



UNIVERSITÀ
DI PAVIA

Department of Economics and Management

Master's Degree Programme in Finance

Efficient Monte Carlo Pricing of Barrier Options

under the OUSV Model:

A Variance Reduction Approach

Supervisor:

Prof. Riccardo Brignone

Degree Thesis

by Alan Leopoldo Sampaolo

Student ID: 555124

Academic Year 2025–2026



UNIVERSITÀ
DI PAVIA

Dipartimento di Scienze Economiche e Aziendali

Corso di Laurea Magistrale in Finance

**Valutazione efficiente mediante metodi
Monte Carlo
di opzioni barriera nel modello OUSV:
un approccio basato sulla riduzione della
varianza**

Relatore:

Chiar.mo Prof. Riccardo Brignone

Tesi di Laurea

di Alan Leopoldo Sampaolo

Matricola: 555124

Anno Accademico 2025–2026

Title and Abstract

Title: Efficient Monte Carlo Pricing of Barrier Options under the OUSV Model: A Variance Reduction Approach

Abstract

This thesis studies the Monte Carlo pricing of barrier options under the Ornstein–Uhlenbeck stochastic volatility model, abbreviated as OUSV. Barrier options are path-dependent derivatives, since their payoff depends not only on the terminal value of the underlying asset, but also on whether the asset price has reached a predetermined barrier during the life of the contract.

The thesis first introduces the main theoretical tools used in the analysis: risk-neutral pricing, Monte Carlo estimation and variance reduction techniques. The methods considered are antithetic variables, control variates and conditional Monte Carlo. The Black–Scholes model is then presented as a benchmark framework, because it provides a simple setting in which exact simulation and vanilla option pricing can be understood before moving to the more general OUSV model.

The main part of the thesis focuses on the OUSV model, where the asset price is driven by a latent Ornstein–Uhlenbeck volatility factor. Since the numerical analysis in this thesis focuses on computationally efficient approximate simulation of the OUSV dynamics, three simulation schemes are considered: the Euler scheme, a scheme based on van Haastrecht, Lord and Pelsner, and the Inverse Gaussian Scheme proposed by Brignone and Sgarra.

These schemes are applied to the pricing of up-and-in and up-and-out call options for different barrier levels. Each barrier is treated as a separate contract and the comparison uses end-to-end computational times, with pathwise early stopping when the payoff or estimator is determined at the first hitting time. The results show that control variates are generally preferred for up-and-in calls and for high-barrier up-and-out calls, while plain Monte Carlo and antithetic variables remain competitive for low-barrier up-and-out contracts. A separate comparison of the

simulation schemes shows that the Inverse Gaussian Scheme provides the smallest observed pricing deviation on both coarse and fine grids, while Euler and VHP become statistically consistent with the Fourier benchmark on the fine grid at substantially lower computational cost.

Titolo e Abstract

Titolo: Valutazione efficiente mediante metodi Monte Carlo di opzioni barriera nel modello OUSV: un approccio basato sulla riduzione della varianza

Abstract

Questa tesi studia la valutazione mediante metodi Monte Carlo di opzioni barriera nel modello di volatilità stocastica Ornstein–Uhlenbeck, abbreviato come OUSV. Le opzioni barriera sono derivati path-dependent, poiché il loro payoff dipende non solo dal valore finale dell’attività sottostante, ma anche dal fatto che il prezzo abbia raggiunto una barriera prefissata durante la vita del contratto.

La tesi introduce innanzitutto gli strumenti teorici principali utilizzati nell’analisi: la valutazione risk-neutral, la stima Monte Carlo e le tecniche di riduzione della varianza. I metodi considerati sono le variabili antitetiche, le variabili di controllo e il conditional Monte Carlo. Il modello di Black–Scholes viene poi presentato come framework di riferimento, poiché fornisce un contesto semplice in cui comprendere la simulazione esatta e la valutazione di opzioni vanilla prima di passare al modello OUSV.

La parte principale della tesi si concentra sul modello OUSV, nel quale il prezzo dell’attività sottostante è guidato da un fattore latente di volatilità con dinamica Ornstein–Uhlenbeck. Poiché l’analisi numerica di questa tesi si concentra su schemi approssimati ed efficienti per la simulazione delle dinamiche OUSV, vengono considerati tre schemi di simulazione: lo schema di Eulero, uno schema basato su van Haastrecht, Lord e Pelsser, e l’Inverse Gaussian Scheme proposto da Brignone e Sgarra.

Questi schemi vengono applicati alla valutazione di opzioni call up-and-in e up-and-out per diversi livelli di barriera. Ogni barriera è trattata come un contratto separato e il confronto utilizza tempi computazionali end-to-end, con arresto anticipato dei percorsi quando il payoff o lo stimatore risultano determinati al primo hitting. I risultati mostrano che le variabili di controllo sono generalmente preferibili per le

up-and-in e per le up-and-out con barriera elevata, mentre Monte Carlo plain e variabili antitetiche rimangono competitivi per le up-and-out con barriera bassa. Un confronto separato tra gli schemi di simulazione mostra inoltre che l'Inverse Gaussian Scheme presenta la minore deviazione di prezzo osservata sia sulla griglia grossolana sia su quella fine, mentre sulla griglia fine Eulero e VHP risultano compatibili con il benchmark Fourier a un costo computazionale sensibilmente inferiore.

Contents

Title and Abstract	i
Titolo e Abstract	iii
Introduction	1
1 Risk-Neutral Pricing and Monte Carlo Methods	4
1.1 Risk-Neutral Pricing	4
1.2 Monte Carlo Estimation	5
1.3 Variance Reduction: General Idea	8
1.4 Antithetic Variables	9
1.5 Control Variates	13
1.6 Conditional Monte Carlo	19
2 The Black–Scholes Model	24
2.1 Risk-Neutral Dynamics	24
2.2 The Black–Scholes Pricing Formula	27
2.3 Monte Carlo Pricing in the Black–Scholes Model	28
2.4 Limitations of the Black–Scholes Model	30
3 The Ornstein–Uhlenbeck Stochastic Volatility Model	33
3.1 Model Dynamics	33
3.2 From Black–Scholes Exact Simulation to OUSV Simulation Schemes .	34
3.3 Fourier Pricing Formula for European Options under OUSV	36
3.4 Simulation Schemes for the OUSV Model	40
3.4.1 Euler Scheme	41
3.4.2 Scheme Based on van Haastrecht, Lord and Pelsser	43
3.4.3 Inverse Gaussian Scheme	46
3.4.4 Comparison of the Simulation Schemes	50
3.5 Monte Carlo Estimators with the OUSV Model	51

3.5.1	Plain Monte Carlo Implementation	51
3.5.2	Antithetic Variables Implementation	52
3.5.3	Control Variates Implementation	53
3.5.4	Conditional Monte Carlo Implementation and In-Out Parity .	54
4	Numerical Results	55
4.1	Numerical Setting	55
4.2	Up-and-In Call Options	56
4.2.1	Euler Scheme	57
4.2.2	VHP Scheme	58
4.2.3	Inverse Gaussian Scheme	59
4.2.4	Summary of the Up-and-In Results	60
4.3	Up-and-Out Call Options	62
4.3.1	Euler Scheme	62
4.3.2	VHP Scheme	63
4.3.3	Inverse Gaussian Scheme	64
4.3.4	Summary of the Up-and-Out Results	65
4.4	Comparison of the Simulation Schemes	67
5	Conclusions	70

Introduction

The pricing of financial derivatives is one of the central problems in quantitative finance. A derivative contract derives its value from an underlying asset, and its price is typically obtained as the discounted expected payoff under a risk-neutral probability measure. Therefore, once a model for the underlying asset has been specified, the pricing problem becomes the computation of an expectation under the risk-neutral measure. For some derivatives and some particular models, this expectation has a closed or semi-closed form. This is the case for European vanilla options in several models.

Barrier options form an important class of path-dependent derivatives. Unlike vanilla options, their activation or termination depends on whether the underlying asset reaches a predetermined price level during the life of the contract. For instance, an up-and-out call option pays the standard call payoff only if the underlying asset has not crossed an upper barrier before maturity; if the barrier is reached, the option is knocked out and its payoff becomes zero. Conversely, an up-and-in call becomes active only after the barrier has been reached. Barrier options are often traded in over-the-counter markets and can be used either to hedge specific risks, for example, an investor exposed to a sharp fall in an asset price may use a down-and-in put that becomes active only after a given downside barrier is reached, or to take positions on whether the underlying asset will reach a given price level, for instance by purchasing an up-and-in call when expecting the asset price to rise above a specified threshold. Compared with the corresponding vanilla options, barrier options generally require a lower initial premium. This is because the barrier condition restricts the circumstances under which the holder receives the payoff. The lower cost is therefore obtained in exchange for an additional source of risk: depending on the path followed by the underlying asset, the option may be deactivated before maturity or may never become active. The same path-dependence that makes barrier options less expensive also makes their valuation more demanding. Their payoff depends not only on the terminal value of the underlying asset, but also on its evolution over the life of the contract. Monte Carlo simulation is therefore a natural pricing tool,

since it generates possible paths of the underlying asset and checks, for each path, whether the barrier condition has been satisfied.

The model considered in this thesis is the Ornstein–Uhlenbeck stochastic volatility model, abbreviated as OUSV. In this model, the asset price is driven by a latent volatility factor following an Ornstein–Uhlenbeck dynamics. The model is more flexible than Black–Scholes because volatility is no longer constant over time. Under the OUSV model, European vanilla options can be priced by using a semi-closed Fourier formula (this formula provides an important benchmark in the thesis. It is used to validate the simulation schemes and also enters some of the variance reduction methods). For barrier options, however, no analogous closed or semi-closed formula is available in the setting considered here. The price must therefore be approximated numerically, and Monte Carlo simulation becomes the central pricing method.

Given a simulated path, the discounted payoff of a discretely monitored up-and-out call option can be written as

$$Y = e^{-rT}(S_T - K)^+ \mathbf{1}_{\{\max_{0 \leq i \leq N} S_{t_i} < B\}},$$

where K is the strike price, B is the barrier level, and

$$0 = t_0 < t_1 < \cdots < t_N = T$$

is the monitoring grid used in the simulation. The Monte Carlo estimator is then given by the sample average

$$\hat{V}_M = \frac{1}{M} \sum_{j=1}^M Y_j,$$

where Y_j denotes the j -th independent copy of the discounted payoff random variable Y . Once a Monte Carlo sample has been generated, the corresponding numerical realization is denoted by y_j .

By the law of large numbers, this estimator converges to the theoretical price as the number of simulations increases, provided that the simulated payoffs are independent and identically distributed with finite expectation. However, the Monte Carlo estimator is itself a random variable. Different simulated samples lead to different price estimates. For this reason, the precision of the estimator is measured through its variance, or equivalently through its standard error. Increasing the number of simulations reduces the standard error, but it also increases the computational cost. Variance reduction techniques aim to improve this trade-off. Their goal is to construct alternative estimators with the same target expectation as plain Monte

Carlo, but with lower variance or better overall efficiency. The methods considered in this thesis are antithetic variables, control variates and conditional Monte Carlo.

The thesis first introduces the main theoretical tools needed for the numerical analysis: risk-neutral pricing, Monte Carlo estimation and variance reduction methods. The Black–Scholes model is then presented as a benchmark framework, because it provides a simple setting in which exact simulation and vanilla option pricing can be understood before moving to the more general OUSV model. The thesis then moves to the OUSV model. Although exact simulation methods exist, they can be computationally demanding for frequently monitored path-dependent options. For this reason, three approximate simulation schemes are considered, providing different trade-offs between accuracy and computational cost. The first one is the Euler scheme, which is the simplest discretization method and provides a natural baseline. The second one is a scheme based on the approach proposed by van Haastrecht, Lord and Pelsser. The third one is the Inverse Gaussian Scheme proposed by Brignone and Sgarra, which is a more refined simulation scheme for the OUSV dynamics.

The numerical part of the thesis applies these three schemes to the pricing of up-and-in and up-and-out call options. For each scheme and for each barrier level, plain Monte Carlo is compared with the variance reduction methods. Each barrier is treated as a separate contract and the comparison is based on the estimated price, the standard error, the end-to-end computational time and an empirical efficiency indicator that combines computational cost and statistical precision. Whenever the payoff or the estimator is already determined at the first barrier hitting time, the implementation uses pathwise early stopping. The objective of the thesis is therefore twofold. First, it explains why Monte Carlo simulation is well suited to the pricing of path-dependent barrier options. Second, and also more important, after fixing the option type, the barrier level and the OUSV simulation scheme, it studies which estimator provides the best trade-off between precision and computational cost. The results show that the effectiveness of a variance reduction method is not automatic. It depends on the type of barrier option, the barrier level and the simulation scheme used to generate the OUSV paths.

Chapter 1

Risk-Neutral Pricing and Monte Carlo Methods

1.1 Risk-Neutral Pricing

The starting point of derivative pricing is the idea that the value of a financial contract is determined by its future payoff. Let H denote the payoff of a derivative at maturity T . The payoff is a random variable, since it depends on the future evolution of the underlying asset.

In a pricing framework without arbitrage, the value of the derivative at time zero is obtained by taking the discounted expected payoff under a risk-neutral probability measure. If the risk-free rate r is assumed to be constant, the pricing formula is

$$V_0 = \mathbb{E}^{\mathbb{Q}} \left[e^{-rT} H \right],$$

where $\mathbb{Q}$ denotes the risk-neutral measure.

This formula is central for the whole thesis. It says that pricing a derivative is not a problem of computing the real-world expected return of the underlying asset. Instead, the payoff is evaluated under a probability measure in which discounted asset prices are martingales and under this measure, the expected growth rate of the underlying asset is the risk-free rate. This does not mean that the asset is expected to grow at the risk-free rate in the real world. It just means that, for pricing purposes, the probability measure is changed. The practical consequence is that the pricing problem becomes the computation of an expectation. Once the model for the underlying asset has been specified under the risk-neutral measure, the price of the derivative is obtained by computing the expected discounted payoff generated by that model.

For a European vanilla call option with strike K , the payoff is

$$H_{\text{call}} = (S_T - K)^+.$$

In this case, the payoff depends only on the terminal value S_T of the underlying asset. The price is therefore

$$C_0 = \mathbb{E}^{\mathbb{Q}} \left[e^{-rT} (S_T - K)^+ \right].$$

Barrier options have a different structure. For example, the payoff of a discretely monitored up-and-out call option can be written as

$$H_{\text{out}} = (S_T - K)^+ \mathbf{1}_{\{\max_{0 \leq i \leq N} S_{t_i} < B\}},$$

where B is the barrier level and

$$0 = t_0 < t_1 < \dots < t_N = T$$

is the monitoring grid.

The corresponding time-zero price is

$$C_0^{\text{out}} = \mathbb{E}^{\mathbb{Q}} \left[e^{-rT} (S_T - K)^+ \mathbf{1}_{\{\max_{0 \leq i \leq N} S_{t_i} < B\}} \right].$$

That said, when a model does not offer a closed or semi-closed form for the risk-neutral expectation of a specific derivative, the expectation must be computed using numerical methods. In this context, Monte Carlo simulation provides a numerical way to approximate this expectation by generating many possible paths of the underlying asset, computing the discounted payoff on each path, and taking their sample average (LLN).

1.2 Monte Carlo Estimation

The risk-neutral pricing formula shows that the price of a derivative can be written as an expectation. If H denotes the payoff at maturity, it is useful to define the discounted payoff

$$Y = e^{-rT} H.$$

Then the time-zero price can be written as

$$V_0 = \mathbb{E}^{\mathbb{Q}}[Y].$$

When this expectation cannot be computed analytically or semi-analytically (and so it doesn't exist a closed or semi-closed form) it can be approximated numerically by Monte Carlo simulation. The basic idea is to generate M independent copies of the random payoff Y , denoted by

$$Y_1, Y_2, \dots, Y_M.$$

Once the Monte Carlo sample has been generated, the lowercase quantities

$$y_1, y_2, \dots, y_M$$

denote the corresponding numerical realizations. The expectation is then approximated by the sample average.

The plain Monte Carlo estimator, denoted by plain MC in this thesis, is therefore defined as

$$\hat{V}_M^{\text{plain}} = \frac{1}{M} \sum_{j=1}^M Y_j.$$

Equivalently, denoting the sample mean random variable by

$$\bar{Y}_M = \frac{1}{M} \sum_{j=1}^M Y_j,$$

then

$$\hat{V}_M^{\text{plain}} = \bar{Y}_M.$$

For a particular generated sample, the corresponding numerical value of the estimator is

$$\hat{v}_M^{\text{plain}} = \frac{1}{M} \sum_{j=1}^M y_j = \bar{y}_M.$$

The realization y_j is obtained from the j -th simulated scenario of the underlying asset. In the case of a European vanilla option, this may only require the simulation of the terminal value S_T , because the payoff depends only on the final value of the underlying asset. For path-dependent derivatives, such as barrier options, obtaining y_j requires the simulation of an entire path along the monitoring grid. This is because the payoff depends not only on S_T , but also on whether the barrier has been reached before maturity.

For example, for a discretely monitored up-and-out call option, the j -th copy of

the discounted payoff random variable is

$$Y_j^{\text{out}} = e^{-rT} \left(S_T^{(j)} - K \right)^+ \mathbf{1}_{\left\{ \max_{0 \leq i \leq N} S_{t_i}^{(j)} < B \right\}},$$

where $S_{t_i}^{(j)}$ denotes the value of the underlying asset at time t_i along the j -th random path. Once that path has been generated, the corresponding numerical payoff realization is denoted by y_j^{out} .

Therefore, in this specific case, the plain MC estimator can be written explicitly as

$$\hat{C}_M^{\text{out,plain}} = \frac{1}{M} \sum_{j=1}^M Y_j^{\text{out}},$$

that is,

$$\hat{C}_M^{\text{out,plain}} = \frac{1}{M} \sum_{j=1}^M e^{-rT} \left(S_T^{(j)} - K \right)^+ \mathbf{1}_{\left\{ \max_{0 \leq i \leq N} S_{t_i}^{(j)} < B \right\}}.$$

The theoretical justification of the Monte Carlo estimator is given by the law of large numbers indeed, if the simulated payoffs are independent and identically distributed and have finite expectation, then

$$\hat{V}_M^{\text{plain}} \longrightarrow V_0$$

as M tends to infinity.

This convergence result explains why Monte Carlo simulation is a valid pricing method.

The variance of the plain MC estimator is

$$\text{Var} \left(\hat{V}_M^{\text{plain}} \right) = \frac{\text{Var}(Y)}{M}.$$

This formula shows that the accuracy of Monte Carlo simulation improves as the number of simulated paths increases. More precisely, the standard error decreases at rate $1/\sqrt{M}$.

In practice, the variance of Y is unknown and must be estimated from the simulated payoffs.

The sample variance computed from the simulated payoff realizations is then

$$s_Y^2 = \frac{1}{M-1} \sum_{j=1}^M (y_j - \bar{y}_M)^2,$$

and the estimated standard error of the Monte Carlo price is

$$\widehat{\text{SE}}\left(\widehat{V}_M^{\text{plain}}\right) = \frac{s_Y}{\sqrt{M}}.$$

A smaller standard error means that the estimator is less sensitive to the particular simulated sample. Equivalently, if the simulation is repeated with a different random seed, the resulting price estimates are expected to fluctuate less around the theoretical value. This fact motivates the use of variance reduction techniques. Instead of only increasing the number of simulations, one can try to construct alternative estimators with the same target expectation but with lower variance. This is the main idea behind the methods considered in the following sections.

1.3 Variance Reduction: General Idea

Plain MC is simple and flexible, but its rate of convergence is relatively slow. The standard error decreases at rate $1/\sqrt{M}$, where M is the number of simulated paths. This means that reducing the standard error by a factor of two requires approximately four times as many simulations. Variance reduction techniques try to improve the standard error of this plain MC estimator. The key idea is to replace the payoff random variable used in the plain Monte Carlo estimator with an alternative payoff random variable having the same expectation but a lower variance.

Suppose that the target price is

$$V_0 = \mathbb{E}^{\mathbb{Q}}[Y].$$

A variance reduction method constructs an alternative payoff random variable Y^{VR} such that

$$\mathbb{E}^{\mathbb{Q}}[Y^{\text{VR}}] = V_0.$$

The corresponding estimator is

$$\widehat{V}_M^{\text{VR}} = \frac{1}{M} \sum_{j=1}^M Y_j^{\text{VR}}.$$

Then

$$\text{Var}\left(\widehat{V}_M^{\text{VR}}\right) = \frac{\text{Var}\left(Y^{\text{VR}}\right)}{M}.$$

The method is effective when

$$\text{Var}(Y^{\text{VR}}) < \text{Var}(Y).$$

In that case, the estimator based on Y^{VR} has a smaller standard error than the plain MC estimator for the same number of simulations. Equivalently, it can reach a given level of precision with fewer simulated paths. Therefore, the performance of a variance reduction method is usually evaluated by comparing the estimated standard errors of the competing estimators.

A useful quantity is the variance reduction factor, defined as

$$\text{VRF} = \frac{\text{Var}(\hat{V}_M^{\text{plain}})}{\text{Var}(\hat{V}_M^{\text{VR}})}.$$

When the same number of simulations is used for both estimators, this can also be estimated through the squared ratio of the standard errors:

$$\widehat{\text{VRF}} = \left(\frac{\widehat{\text{SE}}(\hat{V}_M^{\text{plain}})}{\widehat{\text{SE}}(\hat{V}_M^{\text{VR}})} \right)^2.$$

A value of $\widehat{\text{VRF}} > 1$ means that the variance reduction method has reduced the variance relative to plain MC. The larger the value, the stronger the variance reduction. However, variance reduction is not only about lowering the variance. Some variance reduction methods also increase the running time of each simulation. For this reason, in numerical applications it is important to look at both the standard error and the running time to assess if a variance reduction method is efficient in practice. The variance reduction methods considered in this thesis are antithetic variables, denoted by AV, control variates, denoted by Control Var, and conditional Monte Carlo, denoted by CMC. The following sections introduce these methods in a general way.

1.4 Antithetic Variables

The first variance reduction technique considered in this thesis is the antithetic variables method, denoted by AV. This is a classical variance reduction technique in Monte Carlo option pricing; see Boyle et al. [1997] and Glasserman [2003]. The basic idea is to generate pairs of negatively related simulated paths instead of using only independent paths. In this way, each simulation produces not only one payoff,

but a pair of payoffs which are then averaged.

Suppose that the payoff can be written as a function of a random input Z , where Z represents the random shocks used to simulate the underlying process. The plain MC payoff random variable can then be written as

$$Y = Y(Z).$$

In the antithetic approach, for each simulated random input Z , one also considers the opposite input $-Z$. This produces an antithetic payoff

$$\tilde{Y} = Y(-Z).$$

If Z is standard normal, then $-Z$ has the same distribution as Z . Therefore, Y and $\tilde{Y}$ have the same distribution and the same expectation.

The antithetic payoff random variable is defined as the average of the two payoffs:

$$Y^{\text{AV}} = \frac{Y + \tilde{Y}}{2}.$$

The payoff random variable used by the antithetic variables estimator remains unbiased. To see this, start from the expectation of the antithetic payoff random variable:

$$\mathbb{E}^{\mathbb{Q}} [Y^{\text{AV}}] = \mathbb{E}^{\mathbb{Q}} \left[\frac{Y + \tilde{Y}}{2} \right].$$

By linearity of expectation, the constant $1/2$ can be taken outside the expectation:

$$\mathbb{E}^{\mathbb{Q}} \left[\frac{Y + \tilde{Y}}{2} \right] = \frac{1}{2} \mathbb{E}^{\mathbb{Q}} [Y + \tilde{Y}].$$

Again by linearity of expectation, the expectation of the sum can be written as the sum of the expectations:

$$\frac{1}{2} \mathbb{E}^{\mathbb{Q}} [Y + \tilde{Y}] = \frac{1}{2} \mathbb{E}^{\mathbb{Q}} [Y] + \frac{1}{2} \mathbb{E}^{\mathbb{Q}} [\tilde{Y}].$$

Since Y and $\tilde{Y}$ have the same distribution, they also have the same expectation. Moreover, both are discounted payoffs targeting the same price V_0 . Hence,

$$\mathbb{E}^{\mathbb{Q}} [Y] = \mathbb{E}^{\mathbb{Q}} [\tilde{Y}] = V_0.$$

Substituting this into the previous formula gives

$$\frac{1}{2}\mathbb{E}^{\mathbb{Q}}[Y] + \frac{1}{2}\mathbb{E}^{\mathbb{Q}}[\tilde{Y}] = \frac{1}{2}V_0 + \frac{1}{2}V_0 = V_0.$$

Therefore,

$$\mathbb{E}^{\mathbb{Q}}[Y^{\text{AV}}] = V_0.$$

This proves that the antithetic estimator is unbiased.

If M_{half} antithetic pairs are generated, the AV estimator is

$$\hat{V}_{M_{\text{half}}}^{\text{AV}} = \frac{1}{M_{\text{half}}} \sum_{j=1}^{M_{\text{half}}} Y_j^{\text{AV}}.$$

Equivalently,

$$\hat{V}_{M_{\text{half}}}^{\text{AV}} = \frac{1}{M_{\text{half}}} \sum_{j=1}^{M_{\text{half}}} \frac{Y_j + \tilde{Y}_j}{2}.$$

The variance reduction effect comes from the covariance between the two payoffs in each pair. Indeed,

$$\text{Var}\left(\frac{Y + \tilde{Y}}{2}\right) = \frac{1}{4} \text{Var}(Y + \tilde{Y}).$$

Using the variance of a sum,

$$\text{Var}(Y + \tilde{Y}) = \text{Var}(Y) + \text{Var}(\tilde{Y}) + 2 \text{Cov}(Y, \tilde{Y}).$$

Therefore,

$$\text{Var}\left(\frac{Y + \tilde{Y}}{2}\right) = \frac{1}{4} [\text{Var}(Y) + \text{Var}(\tilde{Y}) + 2 \text{Cov}(Y, \tilde{Y})].$$

Since Y and $\tilde{Y}$ have the same distribution, they have the same variance.

Thus,

$$\text{Var}\left(\frac{Y + \tilde{Y}}{2}\right) = \frac{1}{2} \text{Var}(Y) + \frac{1}{2} \text{Cov}(Y, \tilde{Y}).$$

This formula shows why the method can reduce variance. For example, if the two payoffs are negatively correlated, then the covariance term is negative and the variance of the antithetic average decreases.

In terms of the correlation coefficient

$$\rho_{\text{AV}} = \text{Corr}(Y, \tilde{Y}),$$

the covariance can be written as

$$\text{Cov}(Y, \tilde{Y}) = \rho_{AV} \text{Var}(Y),$$

Substituting this expression gives

$$\text{Var}\left(\frac{Y + \tilde{Y}}{2}\right) = \frac{\text{Var}(Y)}{2} (1 + \rho_{AV}).$$

The more negative the correlation between the original and antithetic payoffs, the stronger the variance reduction.

In option pricing, the antithetic construction is usually implemented by changing the sign of the Gaussian shocks used in the simulation. For example, if a path is generated using shocks

$$Z_1, Z_2, \dots, Z_N,$$

then the corresponding antithetic path is generated using

$$-Z_1, -Z_2, \dots, -Z_N.$$

The AV payoff is obtained by averaging the discounted payoffs of the two paths.

For example, in the case of a discretely monitored up-and-out call option, the two discounted payoffs are

$$Y_j^{\text{out}} = e^{-rT} \left(S_T^{(j)} - K \right)^+ \mathbf{1}_{\left\{ \max_{0 \leq i \leq N} S_{t_i}^{(j)} < B \right\}},$$

and

$$\tilde{Y}_j^{\text{out}} = e^{-rT} \left(\tilde{S}_T^{(j)} - K \right)^+ \mathbf{1}_{\left\{ \max_{0 \leq i \leq N} \tilde{S}_{t_i}^{(j)} < B \right\}}.$$

The corresponding antithetic payoff random variable is

$$Y_j^{\text{out,AV}} = \frac{Y_j^{\text{out}} + \tilde{Y}_j^{\text{out}}}{2}.$$

The AV estimator for the up-and-out call price is therefore

$$\hat{C}_{M_{\text{half}}}^{\text{out,AV}} = \frac{1}{M_{\text{half}}} \sum_{j=1}^{M_{\text{half}}} Y_j^{\text{out,AV}}.$$

The same construction can be applied to up-and-in options. In that case, the indicator is reversed.

The estimated standard error of the AV estimator is computed as

$$\widehat{\text{SE}}\left(\widehat{V}_{M_{\text{half}}}^{\text{AV}}\right) = \frac{s_{\text{AV}}}{\sqrt{M_{\text{half}}}},$$

where s_{AV} is the sample standard deviation of the realized averaged antithetic payoffs:

$$y_1^{\text{AV}}, y_2^{\text{AV}}, \dots, y_{M_{\text{half}}}^{\text{AV}}.$$

Antithetic variables are easy to implement and usually require little additional computational effort. However, their effectiveness is not guaranteed. If the original and antithetic payoffs are not sufficiently negatively correlated, the variance reduction may be small.

1.5 Control Variates

The second variance reduction technique considered in this thesis is the control variates method, denoted by Control Var. This method is based on the idea of using an auxiliary payoff random variable correlated with the target payoff and whose expectation is known. The method is classical in Monte Carlo option pricing; see Boyle et al. [1997] and Glasserman [2003].

Suppose that the target price is

$$V_0 = \mathbb{E}^{\mathbb{Q}}[Y].$$

Let X be another payoff random variable such that its expectation is known:

$$\mu_X = \mathbb{E}^{\mathbb{Q}}[X].$$

The variable X is called a control variate. The basic idea is to construct a new payoff random variable for the estimator by correcting Y through the difference between X and its known expectation.

In this thesis, the control variate payoff random variable is written as

$$Y^{\text{CV}} = Y + \alpha(X - \mu_X),$$

where α is a coefficient to be chosen.

The correction term $X - \mu_X$ has zero expectation. Indeed,

$$\mathbb{E}^{\mathbb{Q}}[X - \mu_X] = \mathbb{E}^{\mathbb{Q}}[X] - \mathbb{E}^{\mathbb{Q}}[\mu_X].$$

Since μ_X is a constant,

$$\mathbb{E}^{\mathbb{Q}}[\mu_X] = \mu_X.$$

Moreover, by definition,

$$\mathbb{E}^{\mathbb{Q}}[X] = \mu_X.$$

Therefore,

$$\mathbb{E}^{\mathbb{Q}}[X - \mu_X] = \mu_X - \mu_X = 0.$$

It follows that the control variate estimator is unbiased. In fact,

$$\mathbb{E}^{\mathbb{Q}}[Y^{\text{CV}}] = \mathbb{E}^{\mathbb{Q}}[Y + \alpha(X - \mu_X)].$$

By linearity of expectation,

$$\mathbb{E}^{\mathbb{Q}}[Y + \alpha(X - \mu_X)] = \mathbb{E}^{\mathbb{Q}}[Y] + \alpha\mathbb{E}^{\mathbb{Q}}[X - \mu_X].$$

Since

$$\mathbb{E}^{\mathbb{Q}}[Y] = V_0$$

and

$$\mathbb{E}^{\mathbb{Q}}[X - \mu_X] = 0,$$

it follows that

$$\mathbb{E}^{\mathbb{Q}}[Y^{\text{CV}}] = V_0 + \alpha \cdot 0 = V_0.$$

Thus, the control variate construction does not change the target price.

If M simulations are used, the Control Var estimator is

$$\hat{V}_M^{\text{CV}} = \frac{1}{M} \sum_{j=1}^M Y_j^{\text{CV}}.$$

Equivalently,

$$\hat{V}_M^{\text{CV}} = \frac{1}{M} \sum_{j=1}^M [Y_j + \alpha(X_j - \mu_X)].$$

The coefficient α is chosen in order to minimize the variance of Y^{CV} . Since

$$Y^{\text{CV}} = Y + \alpha(X - \mu_X),$$

it follows that

$$\text{Var}(Y^{\text{CV}}) = \text{Var}(Y + \alpha(X - \mu_X)).$$

Using the variance of a sum gives

$$\text{Var}(Y + \alpha(X - \mu_X)) = \text{Var}(Y) + \text{Var}(\alpha(X - \mu_X)) + 2 \text{Cov}(Y, \alpha(X - \mu_X)).$$

Now,

$$\text{Var}(\alpha(X - \mu_X)) = \alpha^2 \text{Var}(X - \mu_X).$$

Since subtracting a constant does not change the variance,

$$\text{Var}(X - \mu_X) = \text{Var}(X).$$

Therefore,

$$\text{Var}(\alpha(X - \mu_X)) = \alpha^2 \text{Var}(X).$$

Moreover,

$$\text{Cov}(Y, \alpha(X - \mu_X)) = \alpha \text{Cov}(Y, X - \mu_X).$$

Since subtracting a constant does not change the covariance,

$$\text{Cov}(Y, X - \mu_X) = \text{Cov}(Y, X).$$

Thus,

$$\text{Cov}(Y, \alpha(X - \mu_X)) = \alpha \text{Cov}(Y, X).$$

Putting these results together gives

$$\text{Var}(Y^{\text{CV}}) = \text{Var}(Y) + \alpha^2 \text{Var}(X) + 2\alpha \text{Cov}(Y, X).$$

To find the value of α that minimizes this variance, differentiation with respect to α gives:

$$\frac{\partial}{\partial \alpha} \text{Var}(Y^{\text{CV}}) = 2\alpha \text{Var}(X) + 2 \text{Cov}(Y, X).$$

The minimum is obtained by setting this derivative equal to zero:

$$2\alpha \text{Var}(X) + 2 \text{Cov}(Y, X) = 0.$$

Dividing by 2 gives

$$\alpha \text{Var}(X) + \text{Cov}(Y, X) = 0.$$

Therefore,

$$\alpha \text{Var}(X) = - \text{Cov}(Y, X).$$

If $\text{Var}(X) > 0$, the optimal coefficient is

$$\alpha^* = -\frac{\text{Cov}(Y, X)}{\text{Var}(X)}.$$

Substituting α^* into the variance formula gives the minimum variance. Starting from

$$\text{Var}(Y^{\text{CV}}) = \text{Var}(Y) + \alpha^2 \text{Var}(X) + 2\alpha \text{Cov}(Y, X),$$

and using

$$\alpha^* = -\frac{\text{Cov}(Y, X)}{\text{Var}(X)},$$

it follows that

$$(\alpha^*)^2 = \frac{\text{Cov}(Y, X)^2}{\text{Var}(X)^2}.$$

Hence,

$$(\alpha^*)^2 \text{Var}(X) = \frac{\text{Cov}(Y, X)^2}{\text{Var}(X)}.$$

Moreover,

$$2\alpha^* \text{Cov}(Y, X) = -2\frac{\text{Cov}(Y, X)^2}{\text{Var}(X)}.$$

Therefore,

$$\text{Var}(Y^{\text{CV}}) = \text{Var}(Y) + \frac{\text{Cov}(Y, X)^2}{\text{Var}(X)} - 2\frac{\text{Cov}(Y, X)^2}{\text{Var}(X)}.$$

Combining the last two terms gives

$$\text{Var}(Y^{\text{CV}}) = \text{Var}(Y) - \frac{\text{Cov}(Y, X)^2}{\text{Var}(X)}.$$

This expression shows that the variance is reduced whenever Y and X are correlated.

If $\text{Cov}(Y, X) = 0$, the control variate does not reduce the variance.

The same result can be expressed in terms of the correlation coefficient

$$\rho_{Y,X} = \frac{\text{Cov}(Y, X)}{\sqrt{\text{Var}(Y)}\sqrt{\text{Var}(X)}}.$$

From this definition,

$$\text{Cov}(Y, X) = \rho_{Y,X}\sqrt{\text{Var}(Y)}\sqrt{\text{Var}(X)}.$$

Squaring both sides gives

$$\text{Cov}(Y, X)^2 = \rho_{Y,X}^2 \text{Var}(Y) \text{Var}(X).$$

Therefore,

$$\frac{\text{Cov}(Y, X)^2}{\text{Var}(X)} = \rho_{Y,X}^2 \text{Var}(Y).$$

Substituting into the minimum variance formula gives

$$\text{Var}(Y^{\text{CV}}) = \text{Var}(Y) - \rho_{Y,X}^2 \text{Var}(Y).$$

Thus,

$$\text{Var}(Y^{\text{CV}}) = \text{Var}(Y) (1 - \rho_{Y,X}^2).$$

This formula shows that the method is effective when the absolute correlation between Y and X is high. The sign of the correlation is not important for the variance reduction itself, because the squared correlation appears in the formula. What matters is that the control variate is strongly related to the target payoff.

In practice, let

$$(y_1, x_1), (y_2, x_2), \dots, (y_M, x_M)$$

be the realized payoff pairs obtained in the Monte Carlo run. The corresponding sample means are

$$\bar{y}_M = \frac{1}{M} \sum_{j=1}^M y_j$$

and

$$\bar{x}_M = \frac{1}{M} \sum_{j=1}^M x_j.$$

The sample covariance between Y and X is

$$\widehat{\text{Cov}}(Y, X) = \frac{1}{M-1} \sum_{j=1}^M (y_j - \bar{y}_M) (x_j - \bar{x}_M).$$

The sample variance of X is

$$\widehat{\text{Var}}(X) = \frac{1}{M-1} \sum_{j=1}^M (x_j - \bar{x}_M)^2.$$

Therefore, the sample estimate of the optimal coefficient is

$$\hat{\alpha} = -\frac{\widehat{\text{Cov}}(Y, X)}{\widehat{\text{Var}}(X)}.$$

Substituting the explicit formulas for the sample covariance and the sample variance, the factor $1/(M-1)$ cancels out. Hence,

$$\hat{\alpha} = -\frac{\sum_{j=1}^M (y_j - \bar{y}_M)(x_j - \bar{x}_M)}{\sum_{j=1}^M (x_j - \bar{x}_M)^2}.$$

Once $\hat{\alpha}$ has been estimated, the adjusted control variate realizations are computed as

$$y_j^{CV} = y_j + \hat{\alpha}(x_j - \mu_X), \quad j = 1, \dots, M.$$

The corresponding realized Control Var price estimate is

$$\hat{v}_M^{CV} = \frac{1}{M} \sum_{j=1}^M y_j^{CV} = \frac{1}{M} \sum_{j=1}^M [y_j + \hat{\alpha}(x_j - \mu_X)].$$

The estimated standard error is computed from the realized control variate values

$$y_1^{CV}, y_2^{CV}, \dots, y_M^{CV}.$$

If s_{CV} denotes their sample standard deviation, then

$$\widehat{\text{SE}}(\hat{V}_M^{CV}) = \frac{s_{CV}}{\sqrt{M}}.$$

In the numerical applications of this thesis, a natural control variate for barrier call options is the discounted payoff of the corresponding vanilla call option:

$$X = e^{-rT}(S_T - K)^+.$$

The reason is that the vanilla call payoff is related to the barrier call payoff and expectation is easy to compute. In the Black–Scholes model, the expectation is given by the Black–Scholes closed pricing formula. In the OUSV model, the expectation is computed by the semi-closed Fourier formula.

1.6 Conditional Monte Carlo

The third variance reduction technique considered in this thesis is conditional Monte Carlo, denoted by CMC. The method is based on replacing the payoff random variable by its conditional expectation with respect to a suitable information set. In option pricing, this technique is useful when part of the future randomness of the payoff can be replaced by a conditional price computed through a pricing formula. The method and its application to barrier options are discussed in Boyle et al. [1997].

Suppose that the target price is

$$V_0 = \mathbb{E}^{\mathbb{Q}}[Y].$$

The plain MC estimator is

$$\hat{V}_M^{\text{plain}} = \frac{1}{M} \sum_{j=1}^M Y_j.$$

In CMC, the original payoff random variable is replaced by a different random variable. Let $\mathcal{G}$ be an information set representing part of the information generated during the simulation. The CMC random variable is defined as

$$Y^{\text{CMC}} = \mathbb{E}^{\mathbb{Q}}[Y \mid \mathcal{G}].$$

Let

$$Y_1^{\text{CMC}}, \dots, Y_M^{\text{CMC}}$$

be independent copies of Y^{CMC} . The corresponding CMC estimator is

$$\hat{V}_M^{\text{CMC}} = \frac{1}{M} \sum_{j=1}^M Y_j^{\text{CMC}}.$$

The important point is that Y^{CMC} is not the same payoff random variable as Y . Once a Monte Carlo sample has been generated, y_j and y_j^{CMC} denote their corresponding numerical realizations. In plain MC, y_j is the discounted payoff obtained by simulating the full path and evaluating the final payoff. In CMC, y_j^{CMC} is the realization of the conditional-expectation random variable.

However, Y^{CMC} has the same expectation as the original random variable Y . Indeed,

$$\mathbb{E}^{\mathbb{Q}}[Y^{\text{CMC}}] = \mathbb{E}^{\mathbb{Q}}[\mathbb{E}^{\mathbb{Q}}[Y \mid \mathcal{G}]].$$

By the tower property of conditional expectation,

$$\mathbb{E}^{\mathbb{Q}} \left[\mathbb{E}^{\mathbb{Q}}[Y \mid \mathcal{G}] \right] = \mathbb{E}^{\mathbb{Q}}[Y].$$

Since

$$\mathbb{E}^{\mathbb{Q}}[Y] = V_0,$$

it follows that

$$\mathbb{E}^{\mathbb{Q}} \left[Y^{\text{CMC}} \right] = V_0.$$

Therefore, the payoff random variable Y^{CMC} has the same expectation as Y . Consequently, the conditional Monte Carlo estimator is unbiased.

The variance reduction property follows from a standard property of conditional expectation (see Boyle et al. [1997]):

$$\text{Var} \left(\mathbb{E}^{\mathbb{Q}}[Y \mid \mathcal{G}] \right) \leq \text{Var}(Y).$$

In the present notation, this means that

$$\text{Var} \left(Y^{\text{CMC}} \right) \leq \text{Var}(Y).$$

Thus, replacing the payoff random variable by its conditional expectation does not increase the variance.

The estimated standard error is computed from the realized CMC values

$$y_1^{\text{CMC}}, y_2^{\text{CMC}}, \dots, y_M^{\text{CMC}}.$$

If s_{CMC} denotes their sample standard deviation, then

$$\widehat{\text{SE}} \left(\hat{V}_M^{\text{CMC}} \right) = \frac{s_{\text{CMC}}}{\sqrt{M}}.$$

The application of this conditional-expectation idea to barrier options is described next. Consider a discretely monitored up-and-in call option with upper barrier B . Let

$$\tau_B = \min \{t_i : S_{t_i} \geq B\}$$

be the first monitoring date at which the barrier is reached. If the barrier is never reached before maturity, the convention is

$$\tau_B > T.$$

The discounted payoff of the up-and-in call is

$$Y^{\text{in}} = e^{-rT} (S_T - K)^+ \mathbf{1}_{\{\tau_B \leq T\}}.$$

In plain MC, the simulated realization is

$$y_j^{\text{in}} = e^{-rT} \left(S_T^{(j)} - K \right)^+ \mathbf{1}_{\{\tau_B^{(j)} \leq T\}}.$$

The plain MC estimator is therefore

$$\hat{C}_M^{\text{in,plain}} = \frac{1}{M} \sum_{j=1}^M Y_j^{\text{in}},$$

while its realized numerical value is

$$\hat{c}_M^{\text{in,plain}} = \frac{1}{M} \sum_{j=1}^M y_j^{\text{in}}.$$

The CMC construction changes the random variables that are averaged. For each simulation j , the path is simulated until the first hitting time $\tau_B^{(j)}$, or until maturity if the barrier is never reached.

If the barrier is never reached, then

$$\tau_B^{(j)} > T,$$

and the up-and-in option is never activated. Therefore,

$$y_j^{\text{in,CMC}} = 0.$$

If the barrier is reached, then

$$\tau_B^{(j)} \leq T.$$

At that time, the knock-in option has become a vanilla call option with remaining maturity

$$T - \tau_B^{(j)}.$$

Therefore, instead of continuing the simulation up to maturity and computing the random terminal payoff

$$e^{-rT} \left(S_T^{(j)} - K \right)^+,$$

the CMC method replaces the future payoff by the conditional value of the corre-

sponding vanilla call at the hitting time.

At the theoretical level, this construction follows directly from the conditional-expectation definition of the CMC payoff random variable. In particular, for the up-and-in option,

$$Y^{\text{in,CMC}} = \mathbb{E}^{\mathbb{Q}} \left[Y^{\text{in}} \mid \mathcal{F}_{\tau_B} \right].$$

Since

$$Y^{\text{in}} = e^{-rT} (S_T - K)^+ \mathbf{1}_{\{\tau_B \leq T\}},$$

it follows that

$$Y^{\text{in,CMC}} = \mathbb{E}^{\mathbb{Q}} \left[e^{-rT} (S_T - K)^+ \mathbf{1}_{\{\tau_B \leq T\}} \mid \mathcal{F}_{\tau_B} \right].$$

At the hitting time, the event $\{\tau_B \leq T\}$ is already known. Therefore, its indicator is measurable with respect to $\mathcal{F}_{\tau_B}$ and can be taken outside the conditional expectation:

$$Y^{\text{in,CMC}} = \mathbf{1}_{\{\tau_B \leq T\}} \mathbb{E}^{\mathbb{Q}} \left[e^{-rT} (S_T - K)^+ \mid \mathcal{F}_{\tau_B} \right].$$

Using

$$e^{-rT} = e^{-r\tau_B} e^{-r(T-\tau_B)},$$

and noting that $e^{-r\tau_B}$ is also known at the hitting time, this becomes

$$Y^{\text{in,CMC}} = e^{-r\tau_B} \mathbf{1}_{\{\tau_B \leq T\}} \mathbb{E}^{\mathbb{Q}} \left[e^{-r(T-\tau_B)} (S_T - K)^+ \mid \mathcal{F}_{\tau_B} \right].$$

The conditional expectation in the last expression is the value, at time τ_B , of the corresponding vanilla call option with remaining maturity $T - \tau_B$. Therefore,

$$\mathbb{E}^{\mathbb{Q}} \left[e^{-r(T-\tau_B)} (S_T - K)^+ \mid \mathcal{F}_{\tau_B} \right] = C(\mathcal{S}_{\tau_B}, K, T - \tau_B).$$

Hence,

$$Y^{\text{in,CMC}} = e^{-r\tau_B} C(\mathcal{S}_{\tau_B}, K, T - \tau_B) \mathbf{1}_{\{\tau_B \leq T\}}.$$

Let

$$C \left(\mathcal{S}_{\tau_B}^{(j)}, K, T - \tau_B^{(j)} \right)$$

denote the time- $\tau_B^{(j)}$ price of the corresponding vanilla call option, conditional on the state of the model at the hitting time, with remaining time to maturity $T - \tau_B^{(j)}$. The notation $\mathcal{S}_{\tau_B}^{(j)}$ represents the relevant state variables in the j -th simulation at the hitting time.

Therefore, for the j -th simulation, the numerical realization of the CMC payoff

random variable is

$$y_j^{\text{in,CMC}} = e^{-r\tau_B^{(j)}} C\left(\mathcal{S}_{\tau_B^{(j)}}^{(j)}, K, T - \tau_B^{(j)}\right) \mathbf{1}_{\{\tau_B^{(j)} \leq T\}}.$$

The corresponding CMC estimator for the up-and-in call price is

$$\hat{C}_M^{\text{in,CMC}} = \frac{1}{M} \sum_{j=1}^M e^{-r\tau_B^{(j)}} C\left(\mathcal{S}_{\tau_B^{(j)}}^{(j)}, K, T - \tau_B^{(j)}\right) \mathbf{1}_{\{\tau_B^{(j)} \leq T\}}.$$

This is the key difference between plain MC and CMC. Plain MC averages the terminal payoff realizations y_j^{in} , whereas CMC averages the conditional-expectation realizations $y_j^{\text{in,CMC}}$, in which the continuation value after the barrier hitting time is replaced by a vanilla call price.

For up-and-out calls, the in-out parity relation gives

$$C_0^{\text{vanilla}} = C_0^{\text{in}} + C_0^{\text{out}}.$$

Hence, the up-and-out CMC estimator is constructed as

$$\hat{C}_M^{\text{out,CMC}} = C_0^{\text{vanilla}} - \hat{C}_M^{\text{in,CMC}}.$$

Since C_0^{vanilla} is computed by a pricing formula and is therefore treated as deterministic, the simulation error of the up-and-out estimator comes from the conditional Monte Carlo estimate of the up-and-in component. Therefore,

$$\text{Var}\left(\hat{C}_M^{\text{out,CMC}}\right) = \text{Var}\left(\hat{C}_M^{\text{in,CMC}}\right).$$

The estimated standard error of the up-and-out estimator obtained through in-out parity is therefore the same as the estimated standard error of the corresponding up-and-in CMC estimator.

Chapter 2

The Black–Scholes Model

2.1 Risk-Neutral Dynamics

The Black–Scholes model is presented as a benchmark framework, because it provides a simple setting in which exact simulation and vanilla option pricing can be understood before moving to the more general OUSV model.

The presentation in this chapter follows the treatment of the Black–Scholes model and risk-neutral valuation in Haugh [2016] and MIT OpenCourseWare [2024].

Let S_t denote the price of the underlying asset at time t . Under the real-world probability measure, the Black–Scholes model assumes that the stock price follows a geometric Brownian motion:

$$dS_t = \mu S_t dt + \sigma S_t dW_t,$$

where μ is the real-world drift, $\sigma > 0$ is the constant volatility, and W_t is a Brownian motion under the real-world probability measure.

For derivative pricing, however, the real-world drift μ is not used. The pricing argument is based on the risk-neutral probability measure $\mathbb{Q}$. Under standard no-arbitrage assumptions, there exists an equivalent martingale measure $\mathbb{Q}$.

In the Black–Scholes model, under this risk-neutral measure $\mathbb{Q}$, the drift of the stock price is the risk-free rate r . Therefore, the risk-neutral dynamics are

$$dS_t = r S_t dt + \sigma S_t dW_t^{\mathbb{Q}},$$

where $W_t^{\mathbb{Q}}$ is a Brownian motion under $\mathbb{Q}$.

The Black–Scholes stochastic differential equation admits a closed solution for

the stock price. Its derivation uses Itô's formula. In general, if a process X_t satisfies

$$dX_t = a_t dt + b_t dW_t,$$

then Itô's formula gives

$$df(t, X_t) = \frac{\partial f}{\partial t}(t, X_t) dt + \frac{\partial f}{\partial x}(t, X_t) dX_t + \frac{1}{2} \frac{\partial^2 f}{\partial x^2}(t, X_t) (dX_t)^2.$$

In the present case, Itô's formula is applied to

$$f(S_t) = \log S_t.$$

Since f does not depend explicitly on time, its time derivative is zero:

$$\frac{\partial f}{\partial t} = 0.$$

The first and second derivatives with respect to S_t are

$$f'(S_t) = \frac{1}{S_t}$$

and

$$f''(S_t) = -\frac{1}{S_t^2}.$$

Therefore,

$$d \log S_t = \frac{1}{S_t} dS_t + \frac{1}{2} \left(-\frac{1}{S_t^2} \right) (dS_t)^2.$$

From the risk-neutral Black–Scholes dynamics,

$$dS_t = r S_t dt + \sigma S_t dW_t^{\mathbb{Q}}.$$

The quadratic variation term is obtained by squaring dS_t :

$$(dS_t)^2 = \left(r S_t dt + \sigma S_t dW_t^{\mathbb{Q}} \right)^2.$$

Expanding,

$$(dS_t)^2 = r^2 S_t^2 (dt)^2 + 2r\sigma S_t^2 dt dW_t^{\mathbb{Q}} + \sigma^2 S_t^2 (dW_t^{\mathbb{Q}})^2.$$

Using the Itô multiplication rules

$$(dt)^2 = 0, \quad dt dW_t^{\mathbb{Q}} = 0, \quad (dW_t^{\mathbb{Q}})^2 = dt,$$

it follows that

$$(dS_t)^2 = \sigma^2 S_t^2 dt.$$

Substituting dS_t and $(dS_t)^2$ into Itô's formula gives

$$d \log S_t = \frac{1}{S_t} \left(r S_t dt + \sigma S_t dW_t^{\mathbb{Q}} \right) + \frac{1}{2} \left(-\frac{1}{S_t^2} \right) \sigma^2 S_t^2 dt.$$

Simplifying,

$$d \log S_t = r dt + \sigma dW_t^{\mathbb{Q}} - \frac{1}{2} \sigma^2 dt.$$

Hence,

$$d \log S_t = \left(r - \frac{1}{2} \sigma^2 \right) dt + \sigma dW_t^{\mathbb{Q}}.$$

Integrating both sides from 0 to T gives:

$$\int_0^T d \log S_t = \int_0^T \left(r - \frac{1}{2} \sigma^2 \right) dt + \int_0^T \sigma dW_t^{\mathbb{Q}}.$$

The left-hand side is

$$\int_0^T d \log S_t = \log S_T - \log S_0.$$

The first term on the right-hand side is

$$\int_0^T \left(r - \frac{1}{2} \sigma^2 \right) dt = \left(r - \frac{1}{2} \sigma^2 \right) T.$$

The stochastic integral is

$$\int_0^T \sigma dW_t^{\mathbb{Q}} = \sigma \left(W_T^{\mathbb{Q}} - W_0^{\mathbb{Q}} \right).$$

Since $W_0^{\mathbb{Q}} = 0$, this becomes

$$\int_0^T \sigma dW_t^{\mathbb{Q}} = \sigma W_T^{\mathbb{Q}}.$$

Therefore,

$$\log S_T - \log S_0 = \left(r - \frac{1}{2} \sigma^2 \right) T + \sigma W_T^{\mathbb{Q}}.$$

Equivalently,

$$\log S_T = \log S_0 + \left(r - \frac{1}{2}\sigma^2\right) T + \sigma W_T^{\mathbb{Q}}.$$

Taking the exponential of both sides gives the closed solution

$$S_T = S_0 \exp \left[\left(r - \frac{1}{2}\sigma^2\right) T + \sigma W_T^{\mathbb{Q}} \right].$$

Since

$$W_T^{\mathbb{Q}} \sim \mathcal{N}(0, T),$$

it can be written as

$$W_T^{\mathbb{Q}} = \sqrt{T} Z,$$

where

$$Z \sim \mathcal{N}(0, 1).$$

Thus,

$$S_T = S_0 \exp \left[\left(r - \frac{1}{2}\sigma^2\right) T + \sigma \sqrt{T} Z \right].$$

This formula is especially important for Monte Carlo simulation. It allows one to simulate the terminal value S_T exactly (so without discretization error).

More generally, for a time step $\Delta t = t_{i+1} - t_i$, the exact simulation formula is

$$S_{t_{i+1}} = S_{t_i} \exp \left[\left(r - \frac{1}{2}\sigma^2\right) \Delta t + \sigma \sqrt{\Delta t} Z_{i+1} \right],$$

where

$$Z_{i+1} \sim \mathcal{N}(0, 1).$$

For European vanilla options, the payoff depends only on S_T , so simulating the terminal value is sufficient. For path-dependent options, such as barrier options, the whole path along the monitoring dates must be simulated.

2.2 The Black–Scholes Pricing Formula

The previous section derived the closed solution of the Black–Scholes stochastic differential equation under the risk-neutral measure. The Black–Scholes closed pricing formula for a European vanilla call option is stated below. The formula is reported in Haugh [2016].

Consider a European call option with strike K and maturity T , written on a non-dividend-paying stock. Therefore, in the notation of Haugh [2016], the setting

corresponds to

$$q = 0.$$

At time zero, the Black–Scholes closed pricing formula for the European call is

$$C_0 = S_0 \Phi(d_1) - K e^{-rT} \Phi(d_2),$$

where

$$d_1 = \frac{\log\left(\frac{S_0}{K}\right) + \left(r + \frac{1}{2}\sigma^2\right)T}{\sigma\sqrt{T}}$$

and

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

Equivalently,

$$d_2 = \frac{\log\left(\frac{S_0}{K}\right) + \left(r - \frac{1}{2}\sigma^2\right)T}{\sigma\sqrt{T}}.$$

Here, Φ denotes the cumulative distribution function of a standard normal random variable. The parameters of the formula are the initial stock price S_0 , the strike price K , the maturity T , the risk-free rate r , and the constant volatility σ .

The corresponding European put price can be obtained from put-call parity. For a non-dividend-paying stock, put-call parity is

$$C_0 - P_0 = S_0 - K e^{-rT}.$$

Rearranging,

$$P_0 = C_0 - S_0 + K e^{-rT}.$$

Substituting the Black–Scholes call price gives the closed pricing formula for the European put:

$$P_0 = K e^{-rT} \Phi(-d_2) - S_0 \Phi(-d_1).$$

2.3 Monte Carlo Pricing in the Black–Scholes Model

The closed solution of the Black–Scholes SDE is very useful for Monte Carlo simulation. Since the transition of the underlying asset is known explicitly, the terminal value S_T can be simulated exactly under the risk-neutral measure.

For a European vanilla call option, the payoff depends only on the terminal value of the underlying asset. Therefore, for the j -th simulation, the terminal value is

generated as

$$S_T^{(j)} = S_0 \exp \left[\left(r - \frac{1}{2} \sigma^2 \right) T + \sigma \sqrt{T} Z_j \right],$$

where

$$Z_j \sim \mathcal{N}(0, 1).$$

The corresponding realized discounted payoff is

$$y_j^{\text{call}} = e^{-rT} \left(S_T^{(j)} - K \right)^+.$$

The realized Monte Carlo estimate of the call price is then

$$\hat{C}_M^{\text{plain}} = \frac{1}{M} \sum_{j=1}^M y_j^{\text{call}}.$$

Equivalently,

$$\hat{C}_M^{\text{plain}} = \frac{1}{M} \sum_{j=1}^M e^{-rT} \left(S_T^{(j)} - K \right)^+.$$

In this case, the Black–Scholes closed pricing formula provides a natural theoretical benchmark. Since the exact price C_0 is known (by using the Black–Scholes formula), the Monte Carlo estimate can be compared with it. If the simulation is correctly implemented, then the estimated price should be close to the Black–Scholes price, up to the statistical error of the Monte Carlo estimator (the Black–Scholes price should lie inside the confidence interval of the Monte Carlo estimate).

This validation step is useful because it checks whether the simulation procedure for vanilla options is consistent with the known Black–Scholes price before applying Monte Carlo methods to more complex products where a theoretical benchmark is not available.

For barrier options, the simulation is more involved. A barrier payoff depends not only on S_T , but also on whether the underlying asset has crossed the barrier during the life of the option (therefore, simulating only S_T is not enough).

If the barrier is monitored at the dates

$$0 = t_0 < t_1 < \dots < t_N = T,$$

then the path must be simulated along this grid. In the Black–Scholes model, this can still be done exactly on the discrete grid using the transition formula

$$S_{t_{i+1}}^{(j)} = S_{t_i}^{(j)} \exp \left[\left(r - \frac{1}{2} \sigma^2 \right) \Delta t + \sigma \sqrt{\Delta t} Z_{i+1}^{(j)} \right],$$

where

$$\Delta t = t_{i+1} - t_i$$

and

$$Z_{i+1}^{(j)} \sim \mathcal{N}(0, 1).$$

For example, the realized discounted payoff of a discretely monitored up-and-out call is

$$y_j^{\text{out}} = e^{-rT} \left(S_T^{(j)} - K \right)^+ \mathbf{1}_{\left\{ \max_{0 \leq i \leq N} S_{t_i}^{(j)} < B \right\}}.$$

The corresponding realized plain MC price estimate is

$$\hat{c}_M^{\text{out,plain}} = \frac{1}{M} \sum_{j=1}^M y_j^{\text{out}}.$$

2.4 Limitations of the Black–Scholes Model

The Black–Scholes model is fundamental in option pricing, but it relies on restrictive assumptions. In particular, the model assumes that the underlying asset follows a geometric Brownian motion with constant volatility. This means that the volatility parameter σ is fixed over time. This assumption is convenient from a mathematical point of view, because it leads to a closed-form solution of the stock price stochastic differential equation and to a closed pricing formula for European vanilla options. However, it is not fully consistent with market data. As discussed in Haugh [2016], stock prices do not always move continuously as predicted by the geometric Brownian motion. In practice, asset prices may exhibit sudden jumps and empirical return distributions often display heavier tails than those implied by a constant volatility model. Another important limitation concerns implied volatility. Given the market price of a European call option, one can invert the Black–Scholes pricing formula and obtain the volatility value that makes the Black–Scholes price equal to the market price. This value is called implied volatility.

More precisely, the implied volatility is the value σ_{imp} such that

$$C^{\text{mkt}}(K, T) = BS(S_0, K, T, r, \sigma_{\text{imp}}),$$

where $C^{\text{mkt}}(K, T)$ denotes the market price of a call option with strike K and maturity T , and $BS(\cdot)$ denotes the Black–Scholes closed pricing formula.

If the Black–Scholes model were perfectly consistent with market prices, the same volatility parameter σ should explain option prices across different strikes and

maturities. In other words, the implied volatility should be constant:

$$\sigma_{\text{imp}}(K, T) = \sigma$$

for all strikes K and maturities T . In practice, this is not what is observed. The implied volatility depends on the strike and on the maturity, generating a volatility surface:

$$(K, T) \mapsto \sigma_{\text{imp}}(K, T).$$

This effect is illustrated in Figure 2.1, which shows an example of implied volatility surface reported in Haugh [2016]. The surface is clearly not flat: implied volatility changes across both strike and time-to-maturity.

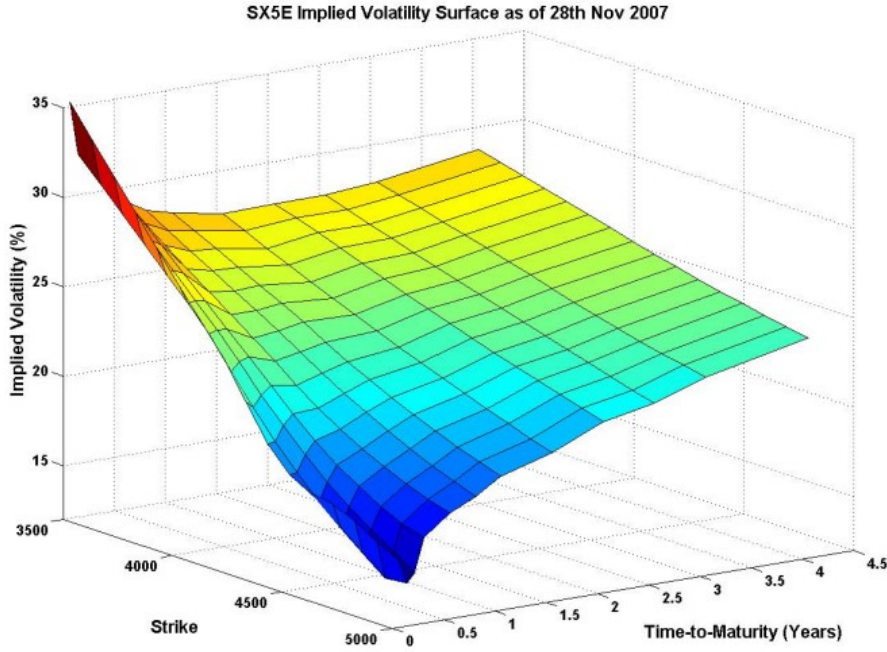


Figure 2.1: Implied volatility surface. Source: Haugh [2016].

(It could also be shown that, for a fixed maturity, implied volatility varies with the strike. This pattern is widely known and is commonly referred to as a volatility smile or volatility skew). The existence of a non-flat implied volatility surface shows that the Black–Scholes model cannot fully capture the behaviour of option prices observed in the market. This limitation motivates the use of more general models. A natural direction is to replace constant volatility with a stochastic volatility process. In such models, volatility is itself random and can evolve over time. This makes the model more flexible and allows it to capture richer dynamics than the constant-volatility Black–Scholes framework. The OUSV model considered in this thesis follows this

idea. Instead of assuming a constant volatility parameter, volatility is driven by a stochastic factor with Ornstein–Uhlenbeck dynamics. This provides a more flexible framework for pricing options, while also making Monte Carlo simulation more involved.

Chapter 3

The Ornstein–Uhlenbeck Stochastic Volatility Model

3.1 Model Dynamics

The limitations of the Black–Scholes model discussed in the previous chapter motivate the use of more flexible models for the dynamics of volatility. The model considered in this thesis is the Ornstein–Uhlenbeck driven stochastic volatility model, abbreviated as OUSV.

The notation used in this chapter follows Brignone and Sgarra [2026]. The analysis is conducted on a filtered probability space

$$(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t \in [0, T]}, \mathbb{Q}),$$

where $\mathbb{Q}$ denotes the risk-neutral probability measure. Under $\mathbb{Q}$, the OUSV model is defined by the following system of stochastic differential equations:

$$\begin{cases} dS_t = rS_t dt + S_t v_t \left(\rho dW_t^{(2)} + \sqrt{1 - \rho^2} dW_t^{(1)} \right), \\ dv_t = k(\theta - v_t) dt + \xi dW_t^{(2)}, \\ dV_t = 2k \left(\frac{\xi^2}{2k} + \theta \sqrt{V_t} - V_t \right) dt + 2\xi \sqrt{V_t} dW_t^{(2)}. \end{cases}$$

Here, S_t denotes the asset price and r is the constant risk-free rate. The processes $W_t^{(1)}$ and $W_t^{(2)}$ are independent standard Brownian motions under the risk-neutral measure.

The process v_t follows an Ornstein–Uhlenbeck dynamics:

$$dv_t = k(\theta - v_t) dt + \xi dW_t^{(2)}.$$

In this thesis, v_t will be called the latent volatility factor. This terminology is important. Although v_t drives the volatility of the asset price, it should not be interpreted directly as the volatility of the underlying asset. The reason is that v_t is Gaussian and can therefore take negative values.

The variance process is defined as

$$V_t = v_t^2.$$

Thus, the true instantaneous volatility of the asset price is

$$\sqrt{V_t} = |v_t|.$$

The parameters of the model have the following interpretation. The parameter $k > 0$ is the speed of mean reversion of the latent volatility factor. The parameter $\theta > 0$ is its long-run mean, while $\xi > 0$ is the volatility of the latent factor. The parameter $\rho \in (-1, 1)$ determines the correlation structure between the asset-price shock and the shock driving the latent volatility factor. The OUSV model is therefore more flexible than the Black–Scholes model.

3.2 From Black–Scholes Exact Simulation to OUSV Simulation Schemes

The transition from the Black–Scholes model to the OUSV model changes the nature of the simulation problem. In the Black–Scholes model, the stock price has a closed-form solution for the stochastic differential equation. Under the risk-neutral measure,

$$S_T = S_0 \exp \left[\left(r - \frac{1}{2} \sigma^2 \right) T + \sigma \sqrt{T} Z \right],$$

where

$$Z \sim \mathcal{N}(0, 1).$$

This formula provides an exact simulation rule for the stock price. More generally, on a time interval $[t_i, t_{i+1}]$, the exact transition is

$$S_{t_{i+1}} = S_{t_i} \exp \left[\left(r - \frac{1}{2} \sigma^2 \right) \Delta t + \sigma \sqrt{\Delta t} Z_{i+1} \right],$$

where

$$\Delta t = t_{i+1} - t_i, \quad Z_{i+1} \sim \mathcal{N}(0, 1).$$

Therefore, the Black–Scholes dynamics can be simulated exactly on any chosen time grid.

The situation is different in the OUSV model. In this case, the asset price is driven by a stochastic volatility factor. The risk-neutral dynamics of the asset price are

$$\frac{dS_t}{S_t} = r dt + v_t \left(\rho dW_t^{(2)} + \sqrt{1 - \rho^2} dW_t^{(1)} \right),$$

where the latent factor v_t follows its own stochastic dynamics.

Applying Itô's formula to the log-return, and using $V_t = v_t^2$, gives

$$X_T = \log \left(\frac{S_T}{S_0} \right) = rT - \frac{1}{2} \int_0^T V_s ds + \rho \int_0^T v_s dW_s^{(2)} + \sqrt{1 - \rho^2} \int_0^T v_s dW_s^{(1)}.$$

This expression shows the key difference with respect to Black–Scholes. In the OUSV model, the log-return depends on integrated quantities involving the latent factor, such as

$$\int_0^T V_s ds,$$

and on stochastic integrals whose integrand is v_s . These quantities depend on the path followed by the latent volatility factor over time.

Unlike the Black–Scholes model, the OUSV model does not provide a simple closed-form transition formula for the asset price. Exact simulation methods nevertheless exist, but they can be computationally demanding, particularly for frequently monitored path-dependent options. For this reason, the numerical analysis in this thesis focuses on approximate simulation schemes that provide different trade-offs between discretization accuracy and computational cost.

If

$$0 = t_0 < t_1 < \dots < t_N = T$$

is the simulation grid and

$$\Delta t = \frac{T}{N},$$

increasing N reduces the time step Δt and usually improves the approximation of the continuous-time dynamics.

3.3 Fourier Pricing Formula for European Options under OUSV

Although the OUSV model is more complex than the Black–Scholes model, European vanilla options can still be priced efficiently. The reason is that the characteristic function of the log-return is available under the OUSV model.

Let the log-return be defined as

$$X_T = \log \left(\frac{S_T}{S_0} \right).$$

Let

$$\Phi(u) = \mathbb{E}^{\mathbb{Q}} \left[e^{iuX_T} \mid v_0 \right]$$

denote the characteristic function of X_T under the risk-neutral measure. Following Brignone and Sgarra [2026], this characteristic function can be written in the numerically continuous form

$$\Phi(u) = \mathbb{E}^{\mathbb{Q}} \left[e^{iuX_T} \mid v_0 \right] = \exp \left[iurT + L(T, \underline{\theta})V_0 + G(T, \underline{\theta})v_0 + H(T, \underline{\theta}) + Q_2 \right],$$

where

$$\underline{\theta} = (k, \theta, \xi, \rho)$$

denotes the vector of OUSV parameters appearing in the functions L , G , H and Q_2 . By defining

$$\zeta_1 = \cosh(\gamma_1 T), \quad \zeta_2 = \sinh(\gamma_1 T),$$

it follows that

$$Q_2 = -\frac{1}{2}\gamma_1 T - \frac{1}{2} \log \left[\frac{1}{2} \left(1 + \gamma_2 + e^{-\gamma_1 T} (1 - \gamma_2) \right) \right].$$

The expression based on Q_2 is algebraically equivalent to the formulation based on

$$Q_1 = -\frac{1}{2} \log (\zeta_1 + \gamma_2 \zeta_2).$$

The advantage of Q_2 is mainly numerical. Although both expressions represent the same theoretical characteristic function, the formula based on Q_1 contains a complex logarithm. When this logarithm is evaluated by numerical software for different values of u , it may produce artificial jumps in the computed characteristic function. The representation based on Q_2 avoids this problem and provides a smoother and more stable numerical evaluation of $\Phi(u)$ [Brignone and Sgarra, 2026].

$$L(T, \underline{\theta}) = \frac{1}{2\xi^2} \left(k - \frac{\gamma_1 \zeta_2 + \gamma_1 \gamma_2 \zeta_1}{\zeta_1 + \gamma_2 \zeta_2} \right) - \frac{\rho i u}{2\xi},$$

$$\begin{aligned} H(T, \underline{\theta}) = & -\frac{1}{2} \rho i u \xi T + \frac{kT}{2} \\ & + \frac{k^2 \theta^2 \gamma_1^2 - \gamma_3^2}{2\xi^2 \gamma_1^3} \left(\frac{\zeta_2}{\zeta_1 + \gamma_2 \zeta_2} - \gamma_1 T \right) \\ & + \frac{(k\theta \gamma_1 - \gamma_2 \gamma_3) \gamma_3}{\xi^2 \gamma_1^3} \left(\frac{\zeta_1 - 1}{\zeta_1 + \gamma_2 \zeta_2} \right), \end{aligned}$$

and

$$G(T, \underline{\theta}) = \frac{1}{\xi^2 \gamma_1} \left(-\theta k \gamma_1 + \frac{k\theta \gamma_1 - \gamma_2 \gamma_3 + \gamma_3 \zeta_2 + \gamma_2 \gamma_3 \zeta_1}{\zeta_1 + \gamma_2 \zeta_2} \right),$$

with

$$\gamma_1 = \sqrt{2\xi^2 \left(\frac{u^2}{2} (1 - \rho^2) + \frac{i u}{2} \left(1 - \frac{2k\rho}{\xi} \right) \right) + k^2},$$

$$\gamma_2 = \frac{1}{\gamma_1} (k - \xi i u \rho),$$

and

$$\gamma_3 = k^2 \theta - \xi i u k \theta \rho.$$

Using this characteristic function, the European call price under the OUSV model can be written through a Fourier inversion formula. In the notation used in this thesis, the OUSV Fourier price of a European call option with strike K and maturity T is

$$\begin{aligned} C_0^{\text{Fourier}} = & \frac{S_0 - K e^{-rT}}{2} \\ & + \frac{e^{-rT}}{\pi} \int_0^{+\infty} \text{Re} \left(\frac{e^{-iu \log\left(\frac{K}{S_0}\right)}}{iu} \Phi(u - i) \right) du \\ & - \frac{K e^{-rT}}{\pi} \int_0^{+\infty} \text{Re} \left(\frac{e^{-iu \log\left(\frac{K}{S_0}\right)}}{iu} \Phi(u) \right) du. \end{aligned}$$

Here, $\text{Re}(\cdot)$ denotes the real part of a complex number. The formula expresses the call price in terms of the characteristic function. Therefore, once the characteristic function is available, the pricing problem is reduced to the numerical evaluation of deterministic integrals. For this reason, the formula is referred to in this thesis as a semi-closed Fourier pricing formula. It is not a closed pricing formula like the Black–Scholes formula, because the integrals must still be computed numerically. In

the numerical implementation, the integrals are evaluated by numerical quadrature. This produces a highly accurate vanilla call price under the OUSV model, which will be used as a benchmark value in the Monte Carlo experiments. This benchmark is important for the thesis for three reasons. First, it allows the OUSV simulation schemes to be validated on a European vanilla call option. Indeed, comparing the Monte Carlo vanilla price with C_0^{Fourier} provides a direct check of the quality of the simulation scheme. Second, the Fourier vanilla call price is used in the Control Var method. Indeed, C_0^{Fourier} is the known expectation of the control variate (the vanilla call).

Third, the same pricing formula is used in the CMC method. When the barrier is reached, the remaining payoff can be replaced by the Fourier formula price of the residual vanilla call option with the remaining time to maturity. The Fourier pricing formula contains two integrals over the interval $[0, +\infty)$. In the numerical implementation, both integrals are evaluated using Gauss–Laguerre quadrature with 64 nodes, following the numerical approach used in Brignone and Sgarra [2026].

Gauss–Laguerre quadrature is designed to approximate integrals of the form

$$\int_0^{+\infty} e^{-u} f(u) du.$$

If $u_1, \dots, u_n$ are the quadrature nodes and $w_1, \dots, w_n$ are the corresponding weights, then

$$\int_0^{+\infty} e^{-u} f(u) du \approx \sum_{j=1}^n w_j f(u_j).$$

The integrals appearing in the Fourier formula are not written directly in this weighted form. They are of the form

$$\int_0^{+\infty} g(u) du.$$

To apply Gauss–Laguerre quadrature, the integral is rewritten as

$$\int_0^{+\infty} g(u) du = \int_0^{+\infty} e^{-u} (e^u g(u)) du.$$

Therefore, the Gauss–Laguerre approximation becomes

$$\int_0^{+\infty} g(u) du = \int_0^{+\infty} e^{-u} (e^u g(u)) du \approx \sum_{j=1}^{64} w_j e^{u_j} g(u_j).$$

In the present case, the first integrand is

$$g_1(u) = \operatorname{Re} \left(\frac{e^{-iu \log\left(\frac{K}{S_0}\right)}}{iu} \Phi(u - i) \right),$$

while the second integrand is

$$g_2(u) = \operatorname{Re} \left(\frac{e^{-iu \log\left(\frac{K}{S_0}\right)}}{iu} \Phi(u) \right).$$

Hence, the two integrals are approximated by

$$\int_0^{+\infty} g_1(u) du = \int_0^{+\infty} e^{-u} (e^u g_1(u)) du \approx \sum_{j=1}^{64} w_j e^{u_j} g_1(u_j)$$

and

$$\int_0^{+\infty} g_2(u) du = \int_0^{+\infty} e^{-u} (e^u g_2(u)) du \approx \sum_{j=1}^{64} w_j e^{u_j} g_2(u_j).$$

Substituting these approximations into the Fourier pricing formula gives the numerical benchmark

$$\begin{aligned} C_0^{\text{Fourier}} &\approx \frac{S_0 - K e^{-rT}}{2} \\ &\quad + \frac{e^{-rT}}{\pi} \sum_{j=1}^{64} w_j e^{u_j} g_1(u_j) \\ &\quad - \frac{K e^{-rT}}{\pi} \sum_{j=1}^{64} w_j e^{u_j} g_2(u_j). \end{aligned}$$

In the MATLAB implementation, the 64 Gauss–Laguerre nodes and weights are computed before evaluating the Fourier pricing formula. Therefore, the quadrature rule is constructed only once and this implementation is efficient because the construction of the quadrature nodes and weights is not repeated for every evaluation of the Fourier price. Once the nodes and weights are available, the Fourier price is obtained by evaluating the characteristic function at the required complex arguments u_j and $u_j - i$, multiplying by the quadrature weights, and summing the resulting terms.

3.4 Simulation Schemes for the OUSV Model

The previous section showed that, under the OUSV model, the asset price does not admit a simple closed-form transition rule analogous to the Black–Scholes one.

Let

$$0 = t_0 < t_1 < \dots < t_N = T$$

be an equally spaced time grid, with

$$\Delta t = \frac{T}{N}.$$

A simulation scheme for the OUSV model provides a rule to move from time t_i to time t_{i+1} . Starting from the initial values S_0 and v_0 , the objective is to generate the values

$$S_{t_i}, \quad v_{t_i}, \quad V_{t_i} = v_{t_i}^2,$$

for

$$i = 0, 1, \dots, N.$$

Once the path has been generated on the grid, the payoff of the option can be computed.

This thesis considers three simulation schemes for the OUSV model. The first one is the Euler scheme, which is the simplest discretization method and provides a natural baseline [Boyle et al., 1997, Glasserman, 2003]. The second one is a scheme based on the approach of van Haastrecht et al. [2014], designed for the Schöbel–Zhu and OUSV framework. The third one is the Inverse Gaussian Scheme proposed by Brignone and Sgarra [2026], which is a low-bias simulation scheme for the OUSV model. The three schemes differ in the way they approximate the OUSV dynamics over each time step. Euler uses a local discretization. The scheme based on van Haastrecht, Lord and Pelsser improves the treatment of the latent factor and of the integrated quantities entering the log-price dynamics. The Inverse Gaussian Scheme further refines this approach by simulating additional integrated quantities and by approximating the conditional distribution of the integrated variance. These differences are important because they affect both accuracy and computational cost. A simpler scheme is generally faster, but it may introduce a larger discretization bias. A more refined scheme may approximate the continuous-time dynamics more accurately, but it usually requires more computational time.

In the numerical part of the thesis, the three schemes are first validated on a European vanilla call option. In that case, the OUSV Fourier price introduced in the previous section provides a deterministic benchmark. After this validation, the

same schemes are applied to up-and-in and up-and-out call options. The following subsections describe the three simulation schemes in more detail.

3.4.1 Euler Scheme

The Euler scheme is the simplest discretization method considered in this thesis and is used as a natural baseline [Boyle et al., 1997, Glasserman, 2003]. It approximates the continuous-time dynamics by dividing the time horizon into small intervals. At each step, the next value of the process is obtained from its current value by adding the corresponding drift term and a random Gaussian shock. This leads to a recursive simulation rule on the time grid

$$0 = t_0 < t_1 < \cdots < t_N = T,$$

where

$$\Delta t = \frac{T}{N}.$$

In the OUSV model, the latent volatility factor satisfies

$$dv_t = k(\theta - v_t) dt + \xi dW_t^{(2)}.$$

Applying the Euler scheme to the latent volatility factor gives

$$v_{i+1} = v_i + k(\theta - v_i)\Delta t + \xi\sqrt{\Delta t} Z_{i+1}^{(2)},$$

The variance process can then be recovered from the simulated latent factor as

$$V_{i+1} = v_{i+1}^2.$$

The asset price dynamics are

$$\frac{dS_t}{S_t} = r dt + v_t \left(\rho dW_t^{(2)} + \sqrt{1 - \rho^2} dW_t^{(1)} \right).$$

Instead of applying Euler directly to S_t , the implementation works with the log-price. Define

$$\ell_t = \log S_t.$$

The dynamics of ℓ_t can now be derived. Consider the function

$$f(S) = \log S.$$

Then

$$f'(S) = \frac{1}{S}, \quad f''(S) = -\frac{1}{S^2}.$$

By Itô's formula,

$$d\ell_t = df(S_t) = \frac{1}{S_t} dS_t - \frac{1}{2S_t^2} (dS_t)^2.$$

From the OUSV asset price dynamics,

$$dS_t = rS_t dt + S_t v_t \left(\rho dW_t^{(2)} + \sqrt{1 - \rho^2} dW_t^{(1)} \right).$$

Therefore,

$$\frac{1}{S_t} dS_t = r dt + v_t \left(\rho dW_t^{(2)} + \sqrt{1 - \rho^2} dW_t^{(1)} \right).$$

The quadratic variation term is

$$(dS_t)^2 = S_t^2 v_t^2 \left(\rho dW_t^{(2)} + \sqrt{1 - \rho^2} dW_t^{(1)} \right)^2.$$

Since $W^{(1)}$ and $W^{(2)}$ are independent Brownian motions,

$$(dW_t^{(1)})^2 = dt, \quad (dW_t^{(2)})^2 = dt, \quad dW_t^{(1)} dW_t^{(2)} = 0.$$

Hence,

$$\left(\rho dW_t^{(2)} + \sqrt{1 - \rho^2} dW_t^{(1)} \right)^2 = \rho^2 dt + (1 - \rho^2) dt = dt.$$

It follows that

$$(dS_t)^2 = S_t^2 v_t^2 dt.$$

Substituting into Itô's formula gives

$$d\ell_t = \left(r - \frac{1}{2} v_t^2 \right) dt + v_t \left(\rho dW_t^{(2)} + \sqrt{1 - \rho^2} dW_t^{(1)} \right).$$

This is the stochastic differential equation for the log-price. Applying the Euler scheme to this equation gives

$$\ell_{i+1} = \ell_i + \left(r - \frac{1}{2} v_i^2 \right) \Delta t + v_i \left(\rho \Delta W_{i+1}^{(2)} + \sqrt{1 - \rho^2} \Delta W_{i+1}^{(1)} \right),$$

where

$$\Delta W_{i+1}^{(1)} = \sqrt{\Delta t} Z_{i+1}^{(1)}, \quad \Delta W_{i+1}^{(2)} = \sqrt{\Delta t} Z_{i+1}^{(2)}.$$

Therefore,

$$\ell_{i+1} = \ell_i + \left(r - \frac{1}{2}v_i^2\right) \Delta t + v_i \sqrt{\Delta t} \left(\rho Z_{i+1}^{(2)} + \sqrt{1 - \rho^2} Z_{i+1}^{(1)}\right),$$

where $Z_{i+1}^{(1)}$ and $Z_{i+1}^{(2)}$ are independent standard normal random variables.

After updating the log-price, the asset price is reconstructed as

$$S_{i+1} = e^{\ell_{i+1}}.$$

This construction is crucial in order to ensure that the simulated asset price remains strictly positive. This is why the log-price is the variable used for the numerical recursion, while the asset price is needed to compute the payoff and to check the barrier condition. The same shock $Z_{i+1}^{(2)}$ is used both in the Euler update of the latent volatility factor and of the asset price. This is important because $W^{(2)}$ is the Brownian motion shared by the latent factor and by the asset price dynamics. The shock $Z_{i+1}^{(1)}$, instead, drives the independent component of the asset price only.

In the numerical implementation, the recursion starts from

$$v_0, \quad S_0, \quad \ell_0 = \log S_0.$$

At each time step, the latent volatility factor is updated using the Euler formula. Then the log-price is updated using the previous value of the latent factor v_i . Finally, the asset price is recovered by exponentiation. This gives the simulated values of S_t along the grid. The Euler scheme is easy to implement and provides a useful benchmark. However, it is only an approximate simulation scheme. Its accuracy depends on the time step Δt , and a smaller time step is generally required to reduce the discretization error.

3.4.2 Scheme Based on van Haastrecht, Lord and Pelsser

The second simulation scheme considered in this thesis is based on the approach proposed by van Haastrecht et al. [2014]. In this thesis, this scheme is also denoted by VHP, from the names of van Haastrecht, Lord and Pelsser. The reason for introducing this scheme is that the Euler scheme is very simple. The VHP scheme improves on Euler in two main directions. First, the latent volatility factor is simulated from its exact conditional distribution, since v_t follows an Ornstein–Uhlenbeck process. Second, when simulating the log-price, the scheme doesn't use anymore only the initial value of the latent factor but it uses even information about the path of the latent factor over the time interval. This is important because the log-price depends

on path-integrated quantities involving v_t and $V_t = v_t^2$ and not only on the value of the latent factor at the beginning of the interval as it was the case in Euler.

This last point is related to the leaking correlation problem discussed by van Haastrecht et al. [2014]. In stochastic volatility models, the correlation between the asset price and the volatility factor is an important driver of option prices and a discretization scheme that does not reproduce this dependence accurately, such as the Euler scheme, may introduce non-negligible pricing bias. The VHP scheme is designed to preserve this dependence more accurately.

The latent volatility factor satisfies

$$dv_t = k(\theta - v_t) dt + \xi dW_t^{(2)}.$$

Since this is an Ornstein–Uhlenbeck process, its exact solution over one time step is known. Following van Haastrecht et al. [2014], it can be written as

$$v_{i+1} = v_i e^{-k\Delta t} + \int_{t_i}^{t_{i+1}} k\theta e^{-k(t_{i+1}-s)} ds + \int_{t_i}^{t_{i+1}} \xi e^{-k(t_{i+1}-s)} dW_s^{(2)}.$$

The first two terms are deterministic once v_i is known. The last term is an Itô integral with deterministic integrand. Therefore, conditional on v_i , this last term is normally distributed. As a consequence, $v_{i+1} \mid v_i$ is normally distributed, because it is obtained by adding deterministic terms to a Gaussian random variable.

More precisely, the conditional mean and conditional standard deviation of v_{i+1} are

$$\mathbb{E}[v_{i+1} \mid v_i] = K_1 v_i + K_2, \quad \text{Std}(v_{i+1} \mid v_i) = K_3.$$

Hence, the exact transition of the latent factor can be written in the following simulation form:

$$v_{i+1} = K_1 v_i + K_2 + K_3 Z_{i+1}^{(2)},$$

where

$$K_1 = e^{-k\Delta t}, \quad K_2 = \theta(1 - e^{-k\Delta t}),$$

and

$$K_3 = \xi \sqrt{\frac{1 - e^{-2k\Delta t}}{2k}}.$$

Here,

$$Z_{i+1}^{(2)} \sim N(0, 1).$$

This is the first main difference with Euler: the latent factor is not approximated by an Euler step, but is sampled exactly from its transition.

The second ingredient of the scheme concerns the log-price update. The starting point is the integrated dynamics of the log-price. Adapting the notation of van Haastrecht et al. [2014] to the OUSV model used in this thesis, and adding the risk-neutral drift r , the log-price over the interval $[t_i, t_{i+1}]$ can be written as

$$\begin{aligned}\ell_{i+1} = \ell_i + r\Delta t - \frac{1}{2} \int_{t_i}^{t_{i+1}} v_s ds + \frac{\rho k}{\xi} \int_{t_i}^{t_{i+1}} (V_s - \theta v_s) ds \\ + \frac{\rho}{2\xi} (V_{i+1} - V_i - \xi^2 \Delta t) + \sqrt{1 - \rho^2} \int_{t_i}^{t_{i+1}} v_s dW_s^{(1)}.\end{aligned}$$

The log-price update depends on the integrated latent factor and on the integrated variance over the interval $[t_i, t_{i+1}]$.

The paper approximates integrals of powers of the latent factor by a linear combination of their values at the two endpoints of the interval:

$$\int_{t_i}^{t_{i+1}} v_s^p ds \approx (\delta_1 v_i^p + \delta_2 v_{i+1}^p) \Delta t, \quad p \in \{1, 2\}.$$

In the central version of the scheme,

$$\delta_1 = \delta_2 = \frac{1}{2}.$$

Thus,

$$\int_{t_i}^{t_{i+1}} v_s ds \approx \frac{1}{2}(v_i + v_{i+1})\Delta t$$

and

$$\int_{t_i}^{t_{i+1}} V_s ds = \int_{t_i}^{t_{i+1}} v_s^2 ds \approx \frac{1}{2}(v_i^2 + v_{i+1}^2)\Delta t.$$

Using these trapezoidal approximations in the previous log-price expression, the one-step update can be written as

$$\begin{aligned}\ell_{i+1} = \ell_i + r\Delta t + C_0 + C_1 v_i + C_2 v_{i+1} + C_3 v_i^2 + C_4 v_{i+1}^2 \\ + C_5 \sqrt{\delta_1 v_i^2 + \delta_2 v_{i+1}^2} Z_{i+1}^{(1)}.\end{aligned}$$

(The term $r\Delta t$ appears because the OUSV dynamics in this thesis are written under the risk-neutral measure with constant risk-free rate r). The remaining coefficients are

$$C_0 = -\frac{1}{2}\rho\xi\Delta t,$$

$$C_1 = -\delta_1 \frac{\rho k \theta}{\xi} \Delta t, \quad C_2 = -\delta_2 \frac{\rho k \theta}{\xi} \Delta t,$$

$$C_3 = -\frac{1}{2}\delta_1\Delta t + \frac{\rho}{\xi}\left(\delta_1 k\Delta t - \frac{1}{2}\right),$$

$$C_4 = -\frac{1}{2}\delta_2\Delta t + \frac{\rho}{\xi}\left(\delta_2 k\Delta t + \frac{1}{2}\right),$$

and

$$C_5 = \sqrt{1 - \rho^2}\sqrt{\Delta t}.$$

Once the log-price has been updated, the asset price is reconstructed as

$$S_{i+1} = e^{\ell_{i+1}}.$$

So to summarize, in Euler, the log-price update uses only v_i . In the VHP scheme, instead, the update uses both v_i and v_{i+1} to approximate the integrated latent factor and the integrated variance (this is a consequence of the trapezoidal approximation of the two integrals). Therefore, it incorporates more information about the behavior of the latent factor during the time step. This is why the scheme is expected to reduce discretization bias with respect to Euler.

In the numerical implementation, the one-step VHP update is coded in a separate function. At each time step, the function first simulates v_{i+1} using the exact Ornstein–Uhlenbeck transition. It then sets

$$\delta_1 = \delta_2 = \frac{1}{2},$$

computes the coefficients $C_0, \dots, C_5$, updates the log-price according to the formula above, and finally reconstructs the asset price by exponentiation. As in the Euler case, the path-generation mechanism is the same for up-and-in and up-and-out options.

3.4.3 Inverse Gaussian Scheme

The third simulation scheme considered in this thesis is the Inverse Gaussian Scheme proposed by Brignone and Sgarra [2026]. This is the main low-bias simulation scheme used in the paper for the OUSV model. The reason for introducing the Inverse Gaussian Scheme is that the VHP scheme still approximates the integrated quantities over each time interval by a trapezoidal rule. The Inverse Gaussian Scheme improves this step by simulating exactly the integrated latent factor jointly with the terminal latent factor, and by treating the integrated variance as a random variable whose conditional distribution is approximated through moment matching. The scheme is

based on three main ideas. First, the pair formed by the terminal latent factor and the integrated latent factor is simulated from its exact bivariate normal distribution. Second, the integrated variance is simulated conditionally by matching its first two moments with an Inverse Gaussian distribution. Third, the log-return over the time step is simulated conditionally on the quantities generated in the previous steps.

Define the integrated latent factor over the interval as

$$I^v = \int_{t_i}^{t_{i+1}} v_s ds.$$

The first step of the Inverse Gaussian Scheme consists in simulating the pair

$$(v_{i+1}, I^v)$$

conditionally on v_i . Since v_t is an Ornstein–Uhlenbeck process, this pair has a bivariate normal distribution:

$$\begin{pmatrix} v_{i+1} \\ I^v \end{pmatrix} \bigg| v_i \sim \mathcal{N}(\mu, \Sigma).$$

The mean vector is

$$\mu = \begin{pmatrix} v_i e^{-k\Delta t} + \theta(1 - e^{-k\Delta t}) \\ \theta\Delta t + (v_i - \theta)\frac{1 - e^{-k\Delta t}}{k} \end{pmatrix}.$$

The covariance matrix is

$$\Sigma = \begin{pmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{12} & \sigma_{22} \end{pmatrix},$$

where

$$\sigma_{11} = \frac{\xi^2}{2k} (1 - e^{-2k\Delta t}),$$

$$\sigma_{12} = \frac{\xi^2}{2k^2} (1 - e^{-k\Delta t})^2,$$

and

$$\sigma_{22} = \frac{\xi^2}{k^2} \left(\Delta t - \frac{1 - e^{-k\Delta t}}{k} \right) - \frac{\xi^2}{2k^3} (1 - e^{-k\Delta t})^2.$$

The second step concerns the integrated variance

$$I^V = \int_{t_i}^{t_{i+1}} V_s ds = \int_{t_i}^{t_{i+1}} v_s^2 ds.$$

In the VHP scheme, this quantity is approximated by a trapezoidal rule. The

Inverse Gaussian Scheme instead considers its conditional distribution given v_i , v_{i+1} and I^v . Following Brignone and Sgarra [2026], define

$$I^c = I^V \Big| (v_i, v_{i+1}, I^v).$$

The exact distribution of I^c is complicated. However, its first two cumulants can be obtained from the Laplace transform. Denoting this Laplace transform by $L(u)$, the mean and variance are

$$\mathbb{E}[I^c] = - \frac{\partial}{\partial u} \log L(u) \Big|_{u=0},$$

and

$$\text{Var}(I^c) = \frac{\partial^2}{\partial u^2} \log L(u) \Big|_{u=0}.$$

Let

$$c_1 = \mathbb{E}[I^c], \quad c_2 = \text{Var}(I^c).$$

The Inverse Gaussian Scheme approximates the conditional distribution of I^c by an Inverse Gaussian distribution whose parameters are chosen by matching these two moments. Therefore,

$$I^c \approx \text{IG}(\nu, \lambda),$$

where

$$\nu = c_1$$

and

$$\lambda = \frac{c_1^3}{c_2}.$$

These are the main differences with respect to VHP. The integrated stochastic volatility factor and the integrated variance are not replaced by a trapezoidal approximation, instead, they are simulated.

The third step consists in simulating the log-return over the interval. In this subsection, following the notation of the Inverse Gaussian Scheme in Brignone and Sgarra [2026], X_{i+1} denotes the one-step log-return over $[t_i, t_{i+1}]$, namely

$$X_{i+1} = \log \left(\frac{S_{i+1}}{S_i} \right).$$

Conditionally on I^V , I^v , v_i and v_{i+1} , this one-step log-return is normally distributed:

$$X_{i+1} \sim \mathcal{N}(m, s^2),$$

where

$$s^2 = (1 - \rho^2)I^V$$

and

$$m = \left(r - \frac{\rho\xi}{2}\right) \Delta t + \frac{\rho}{2\xi} (V_{i+1} - V_i) - \frac{k\theta\rho}{\xi} I^v + \left(\frac{k\rho}{\xi} - \frac{1}{2}\right) I^V.$$

Thus, the simulated one-step log-return is

$$X_{i+1} = m + sZ^X,$$

where

$$Z^X \sim \mathcal{N}(0, 1).$$

Finally, the asset price is updated as

$$S_{i+1} = S_i e^{X_{i+1}},$$

In the numerical implementation, the one-step Inverse Gaussian Scheme update is coded in a separate function. At each time step, the function first simulates

$$(v_{i+1}, I^v)$$

from the bivariate normal distribution described above. This is implemented through the Cholesky decomposition of the covariance matrix Σ .

Then, the conditional mean c_1 and variance c_2 of the integrated variance are computed using a function generated symbolically from the derivatives of the Laplace transform. These quantities are used to construct the Inverse Gaussian parameters

$$\nu = c_1, \quad \lambda = \frac{c_1^3}{c_2}.$$

The conditional integrated variance I^V is then sampled from the moment-matched Inverse Gaussian distribution and this sampling requires one standard normal random variable and one uniform random variable.

Finally, the conditional mean m and variance s^2 of the one-step log-return are computed, the log-return X_{i+1} is simulated from the normal distribution, and the asset price is updated through

$$S_{i+1} = S_i e^{X_{i+1}}.$$

As for the Euler and VHP schemes, the path-generation mechanism is the same for up-and-in and up-and-out options (what changes is only the payoff construction and the barrier condition). Overall, the Inverse Gaussian Scheme is the most refined simulation scheme considered in this thesis.

3.4.4 Comparison of the Simulation Schemes

The three simulation schemes described above differ in the amount of information they use over each time step. The Euler scheme is the simplest approach. It updates the latent volatility factor through a local Euler discretization and then updates the log-price using the value of the latent factor at the beginning of the interval. This makes Euler easy to implement, but also the most basic of the three schemes.

The VHP scheme improves on Euler in two ways. First, the latent volatility factor is simulated from the exact transition of the Ornstein–Uhlenbeck process. Second, the integrated latent factor and the integrated variance are approximated using both endpoints of the interval. In particular, the scheme uses a trapezoidal approximation of the form

$$\int_{t_i}^{t_{i+1}} v_s ds \approx \frac{1}{2}(v_i + v_{i+1})\Delta t$$

and

$$\int_{t_i}^{t_{i+1}} V_s ds \approx \frac{1}{2}(V_i + V_{i+1})\Delta t.$$

Thus, VHP uses more information and so it improves Euler, since it incorporates both the initial and final values of the latent volatility factor over the interval.

The Inverse Gaussian Scheme goes one step further. It does not approximate the integrated latent factor by a trapezoidal rule. Instead, it simulates the pair

$$\left(v_{i+1}, \int_{t_i}^{t_{i+1}} v_s ds \right)$$

from its exact bivariate normal distribution. Then, the conditional integrated variance is simulated approximately through a moment-matched Inverse Gaussian distribution. Therefore, the Inverse Gaussian Scheme improves over Euler and VHP.

3.5 Monte Carlo Estimators with the OUSV Model

The variance reduction techniques used in this thesis were introduced in Chapter 1. The purpose of this section is not to repeat their general theoretical properties, but to describe how the corresponding estimators are implemented in the OUSV barrier option setting.

In all the numerical experiments, the asset price and the latent volatility factor are simulated on the monitoring grid

$$0 = t_0 < t_1 < \dots < t_N = T.$$

The simulated path is generated by one of the three OUSV simulation schemes described in the previous section: Euler, VHP or the Inverse Gaussian Scheme. Each barrier level is treated as a separate contract. Consequently, the simulation is run independently for every combination of option type, barrier, simulation scheme and estimator. Pathwise early stopping is used whenever the payoff or the estimator is already determined by the first barrier hitting time.

3.5.1 Plain Monte Carlo Implementation

The plain MC estimator is the direct benchmark against which the variance reduction methods are compared. In the OUSV setting, its implementation is straightforward: for each simulation, one of the three schemes described above is used to generate a path of the latent volatility factor and of the asset price along the discrete monitoring grid.

For the j -th simulated path, the scheme produces

$$S_{t_0}^{(j)}, S_{t_1}^{(j)}, \dots, S_{t_N}^{(j)}.$$

The running maximum is then computed as

$$M_T^{(j)} = \max_{0 \leq i \leq N} S_{t_i}^{(j)}.$$

For an up-and-in call option, the simulated discounted payoff realization is

$$y_j^{in} = e^{-rT} \left(S_T^{(j)} - K \right)^+ \mathbf{1}_{\{M_T^{(j)} \geq B\}}.$$

For an up-and-out call option, the simulated discounted payoff realization is

$$y_j^{out} = e^{-rT} \left(S_T^{(j)} - K \right)^+ \mathbf{1}_{\{M_T^{(j)} < B\}}.$$

The only difference between the two cases is the barrier indicator.

3.5.2 Antithetic Variables Implementation

The AV method was introduced in Chapter 1 as a general variance reduction technique. In the OUSV implementation, the main point is how the antithetic paths are generated for each simulation scheme.

For the Euler scheme, the construction is direct. Each path is generated using two independent Gaussian shocks at each time step, one for the update of the the latent volatility factor and one for the update of the log-price. If the original path is generated using

$$Z_{i+1}^{(1)} \quad \text{and} \quad Z_{i+1}^{(2)},$$

then the antithetic path is generated using

$$-Z_{i+1}^{(1)} \quad \text{and} \quad -Z_{i+1}^{(2)}.$$

The same idea is used for the VHP scheme.

For the Inverse Gaussian Scheme, the antithetic construction is slightly more involved, because the one-step simulation uses more random inputs. In particular, the simulation requires two Gaussian shocks for the bivariate normal sampling of

$$(v_{i+1}, I^v),$$

a Gaussian shock and a uniform random variable for the sampling of the moment-matched Inverse Gaussian distribution, and a Gaussian shock for the conditional simulation of the one-step log-return.

Therefore, in the Inverse Gaussian Scheme implementation, the antithetic path is generated by using the opposite Gaussian shocks and the complementary uniform random variable. In particular, if the original path uses

$$Z_1, \quad Z_2, \quad Z^{IG}, \quad U^{IG}, \quad Z^X,$$

then the antithetic path uses

$$-Z_1, \quad -Z_2, \quad -Z^{IG}, \quad 1 - U^{IG}, \quad -Z^X.$$

For each antithetic pair, the original path and the antithetic path are both checked against the barrier. This means that the two paths may lead to different barrier events.

3.5.3 Control Variates Implementation

The Control Var method is implemented by using the corresponding European vanilla call option as control variate. The target payoff is the payoff of the barrier option, while the control variate is the discounted payoff

$$X = e^{-rT}(S_T - K)^+.$$

Under the OUSV model, the expectation of the control variate is known through the Fourier pricing formula introduced earlier. Therefore,

$$\mu_X = \mathbb{E}^Q[X] = C_0^{\text{Fourier}}.$$

In the notation of Chapter 1, the control variate payoff random variable is written as

$$Y^{CV} = Y + \alpha(X - \mu_X).$$

With this convention, the estimated coefficient is

$$\hat{\alpha} = -\frac{\widehat{\text{Cov}}(Y, X)}{\widehat{\text{Var}}(X)}.$$

In the numerical implementation of this thesis, the coefficient $\hat{\alpha}$ is estimated using an independent pilot sample. More precisely, for each simulation scheme and for each barrier level, a preliminary Monte Carlo simulation with $M_{\text{pilot}} = 1000$ independent paths is used to estimate the covariance between the barrier payoff and the vanilla payoff, and the variance of the vanilla payoff. The resulting coefficient is then kept fixed in the main Monte Carlo simulation used to compute the final price estimate.

The main simulation is performed with a new independent sample of paths. For each path in the main simulation, the realized barrier payoff y_j and the realized vanilla payoff x_j are computed, and the adjusted realization is

$$y_j^{CV} = y_j + \hat{\alpha}(x_j - \mu_X), \quad j = 1, \dots, M.$$

3.5.4 Conditional Monte Carlo Implementation and In-Out Parity

The CMC method is implemented by exploiting the fact that, once the upper barrier has been reached, an up-and-in call option becomes a standard European vanilla call option with the remaining time to maturity.

For each simulated path, the asset price is monitored along the grid. If the barrier is never reached, then the up-and-in option is never activated and the CMC realization is set equal to zero. If the barrier is reached, the first hitting time on the grid is recorded:

$$\tau_B = \min\{t_i : S_{t_i} \geq B\}.$$

At this time, the relevant state variables of the OUSV model are known from the simulation. In particular, the path provides the asset price S_{τ_B} and the latent volatility factor v_{τ_B} . Conditional on this information, the remaining payoff is the payoff of a European vanilla call option with initial asset price S_{τ_B} , initial latent factor v_{τ_B} , strike K , and remaining maturity $T - \tau_B$.

Therefore, in the OUSV implementation, the CMC payoff random variable is

$$Y^{in,CMC} = e^{-r\tau_B} C^{\text{Fourier}}(S_{\tau_B}, v_{\tau_B}, K, T - \tau_B) \mathbf{1}_{\{\tau_B \leq T\}}.$$

Here,

$$C^{\text{Fourier}}(S_{\tau_B}, v_{\tau_B}, K, T - \tau_B)$$

denotes the OUSV Fourier price of the residual vanilla call option, computed at the hitting time with the remaining maturity $T - \tau_B$. The discount factor $e^{-r\tau_B}$ brings this conditional value back to time zero.

For up-and-out options, the CMC method is implemented through the in-out parity relation. Since

$$C_0^{\text{vanilla}} = C_0^{\text{in}} + C_0^{\text{out}},$$

it follows that

$$C_0^{\text{out}} = C_0^{\text{vanilla}} - C_0^{\text{in}}.$$

and so

$$\hat{C}_0^{\text{out}} = C_0^{\text{vanilla}} - \hat{C}_0^{\text{in}}.$$

Chapter 4

Numerical Results

4.1 Numerical Setting

This chapter presents the numerical results obtained for up-and-in and up-and-out call options under the OUSV model. The objective is to compare the performance of the variance reduction techniques. The simulation schemes considered are the three schemes described in Section 3.4: the Euler scheme, the scheme based on van Haastrecht, Lord and Pelsser, and the Inverse Gaussian Scheme proposed by Brignone and Sgarra. For each simulation scheme, plain MC, antithetic variables, control variates and conditional Monte Carlo are compared separately for up-and-in and up-and-out call options. The common model and simulation parameters are reported in Table 4.1. The vanilla call price computed by the OUSV Fourier formula is also reported, since it is used both as a benchmark and as the known expectation of the control variate.

The barrier levels considered in the numerical analysis are

$$B \in \{105, 110, 120, 130, 140, 150, 160, 190\}.$$

These barriers allow the analysis of different regimes.

Each barrier is priced as an independent contract. The computational time is measured end-to-end and includes the generation of the random inputs, the simulation of the paths and the estimator-specific operations. For Control Var, the time also includes the independent pilot sample. For CMC, it includes the Fourier evaluations performed at the hitting times. The early-stopping rules used for the computational time are summarized in Table 4.2.

For each Monte Carlo estimator, the estimated price, the estimated standard error, the end-to-end running time and the empirical efficiency indicator were computed.

Parameter	Value
S_0	100
K	100
r	0.05
T	1
v_0	0.25
k	8
θ	0.25
ξ	0.30
ρ	-0.60
N	504
M	10000
M_{half}	5000
M_{pilot}	1000
n_{GL}	64
C_0^{Fourier}	12.6928

Table 4.1: Model and simulation parameters used in the numerical experiments.

Estimator	Up-and-in	Up-and-out
Plain MC	Simulated up to T	Stopped at first hitting
Antithetic	Both legs simulated up to T	Each leg stopped separately at first hitting
Control Var	Simulated up to T	Simulated up to T
CMC	Stopped at first hitting	Stopped at first hitting and use of in-out parity

Table 4.2: Pathwise early-stopping rules used in the numerical experiments.

For estimator m and barrier B , the efficiency indicator is

$$\text{Efficiency}_m(B) = t_m(B) SE_m(B)^2,$$

where $t_m(B)$ is the time measured for that specific contract and implementation. A smaller value of this efficiency indicator means that the estimator achieves a better trade-off between statistical precision and computational cost. For compactness, the tables in the following sections report, for each simulation scheme and barrier level, only the estimator with the smallest efficiency value.

4.2 Up-and-In Call Options

The up-and-in payoff is

$$Y^{in} = e^{-rT} (S_T - K)^+ \mathbf{1}_{\{\tau_B \leq T\}}.$$

As the barrier increases, the knock-in event becomes less frequent. Hence, the price of the up-and-in option decreases with the barrier level. This monotonic behaviour is observed in all the simulation schemes.

4.2.1 Euler Scheme

Table 4.3 reports the most efficient estimator for each barrier level when the OUSV dynamics are simulated with the Euler scheme.

Barrier	Best estimator	Price	SE	Time	Efficiency
105	Control Var	12.683373	0.001475	0.267594	$5.820572 \cdot 10^{-7}$
110	Control Var	12.589694	0.007131	0.224080	$1.139514 \cdot 10^{-5}$
120	Control Var	11.702954	0.030653	0.231842	$2.178388 \cdot 10^{-4}$
130	Control Var	9.746138	0.060881	0.242288	$8.980359 \cdot 10^{-4}$
140	Control Var	7.274012	0.087918	0.223387	$1.726667 \cdot 10^{-3}$
150	Control Var	4.779112	0.103455	0.236985	$2.536424 \cdot 10^{-3}$
160	Control Var	2.806776	0.103876	0.229036	$2.471327 \cdot 10^{-3}$
190	Plain MC	0.354763	0.056054	0.201509	$6.331565 \cdot 10^{-4}$

Table 4.3: Most efficient estimator for up-and-in call options under the Euler scheme.

Under the Euler scheme, Control Var is the most efficient estimator from $B = 105$ to $B = 160$. The result is particularly strong for the lowest barriers, where the up-and-in payoff is very close to the vanilla payoff and the standard error is reduced substantially. As the barrier increases, the effectiveness of the control becomes weaker. At $B = 190$ plain MC becomes marginally preferable. CMC reduces the standard error with respect to plain MC for most barriers, but the repeated Fourier evaluations make it computationally expensive.

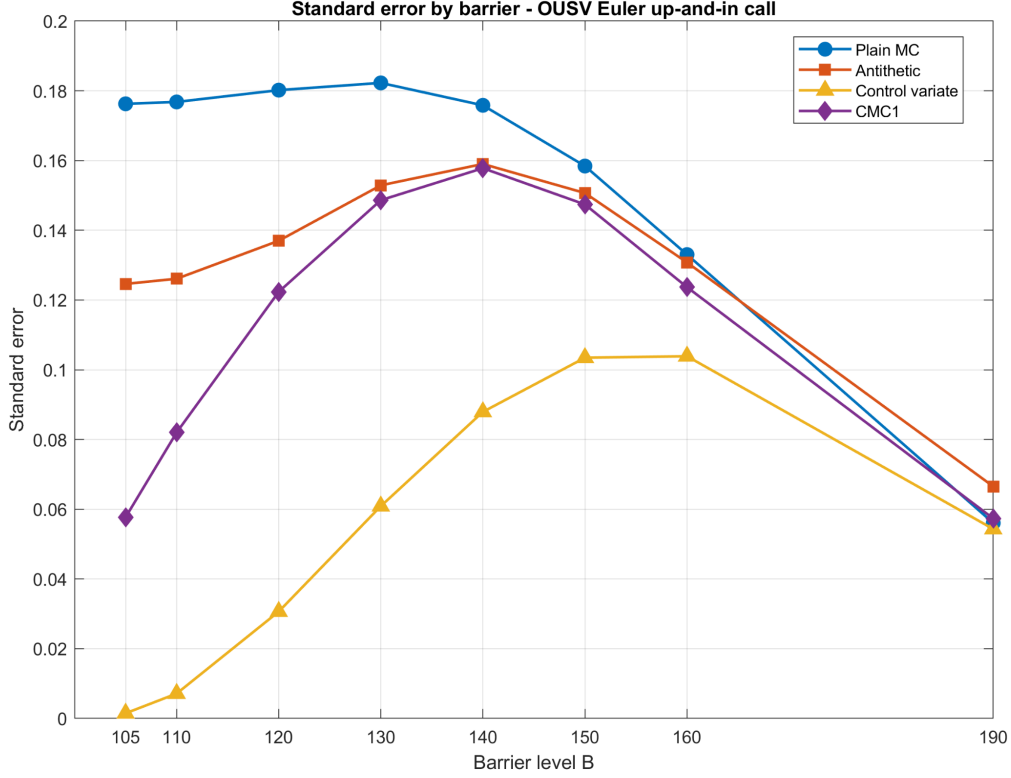


Figure 4.1: Standard error as a function of the barrier under the Euler scheme for up-and-in calls.

4.2.2 VHP Scheme

The same comparison is repeated using the scheme based on van Haastrecht, Lord and Pelsser. The most efficient estimators are reported in Table 4.4.

Barrier	Best estimator	Price	SE	Time	Efficiency
105	Control Var	12.683499	0.001494	0.318318	$7.102865 \cdot 10^{-7}$
110	Control Var	12.586012	0.007293	0.312058	$1.659991 \cdot 10^{-5}$
120	Control Var	11.722833	0.030092	0.307395	$2.783608 \cdot 10^{-4}$
130	Control Var	9.738827	0.061069	0.288195	$1.074804 \cdot 10^{-3}$
140	Control Var	7.265563	0.088021	0.257682	$1.996452 \cdot 10^{-3}$
150	Control Var	4.767367	0.103516	0.268166	$2.873543 \cdot 10^{-3}$
160	Control Var	2.846058	0.103846	0.288842	$3.114891 \cdot 10^{-3}$
190	Plain MC	0.354231	0.056009	0.233636	$7.329179 \cdot 10^{-4}$

Table 4.4: Most efficient estimator for up-and-in call options under the VHP scheme.

The VHP results are similar to those obtained with Euler. Control Var is the most efficient estimator from $B = 105$ to $B = 160$, while plain MC becomes marginally preferable at $B = 190$. AV provides a moderate reduction in standard error, but it is not selected as the most efficient method. CMC benefits from early stopping, especially when the barrier is low, but many paths then require an OUSV Fourier

evaluation at the hitting time. This additional cost prevents CMC from dominating the efficiency comparison.

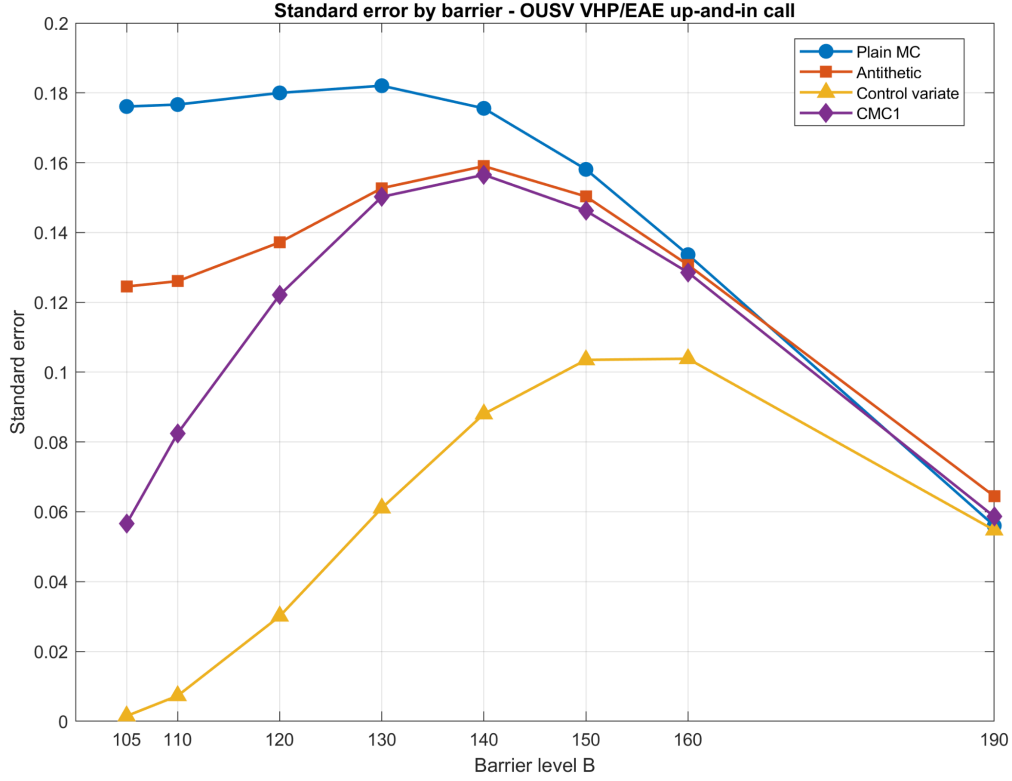


Figure 4.2: Standard error as a function of the barrier under the VHP scheme for up-and-in calls.

4.2.3 Inverse Gaussian Scheme

Finally, the Inverse Gaussian Scheme is considered. The most efficient estimators are shown in Table 4.5.

Barrier	Best estimator	Price	SE	Time	Efficiency
105	Control Var	12.685755	0.001343	1.771517	$3.194332 \cdot 10^{-6}$
110	Control Var	12.581352	0.007571	1.832058	$1.050132 \cdot 10^{-4}$
120	Control Var	11.661401	0.031487	1.796887	$1.781479 \cdot 10^{-3}$
130	Control Var	9.707184	0.061225	1.686581	$6.322111 \cdot 10^{-3}$
140	Control Var	7.243111	0.086711	1.719075	$1.292528 \cdot 10^{-2}$
150	Control Var	4.871724	0.101830	1.694773	$1.757361 \cdot 10^{-2}$
160	Control Var	2.931058	0.103006	1.760469	$1.867895 \cdot 10^{-2}$
190	Plain MC	0.320301	0.052767	1.607019	$4.474442 \cdot 10^{-3}$

Table 4.5: Most efficient estimator for up-and-in call options under the Inverse Gaussian Scheme.

Control Var is again the most efficient estimator from $B = 105$ to $B = 160$. For

$B = 190$, plain MC is marginally preferred to Control Var. CMC remains penalized by the repeated Fourier evaluations, while AV offers a cheaper but weaker variance reduction.

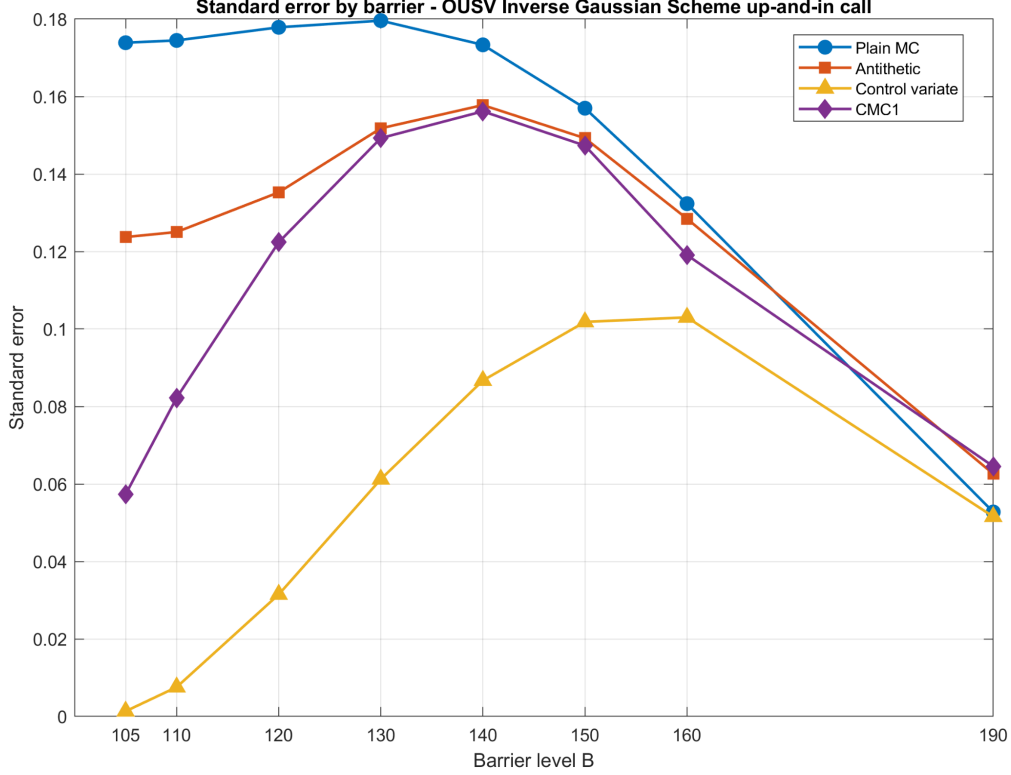


Figure 4.3: Standard error as a function of the barrier under the Inverse Gaussian Scheme for up-and-in calls.

4.2.4 Summary of the Up-and-In Results

The up-and-in results show that Control Var is the preferred estimator for low, medium and moderately high barriers under all three simulation schemes. In this region, the up-and-in payoff remains strongly related to the vanilla payoff. At the highest barrier, the efficiency values of plain MC and Control Var are very close. Plain MC is selected under all three simulation schemes.

CMC generally reduces the standard error relative to plain MC. Its pathwise early stopping is strongest for low barriers, because the barrier is reached rapidly by many paths. However, each hitting path also requires the Fourier price of the residual vanilla call. In the present implementation, this cost more than offsets the saved simulation steps. Figure 4.4 illustrates this trade-off for the lower barriers under the Euler scheme.

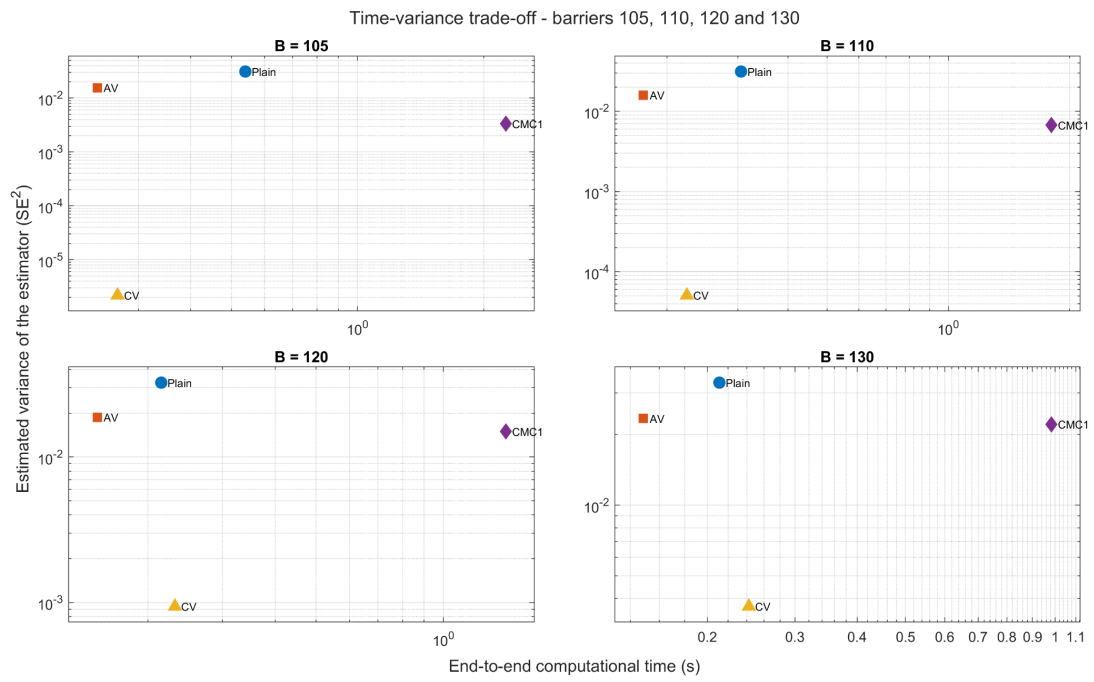


Figure 4.4: Representative time-variance trade-off for lower up-and-in barriers under the Euler scheme. Points closer to the lower-left corner provide a better combination of computational time and estimator variance.

4.3 Up-and-Out Call Options

The up-and-out payoff is

$$Y^{out} = e^{-rT}(S_T - K)^+ \mathbf{1}_{\{\tau_B > T\}}.$$

The option pays the vanilla call payoff only if the upper barrier is not reached before maturity. As the barrier increases, the knock-out event becomes less frequent and the price of the up-and-out option increases.

4.3.1 Euler Scheme

Table 4.6 reports the most efficient estimator for each barrier level under the Euler scheme.

Barrier	Best estimator	Price	SE	Time	Efficiency
105	Plain MC	0.010845	0.001646	0.186447	$5.051513 \cdot 10^{-7}$
110	Control Var	0.103110	0.007131	0.234645	$1.193238 \cdot 10^{-5}$
120	Control Var	0.989850	0.030653	0.234272	$2.201221 \cdot 10^{-4}$
130	Control Var	2.946666	0.060881	0.223406	$8.280526 \cdot 10^{-4}$
140	Control Var	5.418792	0.087918	0.228298	$1.764629 \cdot 10^{-3}$
150	Control Var	7.913692	0.103455	0.226011	$2.418968 \cdot 10^{-3}$
160	Control Var	9.886028	0.103876	0.237247	$2.559927 \cdot 10^{-3}$
190	Control Var	12.350343	0.054323	0.233334	$6.885717 \cdot 10^{-4}$

Table 4.6: Most efficient estimator for up-and-out call options under the Euler scheme.

At $B = 105$, plain MC and Control Var have almost identical efficiency values, with plain MC selected as the best estimator. From $B = 110$ to $B = 190$, Control Var provides the best time-variance trade-off. For low barriers, the standard errors of plain MC, AV and Control Var are already small because many paths are knocked out. At high barriers, the up-and-out payoff becomes close to the vanilla payoff, which makes the control variate particularly effective. CMC through in-out parity is not competitive for low barriers but becomes more effective as the barrier rises.

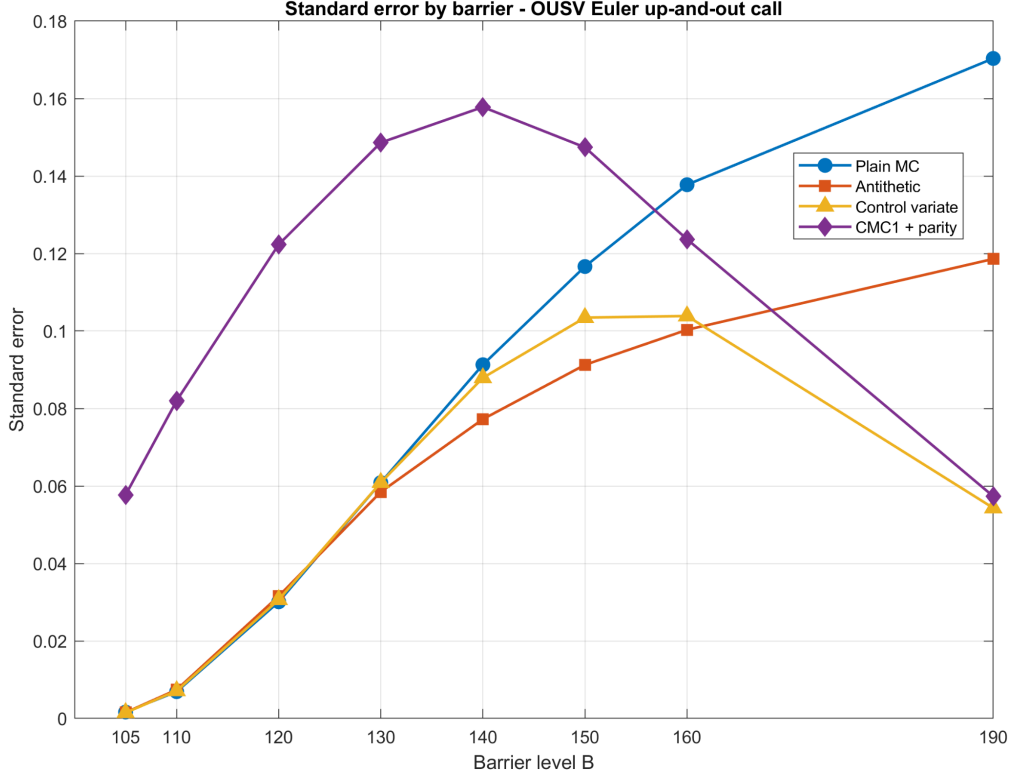


Figure 4.5: Standard error as a function of the barrier under the Euler scheme for up-and-out calls.

4.3.2 VHP Scheme

The VHP results are reported in Table 4.7.

Barrier	Best estimator	Price	SE	Time	Efficiency
105	Plain MC	0.008686	0.001532	0.154097	$3.615817 \cdot 10^{-7}$
110	Plain MC	0.115585	0.007753	0.241873	$1.453881 \cdot 10^{-5}$
120	Control Var	0.969971	0.030092	0.340058	$3.079388 \cdot 10^{-4}$
130	Control Var	2.953977	0.061069	0.251138	$9.366033 \cdot 10^{-4}$
140	Control Var	5.427241	0.088021	0.260720	$2.019984 \cdot 10^{-3}$
150	Control Var	7.925437	0.103516	0.266693	$2.857766 \cdot 10^{-3}$
160	Control Var	9.846746	0.103846	0.262420	$2.829949 \cdot 10^{-3}$
190	Control Var	12.346244	0.054755	0.255892	$7.671875 \cdot 10^{-4}$

Table 4.7: Most efficient estimator for up-and-out call options under the VHP scheme.

For $B = 105$ and $B = 110$, plain MC is selected. From $B = 120$ onward, Control Var is the most efficient estimator. The pattern is therefore similar to the Euler case: early stopping makes plain MC highly competitive when the option is frequently knocked out, while the vanilla control becomes increasingly useful as the barrier rises. CMC remains slower.

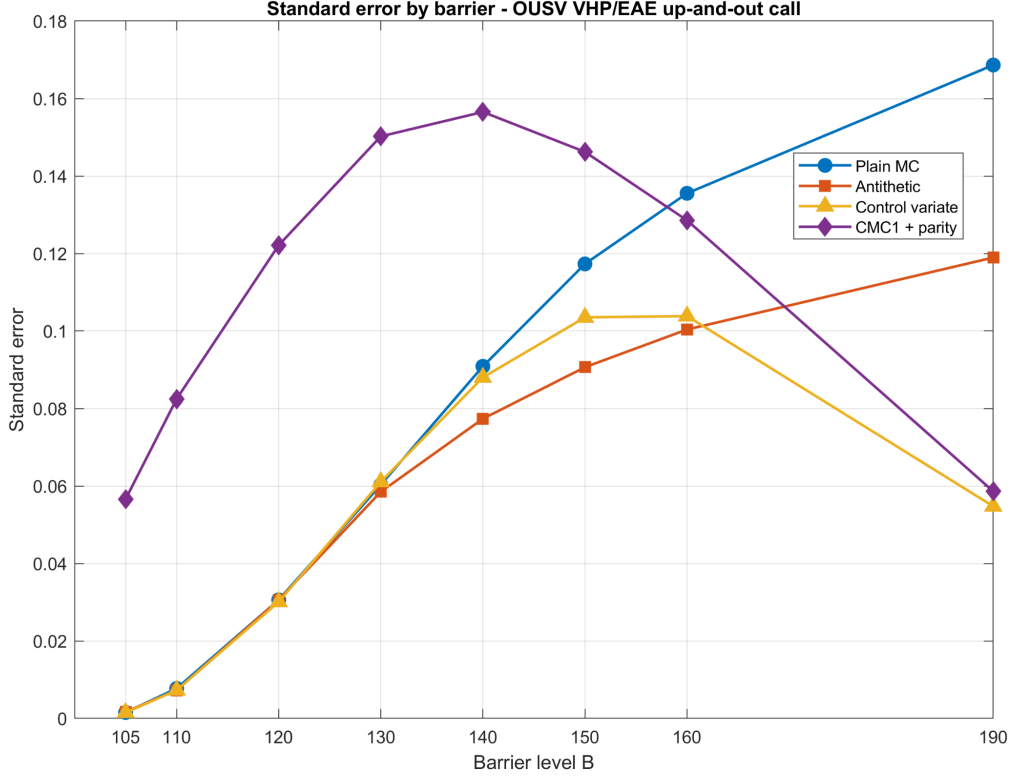


Figure 4.6: Standard error as a function of the barrier under the VHP scheme for up-and-out calls.

4.3.3 Inverse Gaussian Scheme

Finally, Table 4.8 reports the results under the Inverse Gaussian Scheme.

Barrier	Best estimator	Price	SE	Time	Efficiency
105	AV	0.010135	0.001652	0.679717	$1.854020 \cdot 10^{-6}$
110	AV	0.103942	0.007249	1.003138	$5.271279 \cdot 10^{-5}$
120	Plain MC	0.990184	0.030837	1.280653	$1.217765 \cdot 10^{-3}$
130	Plain MC	2.905065	0.061419	1.395335	$5.263612 \cdot 10^{-3}$
140	AV	5.468001	0.077246	1.833398	$1.093991 \cdot 10^{-2}$
150	AV	7.832777	0.089591	1.719583	$1.380244 \cdot 10^{-2}$
160	Control Var	9.761746	0.103006	1.761678	$1.869178 \cdot 10^{-2}$
190	Control Var	12.373345	0.051579	1.753018	$4.663748 \cdot 10^{-3}$

Table 4.8: Most efficient estimator for up-and-out call options under the Inverse Gaussian Scheme.

The classification is more heterogeneous. AV is selected at $B = 105$, $B = 110$, $B = 140$ and $B = 150$, while plain MC is selected at $B = 120$ and $B = 130$. Control Var becomes preferable at $B = 160$ and clearly dominates at $B = 190$.

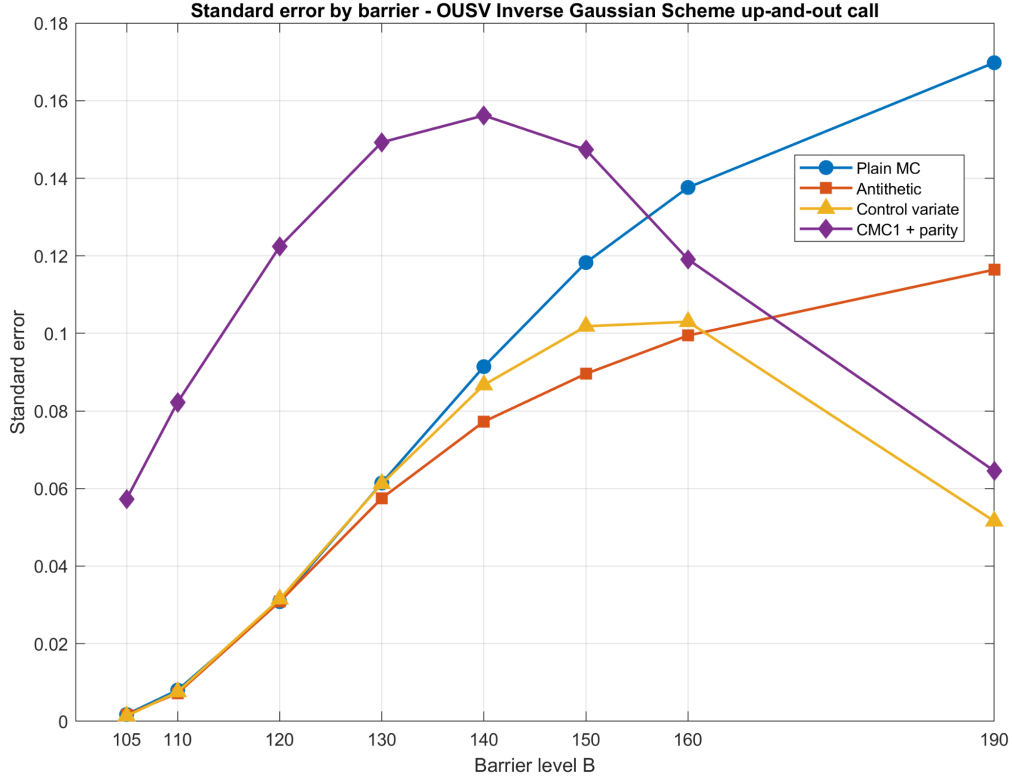


Figure 4.7: Standard error as a function of the barrier under the Inverse Gaussian Scheme for up-and-out calls.

4.3.4 Summary of the Up-and-Out Results

For low barriers, up-and-out paths are frequently knocked out and plain MC or AV can exploit early stopping. For higher barriers, the option is rarely knocked out and the payoff becomes close to the vanilla payoff. Control Var then becomes increasingly effective and is selected for $B = 190$ under all three schemes.

CMC through in-out parity is generally not competitive across the barrier levels considered.

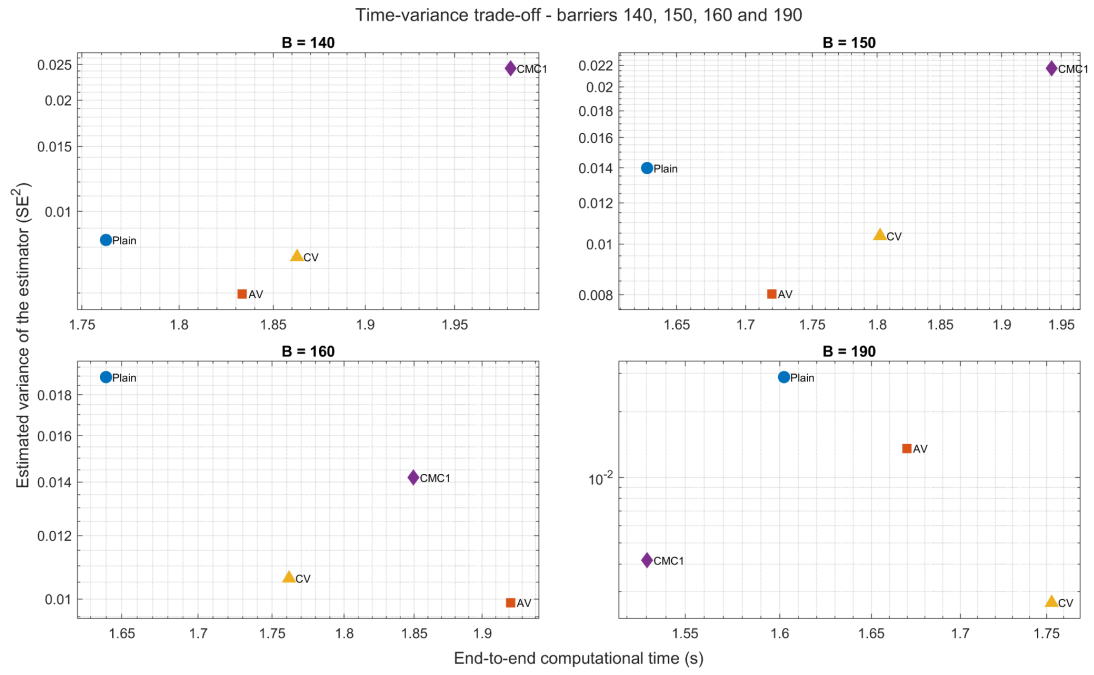


Figure 4.8: Representative time-variance trade-off for higher up-and-out barriers under the Inverse Gaussian Scheme. Points closer to the lower-left corner provide a better combination of computational time and estimator variance.

4.4 Comparison of the Simulation Schemes

The previous sections compared the Monte Carlo estimators while fixing the simulation scheme, the option type and the barrier level. The purpose of this section is different: the three OUSV simulation schemes are compared directly in terms of computational cost and numerical accuracy. To avoid introducing differences related to the barrier structure, both comparisons are performed on the same European vanilla call option.

The first comparison concerns computational cost. A European vanilla call is priced under the same model parameters used throughout the numerical analysis, with $M = 10,000$ Monte Carlo paths. Two different time grids are considered, $N = 12$ and $N = 504$. For all three schemes and both grids, every simulated path is propagated up to maturity T .

N	Simulation scheme	Running time (seconds)
12	Euler	0.013327
12	VHP	0.009489
12	Inverse Gaussian Scheme	0.047881
504	Euler	0.26005
504	VHP	0.37282
504	Inverse Gaussian Scheme	1.8193

Table 4.9: End-to-end running time for pricing the same European vanilla call with $M = 10,000$ Monte Carlo paths for two different time grids. All paths are simulated up to maturity.

Euler and VHP exhibit broadly comparable computational costs, while the Inverse Gaussian Scheme is consistently more expensive. For $N = 12$, all three schemes require only a small absolute amount of computational time, although the Inverse Gaussian Scheme is already more expensive than Euler and VHP. When the grid is refined to $N = 504$, the difference becomes more relevant in absolute terms. The relative ordering of Euler and VHP changes across the two time grids, and such small differences should not be overinterpreted, since sub-second running times can be affected by ordinary runtime fluctuations and implementation specific effects. The relevant result is that Euler and VHP remain of the same order of computational cost, whereas the more elaborate one-step construction of the Inverse Gaussian Scheme requires substantially more computation.

Computational cost alone, however, is not sufficient to compare the simulation schemes. A faster discretization may also provide a less accurate approximation of the continuous-time OUSV dynamics. For this reason, the three schemes are also validated on the same European vanilla call option using the OUSV Fourier

price as a deterministic benchmark. The validation is performed with $M = 500,000$ Monte Carlo paths for the same two time grids, $N = 12$ and $N = 504$. The Fourier benchmark is

$$C^{\text{Fourier}} = 12.692804024535.$$

For each Monte Carlo estimate, the observed pricing difference is defined as

$$\text{Observed pricing difference} = C^{\text{Fourier}} - \hat{C}^{\text{MC}},$$

while its absolute value is

$$\text{Absolute deviation} = |C^{\text{Fourier}} - \hat{C}^{\text{MC}}|.$$

N	Scheme	Fourier price	MC price	SE	95% CI	Absolute deviation
12	Euler	12.693	12.860	0.025340	[12.810, 12.910]	0.16734
12	VHP	12.693	12.840	0.024870	[12.792, 12.889]	0.14747
12	Inverse Gaussian Scheme	12.693	12.679	0.024687	[12.631, 12.728]	0.013336
504	Euler	12.693	12.728	0.024746	[12.679, 12.776]	0.035003
504	VHP	12.693	12.723	0.024734	[12.675, 12.771]	0.030209
504	Inverse Gaussian Scheme	12.693	12.695	0.024671	[12.646, 12.743]	0.001699

Table 4.10: Numerical validation of the three OUSV simulation schemes on a European vanilla call option using $M = 500,000$ Monte Carlo paths.

The results for the coarse grid $N = 12$ show clear differences among the three schemes. For Euler and VHP, the Fourier benchmark lies outside the corresponding 95% Monte Carlo confidence intervals, and the observed absolute deviations are 0.16734 and 0.14747, respectively. By contrast, the Inverse Gaussian Scheme produces an estimate much closer to the Fourier benchmark, with an absolute deviation of 0.013336, and the benchmark lies inside its confidence interval. This greater observed accuracy is obtained at a higher computational cost, but for such a coarse grid the absolute running time remains small for all three schemes.

When the number of time steps is increased to $N = 504$, the results of all three schemes become compatible with the Fourier benchmark. In particular, the Fourier price lies inside the 95% Monte Carlo confidence interval for Euler, VHP and the Inverse Gaussian Scheme. The corresponding absolute deviations decrease to 0.035003, 0.030209 and 0.001699, respectively. Thus, the validation provides numerical evidence that the discretization $N = 504$ produces Monte Carlo prices consistent with the Fourier benchmark for all three simulation schemes. The Inverse Gaussian Scheme still exhibits the smallest observed deviation, but its computational cost is substantially higher than that of Euler and VHP.

Taken together, the two comparisons illustrate the trade-off between computational cost and numerical accuracy. For a coarse time grid such as $N = 12$, the additional computational cost of the Inverse Gaussian Scheme may be justified by the much smaller observed pricing deviation, especially since the absolute running time remains limited. Conversely, when a fine grid such as $N = 504$ is used, Euler and VHP already produce estimates statistically consistent with the Fourier benchmark while requiring considerably less computational time. In this numerical setting, they therefore represent a more attractive choice when computational efficiency is a primary concern.

Chapter 5

Conclusions

This thesis studied the Monte Carlo pricing of discretely monitored barrier options under the Ornstein–Uhlenbeck stochastic volatility model, with particular attention to the trade-off between statistical precision and computational cost. The analysis combined three different OUSV simulation schemes: Euler, VHP and the Inverse Gaussian Scheme with four Monte Carlo estimators: plain Monte Carlo, antithetic variables, control variates and conditional Monte Carlo. Up-and-in and up-and-out call options were considered over a range of barrier levels, with each barrier treated as a separate contract and computational time measured end-to-end.

The numerical results show that there is no variance reduction technique that is uniformly preferable across all contracts. For up-and-in calls, the control variate method provides the best efficiency over most of the barrier range under all three simulation schemes. This result is especially strong for low barriers, where the up-and-in payoff is closely related to the payoff of the corresponding vanilla call. As the barrier increases and the knock-in event becomes less frequent, this relation weakens and the advantage of the control variate decreases. At the highest barrier, plain Monte Carlo is selected under all three simulation schemes.

The behaviour of up-and-out options is different. For low barriers, the option is frequently knocked out and plain Monte Carlo or antithetic variables can be very competitive. As the barrier increases, the up-and-out payoff becomes progressively closer to the vanilla call payoff and the control variate becomes more effective. At the highest barrier considered in the analysis, the control variate is selected under all three simulation schemes.

Conditional Monte Carlo for both up-and-in and up-and-out options generally reduces the standard error relative to plain Monte Carlo, but in the present implementation its repeated OUSV Fourier evaluations at the barrier hitting times introduce a sufficiently large computational cost to prevent it from dominating the

overall efficiency comparison.

The comparison of the simulation schemes highlights a second trade-off, this time between discretization accuracy and computational cost. On the coarse grid $N = 12$, the Inverse Gaussian Scheme produces a price much closer to the OUSV Fourier benchmark than Euler and VHP, while its absolute computational time remains limited. Its more refined construction can therefore be valuable when only a small number of time steps is used. When the grid is refined to $N = 504$, however, all three schemes produce Monte Carlo confidence intervals containing the Fourier benchmark. In this case, Euler and VHP require substantially less computational time than the Inverse Gaussian Scheme and therefore provide an attractive compromise between accuracy and cost. The Inverse Gaussian Scheme remains the one with the smallest observed pricing deviation, but this additional accuracy comes at a considerably higher computational expense.

Overall, the main conclusion of the thesis is that efficient Monte Carlo pricing cannot be separated from the characteristics of the contract and from the numerical implementation used to generate the paths. The effectiveness of a variance reduction method depends on the type of the option, the barrier level and on the simulation scheme used to generate the OUSV paths, while the scheme used to generate the paths depends on the desired balance between discretization accuracy and computational cost.

Ringraziamenti

Infine, vorrei ringraziare tutte le persone che hanno reso possibile questo momento.

Per prima cosa, ci tengo a ringraziare il mio relatore, il Professor Riccardo Brignone. Quando abbiamo iniziato a lavorare su questa tesi, avevo una conoscenza veramente scarsa dell'argomento; oggi sono arrivato al punto di aver scritto un'intera tesi di laurea magistrale su di esso. Senza il suo aiuto, con ogni probabilità non sarei riuscito a fare tutto questo. Oltre alla sua competenza, sulla quale sarebbe superfluo soffermarsi, ciò che ho apprezzato di più è stato il suo supporto umano. Sempre disponibile e affabile nei modi, mi ha semplificato moltissimo il lavoro nei momenti in cui la difficoltà degli argomenti mi disorientava. Per tutto questo gli sono estremamente grato, non aveva alcun motivo per farlo eppure lo ha fatto.

Vorrei poi ringraziare tutte le persone che mi hanno accompagnato durante questo percorso.

Ringrazio mia madre Sharon e mio padre Emanuele, sempre presenti, soprattutto nei momenti difficili, quando esserci è sicuramente meno semplice e ancora più importante. Ringrazio poi mia sorella Laurely e mio zio Daniele, senza i quali ormai non riuscirei a immaginare la mia vita e che sono parte delle mie decisioni, anche delle più piccole prese nel quotidiano. Ringrazio mia nonna Giuliana e mio nonno Giuseppe che non mi hanno mai fatto mancare il loro grande affetto. Ultimo, ma certamente non per importanza tra i membri della mia famiglia, ringrazio il mio cucciolo Billy, che da quando avevo undici anni riempie le mie giornate di amore.

Ringrazio poi i miei amici di una vita, Cristian, Esteban, Lorenzo, Nicolò, Simone e Tommaso, con i quali da sempre condivido successi e insuccessi.

Un ringraziamento speciale va anche a tutte le persone con cui ho condiviso la quotidianità di questi ultimi due anni di laurea magistrale e che hanno contribuito a renderli così belli.

Grazie a Dario, che considero la mia anima gemella e un fratello: una delle persone più intelligenti che conosca e, allo stesso tempo, capace di essere incredibilmente stupido. Con lui ho condiviso praticamente ogni momento di questa magistrale, e difficilmente avrei potuto desiderare un compagno di viaggio migliore.

Grazie a Yuri e Kama, che sono stati una costante fonte di spensieratezza proprio nei momenti in cui i pensieri più pesanti iniziavano a farsi sentire. Grazie anche a Ciccio, Omar e Rebecca per tutte le risate, le giornate insieme e i bei momenti condivisi. Infine, un grande grazie a Cristina per i ancor pochi, ma molto intensi momenti trascorsi insieme.

Un grazie enorme va poi ai miei amici dell'Erasmus a Tübingen, che continuo tuttora a sentire spesso: Biagio, Davide, Giorgia, Sebastian e Samuele. Avete reso quell'esperienza una bellissima parte del mio percorso (salvandomi dall'esaurimento nervoso). E, anche a distanza di tre anni, un pensiero speciale va ai miei amici di Cádiz, che continuano ad avere un posto nel mio cuore.

Grazie davvero a tutti voi. Se sono arrivato a questo punto, lo devo soltanto a voi.

Bibliography

- Phelim Boyle, Mark Broadie, and Paul Glasserman. Monte Carlo methods for security pricing. *Journal of Economic Dynamics and Control*, 21(8–9):1267–1321, 1997.
- Riccardo Brignone and Carlo Sgarra. Full calibration, fast simulation, and exotic options pricing under the Ornstein–Uhlenbeck driven stochastic volatility model. *European Journal of Operational Research*, 2026. doi: 10.1016/j.ejor.2026.08.010.
- Paul Glasserman. *Monte Carlo Methods in Financial Engineering*. Springer, 2003.
- Martin Haugh. The Black–Scholes model. IEOR E4706: Foundations of Financial Engineering, Columbia University, 2016. URL <https://www.columbia.edu/~mh2078/FoundationsFE/BlackScholes.pdf>.
- MIT OpenCourseWare. Risk neutral valuation, black–scholes equation. Lecture notes for 18.642 Topics in Mathematics with Applications in Finance, 2024. URL https://ocw.mit.edu/courses/18-642-topics-in-mathematics-with-applications-in-finance-fall-2024/mit18_642_f24_lec21.pdf. Fall 2024.
- Alexander van Haastrecht, Roger Lord, and Antoon Pelsser. Monte Carlo pricing in the Schöbel–Zhu model and its extensions. *Journal of Computational Finance*, 17: 57–86, 2014.