ESCP Business School

ENSAE Paris

Pricing Asian Options Using Monte Carlo and Quasi-Monte Carlo Methods: An Application to Brent Crude

Author: Simon Duong

February 20, 2025

Abstract

The valuation of path-dependent options, such as Asian options, are crucial tasks in financial markets, especially for managing the risks associated with derivative products. In this document, we will first consider an overview of Asian options, discussing their unique characteristics, types, and practical applications. This background sets the stage for understanding the specific challenges and strategies involved in managing these instruments.

The project is then structured in two main phases. The first phase focuses on the pricing theory for Asian options using the Black-Scholes-Merton model. In the second phase, we price Asian options using Monte Carlo (MC) and random quasi-Monte Carlo (RQMC) methods. Different variance reduction techniques are applied within these methods to enhance pricing accuracy and efficiency, making it possible to obtain more precise estimates of the option's expected payoff. By evaluating and comparing these MC and RQMC approaches, we aim to identify the most effective techniques for pricing Asian options.

This project provides a valuable opportunity to apply theoretical knowledge to real-world financial applications, bridging the gap between academic concepts and practical finance. By exploring both pricing methods for an Asian option, this study not only contributes to deepen my understanding of market finance but also offers me insights into the challenges and methodologies involved in managing path-dependent options.

Contents

List of Figures

List of Tables

1

Asian Options Overview

Asian options are exotic options whose payoff depends on the average price of the underlying asset. In that sense, they are classified as path-dependent options. The average price is calculated using prices of the underlying asset recorded on specific and periodic observation dates defined in the contract. This average price is therefore only computable at maturity.

Asian options can be either call or put options, but for the sake of this project, we will focus only on Asian call options. Therefore, in the rest of this report it is implied that Asian options are of the call type.

The payoff of the Asian option is the maximum between the average price computed at maturity minus the strike price, and zero. It can be written as follows:

$$\left(\frac{1}{T} \int_0^T S_t \, dt - K \right)^+$$

where T is the maturity date (in years), S_t is the price of the underlying at date t, K is the strike price, and $(x)^+ = \max(x, 0)$.

There are also Asian options where the average is used as the strike price. Their payoff is:

$$\left(S_T - \frac{1}{T} \int_0^T S_t \, dt \right)^+$$

However, this type will not be covered in this document.

Asian options are flexible derivatives as the agent can choose the following characteristics:

- **Type of average:** usually, Asian option payoffs are computed with an arithmetic average, but it is also possible to use a geometric average, and even to apply different weights to the observation dates. This project focuses on arithmetic average options.
- **Observation periods:** the mean can be calculated over the entire life of the option or over a narrower period.
- **Observation frequency:** it is defined at the beginning of the contract and can be daily, weekly, monthly, annually, quarterly, etc.

Let's take the example of a 1-year arithmetic average Asian call option with quarterly observations over the entire life of the derivative and a strike price of 2500 €. Suppose the prices gathered are 2250, 3050, 3700, 3200, and 3000. At maturity, the average price is 3040 €. The payoff is therefore $3040 - 2500 = 540$ €.

Some alternative forms of Asian options exist. We will mention them here, but not in detail.

Floored Asian options The underlying asset's prices are considered in the computation of the average only if they are higher than the floor price. Otherwise, they are replaced by a reference value. If we take the same example as before with a floor price equal to the strike price, and with the floor price as the reference price, we have the following average:

$$\frac{1}{5}(2500 + 3050 + 3700 + 3200 + 3000) = 3090 \text{ €}$$

The payoff is therefore $3090 - 2500 = 590$ €. Here, 2250 has been replaced by 2500 because it is lower than the floor price. This type of Asian option guarantees a better or equal payoff, so they are more expensive.

It is important to note that this is different from Asian options where the final average itself is floored. In that case, with a floor at 3100, the payoff would have been

$$\max\left(\frac{1}{T}\int_0^T S_t\,dt - K, F - K\right) = \max(3040 - 2500, 3100 - 2500) = 600 \text{ €}$$

Where F denotes the floor price.

Super-Asian options Any observation price lower than the strike price is not included in the computation of the average. In the previous example, the average at maturity is:

$$\frac{1}{4}(3050 + 3700 + 3200 + 3000) = 3237.5 \text{ €}$$

Here, 2250 is not integrated as it is lower than 2500. Once again, this variant offers a higher payoff than standard Asian and floored Asian options, but it is also more expensive.

Pyramid Asian options In pyramid Asian options, each observed price is assigned a different weight, with more recent prices generally receiving higher weights. The weights can increase exponentially, linearly, and so forth, making the option more sensitive to recent variations in the underlying price. The payoff is

$$\left(\frac{\int_0^T w_t S_t \, dt}{\int_0^T w_t \, dt} - K\right)^+$$

Where w_t is the weight assigned to $S_t, \forall t \in [0, T]$

For a linearly weighted average in the same example, the average is:

$$\frac{1 \times 2250 + 2 \times 3050 + 3 \times 3700 + 4 \times 3200 + 5 \times 3000}{1 + 2 + 3 + 4 + 5} = 3150 \text{ €}$$

And the payoff is 650 €. Pyramid Asian options are useful for investors who believe that recent movements are more indicative of future trends.

These three variants leverage the flexibility of average computation. The next and last one uses the flexibility of the observation dates.

Maxi-Asian (or lookback Asian) options To compute the average, the highest observed prices between two observation dates are retained. The initial observation can be included, but is usually omitted if it is not the maximum for the first observation period. Consider the same example, but with the collected prices (i.e., the maxima over the periods) as follows: 3200, 3700, 3700, and 3500. The average is:

- If the initial price is included:

$$\frac{1}{5}(2250 + 3200 + 3700 + 3700 + 3500) = 3270 \text{ €}$$

- If the initial price is not included:

$$\frac{1}{4}(3200 + 3700 + 3700 + 3500) = 3525 \text{ €}$$

The payoffs are therefore 770 € and 1025 €, respectively. The price of this option is higher than that of a standard Asian option because it offers a higher potential payoff.

Type	**Payoff**	**Relative value**
Standard Asian	$\left(\frac{1}{T}\int_0^T S_t\,dt - K\right)^+$	Cheaper than vanilla call options due to averaging effect.
Floored Asian	$(\overline{S_T} - K)^+$ where $\overline{S_T}$ is the average correcting prices under the floor.	More expensive than standard Asian options, as lower prices are replaced by a reference value.
Super-Asian	$(\overline{S_T} - K)^+$ where $\overline{S_T}$ is the average excluding prices under the strike.	More expensive than standard and floored-Asian options, as prices below the strike are excluded from the average.
Pyramid Asian	$\left(\frac{\int_0^T w_t S_t\,dt}{\int_0^T w_t\,dt} - K\right)^+$	More sensitive to recent price changes, typically more expensive than standard Asian options.
Maxi-Asian (or Lookback Asian)	$\left(\overline{S_T} - K\right)^+$ where $\overline{S_T}$ is the average on the maximum observed prices over each observation period.	More expensive than standard Asian options, as it uses the maximum price over each observation period.

Table 1.1: Types of Asian options

Now that we have an overview of the different types of Asian options and some of their characteristics, we will focus our study on standard Asian call options.

Advantages	**Disadvantages**
Smooths out extreme price fluctuations	Limits potential gains in case of a price spike
Cheaper than vanilla options	Less suitable for short-term speculative strategies
Useful for hedging and long-term strategies	Less liquid than vanilla options, often traded OTC

Table 1.2: Advantages and Disadvantages of Asian options

2

Pricing Theory

Let's consider a financial market composed of two assets, and a time interval $[0, T]$, $T > 0$, where T is the maturity date of the option. The first asset is a risk-free asset (bond) denoted by B, with price $B_t = e^{rt}$ for $t \in [0, T]$, where r is a positive constant representing the risk-free interest rate. The second asset, denoted by S, is a risky asset (underlying asset of the options) with price S_t at time $t \in [0, T]$. We assume that the stochastic process $(S_t)_{t\in[0,T]}$ is defined on a probability space $(\Omega, \mathcal{A}, \mathbb{P})$, equipped with a standard Brownian motion $(W_t)_{t\in[0,T]}$. We denote $(\mathcal{F}_t)_{t\in[0,T]}$ as the natural filtration of $(W_t)_{t\in[0,T]}$. We assume that the initial price $S_0 > 0$ is constant and that the evolution of the process $(S_t)_{t\in[0,T]}$ is governed by the Black-Scholes model:

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

with $\mu \in \mathbb{R}$ and $\sigma > 0$. Therefore, r, μ and σ are deterministic in this model.

Intuitively, this equation characterizes the evolution of S_t because the spot's return, dS_t/S_t, is approximately equal to μdt, where μ is the asset's drift, and a (small) random disturbance σdW_t. The range of this noise is measured by the volatility σ.

In this section, we study the price of an Asian call option whose payoff at maturity T is given by

$$h = \left(\frac{1}{T} \int_0^T S_t \, dt - K \right)^+$$

From the previous SDE, we see that S is an unidimensional Ito's process, so we can compute an explicit expression for the spot price using Ito's lemma with $f(x) = \ln(x)$, which is of class C^2 on $\mathbb{R}_+^*$.

$$
\begin{aligned}
d\ln(S_t) &= \frac{1}{S_t}dS_t - \frac{1}{2S_t}\sigma^2 S_t^2 dt \\
&= \mu dt + \sigma dW_t - \frac{\sigma^2}{2}dt \\
&= \left(\mu - \frac{\sigma^2}{2}\right) dt + \sigma dW_t
\end{aligned}
$$

Therefore,

$$
\ln(S_t) - \ln(S_0) = \left(\mu - \frac{\sigma^2}{2}\right) t + \sigma W_t
$$

And,

$$
S_t = S_0 e^{\left(\mu - \frac{\sigma^2}{2}\right)t + \sigma W_t}
$$

To determine the theoretical price of the Asian option, we need to evaluate the expected payoff of a replication portfolio under the risk-neutral measure. According to the first fundamental theorem of asset pricing, in the absence of arbitrage, there exists an equivalent risk-neutral measure $\hat{\mathbb{P}}$ under which all discounted asset prices are martingales. This allows us to calculate option prices by discounting the expected payoff under $\hat{\mathbb{P}}$, ensuring consistency with the no-arbitrage condition. Furthermore, if we assume the market is complete, the second fundamental theorem of asset pricing guarantees the uniqueness of this risk-neutral measure $\hat{\mathbb{P}}$. This uniqueness ensures that there is a single, arbitrage-free price for the Asian option, which can be determined through replication.

Let's assume that this market is complete, so that we only have to find one risk-neutral measure to be certain that it is the only existing one.

Let's define $\lambda = \frac{\mu - r}{\sigma}$. W_t is a Brownian motion so $Z_t = e^{-\lambda W_t - \frac{\lambda^2}{2}t}$ is a martingale. This implies that $\mathbb{E}[Z_t] = 1$. Therefore, we define

$$
\frac{d\hat{\mathbb{P}}}{d\mathbb{P}} = Z_t
$$

And Girsanov's theorem states that $\hat{W}_t$ is a Brownian motion on $(\Omega, \mathcal{A}, \hat{\mathbb{P}})$, where

$$
\begin{aligned}
\hat{W}_t &= W_t + \int_0^t \lambda \, ds \\
&= W_t + \lambda t
\end{aligned}
$$

The dynamics of S_t under $\hat{\mathbb{P}}$ become:

$$dS_t = S_t(r\,dt + \sigma\,d\hat{W}_t),$$

This means that under $\hat{\mathbb{P}}$, the drift of S_t is the risk-free rate r, which is consistent with the pricing of derivative products in a risk-neutral world.

Let's now define an auto-financing replicating portfolio with strategy (α, β) whose value process is denoted by $(V_t)_{t\in[0,T]}$. This means that

$$V_t = \alpha_t S_t + \beta_t B_t$$

The present value of this portfolio is defined by $\tilde{V}_t = e^{-rt}V_t$ and its dynamics can be computed using Ito's formula with $f(x,t) = e^{-rt}x$

$$\begin{aligned} d\tilde{V}_t &= -re^{-rt}V_t dt + e^{-rt}dV_t \\ &= \alpha_t(-r\tilde{S}_t dt + e^{-rt}dS_t) \end{aligned}$$

where $\tilde{S}_t = e^{-rt}S_t$. Yet, $d\tilde{S}_t = -r\tilde{S}_t dt + e^{-rt}dS_t$ (Ito's formula). Therefore,

$$d\tilde{V}_t = \alpha_t d\tilde{S}_t$$

and $\tilde{V}$ is a local martingale because it is a stochastic integral.

V is a replication portfolio of an Asian option, so $V_T = h$. $\tilde{V}_T$ is square-integrable under $\hat{\mathbb{P}}$ because $\mathbb{E}[\tilde{V}_T^2] = e^{-2rT}h^2$ and

$$\begin{aligned} \mathbb{E}[h^2] &\leq \frac{1}{T^2}\int_0^T \hat{\mathbb{E}}\left[e^{2(r-\sigma^2/2)t+2\sigma W_t}\right] dt \\ &\leq S_0^2 e^{(2r+\sigma^2)T} \\ &< +\infty. \end{aligned}$$

From Doob's inequality,

$$\mathbb{E}\left[\sup_{t\in[0,T]} \tilde{V}_t^2\right] < +\infty.$$

and $\tilde{V}$ is a martingale under $\hat{\mathbb{P}}$. Therefore,

$$\begin{aligned} &\tilde{V}_t = \hat{\mathbb{E}}[\tilde{V}_T \mid \mathcal{F}_t] \\ \iff &e^{-rt}V_t = \hat{\mathbb{E}}[e^{-rT}V_T \mid \mathcal{F}_t] \\ \iff &V_t = e^{-r(T-t)}\hat{\mathbb{E}}[V_T \mid \mathcal{F}_t] \\ \iff &V_t = e^{-r(T-t)}\hat{\mathbb{E}}\left[\left(\frac{1}{T}\int_0^T S_u\,du - K\right)^+ \middle| \mathcal{F}_t\right] \end{aligned}$$

Finally, if we compute the value of this replication portfolio at time $t = 0$, we have a mathematical formula for the price of the Asian call option.

$$V_0 = e^{-rT}\hat{\mathbb{E}}\left[\left(\frac{1}{T}\int_0^T S_u\,du - K\right)^+ \middle| \mathcal{F}_0\right]$$

This formula cannot be simplified, and this is not enough to compute the price of the option. In the next chapter, we use MC and QMC algorithms to approximate this value.

3

Application to Brent Crude: Pricing Asian Options

3.1 Brent Crude Overview

Brent Crude is one of the most widely recognized benchmarks for oil prices globally. It serves as a pricing reference for approximately two-thirds of the world's internationally traded crude oil. Originating from the North Sea, Brent Crude is light and sweet, meaning it has low sulfur content and density, making it ideal for refining into high-demand products such as gasoline and diesel. Its global importance as a benchmark stems from its liquidity, transparency, and relevance to energy markets across Europe, Asia, and the Americas. The choice of Brent Crude for this project is justified by several characteristics.

Economic Relevance: Brent Crude plays a pivotal role in global energy markets, influencing prices across multiple sectors such as transportation, manufacturing, and power generation. Given its significance, analyzing financial instruments tied to Brent provides valuable insights into the pricing and hedging of commodities, particularly in volatile and dynamic markets.

Volatility Characteristics: Oil prices, including Brent Crude, are often characterized by high volatility due to geopolitical risks, supply-demand imbalances, and macroeconomic factors. This inherent volatility makes Brent Crude an excellent candidate for studying the pricing and hedging of options, as such conditions test the robustness of Monte Carlo and randomised quasi-Monte Carlo simulation methods.

Practical Application: Asian options, which are path-dependent derivatives where the payoff is based on the average price of the underlying asset, are particularly relevant in the context of commodities like Brent Crude. By averaging the price over time, Asian options mitigate the effects of extreme price fluctuations, making them appealing to energy producers, consumers, and investors seeking to hedge against price volatility. They are commonly used in the oil industry to reduce exposure to short-term price spikes or crashes.

From Yahoo Finance, we import Brent Crude (ticker BZ=F) daily data from 01/01/2018 to 28/10/2024. This index is based on Brent Crude's future prices. After a little bit of data cleaning, we obtain the following summary:

	Adj Close	Close	High	Low	Open	Volume
count	1738.000000	1738.000000	1738.000000	1738.000000	1738.000000	1738.000000
mean	72.955138	72.955138	74.082445	71.763360	72.936881	30915.693901
std	17.913406	17.913406	18.199503	17.585558	17.883166	12270.559135
min	19.330000	19.330000	21.270000	16.000000	19.559999	0.000000
25%	63.389999	63.389999	64.357498	62.385001	63.290001	23159.750000
50%	74.255001	74.255001	75.155003	73.160004	74.300003	30319.000000
75%	83.267498	83.267498	84.277500	82.255001	83.285000	37176.500000
max	127.980003	127.980003	137.000000	122.500000	129.570007	90111.000000

Table 3.1: Summary of Brent Crude data

We can clearly observe a strong volatility on this index, particularly on the Covid-19's period. Otherwise, the prices are mainly between \$60 and \$100. In this project, we will price an Asian Call Option on Brent Crude with start date on 02/10/2023, maturity one year, strike price \$80 and weekly observations.

3.2 Estimation of Parameters

In order to price this option on Brent Crude, we use Black and Scholes' model to simulate the asset path. Therefore, the volatility σ and the risk-free interest rate r are constant and deterministic and we need to compute them. This hypothesis is not very realistic but it is a reasonable approximation for reflection. There is two methods to obtain σ.

Historical Volatility: Past prices are used to calculate the standard deviation over a defined period.

Implied Volatility: The volatility parameter is calculated by collecting prices of vanilla options on Brent Crude and inverting Black and Scholes' formula.

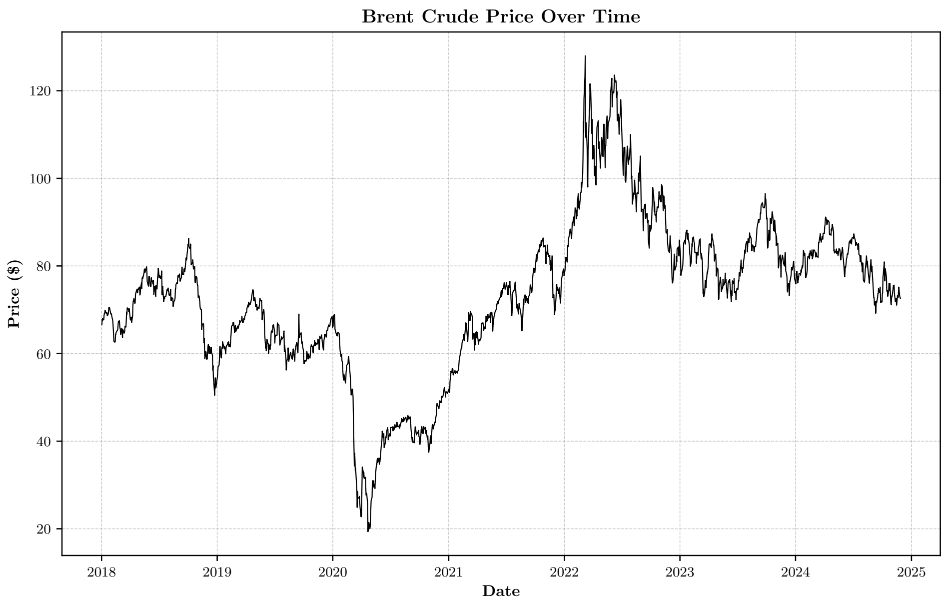

Figure 3.1: Brent Crude price over time

Implied volatility appears to be more accurate as it based on the anticipation rather than the past movements. However, it is harder to collect data on option price. In our specific case, option prices on Brent Crude are not available on Yahoo Finance, so for convenience reason we will use the historical volatility for this project. This will not prevent us from comparing MC and RQMC methods.

We compute the standard deviation of log-returns $r_t = \ln(S_t/S_{t-1})$ based on the Brent Price between 02/10/2022 and the last date available when we are on the 02/10/2022 (i.e. the 01/10/2022 or before if the markets were closed on that day). We have now the annual volatility and we derive the daily volatility by multiplying it with $\sqrt{252}$, because there is approximately 252 trading days in a year. By implementing it with Python, we have $\sigma \approx 30.93\%$.

Finally, we collect the one-year risk-free interest rate from the website of the U.S. Department Of The Treasury[1], at the last date available when we are on the 02/10/2022. We have $r = 5.46\%$

[1] https://home.treasury.gov/resource-center/data-chart-center/interest-rates/TextView?type=daily_treasury_yield_curve&field_tdr_date_value=2024

3.3 Basic Monte Carlo Method

In the last chapter, we express the price of the option as

$$V_0 = e^{-rT}\hat{\mathbb{E}}\left[\left(\frac{1}{T}\int_0^T S_u\,du - K\right)^+ \middle| \mathcal{F}_0\right]$$

To compute it with Python, we will discretize this expression in the following way

$$AC = e^{-rT}\hat{\mathbb{E}}\left[\left(\frac{1}{M}\sum_{i=1}^{M} S(t_i, Z_i) - K\right)^+\right]$$

where $M \in \mathbb{N}^*$ is the number of time-steps, $t_0 = 0 < t_1 < ... < t_M = T$, and AC stands for "Asian Call". And

$$S(t_i, Z_i) = S(t_{i-1}, Z_{i-1})\exp\left(\left(r - \frac{\sigma^2}{2}\right)dt + \sigma\sqrt{dt}Z_i\right), \qquad 1 \leq i \leq M$$

where $(Z_i)_{1\leq i\leq M}$ have the same distribution than $Z \sim \mathcal{N}(0,1)$ and $dt = T/M$.

Finally, we approximate this expectancy with

$$\hat{AC}_I = \frac{e^{-rT}}{I}\sum_{k=1}^{I}\phi(S_k, Z_k)$$

where I is the number of MC simulations, $\phi(S_k, Z_k) = \left(\frac{1}{M}\sum_{i=1}^{M} S_k(t_i, Z_{k,i}) - K\right)^+$ and S_k is the k-th simulation of the underlying's price and Z_k is a vector of standard normal distributions.

This estimator is unbiased because $(S_k)_{1\leq k\leq I}$ are i.i.d. so $\hat{\mathbb{E}}[\hat{AC}_n] = AC$, and the law of large numbers ensures that it is consistent:

$$\hat{AC}_I \xrightarrow[I\to\infty]{\text{p.s.}} AC.$$

Its variance is $\mathrm{Var}(\hat{AC}_I) = \frac{e^{-2rT}}{I}\mathrm{Var}(\phi(S,Z))$, so its convergence speed is $\mathcal{O}\left(\frac{1}{\sqrt{I}}\right)$.

```python
def MC_Asian_price(S0, K, T, r, sigma, I, M, dt):
    S = np.zeros((M+1,I))
    S[0] = S0
    for t in range(1,M+1):
        S[t] = S[t-1] * np.exp((r - sigma**2 / 2) * dt + sigma *
            np.sqrt(dt) * np.random.standard_normal(I)) #Simulation
            of the underlying's price using B&S' model

```

```
    hT = np.maximum(np.mean(S[1:], axis=0) - K, 0) #Payoffs of the I simulations

    price_estimate = np.exp(-r * T) * np.mean(hT) #Estimated price
    price_variance = np.exp(-2 * r * T) * np.var(hT) / I
    price_std = np.sqrt(price_variance)
    return price_estimate, price_variance, price_std
```

Code 3.1: Basic Monte Carlo Simulation Function

This function estimate the price of the option, but also its variance and standard deviation, which will be useful for further analysis of variance reduction. In our situation, $S_0 \approx \$95.31$, $K = 80$, $T = 1$, $r \approx 5.46\%$, $\sigma \approx 30.93\%$, and $M = 52$.

I	Estimated Price	Standard Deviation
100	18.4432	1.5130
1000	18.0439	0.5044
5000	17.9491	0.2231
10000	18.1532	0.1570
50000	18.0116	0.0707
100000	17.9691	0.0501
250000	18.0273	0.0317
750000	18.0212	0.0183

Table 3.2: Basic Monte Carlo Estimated Prices

Therefore, the price of the option is around \$18. As we would have expected, the standard deviation of the estimated price decreases when we increase the number of simulation. However, we have to run a minimum of 250,000 simulations to obtain a standard deviation of 0.03. We can reasonably wonder if it is possible to have this accuracy with less simulations. To this aim, variance reduction methods are useful tools.

3.4 Variance Reduction Methods

In this section, we will explore two variance reduction methods: antithetic variables and control variates.

3.4.1 Antithetic Variables

We know that Z and $-Z$ follow the same distribution. The idea of the antithetic variable method is to introduce a new estimator using this property. Z and $-Z$ will therefore compensate each other, reducing the estimator's variance. Intuitively, this estimator will be more stable because the trajectories of these two random variables are built to compensate each other.

$$\hat{AC}_I^{\text{anti}} = \frac{e^{-rT}}{2I} \sum_{k=1}^{I} (\phi(S_k, Z_k) + \phi(S_k, -Z_k))$$

One lemma ensure that $Var(\hat{AC}_I^{\text{anti}}) \leq Var(\hat{AC}_I)$. The cost of this variance reduction is that we have twice as many trajectories to compute, while keeping the same number of random generations of Z.

```python
def MC_anti_Asian_price(S0, K, T, r, sigma, I, M, dt):
    S = np.zeros((M+1,I))
    P = np.zeros((M+1,I))
    S[0] = S0
    P[0] = S0

    for t in range(1,M+1):
        Z = np.random.standard_normal(I)
        S[t] = S[t-1] * np.exp((r - sigma**2 / 2) * dt + sigma *
            np.sqrt(dt) * Z)
        P[t] = P[t-1] * np.exp((r - sigma**2 / 2) * dt + sigma *
            np.sqrt(dt) * (-Z))

    hT_S = np.maximum(np.mean(S[1:], axis=0) - K, 0) #Payoffs of
        the I simulations using Z
    hT_P = np.maximum(np.mean(P[1:], axis=0) - K, 0) #Payoffs of
        the I simulations using -Z
    hT_anti = (hT_S + hT_P) / 2

    price_estimate = np.exp(-r * T) * np.sum(hT_anti) / I
        #Estimated price
    price_variance = np.exp(-2 * r * T) * np.var(hT_anti) / I
    price_std = np.sqrt(price_variance)
    return price_estimate, price_variance, price_std
```

Code 3.2: MC Antithetic Variables Simulation Function

I	Estimated Price	Standard Deviation
100	18.0677	0.4218
1000	18.1557	0.1512
5000	17.9371	0.0637
10000	18.0133	0.0460
50000	18.0067	0.0203
100000	18.0100	0.0144
250000	17.9904	0.0091
750000	18.0019	0.0053

Table 3.3: MC Antithetic Variables Estimated Prices

The price of the option is similar than before, around \$18. However, we notice a significant change in standard deviation. Antithetic variables allow us to reach a standard deviation below 0.03 with only 50,000 simulations, compared to 250,000 with basic MC. This result is satisfactory but the issue of computing time remains. We will address it after implementing the control variate method.

3.4.2 Control Variate

The control variate method introduce a random variable Y, such that $\mathbb{E}[Y] = 0$, and the estimator

$$\hat{AC}_I^{\text{cv}} = \frac{e^{-rT}}{I} \sum_{k=1}^{I} \left(\phi(S_k, Z_k) + \beta Y_k\right)$$

with $\beta \in \mathbb{R}$. This estimator is unbiased for any value of β, but we would like to find the β that minimizes its variance.

The variance of $\hat{AC}_I^{\text{cv}}$ is given by:

$$\text{Var}(\hat{AC}_I^{\text{cv}}) = e^{-2rT} \frac{1}{I} \text{Var}\left(\phi(S, Z) + \beta Y\right).$$

Yet,

$$\text{Var}\left(\phi(S, Z) + \beta Y\right) = \text{Var}(\phi(S, Z)) + \beta^2 \text{Var}(Y) + 2\beta \text{Cov}(\phi(S, Z), Y).$$

Therefore,

$$\text{Var}(\hat{AC}_I^{\text{cv}}) = e^{-2rT} \frac{1}{I} \left(\text{Var}(\phi(S, Z)) + \beta^2 \text{Var}(Y) + 2\beta \text{Cov}(\phi(S, Z), Y)\right).$$

To minimize the variance, we differentiate with respect to β and set it to zero:

$$\begin{aligned}&\frac{\partial}{\partial\beta}\left(\mathrm{Var}(\phi(S,Z)) + \beta^2\mathrm{Var}(Y) + 2\beta\mathrm{Cov}(\phi(S,Z),Y)\right) = 0\\ \iff\ & 2\beta\mathrm{Var}(Y) + 2\mathrm{Cov}(\phi(S,Z),Y) = 0\end{aligned}$$

Thus, the optimal value of β is:

$$\beta^* = -\frac{\mathrm{Cov}(\phi(S,Z),Y)}{\mathrm{Var}(Y)}.$$

Substituting β^* back into the expression for the variance, we obtain:

$$\mathrm{Var}(\hat{AC}_I^{\mathrm{cv}}) = e^{-2rT}\frac{1}{I}\left(\mathrm{Var}(\phi(S,Z)) - \frac{\mathrm{Cov}(\phi(S,Z),Y)^2}{\mathrm{Var}(Y)}\right).$$

Finally, using the definition of the correlation coefficient ρ between $\phi(S,Z)$ and Y:

$$\rho = \frac{\mathrm{Cov}(\phi(S,Z),Y)}{\sqrt{\mathrm{Var}(\phi(S,Z))\mathrm{Var}(Y)}},$$

the variance can be rewritten as:

$$\mathrm{Var}(\hat{AC}_I^{\mathrm{cv}}) = e^{-2rT}\frac{1}{I}\mathrm{Var}(\phi(S,Z))\left(1-\rho^2\right).$$

This shows that the variance reduction depends on the correlation ρ between $\phi(S,Z)$ and Y. The closer ρ is to 1, the greater the reduction in variance.

Therefore, we need to chose a variable that is highly correlated with an Asian option. The candidates are the following: a geometric Asian option, and the empirical mean of the underlying's price. Each one of them has advantages and disadvantages.

Geometric Asian Option: It is very close to arithmetic Asian options so guarantees a high correlation but it requires the computation of a second option payoff. Therefore, it can be costly in computing time.

Arithmetic Empirical Mean: This variable differentiates because it is not an option. It is easier to compute and highly correlated because the payoff of the Asian option is based on this mean.

Let's compare the geometric Asian call option (GAC) and arithmetic empirical mean ($\bar{S}$) to see which one is better for our control variate method. These two variables are defined as follows:

$$\bar{S} = \frac{1}{M}\sum_{i=0}^{M} S(t_i)$$

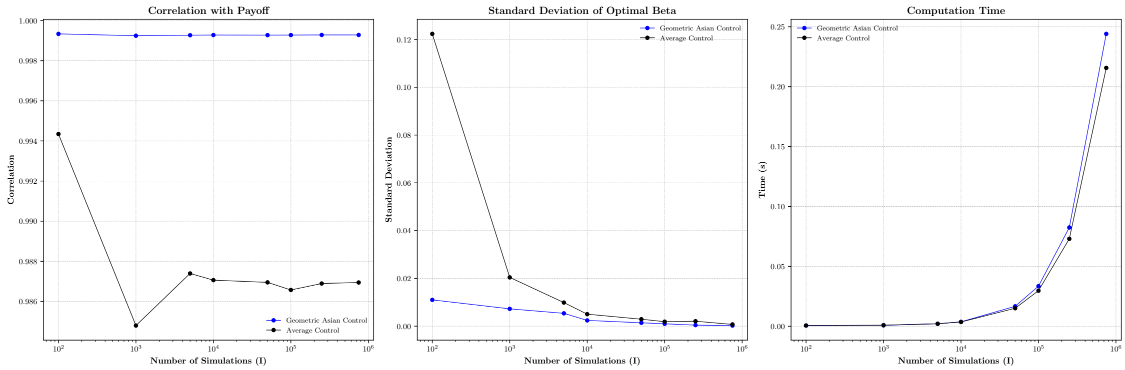

Figure 3.2: Comparison of control variates (Geometric Asian Option and Arithmetic Empirical Average)

$$GAC = \frac{e^{-rT}}{I}\sum_{k=1}^{I}\psi(S_k, Z_k)$$

with $\psi(S_k, Z_k) = \left(\exp\left(\frac{1}{M}\sum_{i=1}^{M}\log(S_k(t_i), Z_{k,i})\right). - K\right)^+$

These graphs shows that the correlation obtained with GAC is higher and more stable than that of $\bar{S}$. GAC is also more robust, as the standard deviation of β^* calculated on subsamples are lower, at least for low values of I. Finally, the computing time for GAC is slightly longer than $\bar{S}$. We will consider that this difference is negligible considering GAC's performance on correlation and β^*'s stability.

Therefore, we choose GAC as control variate for the rest of this report. Let's compute the price of our Asian option with this new control variate method.

```python
def MC_control_Asian_price_geom(S0, K, T, r, sigma, I, M, dt):
    S = np.zeros((M+1, I))
    S[0] = S0

    for t in range(1, M+1):
        Z = np.random.standard_normal(I)
        S[t] = S[t-1] * np.exp((r - sigma**2 / 2) * dt + sigma *
            np.sqrt(dt) * Z)

    hT = np.maximum(np.mean(S[1:], axis=0) - K, 0) #Payoffs of the
        asian option using I simulations
    hT_geom = np.maximum(np.exp(np.mean(np.log(S[1:]), axis=0)) -
        K, 0) #Payoffs of the asian geometric call option using I
        simulations
```

```

    Z_control = hT_geom - np.mean(hT_geom)

    cov = np.cov(hT, Z_control, ddof=1)[0, 1] #Computing the covariance
    var_Z = np.var(Z_control, ddof=1)

    beta_opt = -cov / var_Z

    price_estimate = np.exp(-r * T) * np.sum(hT + beta_opt * Z_control) / I
    price_variance = np.exp(-2 * r * T) * np.var(hT + beta_opt * Z_control) / I
    price_std = np.sqrt(price_variance)
    return price_estimate, price_variance, price_std
```

Code 3.3: MC Control Variate Simulation Function

I	Estimated Price	Standard Deviation
100	15.1251	0.0514
1000	18.1891	0.0185
5000	17.8066	0.0081
10000	17.9023	0.0058
50000	18.0767	0.0027
100000	17.9886	0.0019
250000	18.0253	0.0012
750000	18.0383	0.0007

Table 3.4: MC Control Variate Estimated Prices

Once again, the estimated price of the option is around \$18. However, with this method the standard deviation fall under 3% with only 1,000 simulations.

3.4.3 Comparison of Methods

Now, let's compare these three methods on several criteria: precision (standard deviation), convergence (estimated price), computation time, and efficiency. We define efficiency as:

$$\text{Efficiency} = \frac{1}{\text{Variance} \times \text{Computing Time}}$$

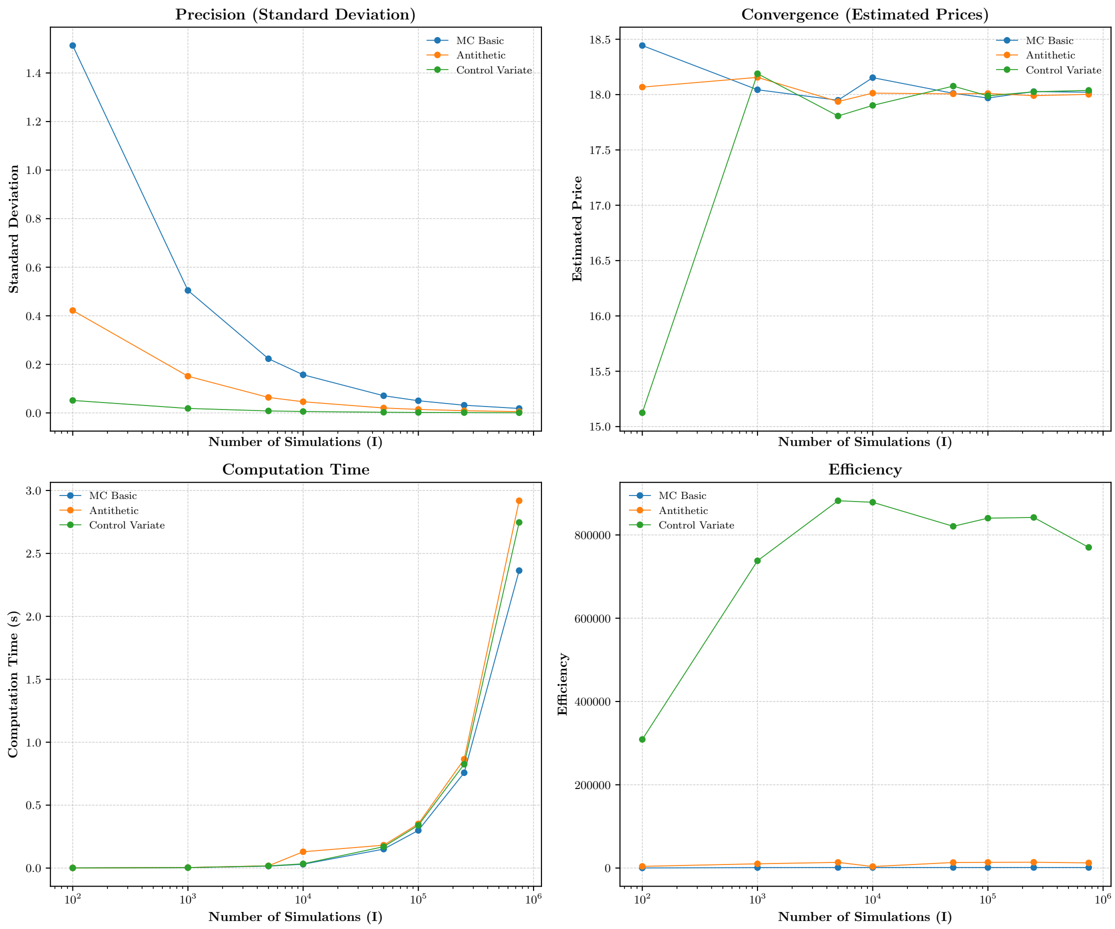

Figure 3.3: Comparison of MC methods

The basic Monte Carlo method is the simplest to use but has the highest variance, meaning its results are less precise. It takes longer to converge, requiring a lot more simulations to reach a good level of accuracy. Because of this, its efficiency is the lowest among the three methods.

The antithetic variates method improves on the basic Monte Carlo by using pairs of opposite simulations, which helps reduce the variance. This makes it more precise without increasing the computation time by much, even though we have to make twice as many simulations. Its efficiency is better than the basic Monte Carlo, but the amount of variance reduction it can achieve is limited. This is because it relies on symmetry rather than directly targeting specific sources of variance, so its advantage becomes less noticeable with a large number of simulations.

The control variates method is the most efficient of the three. By adjusting the simulated payoff using an auxiliary variable that is strongly correlated with the payoff , it reduces variance much more effectively than the antithetic variates method. Even though the control variates method involves extra calculations, like finding the optimal value β^*, it doesn't add much to the computing time. As a result, its precision and efficiency are much higher.

In summary, the control variates method is the best choice for high precision. The antithetic variates method is simpler and still improves results compared to basic Monte Carlo, but it's less effective for reducing variance.

Based on these graphs, we can now choose an "optimal" number of simulations for the control variate method. The efficiency is at its peak at $I = 10^3$ but remains very close to its maximum at $I = 10^4$. The estimated price is closer to the horizontal asymptote at $I = 10^4$ than at $I = 10^3$. Additionally, the standard deviation flattens approximately from 10^4, indicating diminishing returns in accuracy with larger I. Finally, the computing time between $I = 10^3$ and $I = 10^4$ are almost the same. To ensure greater stability, we choose $I^* = 10^4$, as 10^4 is still a small number of simulations.

3.5 Randomised Quasi-Monte Carlo Methods

Now, let's explore QMC methods. We are motivated by the fact that QMC methods have a convergence speed of $\mathcal{O}\left(\frac{\log(I^M)}{I}\right)$, which is better than MC methods. However, QMC can struggle with large value of M as the quasi-uniform structure of Sobol sequences loses efficiency as M increases. This means that a significantly larger number of simulations I may be required for QMC to outperform MC. In our case, $M = 52$, which is at the limit where this degradation can occur. As a result, we will empirically test whether QMC still provides an advantage over MC in this context.

In MC methods, we estimate the price with

$$\hat{C}_I = \frac{e^{-rT}}{I} \sum_{k=1}^{I} \phi(S_k, Z_k)$$

where Z_k is simulated with numpy's function `np.random.standard_normal()`. This function relies on a pseudo-random number generator, which can lead to some clustering of points and less uniform coverage of the sample space.

Instead, QMC methods simulate Z by applying the inverse cumulative distribution function of the standard normal distribution to quasi-random uniform variables. This ensures a more even distribution of points over the definition domain, improving the convergence rate. Quasi random generators are deterministic sequences that have low discrepancy. In our case, we will use the Sobol sequence. This means that

$$Z_k = \Phi^{-1}(U_k)$$

where U_k is the k-th term of the Sobol sequence and Φ^{-1} is the ICDF of a standard normal distribution.

The issue with this QMC method is that it is deterministic, so it makes the evaluation of numerical error complex. To handle it, we can randomise $(U_n)_{n\in\mathbb{N}^*}$ so that

- $U_n \sim \mathcal{U}([0,1]^M)$ marginally.
- U_n is still a low-discrepancy sequence.

Therefore, we have

$$\hat{\mathbb{E}}\left[\frac{e^{-rT}}{I}\sum_{k=1}^{I}\phi(S_k,\Phi^{-1}(U_k))\right] = e^{-rT}\hat{\mathbb{E}}[\phi(S_k,\Phi^{-1}(U_k))]$$

and we can evaluate the numerical error with the empirical variance. This type of method is called Randomised Quasi-Monte Carlo (RQMC). We will implement this method.

The `scipy.stats.qmc` library integrates a scrambled Sobol sequence that randomizes it, so we don't have to address this issue by ourselves. Let's consider our RQMC estimator:

$$\tilde{AC}_I = \frac{e^{-rT}}{I}\sum_{k=1}^{I}\phi(S_k,\Phi^{-1}(U_k))$$

with the same notations as before.

3.5.1 Implementation

The implementation of RQMC methods is quite simple as it is mainly the same as Code 3.1, Code 3.2, and Code 3.3. The only change is the simulation of the gaussian distribution. Instead of using `np.random.standard_normal()`, we use the aforementioned method to generate path simulations:

```
sampler = qmc.Sobol(self.M, scramble=True)
quasi_uniform = sampler.random(I)
quasi_normal = norm.ppf(np.clip(quasi_uniform, 1e-10, 1 - 1e-10))
```

```

S = np.zeros((self.M + 1, I))
S[0] = self.option.S0
for t in range(1, self.M + 1):
        Z = quasi_normal[:, t-1]
        S[t] = S[t - 1] * np.exp((self.option.r -
            self.option.sigma**2 / 2) * self.dt + self.option.sigma
            * np.sqrt(self.dt) * Z)
```

Code 3.4: RQMC Paths Generation

The results obtained are the following:

I	Estimated Price	Standard Deviation
100	18.7279	1.8169
1000	17.9756	0.5124
5000	18.0177	0.2234
10000	18.0171	0.1576
50000	18.0006	0.0708
100000	18.0091	0.0500
250000	18.0072	0.0316
750000	18.0076	0.0183

Table 3.5: RQMC Basic Estimated Prices

I	Estimated Price	Standard Deviation
100	16.9622	0.2271
1000	18.0729	0.1450
5000	17.9931	0.0663
10000	18.0017	0.0451
50000	18.0114	0.0203
100000	18.0099	0.0144
250000	18.0077	0.0091
750000	18.0077	0.0053

Table 3.6: RQMC Antithetic Estimated Prices

I	Estimated Price	Standard Deviation
100	17.5641	0.0633
1000	18.0116	0.0213
5000	17.9895	0.0083
10000	17.9996	0.0061
50000	18.0071	0.0027
100000	18.0107	0.0019
250000	18.0087	0.0012
750000	18.0075	0.0007

Table 3.7: RQMC Control Variate Estimated Prices

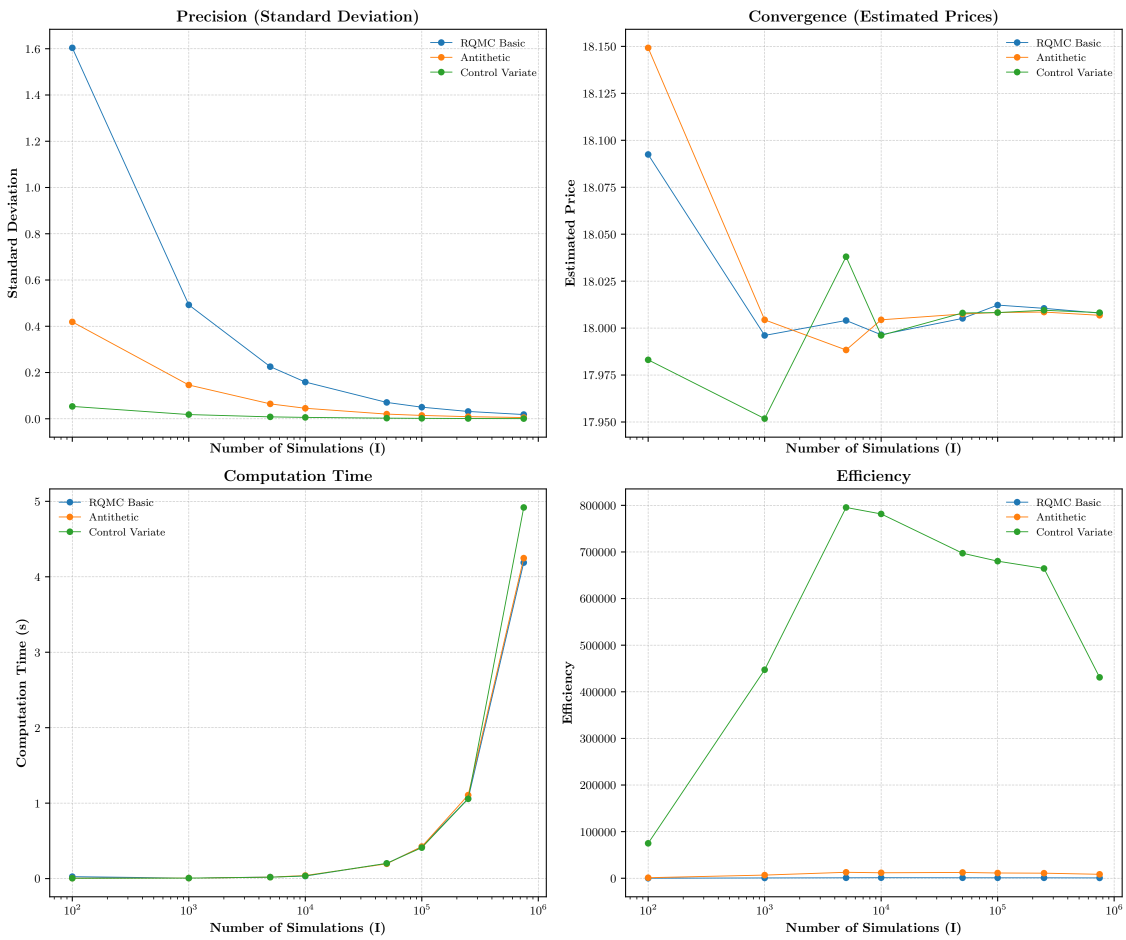

Figure 3.4: Comparison of RQMC methods

3.6 Analysis of Results

Based on Figure 3.3 and Figure 3.4, we can draw some conclusions on the different criteria.

Precision (Standard Deviation) There is no significant difference between MC and RQMC methods in terms of standard deviatioin. This is often observed in high dimension, because the quasi-uniform structure of the Sobol sequence loses efficiency when M is large.

Convergence (Estimated Prices) In both methods, the estimated prices converge approximately to 18.0, demonstrating the stability and consistency of the algorithm. The convergence appears noisier with MC than with RQMC, but as the number of simulations I tends to infinity, the benefits of RQMC become negligible. Contrary to our expectations, RQMC does not significantly improve convergence in this case.

Computation Time MC and RQMC have similar computation time. This is a positive result for RQMC, as it could be more computationally expensive for several reasons. First, the generation of the Sobol sequence is more complex than optimized pseudo-random number generators like `np.random.standard_normal()` ($\mathcal{O}(I \times M)$ against $\mathcal{O}(I)$). The scrambling applied to the Sobol sequence increases the computational cost too. Second, transforming Sobol sequence values into a Gaussian distributiion requires the inverse cumulative function method, which is less optimized than `Numpy`'s generators and even more computationally expensive in high dimension. Fortunately, we observe neither a gain nor a loss here.

Efficiency Once again, the results are quite similar in terms of efficiency for both MC and RQMC. The most impactful method to improve efficiency remains variance reduction through control variates.

Overall, RQMC does not provide any significant advantage over MC methods based on these criteria. The most effective method for pricing Asian call options in this specific context is Monte Carlo with geometric Asian call option as a control variate. We found that $I^* = 10^4$ is the optimal number of simulations for an optimal trade-off between precision and computation time.

References

[1] Idris Kharroubi. **Calcul Stochastique et Introduction au Contrôle Stochastique**. Lecture notes, Sorbonne Université, 2024.

[2] Nicolas Chopin. **Simulation and Monte Carlo Methods**. Lecture notes, ENSAE Paris, 2025.