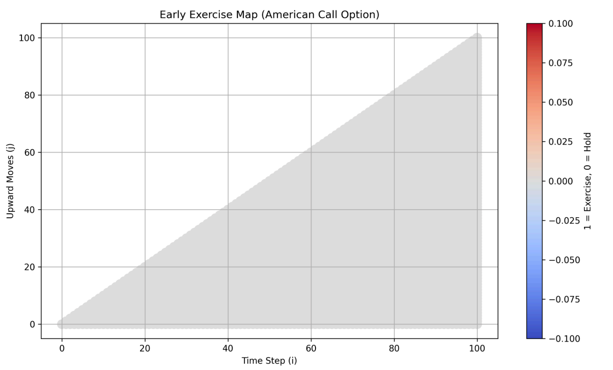

Fig. 4. American-style option vs European-style option option exercise [Source: Author's visualization using Matplotlib]

Figure 4 visualizes the early exercise decision across all nodes in the binomial tree:

1) The x-axis represents the time steps i from initiation to maturity.

2) The y-axis represents the number of up-movements j at each step.

3) Node colors indicate the exercise behavior:

a) *Gray (0):* The option is held (not exercised).

b) *Red (1):* The option is exercised early.

In this case, all nodes are gray, indicating that early exercise is never optimal at any point in the tree. This result aligns with theoretical expectations, i.e. for American call options without dividends, it is never optimal to exercise early. As a consequence, the early value is zero, and the American option behaves identically to its European counterpart, validating the correctness and consistency of the model implementation.

B. Number of Discrete Time Steps Sensitivity Test

To examine the impact of the number of discrete time steps on the valuation accuracy of the binomial model, we conduct a sensitivity analysis on an Apple (AAPL) European-style call option to ensure that the pricing behavior (specifically the valuation at expiry) is uniform and consistent. The input parameters used for this test are summarized in Table III, and are consistent with real market data as in the TSLA case, except for the number of time steps N, which is varied across a wide range for convergence analysis.

TABLE III

APPLE BINOMIAL OPTION PRICING MODEL INPUTS

Variables[a]	*Value*[b]
Spot Price (S_0)	\$201.425
Strike Price (K)	\$250.000
Volatility (σ)	0.411
Risk-Free Rate (r)	0.04273
Time to Maturity (T)	178/365
Dividend Yield (q)	0.005

The number of discrete steps tested includes N = 10, N = 50, N = 100, N = 250, N = 500, N = 1000, N = 2500, and N = 5000. The resulting option premiums for each case are documented in Table IV.

TABLE IV

DISCRETE TIME STEPS TO EUROPEAN-STYLE CALL OPTION SENSITIVITY TEST RESULT

Number of Time Steps[a]	*Valuated Premium*[b]
10	\$9.1679
50	\$9.2786
100	\$9.2314
250	\$9.2023
500	\$9.2261
1000	\$9.2166
2500	\$9.2191
5000	\$9.2198

To visualize the convergence behavior, the results are also plotted in Figure 5.

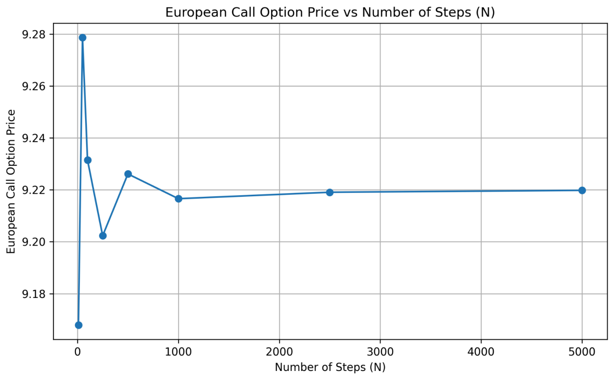

Fig. 5. Discrete time steps to European-style call option price [Source: Author's visualization using Matplotlib]

As shown in Figure 5, the option price initially fluctuates across low values of N, with noticeable discrepancies between $N = 10$ and $N = 250$. These oscillations reflect the discretization error inherent in low-resolution trees. However, as N increases beyond 1000, the option price stabilizes and converges towards approximately $9.22, indicating that the model becomes numerically stable at finer resolutions. This confirms a well-known property of the binomial model: as the number of time steps increases, the model's valuation converges to its theoretical continuous-time limit (as approximated by the Black-Scholes model for European options). Therefore, selecting a sufficiently large N is essential to ensure accuracy, while avoiding unnecessary computational overhead.

V. Conclusion

This paper has successfully demonstrated the implementation of the Binomial Option Pricing Model using a bottom-up dynamic programming strategy for valuing both European and American-style call options. The application of this approach to real-world market data for stocks like Tesla (TSLA) and Apple (AAPL) yielded several key insights. First, the model correctly validated the financial principle that for an American call option on a non-dividend-paying stock, its value is identical to its European counterpart as there is no optimal time for early exercise. This was confirmed through both the valuation results and the early exercise decision map.

Second, the sensitivity analysis on the number of discrete time steps (N) empirically verified the model's convergence property. As N increases, the calculated option price stabilizes, highlighting the trade-off between computational granularity and accuracy. The study ultimately affirms that dynamic programming is a highly effective and efficient algorithmic approach for option valuation. By reducing the problem's time complexity from an exponential $O(2^N)$ to a polynomial $O(N^2)$, it transforms a theoretically elegant model into a computationally feasible tool, perfectly suited for practical applications in quantitative finance.

VI. Acknowledgment

The author expresses heartfelt gratitude to God Almighty for granting the strength, perseverance, and opportunity to successfully complete this paper. The author also wishes to express profound gratitude to Dr. Nur Ulfa Maulidevi, S.T, M.Sc., the lecturer for the IF2211 Algorithm Strategy course, for her unwavering guidance and inspiration throughout his time teaching the students.

VII. Appendix

The complete source code used for the implementation of the binomial option pricing model is available on GitHub. Access the code repository here: GitHub Repository Link.

References

[1] A. Hayes, "Black-Scholes Model: What It Is, How It Works, Options Formula," *Investopedia*, May 17, 2025. [Online]. Available: https://www.investopedia.com/terms/b/blackscholes.asp. [Accessed: Jun. 21, 2025].

[2] J. Chen, "How the Binomial Option Pricing Model Works," *Investopedia*, Oct. 10, 2024. [Online]. Available: https://www.investopedia.com/terms/b/binomialoptionpricing.asp. [Accessed: Jun. 21, 2025].

[3] J. Fernando, "Understanding Derivatives: A Comprehensive Guide to Their Uses and Benefits," *Investopedia*, Jan. 23, 2025. [Online]. Available: https://www.investopedia.com/terms/d/derivative.asp. [Accessed: Jun. 21, 2025].

[4] A. Loo, "Equity Derivatives," *Corporate Finance Institute*. [Online]. Available: https://corporatefinanceinstitute.com/resources/derivatives/equity-derivatives/. [Accessed: Jun. 21, 2025].

[5] J. Chen, "What Are Stock Options? Parameters and Trading, With Examples," *Investopedia*, Apr. 2, 2024. [Online]. Available: https://www.investopedia.com/terms/s/stockoption.asp. [Accessed: Jun. 21, 2025].

[6] A. Farley, "Factors That Determine Option Pricing," *Investopedia*, Dec. 27, 2023. [Online]. Available: https://www.investopedia.com/trading/factors-determine-option-pricing/. [Accessed: Jun. 21, 2025].

[7] CFI Team, "Option Pricing Models," *Corporate Finance Institute*. [Online]. Available: https://corporatefinanceinstitute.com/resources/derivatives/option-pricing-models . [Accessed: Jun. 21, 2025].

[8] Student Learning Advisory Service, University of Kent, "Binomial models for option pricing," Oct. 2013. [Online]. Available: https://www.kent.ac.uk/learning/documents/slas-documents/Binomial_models.pdf. [Accessed: Jun. 22, 2025].

[9] A. Hayes, "Understanding the Binomial Option Pricing Model," *Investopedia*, Aug. 19, 2024. [Online]. Available: https://www.investopedia.com/articles/investing/021215/examples-understand-binomial-option-pricing-model.asp#toc-black-scholes-vs-binomial-option-pricing-model [Accessed: Jun. 22, 2025].

[10] GeeksforGeeks, "Introduction to Dynamic Programming - Data Structure and Algorithm Tutorials," Jun. 18, 2024. [Online]. Available: https://www.geeksforgeeks.org/dsa/introduction-to-dynamic-programming-data-structures-and-algorithm-tutorials/. [Accessed: Jun. 22, 2025].

[11] [2] Barchart, "Most Active Options." [Online]. Available: https://www.barchart.com/options/most-active/stocks. [Accessed: Jun. 23, 2025].

STATEMENT

I hereby declare that this paper is my own work, not a paraphrase or translation of someone else's paper, and not plagiarism.

Bandung, 24 June 2025

Nathaniel Jonathan Rusli
13523013