IB9JHO: Barrier Option Pricing

Hamza Sajjad, Inda Basra, Kejing Wang, Younes Petersen

March 12, 2025

Abstract

This report employs Monte Carlo simulations to price knock-in and knock-out barrier options. An object-oriented C++ framework separates barrier logic from standard option features, and several variance reduction techniques (antithetic variates, importance sampling, control variates, and stratified sampling) are evaluated alongside a naive Monte Carlo baseline. The naive baseline was tested against online resources and a complementary Python script to verify the pricing of the Knock-Out Call option, which is the focus of our main analysis.

Testing across various simulation counts and volatility levels shows that most methods converge reliably, with **control variates** offering the most stable and accurate results. Stratified sampling occasionally deviates with a higher final price, indicating the partitioning strategy may need refinement. Overall, these findings highlight the effectiveness of Monte Carlo models for exotic derivatives, especially when paired with robust variance reduction techniques.

Contents

1 Introduction

A barrier option is an exotic derivative whose payoff depends on whether the underlying asset reaches a predetermined price level. The two most common types are knock-in and knock-out options. A knock-out option expires worthless if the barrier is breached, while a knock-in option only generates a payoff once the barrier is reached.

Pricing barrier options is challenging due to the need to consistently enforce barrier conditions. This project focuses on pricing knock-out barrier options using five Monte Carlo methods (Naive Monte Carlo, Antithetic Variates, Importance Sampling, Control Variates, and Stratified Sampling), comparing their performance in terms of accuracy, variance reduction, and computational efficiency. Our goal is to identify which method converges most reliably under barrier constraints, noting that some methods may exhibit deviations warranting further investigation.

2 Methodology

2.1 How We Chose the Topic and Delegated Project Tasks

Our topic was chosen after group discussions exploring various stochastic techniques, such as the Black–Scholes model and SABR. We initially considered several ideas, but some were too complex for our timeline. Inspired by the lecture slides on pricing Asian options, we decided to follow a similar framework with a similar idea but with a different option, the Barrier Option, as well as introducing different techniques.

Project tasks were allocated based on individual strengths:

- **Hamza:** Developed the general framework, class structure, and fundamental techniques.
- **Younes:** Integrated variance reduction techniques.
- **Inda:** Conducted comprehensive testing and produced convergence results.
- **Kejing:** Performed the final analysis and compiled the documentation.

Each member also documented their contributions, ensuring a balanced and collaborative final project.

2.2 Theory and Modelling Approach

The project utilises Monte Carlo simulations to price barrier options. This choice was driven by the flexibility and generality that simulation methods offer for handling various exotic payoff structures. The modelling approach proceeds by simulating a large number of potential price paths for the underlying asset and then averaging the discounted payoffs, subject to the barrier conditions.

Numerical Recipes and Algorithms. Five primary methods were implemented to estimate the option price:

- *Naive Monte Carlo:* A straightforward simulation of random price paths to obtain the mean payoff.
- *Antithetic Variates:* A variance reduction technique that pairs each random draw with its negative counterpart, improving the stability of the estimates.
- *Importance Sampling:* Focuses sampling efforts on the more significant regions of the payoff distribution to reduce variance.

- *Stratified Sampling:* Partitions the uniform random draws into subintervals, ensuring more uniform coverage of the distribution and reducing variance in the estimated payoff.
- *Control Variates:* Leverages a correlated option (with a known expected value) to adjust the estimates, further decreasing the variance.

Strengths and Weaknesses. The Monte Carlo framework is flexible and can easily incorporate different payoff structures and variance reduction techniques. However, it can be computationally expensive for high accuracy requirements, particularly for complex barrier conditions. Potential numerical issues, such as slow convergence or sensitivity to volatility and barrier parameters, can arise if the variance reduction methods are not carefully tuned. Comparing models is achieved by analyzing their **convergence**. In theory, as we increase the number of simulations, each model's estimate should converge toward the true solution. By comparing the prices obtained at different simulation counts, we can assess how quickly and stably each model converges. This convergence behavior serves as a basis for evaluating model performance.

Another important point to mention is that some of the classes, such as **Stratified Sampling and Importance Sampling**, are heavily dependent on the specific parameters provided. This makes each model potentially unstable, not due to the code design itself, but rather due to the nature of their numerical implementations. Furthermore, for the purpose of analysis and testing, we found it more effective and beneficial to fix the random seed to achieve consistent and reproducible results. Although some randomness remains in the project (as can be observed), fixing the seed ensures that the code remains stable and that results can be compared meaningfully.

In our analysis, we perform extensive convergence studies and sensitivity analyses, such as in `convergence_analysis.cpp`, `spot_price_sensitivity.cpp` , among others. However, these simulations require a significant amount of computation time. While our code fully supports pricing for both Knock-In and Knock-Out Call and Put options as supported by our test cases, running all analyses for every type would be computationally expensive and unnecessary for the purpose of the current analysis. Therefore, all the analysis results presented in this report are focused on the **Knock-Out Call option**. Nevertheless, due to the modular class structure of the project, extending the analysis to other option types (e.g., Knock-In or Puts) would only require minimal adjustments, typically adding just a few lines of code in `main.cpp` if so required.

3 Empirical Results and Discussion

3.1 Data and Analysis

The convergence results, found in `src/output.csv`, were obtained by running simulations with various numbers of Monte Carlo paths. The output comprises option prices computed using five methods: the naive Monte Carlo approach, antithetic variates, control variates, importance sampling, and stratified sampling. The simulation was conducted with counts of 100, 1,000, 5,000, 10,000, 50,000, and 100,000. Table 1 presents a sample of the results:

Table 1: *Convergence results of barrier option prices with increasing simulation counts (from src/output.csv)*

Sims	Naive	Antithetic	Control Variates	Importance	Stratified
100	0.008626	0.008292	0.058653	0.021398	0.009468
1000	0.019142	0.023009	0.026563	0.016228	0.026097
5000	0.021548	0.023249	0.020552	0.024702	0.028393
10000	0.028721	0.024176	0.026561	0.018916	0.031624
50000	0.024430	0.023538	0.025374	0.022401	0.028463
100000	0.023363	0.022245	0.024137	0.024368	0.028332

Figures 1 and 2 illustrate the convergence behaviour for the five pricing methods on a linear scale and a log-log scale, respectively. At 100 simulations, the *Control Variates* method yields the highest estimate (approximately 0.0587), likely due to small-sample variability. In contrast, the naive, antithetic, and stratified approaches all begin near 0.008–0.009, while *Importance Sampling* starts at about 0.0214. As the number of simulations increases, the naive, antithetic, control variates, and importance sampling methods gradually converge to a similar price range (roughly 0.022–0.024) by 100,000 simulations.

Stratified Sampling, however, ends at about 0.0283 at 100,000 simulations, which is noticeably higher than the other methods. This outcome may reflect the particular partitioning strategy used or the need for further refinement of strata. Nonetheless, stratified sampling remains competitive at lower simulation counts, indicating that more sophisticated stratification schemes could potentially reduce variance and produce estimates closer to the other methods in the long run.

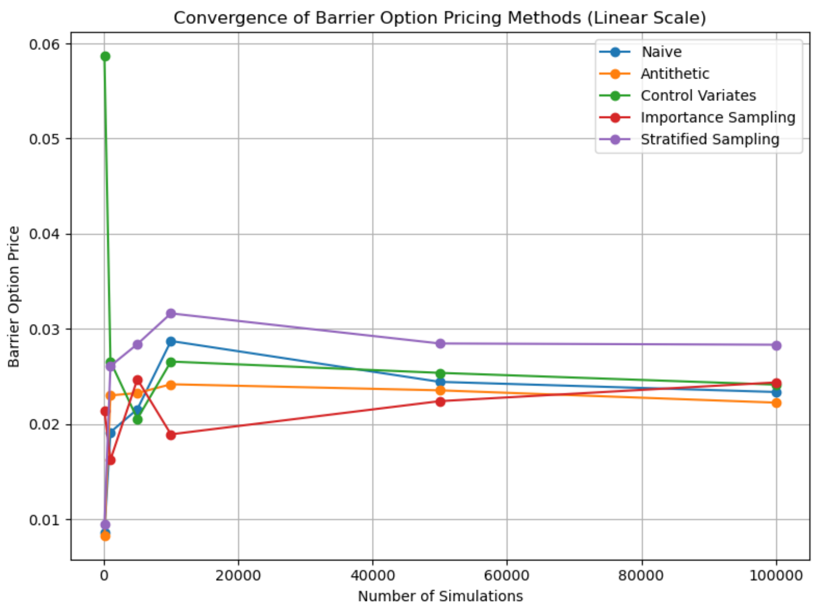

Figure 1: *Convergence of barrier option pricing methods (linear scale)*

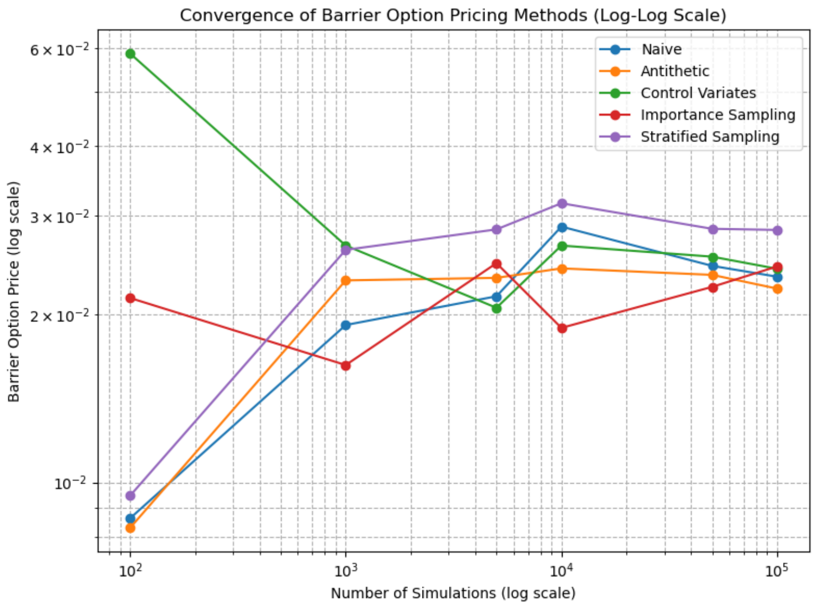

Figure 2: *Convergence of barrier option pricing methods (log-log scale)*

Overall, the *Control Variates* method demonstrates relatively stable convergence once the simulation count exceeds a few thousand, while antithetic variates and importance sampling also reduce variance compared to the naive approach. In practical applications, control variates is often preferable for pricing barrier options, given its balance of variance reduction and convergence speed. Nonetheless, stratified sampling may offer advantages if its partitioning strategy is refined to avoid the higher final estimates observed here.

3.2 Analysis

All of analysis results can be found within `../build/results`. Note that this is different to the `output.csv` shown before. The following inputs were used for all analyses:

Table 2: *Input Parameters for All Analyses*

Parameter	Value	Description
S_0	100.0	Underlying asset price
K	105.0	Strike price
r	0.05	Annual risk-free rate
σ	0.2	Annualized volatility
T	1.0	Time to maturity (years)
B	110.0	Knock-out barrier level
Simulations	100,000	Base simulation count

1. Convergence Analysis

The convergence analysis evaluates the impact of simulation steps on option price estimation. The results show how the price stabilises with increasing simulations.

Table 3: *Convergence Analysis of Different Pricing Methods (from build/results/convergence_analysis.csv)*

Simulations	Naive	Antithetic	ImportanceSampling	ControlVariates	Stratified
1000	0.0308518	0.0251786	0.0161412	0.0358608	0.0439478
5000	0.021727	0.027576	0.0297005	0.0256943	0.035216
10000	0.0250735	0.0270525	0.0299438	0.0263271	0.0300843
50000	0.0250406	0.0251895	0.0246315	0.0256104	0.0310876
100000	0.0248413	0.0250131	0.0225472	0.0244411	0.0314395

As the number of simulations increases, the option price converges to a stable value. The results become stable with more simulations, showing that the chosen simulation range (1,000 to 100,000) accurately captures the dynamics of the option price. Importance sampling estimates occasionally deviate from other methods at certain simulation counts, which may reflect the sampling distribution or small effective sample sizes, especially at lower numbers of simulations.

2. Error Tolerance Analysis

The error tolerance analysis compares the estimated option prices against the reference price at various simulation steps. The results show the convergence of error with increasing simulations:

Table 4: *Error Tolerance Analysis for Different Pricing Methods*

Method	Simulations	Price	Error
Naive	10000	0.0246644	0.000530959
Antithetic	5000	0.0287386	0.00449849
Antithetic	10000	0.0223947	0.00184539
Antithetic	15000	0.0251171	0.000876937
ImportanceSampling	5000	0.0256617	0.000441499
ControlVariates	10000	0.0225877	0.00237014
ControlVariates	20000	0.0244463	0.00051155
Stratified	10000	0.0375808	0.00577001
Stratified	20000	0.0300619	0.00174895
Stratified	30000	0.0265462	0.00526461
Stratified	40000	0.0324844	0.000673575

As the number of simulations increases, the error decreases across all methods. The Antithetic and Importance Sampling methods show lower errors and provide more accurate results with fewer simulations, especially at smaller simulation sizes.

3. Pricing Efficiency Comparison

The efficiency analysis compares the execution time and option prices of each method. The results are as follows:

Table 5: *Pricing Efficiency Comparison for Different Methods*

Method	Price	Time (ms)
Naive	0.0250385	2326

Method	Price	Time (ms)
Antithetic	0.0234217	2333
ImportanceSampling	0.0252263	1269
ControlVariates	0.0243332	2332
Stratified	0.0302289	1525

The Importance Sampling method achieves the fastest execution time (1269 ms) while still providing relatively accurate pricing. Naive, Antithetic, and Control Variates methods have similar execution times. Stratified is only slower than Importance Sampling making it relativity quick however, its pricing accuracy is significantly worse in comparison to the other models.

4. Spot Price Sensitivity Analysis

This analysis explores how changes in the spot price affect the option price. The results show the expected non-linear behavior as the spot price approaches the barrier level:

Table 6: *Spot Price Sensitivity Analysis for Different Pricing Methods*

Spot Price	**Naive**	**Antithetic**	**Importance Sampling**	**Control Variates**	**Stratified**
80	0.0258869	0.0268408	0.0434399	0.0261503	0.0314758
90	0.0303074	0.0313509	0.0394445	0.0304379	0.0393192
100	0.0216917	0.0223500	0.0208087	0.0224567	0.0278191
110	0	0	0	0	0.00336168
120	0	0	0	0	0

As the spot price increases, the option price initially rises and then falls. When the spot price approaches or exceeds the barrier level, the option price becomes zero, reflecting the option's expiry. The only exception is **Stratified Sampling** where when at the spot price the value is not 0. This is due to the random sampling aspect of this model and cannot be phased. This makes it another reason why this model is unreliable.

5. Volatility Sensitivity Analysis

This analysis examines how changes in volatility affect the option price:

Table 7: *Volatility Sensitivity Analysis for Different Pricing Methods*

Volatility	Naive	Antithetic	ImportanceSampling	ControlVariates	Stratified
0.1	0.133142	0.132644	0.137166	0.133352	0.147213
0.15	0.0523417	0.0511555	0.0500498	0.0524085	0.0607496
0.2	0.0247708	0.0262098	0.024878	0.0252907	0.0314401
0.25	0.0152231	0.0134306	0.0153753	0.0140736	0.0187806
0.3	0.00914923	0.00962863	0.00816996	0.00933677	0.0118143

We can clearly see that as volatility increases, the option price decreases. All methods exhibit this behavior, reflecting the increasing likelihood of the option triggering the barrier.

4 Conclusion

Antithetic variates improve accuracy compared to the naive approach but do not achieve the same level of variance reduction as control variates. **Importance sampling** provides the fastest execution, making it appealing when *computational efficiency* is critical. In contrast, **stratified**

sampling produces higher final prices than other methods, suggesting that the chosen partitioning strategy may require refinement to fully realize its variance reduction potential.

Sensitivity analyses on *volatility* and *spot price* revealed that higher volatility typically lowers the option price due to a greater chance of hitting the barrier, while spot prices exceeding the barrier render the option worthless. These results highlight the importance of robust variance reduction techniques to ensure both accuracy and efficiency in barrier option pricing.

In summary, the methods explored, particularly **control variates**, demonstrate reliable convergence and provide practitioners with effective tools for pricing exotic barrier options under various market conditions. This is also supported by its widespread use in practice, reinforcing the validity of our results. Future work could explore hybrid approaches, such as combining stratified sampling with other variance reduction methods, to further enhance stability and performance.