Numerical Comparison, Complexity Analysis, and Convergence Verification of Trinomial Tree Option Pricing Models

Abstract

This paper investigates the pricing accuracy of the trinomial tree option pricing model as compared to the binomial tree model, sets up various comparisons for European options, American options and with and without dividends, and discusses the speed of convergence and computational complexity of the model on this basis. The paper uses Python to write the pricing formula function calculations and uses the Kamrad and Ritchken's risk-neutral probabilities to construct the model and finally estimates the order of convergence by three-step differencing. As a result, we find that trinomial trees converge faster, but at a higher computational cost per step, compared to binomial tree models.

1. Introduction

Option pricing is a broad and common yet limited problem in financial mathematics. While analytical solutions like the Black-Scholes-Merton model exist for simple European options, many real-world options have features that make closed-form solutions difficult or impossible to derive. Numerical methods like tree models offer practical alternatives for these complex scenarios.

Tree models discretize the time and underlying asset price movements and represent the changes in asset price movements in a lattice and tree diagram structure. While binary trees (e.g., CRR binary trees) are more widely used, trinomial trees offer potential advantages in terms of flexibility and speed of convergence. In a trinomial tree, the underlying asset can move up, down, or remain at approximately the same level within each time step, thus providing a more nuanced representation of price dynamics.

This paper focuses on implementing and analyzing trinomial tree models for option pricing. We divide the pricing model into four scenarios and discuss more about their features:

a. European options (with and without dividends)
b. American options (with and without dividends)
c. Convergence properties
d. Computational complexity

2.Methodology

2.1 Trinomial Tree Model Construction

The trinomial tree model extends the binomial model by allowing three possible movements in the asset price at each time step. We implement the Kamrad and Ritchken(1993) parameterization, which introduces a stretch parameter λ that can be adjusted to optimize convergence.

According to the Kamrad and Ritchken have chosen to approximate $S_{n\Delta T}$ (0≤n≤N) by a symmetric 3-point Markov chain $\bar{S}_n$(0≤n≤N):

$$\bar{S}_{n+1} = \bar{S}_n * u, \quad p = p_u$$

$$\bar{S}_{n+1} = \bar{S}_n, \quad p = p_m$$

$$\bar{S}_{n+1} = \bar{S}_n * d, \quad p = p_d$$

Like the probability of price increase or decrease in binomial model, trinomial option price model also divides the risk-neutral probability into three steps:

$$p_u = \frac{1}{2\lambda^2} + \frac{\left(r - \frac{\sigma^2}{2}\right) * \sqrt{\Delta T}}{2\lambda\sigma}$$

$$p_m = 1 - \frac{1}{\lambda^2}$$

$$p_u = \frac{1}{2\lambda^2} - \frac{(r - \frac{\sigma^2}{2}) * \sqrt{\Delta T}}{2\lambda\sigma}$$

We can calculate the price at time t based on above probability, for a European Options, the price of it is equal to the sum of present value under each situation:

$$V_t = e^{-r\Delta T}(p_u V_{t+\Delta T}^u + p_m V_{t+\Delta T}^m + p_d V_{t+\Delta T}^d)$$

American Options is different because it can be executed at any point in time (prior to the option's expiration time), its pricing consists of two components: the normal expiration execution and the payoff obtained from the early exercise of the option:

$$V_t = max\,(e^{-r\Delta T}(p_u V_{t+\Delta T}^u + p_m V_{t+\Delta T}^m + p_d V_{t+\Delta T}^d), payoff_{t+\Delta T})$$

Where the payoff depends on whether it's a call or put option:

$$Call: Payoff(i,j) = max\{S(i,j) - K, 0\}$$

$$Put: Payoff(i,j) = max\{K - S(i,j), 0\}$$

2.2 Practical Analysis of Price Calculation in Trinomial Tree Models

The implementation is in Python, using NumPy for numerical computations and Matplotlib for visualization. Assume an option has below parameters:

- Initial stock price (S_0): $90
- Strike price (K): $100
- Risk-free rate (r): 5%
- Volatility (σ): 30%
- Time to maturity (T): 5 years
- Dividend yield (q): 0% or 2%

After taking these parameters into python code, we generate the price of trinomial model with 1600 steps under different types of options, and compare it with the Black-Scholes-Merton model and binomial pricing model's result:

Model	dividend	European option price	American option price
Black-Scholes	Q = 0	28.600335	N/A
Black-Scholes	Q = 0.02	22.747072	N/A
Binomial	Q = 0	28.599524	28.599524
Binomial	Q = 0.02	22.746607	22.831403
Trinomial	Q = 0	28.599589	28.599589
Trinomial	Q = 0.02	22.746957	22.831793

Table 1. Price calculation for three models

For European options, our trinomial tree model converges to the Black-Scholes price, confirming the implementation's accuracy. As expected, American put options have higher prices than their European counterparts due to the early exercise premium.

Notably, for American call options without dividends, the price matches the European call price, confirming the well-known result that early exercise is not optimal for American calls without dividends. However, when dividends are introduced, early exercise may become optimal for American calls.

2.3 Convergence analysis of trinomial model

Based on Abdurakhman's (2023) three-step differencing method for analyzing the order of convergence of a trinomial tree, this paper uses extended generalized three-step differencing in the following way: for three consecutive steps (N, M, P), compute the corresponding option prices, V_N, V_M, and V_P, and estimate the local order of convergence, p, by the following equation:

$$P = \frac{ln(\frac{V_M - V_N}{V_P - V_M})}{ln(M/N)}$$

where a threshold judgment is introduced to skip cases where the difference value is too small and avoid numerical instability. Ultimately the average of all valid local convergence orders is taken as the local convergence order estimate.

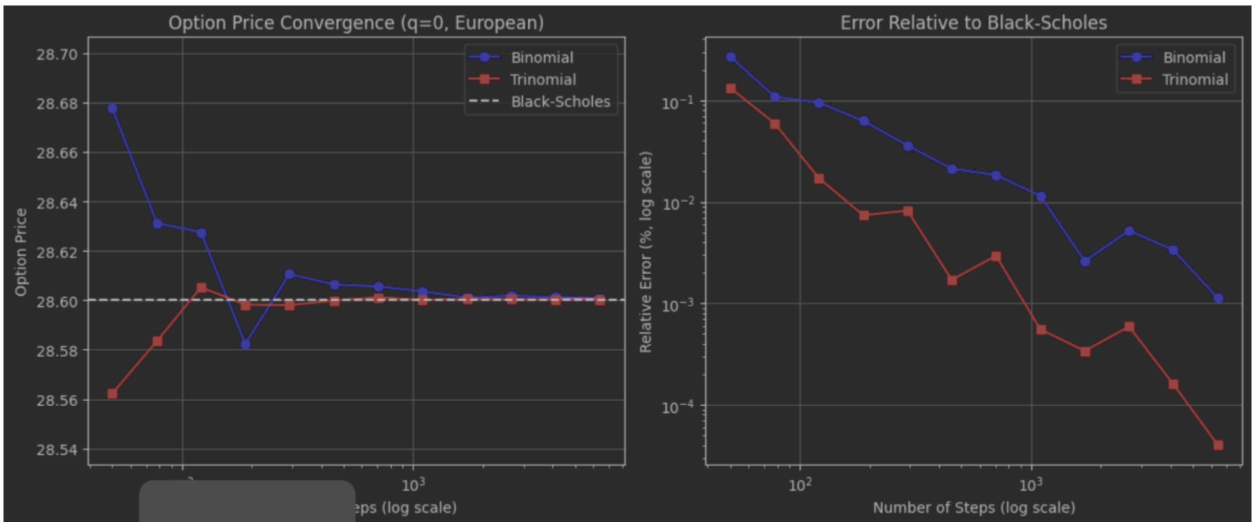

Figure 1. Convergence and error analysis

By analyzing the convergence of the dividend-free European option, we can obtain figure1 and the regression results:

In terms of the convergence trajectory, the trinomial tree model exhibits a more stable convergence process with less oscillations, while the binomial tree model shows obvious oscillations in the convergence process. This indicates that the trinomial tree model has an advantage in numerical stability.

In terms of relative error, the trinomial tree model achieves a much lower error level for the same number of steps. At the maximum number of steps (6400), the error of the trinomial tree model with respect to the Black-Scholes parsing solution is about 0.01%, while the error of the binomial tree model is about 0.1%, which is an order of magnitude difference.

Regarding the regression results on the convergence order, the empirical analysis shows that the average local convergence order of the binary tree model is 0.9657 and the global convergence order is 1.0417, which is in line with the theoretically expected first-order convergence; the average local convergence order of the trinomial tree model is 1.3422 and the global convergence order is 1.4971, which is close to the theoretically expected first-order convergence of 1.5. This result verifies the theoretical advantage of the trinomial tree model in terms of convergence speed.

2.4 Computational complexity analysis of trinomial model

The computational complexity of the models in this paper is evaluated by the actual running time of python functions as follows:

For a range of step lengths (from 50 to 6400), the execution time of the binary tree and trinomial tree models were measured using time.time(), respectively. Subsequently, log-log graphs are used to show the relationship between step length and running time, and the growth rate of time complexity is analyzed. Finally, a computational efficiency metric, defined as the ratio of accuracy to computation time (1/error/time), is introduced to comprehensively evaluate the algorithm performance. The results are listed below:

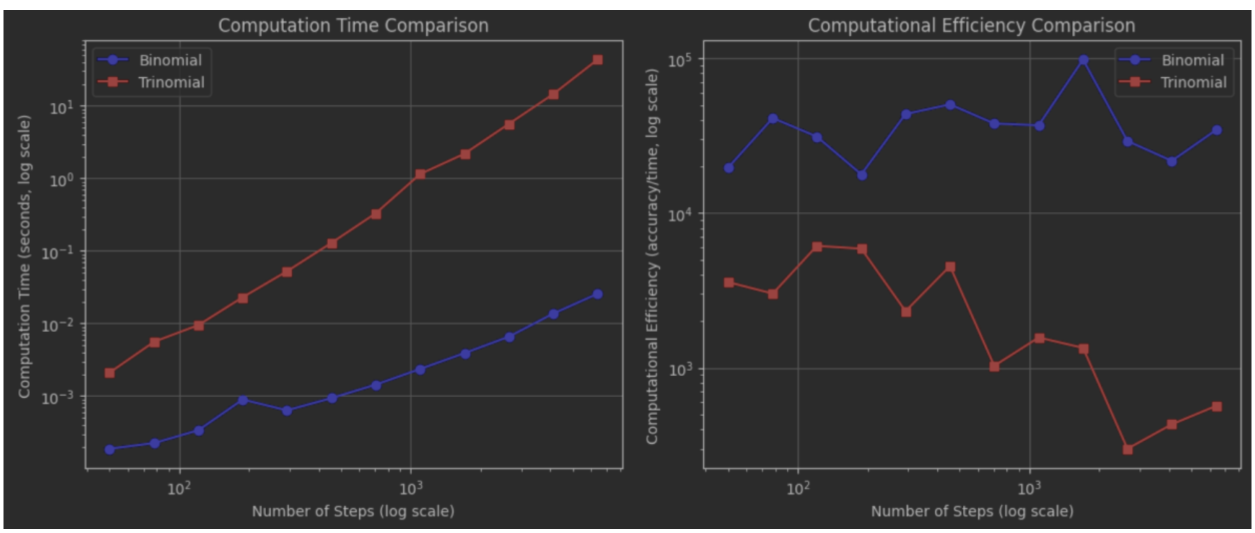

Figure 2. Computational complexity analysis

For the computation time, as shown in the first panel of Figure 2, the trinomial tree model has significantly higher computation time than the binary tree model for the same number of steps.

This is due to the fact that the trinomial tree model has three branches at each node, which is significantly more computational than the two branches of the binary tree model.
In terms of efficiency, although the trinomial tree model provides higher accuracy, the computational efficiency analysis shows that the binary tree model exhibits a better efficiency metric (accuracy/time) at most steps. This suggests that the improvement in accuracy brought about by the use of trinomial trees is not enough to cover the loss in computational time, and that binary trees are still more efficient in practical applications.

3. Conclusion

This paper has conducted a comprehensive analysis of trinomial and binomial tree models for option pricing across various scenarios. Through rigorous numerical experiments, we have demonstrated the trade-offs between accuracy, convergence speed, and computational complexity inherent in these models.
The implementation methodology outlined in this paper, particularly the Kamrad-Ritchken parameterization for trinomial trees, provides a robust framework for option pricing that can be extended to more complex financial instruments.
These findings confirm that trinomial tree models consistently achieve higher convergence rates compared to binomial models, with empirical convergence orders of approximately 1.5 and 1.0 respectively. For European options, the trinomial model converges to the Black-Scholes analytical solution with smaller error margins at equivalent step counts, exhibiting more stable convergence behavior with reduced oscillations. This advantage is particularly notable when pricing options with dividend considerations or early exercise features.
However, this improved accuracy comes at a significant computational cost. Our complexity analysis reveals that trinomial models require substantially more processing time than binomial models at the same number of steps. The efficiency metric (accuracy/computation time) suggests that in many practical scenarios, binomial trees may offer a more favorable balance between precision and computational resources.
In conclusion, the choice between trinomial and binomial tree models should be guided by the specific requirements of the application at hand, considering the balance between pricing accuracy and computational efficiency.

Reference:

[1] Abdurakhman. (2023). Accelerating convergence in trinomial option pricing: Recursive incremental value ordering with repeated Richardson extrapolation, Mathematical Modelling of Engineering Problems, 10(6), 2179-2184

[2] Chen, T., Joshi, M. S. (2008). Trinomial or binomial: Accelerating American put option price on trees, SSRN Electronic Journal, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1261745

[3] Joshi, M. S. (2007). The convergence of binomial trees for pricing the American put, SSRN Electronic Journal, available from SSRN, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1030143

[4] Leduc, G., Palmer, K. (2022). The convergence rate of option prices in trinomial trees, SSRN Electronic Journal, available from SSRN. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4181597

[5] Lelong, J., Zanette, A. (2010). Tree methods. In R. Cont (Ed.), Encyclopedia of Quantitative Finance, John Wiley & Sons, Ltd.