Monte Carlo for the Margrabe option

Bocconi University
APPLIED NUMERICAL FINANCE
- Code 20247 - Fall 2023

Davide Mezzadri - Gruppo 12

1 Introduction

Consider two risky assets S_1 and S_2, driven by

$$\frac{dS(t)}{S(t)} = (r - q_i)\, dt + \sigma_i^T\, dW^Q(t) \tag{1}$$

for $i = 1, 2$ where q_i is the dividend yield, W^Q is a 2-dimensional standard Brownian motion, $\sigma_1^T = [v_1, 0]$ whereas $\sigma_2^T = [\rho\, v_2, \sqrt{1-\rho^2}\, v_2]$ for $\rho \in (-1; 1)$, and $v_1, v_2 > 0$. r is the riskless interest rate.
The Margrabe option is an exotic one, that gives the buyer the right, but not the obligation, to exchange the second asset for the first at the time of maturity T. From a mathematical point of view, the payoff at T is

$$X(T) = \max\{S_1(T) - S_2(T); 0\} \tag{2}$$

One of the main point of the project is leveraging Monte Carlo techniques to compute the price of the option S_X and to conduct further analysis.

In addition, we consider the Margrabe's formula below, namely the closed-form solution for the price of the Margrabe option at $t = 0$.

$$S_X(0) = S_1(0)\, e^{-q_1 T}\, N(d_1) - S_2(0)\, e^{-q_2 T}\, N(d_2) \tag{3}$$

where

$$d_1 = \frac{\ln \frac{S_1(0)}{S_2(0)} + (q_2 - q_1 + \frac{\bar{\sigma}^2}{2})T}{\bar{\sigma} T}$$
$$d_2 = d_1 - \bar{\sigma}\sqrt{T}$$
$$\bar{\sigma}^2 = v_1^2 + v_2^2 - 2\rho v_1 v_2$$

and $N(x)$ is the value of the cumulative distribution function of a standard normal computed at x.

The closed-form solution is useful for a comparative analysis with Monte Carlo, in particular to validate and refine the simulation models.

The report first considers the numerical and financial aspects relating to the execution of the project. Subsequently, it focuses on the development of the code: the main choices, the structure, possible issues and future enrichments. In addition, the report provides a collection of graphical attachments to illustrate some results of the analyzes conducted.

2 Financial and Numerical Aspects

Firstly, it is required to simulate the NA values of securities at maturity based on specified parameters, namely initial prices of the securities ($S_i(0)$), dividend yields (q_i), volatilities (σ_i), correlation (ρ) and maturity (T). This simulation is accomplished through the following formula:

$$S(t) = S(0)e^{(r-q_i-\frac{v_i^2}{2})t+\sigma_i \cdot W^Q(t)} \tag{4}$$

In particular, the simulation necessitates generating a 2-dimensional standard Brownian motion, which boils down to simulate $[\sqrt{t} \cdot Z_1, \sqrt{t} \cdot Z_2]$, where Z_i are independent standard Gaussians.

To comprehend the impact of the parameters on the trajectories of the two securities, the corresponding paths are plotted. For this purpose, time discretization is required based on a predefined number of monitoring dates.

The simulated prices of securities S_1 and S_2 are utilized to compute the payoff of the Margrabe option at maturity:

$$X(T) = \max\{S_1(T) - S_2(T); 0\} \tag{5}$$

The payoff is discounted using the risk-free interest rate (r) and the time-to-maturity (T), where the discount rate at T is e^{-rT}. The sample average of N different simulations of discounted payoffs provides the Monte Carlo estimate of the fair price of the Margrabe option at $t = 0$.

Additionally, the estimate is juxtaposed with the option price derived using the closed-form formula. Confidence intervals for the Monte Carlo estimate are also constructed. In particular, we use the standard deviation estimated from the simulated discounted payoffs and the number of replications.

Figure 1 illustrates the Monte Carlo estimate, confidence interval bounds, and the price calculated using the closed-form solution for varying numbers of replications. Notably, the interval radius diminishes as the number of replications escalates.

Subsequent estimations were conducted employing **variance reduction techniques**, specifically the Control Variate (CV) and Antithetic Variates (AV) methods.

Concerning the CV technique, $S_1(T)$ was selected as the control variate, whose (known) expected value under the risk neutral probability measure in the Black-Scholes multivariate framework is $E^Q[S(T)] = e^{(r-q_i)T}$. A priori, we can say that this method mitigates variance since $S_1(T)$ correlates with the option price at $t = 0$ (in fact, as the value of $S_1(T)$ increases, the value of the payoff at expiration increases and therefore the price of the option). Concerning b, it is estimated through the slope of the least-square regression line of the "unknown" Y (the option payoff) on the "known" X ($S_1(T)$). To have an unbiased estimator, $\hat{b}_n$ is calculated using replications distinct from those used for the final option price estimate.

In the context of the AV method, variance reduction is achieved in comparison to the traditional Monte Carlo estimate when the antithetic variables (in this case we have antithetic payoffs), exhibit negative correlation. This arises when the simulation map is monotone. However, for the Margrabe option, confirming this becomes challenging and is contingent upon the specific parameters involved. Consequently, our subsequent analyses will focus on evaluating this variance reduction through dedicated simulations.

To provide a comparative assessment across the methodologies, different simulations were juxtaposed. Specifically, we examined the distribution of absolute errors, obtained from a particular number of estimates, between the estimates generated by the method in consideration and the closed-formula solution. In Figure 2, we depict the distribution of absolute errors pertaining to the MC method. In contrast, Figure 3 illustrates the analogous distribution for the AV method. Both are realized on 100 estimates with 1,000 replications and the same initial data configurations. Notably, the distribution associated with the AV method manifests a more pronounced skewness towards zero (it has not produced absolute errors > 0.4, differently from MC), underscoring its superior performance efficacy.

Lastly, Figure 4 synthesizes the estimates derived from three distinct methodologies, inclusive of their respective confidence interval bounds and the values derived via the closed-formula. The figure reveals that the confidence interval radius associated with both the CV and the AV methodologies is notably narrower than that obtained through the conventional Monte Carlo approach. This outcome substantiates the efficacy of CV and AV methods in mitigating variance, thereby enhancing the precision of our estimations.

Finally, we conducted a **sensitivity analysis** on the option price concerning its underlying parameters. This was achieved by computing the value of the option, both via MC simulations and the closed-form solution, across a spectrum of parameter values. Users can define these values, by specifying the minimum and maximum bounds of an interval alongside the desired number of subdivisions. This approach facilitated the generation of specialized tables and graphs tailored to the sensitivity analysis. Furthermore, to enhance our analytical depth, we also explored the sensitivity concerning two parameters at the same time, by the creation of contour plots. Notably, our focus was about volatilities and correlations, aligning with primary objectives of the project.

Concerning the volatility, see Figures 5 and 6: the option price grows as v_1 and v_2 increase (at least for the starting inputs, in which the initial values of the securities are equal). Intuitively, this can be verified in the following way: as volatility increases, the probability that the value $S_1(T)$ is high or $S_2(T)$ is low increases. Consequently, the likelihood of a high payoff escalates, leading to a higher option price. It's worth noting that while the probability that $S_1(T)$ is low or $S_2(T)$ is high also rises, its impact remains constrained by the potential non-exercise of the option.

Regarding correlation, as depicted in Figure 7, within the interval $[-1, 1]$, the option value demonstrates a decreasing trend (at least for the starting inputs, in which the initial values of the securities are equal). Intuitively, it could be verified by considering that if two securities are strongly positively correlated, it is more likely that the value is very similar, lowering the value of the payoff. Instead, when they are strongly negatively correlated it is more likely that the payoff is high, increasing the value of the option (obviously it is also more likely that the difference between the prices of the securities at maturity is low, but this is mitigated by the possibility of not exercising the option).

The two-dimensional analysis verifies the previous considerations: see the contour plots in 8 and 9, which focus on v_2 and ρ. The difference between 8 and 9 is that in the first case the prices are simulated, in the second calculated with the closed-form solution: in fact, in the first case you can see how the contours are irregular. Crucially, these graphs underscore that, within the specified intervals, v_2 exerts a more pronounced influence than ρ on the variations observed in the option price.

3 The code

I have decided to use a Jupyter Notebook for Python to develop the code, for various reasons.

First, it allows to write inline comments, documentation and explanations alongside code cells in a nice format. This feature promotes a more effective communication of the results and of the purpose, the logic and the usage of the code. In addition, Jupyter Notebooks allow to execute Python code cells individually or collectively, producing immediate outputs, results, plots, tables, statistics. For this motive, Jupyter Notebooks facilitate data visualization, exploration and analysis: crucial points of this project.

Even if Jupyter Notebooks may not be the ideal platform for creating highly modular software solutions, it is possible to create separate modules by grouping similar functions in the same sections.
The libraries used for the development of the code are:

- **scipy.stats** for computing cumulative probabilities, quantiles associated with the normal distribution, necessary in option pricing and in calculation of confidence intervals.
- **numpy**: for array manipulations, mathematical computations, and efficient numerical operations crucial for simulations and data processing.
- **matplotlib.pyplot**: essential for plotting.
- **pandas**: assists in reading data, preprocessing datasets, and creating tables.

Concerning the project, I have divided the code into an operational panel and a set of functions. In the operational panel, we can initialize the inputs, execute the functions, publish the outputs and produce the plots. In the second part there are the functions developed for the realization of the project, separated into modules, with an extensive documentation (that permits a clear understanding of the project, facilitating possible future extensions). In this regard, you should check the notebook.

4 Conclusion and possible extensions

The report offers a comprehensive description of the project about the simulation of a Margrabe option, leveraging both mathematical aspects and techniques for estimation. We have shown the effectiveness in price evaluation, even under different scenarios. Utilizing a Jupyter Notebook in Python, the code development enhances transparency, facilitating effective communication and visualization of results.

Regarding possible extensions, it could be interesting to consider new estimation methods, or consider other parameters with respect to which a sensitivity analysis can be performed. This is facilitated by specific sections of the code where new functions can be added. Another potentially important analysis could be the estimation of Greeks for the Margrabe option. Finally, another interesting feature could be the realization of interactive dashboards and visualization tools using libraries like Plotly to provide effective insights into the valuation of the Margrabe option.

5 Attachments

The following plots have been produced with the following inputs: $S_1(0) = 60$, $S_2(0) = 60$, $q_1 = 0.01$, $q_2 = 0.01$, $v_1 = 0.05$, $v_2 = 0.25$, $\rho =$ -0.9, $r = 0.05$, $T = 1$. Regarding the confidence intervals, $\alpha = 5\%$.

Concerning the sensitivity analysis, I systematically varied the parameter (or the parameters) of interest while keeping all other factors constant to the values above.

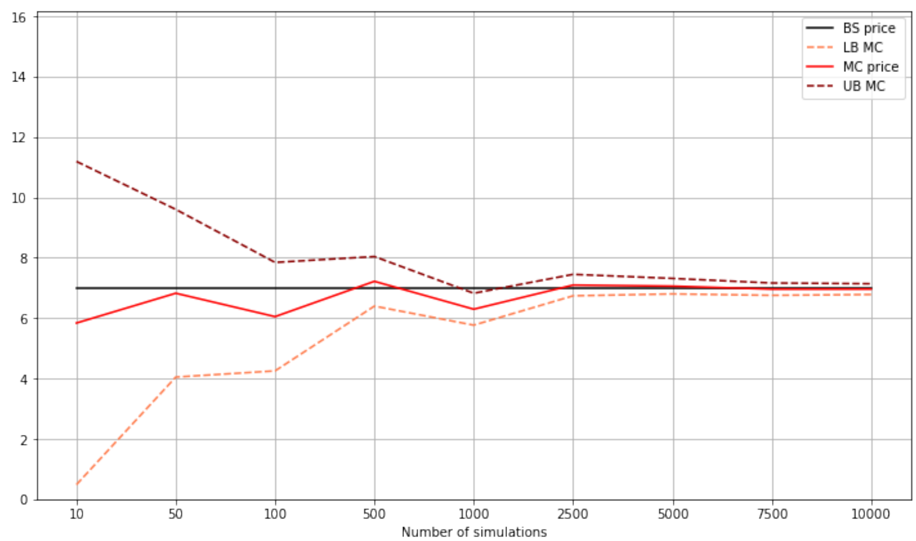

Figure 1

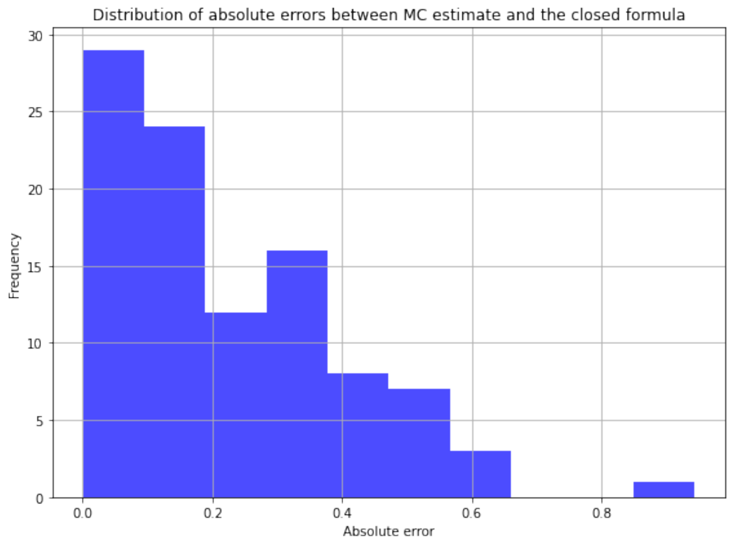

Figure 2

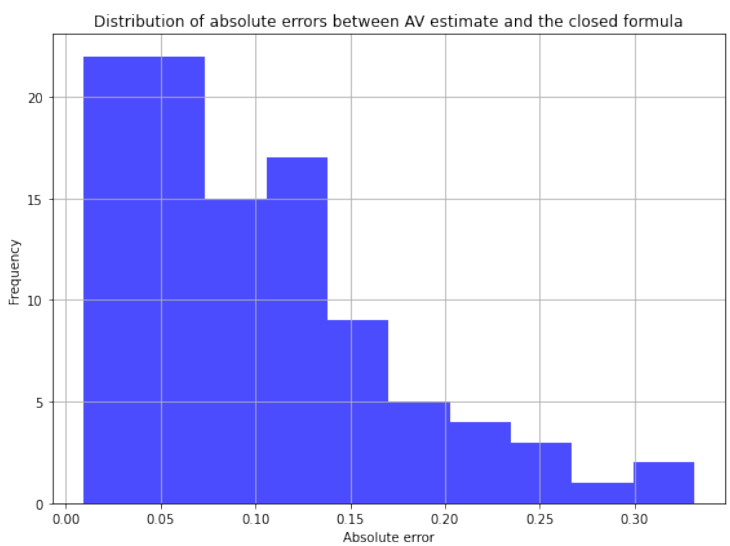

Figure 3

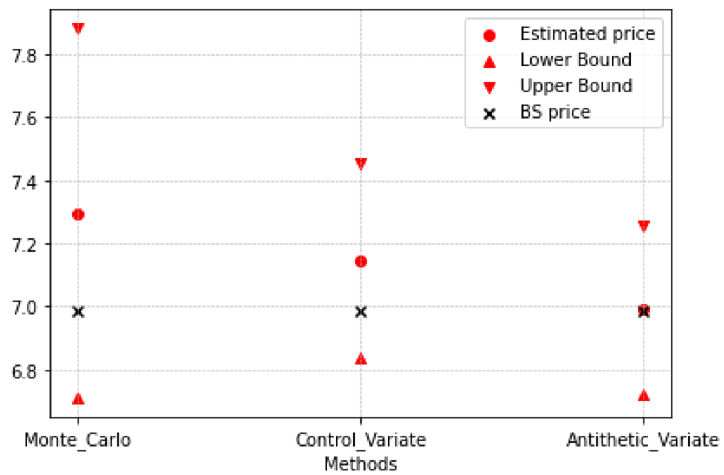

Figure 4

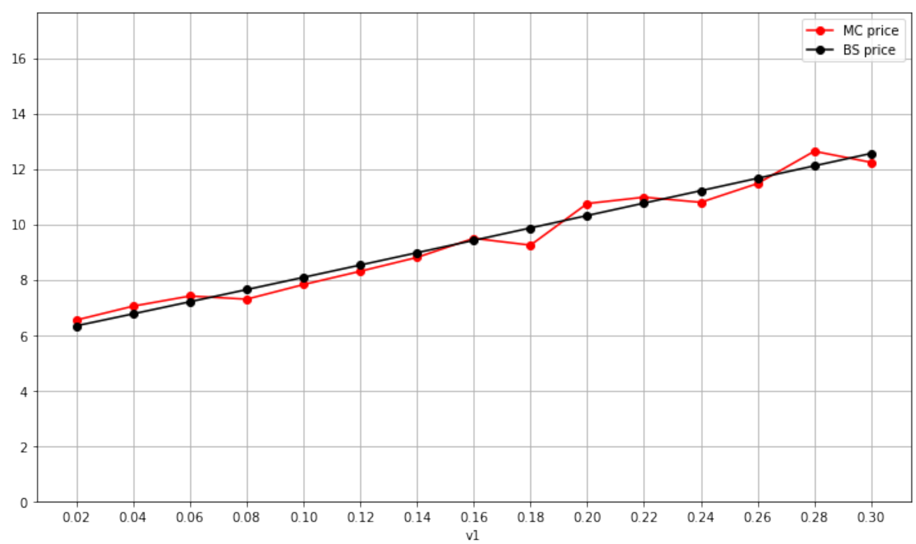

Figure 5

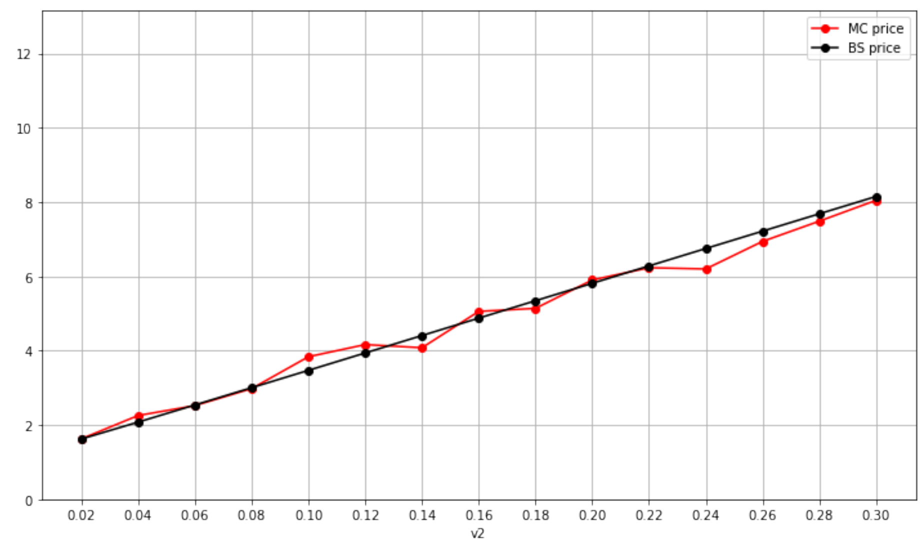

Figure 6

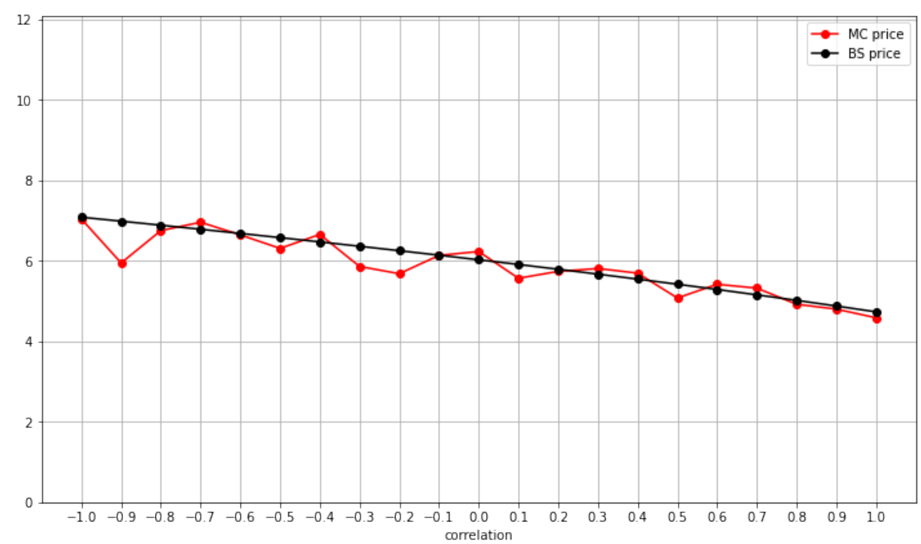

Figure 7

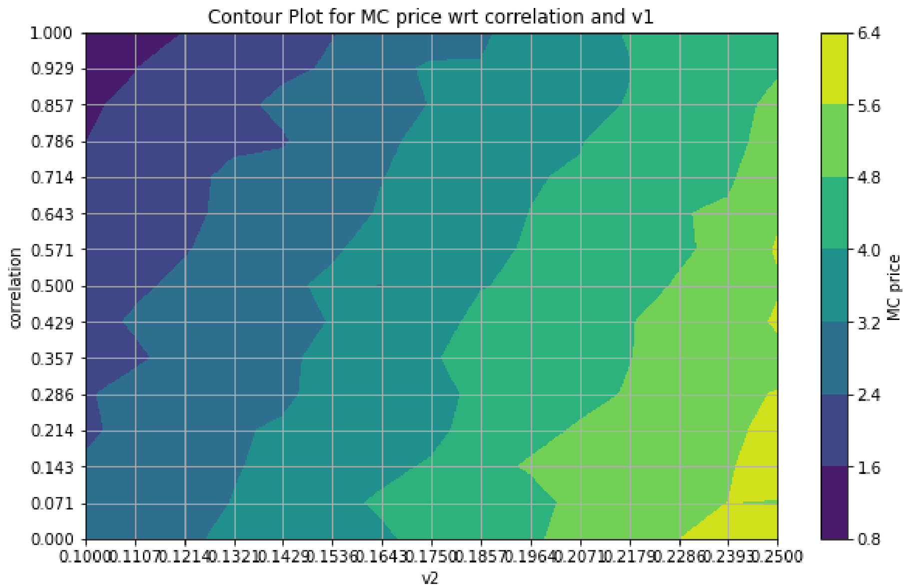

Figure 8

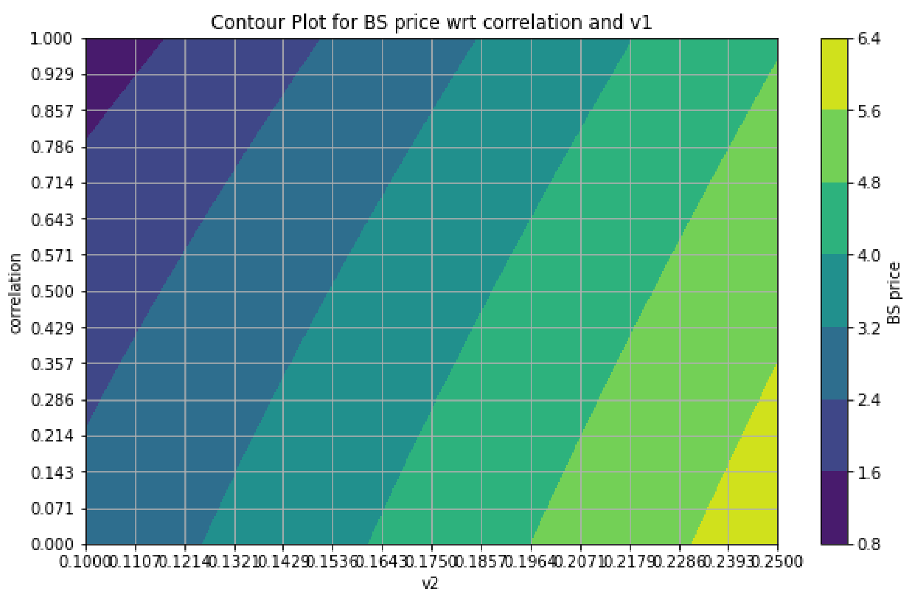

Figure 9