ASSIGNMENT 5

Monte Carlo for the Margrabe option

Group 12:

Christian Gaggiotti 3119139

Pier Francesco Prodani 3004937

Summary

1. Introduction

The aim of the report is to price a European Margrage option, both applying the Margrabe closed form solution for exchange options and by means of a Monte Carlo simulation. A sensitivity analysis will be also implemented, to assess how the option price changes in response to movements in a series of parameters. In addition, we will test our Monte Carlo simulation through all these different scenarios to see if we can find any sign of over or underperformance with respect to changes in said parameters.

The analysis is carried out using the Python programming language, which has been preferred for some specific reasons: it has a very practical and user-friendly syntax which makes coding way easier and it allows for a high computational capability. For all these reasons it is probably the most used programming language within the finance world and beyond, which is why we can find an almost infinite number of packages easily downloadable from the internet. Python is therefore a really efficient language, suitable for pretty much all possible scopes.

Models and Inputs

Exchange options are options to exchange one asset for another, in this case stocks. Therefore if the option is of the European type, like this one, the payoff at maturity is simply:

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

Whose risk-neutral discounted expectation constitutes the value of the option at time equal to zero. In 1978 the economist W. Margrabe was able to compute a closed-form solution to price options of this kind, which is now known as Margrabe's formula, and allows us to calculate the today's price of an option with that specific payoff at maturity as follow:

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

$$d_1 = \frac{\ln \frac{S_1(0)}{S_2(0)} + \left(q_2 - q_1 + \frac{\overline{\sigma}^2}{2}\right) T}{\overline{\sigma}\sqrt{T}}$$

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

$$\overline{\sigma}^2 = v_1^2 + v_2^2 - 2\rho v_1 v_2$$

Where, with respect to the two underlying assets, $S_1(0)$ and $S_2(0)$ are the today's prices, q_1 and q_2 are the continuous dividend yields, v_1 and v_2 are the volatilities, ρ is the correlation coefficient and T is the time to maturity. This formula is obtained under the assumption that the underlings follow a lognormal process of the Black and Scholes type, which we also use to carry out the simulation. Indeed, the prices of two stocks S_i at time t can be easily derived in a bivariate Black Scholes market under the risk-neutral Q probability measure:

$$S_1(t) = S_1(0) \cdot Exp\left[\left(r - \frac{\sigma_1^2}{2}\right) \cdot t + \sigma_1 \cdot \sqrt{t} \cdot Z_1(0,1)\right]$$
$$S_2(t) = S_2(0) \cdot Exp\left[\left(r - \frac{\sigma_2^2}{2}\right) \cdot t + \sigma_2 \cdot \sqrt{t} \cdot \left(\rho \cdot Z_1(0,1) + \sqrt{1-\rho^2} Z_2(0,1)\right)\right]$$

Where Z_i are independent standard normal random variables. In this way the final distribution of the two stocks' values approximate the lognormal distribution which is implied by the Black Scholes market. We now have all the iputs to start our analysis. In the next section we will price first the option with the closed formula and then by simulating the stocks' prices at maturity. The today's price will then be computed as:

$$S_X(O) = \frac{e^{-rT}\sum_{i=1}^{n} \max\left[\left(S_1(T)(i) - S_2(T)(i)\right);\ 0\right]}{n}$$

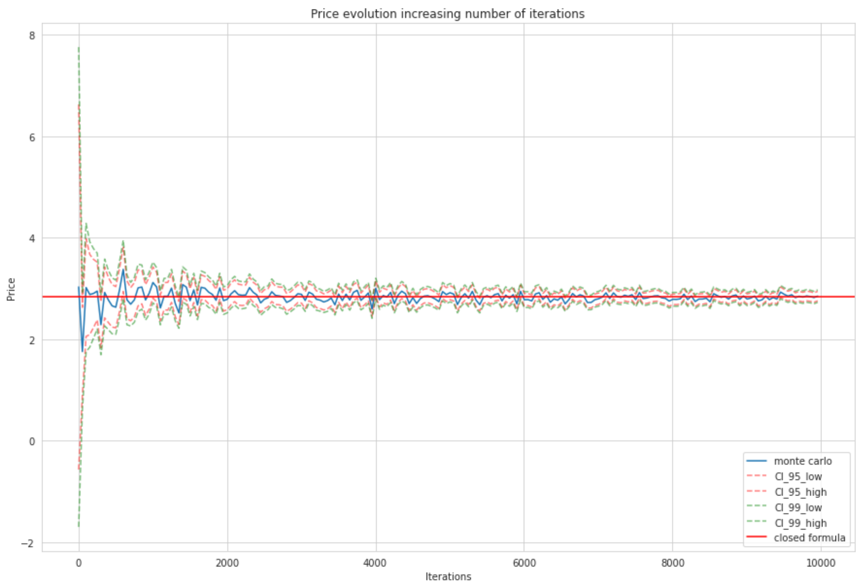
Which is simply the average of the discounted values of the payoff obtained in each simulation *i*. Afterwards we will assess the quality of our Montecarlo simulation in trying to approximate the true value of the option.

2. Monte Carlo simulation

In order to carry out the simulation we assign some starting inputs to our model. For the sake of simplicity the option is set to be ATM with today's stock values both equal to 20. Stock 1 has a volatility of 30%, higher than Stock 2 which has a volatility of 10%. Correlation is negative and equal to -0.5 (we will see that this increases significantly the value of the option). Dividend yield is equal to 1% for both stocks, and time to maturity is 1 year. The risk free rate is equal to 3%. With these inputs we obtain a true value using the Margrabe's formula of 2.833.

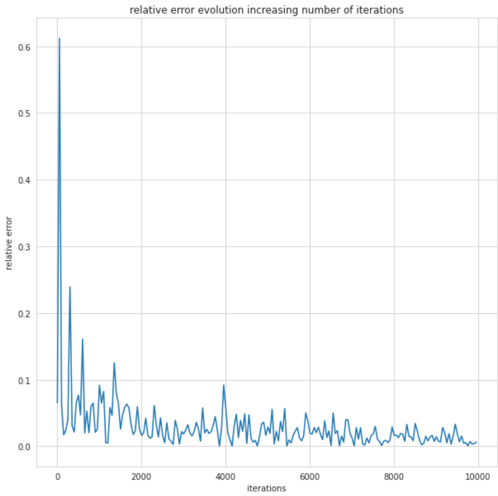
We now create a loop function, named Monte, that given a n number of repetitions generates two standardized normal random numbers (Z_i), computes the difference of the stock values at maturity, calculated according to the formula specified in the previous paragraph, and then applies an if statement that returns the discounted payoff if that difference is positive and 0 otherwise. The function performes the process n times and saves the results in a vector. At this point we create a loop that runs the function with n (number of repetitions) going from 5 to 10,000 by steps of 50, and each time computes the mean over the n simulations. We insert directly into the loop also the computation of the upper and lower bounds of the 95% and 99% confidence intervals, which are given by:

$$CI = \bar{X} \pm \frac{1.96 \times \sigma_x}{\sqrt{n}}$$



Where $\bar{X}$ in our case is the option value, 1.96 is the Z_i associated with the 95% level (2.576 for the 99%), σ_x is the volatility of the payoffs obtained and n is the number of simulations. Hence we come up with a table displaying for each Montecarlo simulation the number of repetitions implemented, the option price and its 95% and 99% confidence interval extremes. We now plot everything in a graph, adding a red line that represents the true price.

It is clear that as the number of simulations increases, our model is more and more capable of approximating the true value and already from 2000 repetitions onwards we experience very few and little fluctuations around the correct value. The same goes for both confidence intervals, which shrink as the number of simulations increases.



In order to have an even better picture of the speed of convergence, we compute the relative error for each one of our estimates as the absolute value of the difference between the simulated value and the true one, divided by the latter, and again we report them in a graph. What we claimed before is confirmed, even though we can now state that in order to have no swings at all, or almost no swings at all, we need to reach about 10,000 repetitions.

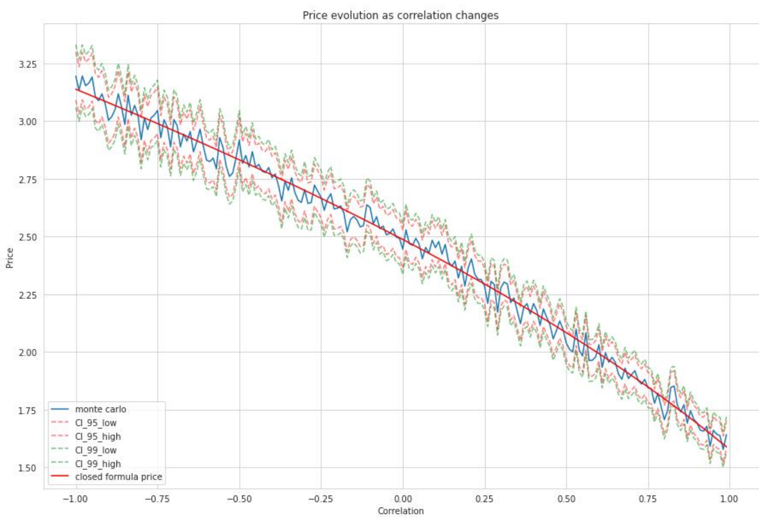
Finally for such number of simulations onwards we can definitely conclude that our method is both solid and precise.

3. Sensitivity analysis

In this section we will test the sensitivity of both the option price and its simulated price, with respect to the main drivers. It is interesting to note that the option price closed formula is independent of the risk-free rate r. This is because, as r increases, the growth rate of both asset prices in a risk-neutral world increases, but this is exactly offset by an increase in the discount rate. Hence, we disregard the sensitivity to the risk free rate, and we focus on the relevant variables, that are in order: correlation (ρ), volatilities ($\sigma_{1,2}$), Stocks' spot prices ($S_{1,2}$) and continuous dividend yields ($q_{1,2}$).

Correlation

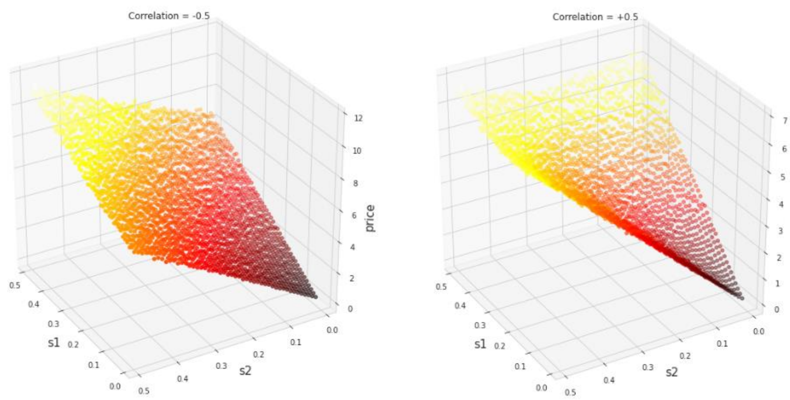
In order to test for correlation shifts we create a list of values for correlation ranging from -1 to +1 with steps of 0.01. Then we run a loop that computes the Montecarlo simulation as explained in the previous section, for each of the values in the correlation list. The number of simulations "n" is fixed and equal to 10,000, since we have already assessed that it is the proper number of repetitions for a good evaluation. In this way we get a table where for each correlation coefficient ρ we have a simulated price with its confidence intervals extremes (95% and 99%). Then we add a column with the right price, calculated with the closed form formula, corresponding to each ρ. Finally we just report everything in a graph.



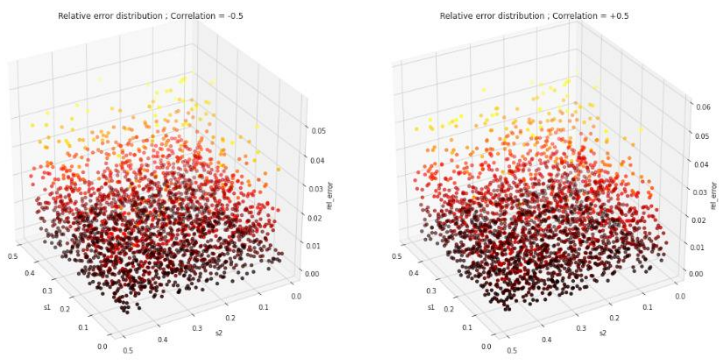
As predicted, the price of the option is higher for lower values of correlation. This is because a lower correlation implies that there is a higher probability that the two underlyings will move in opposite directions, increasing their difference in values and thus the final payoff of the option. If we focus instead on the performance of our simulation, we see that the values are always pretty close to the true ones, even though there are recurring fluctuations. Those fluctuations seem to be apparently decresing in magnitude as correlation increases, but this is probably due to the fact that for very high values of ρ the value of the option is very low, hence the swings are consequently lower in absolute terms. Therefore we can not say that the changing correlation alone has any significant impact on the performance of the Monte Carlo simulation, which remains pretty solid.

Volatilities

For what concerns the sensitivity to the volatilities of the two underlyings ($\sigma_{1,2}$), we basically implement the same procedure seen before for the correlation. There are just two major differences that we want to point out. Here we have two values that can vary at the same time, hence we have to create two lists of possible values for volatilities (for simplicity ranging from 1% to 50% with steps of 1%) and then two nested loops in the code in order to consider all possible combinations of values. Furthermore, we want to investigate also the cross impact that high or low values of correlation might have on the sensitivity to volatilities, thus we run the analysis assuming first a ρ of -0.5 and then of +0.5 and we compare the results.



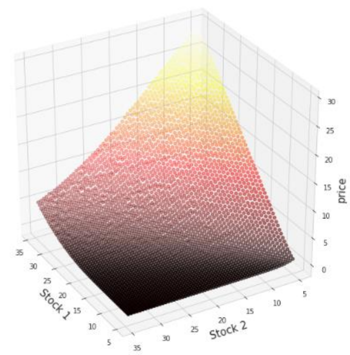
The first thing we notice is that they look very similar, meaning that they share the same trend with respect to the sensitivity to volatility changes. In fact the effect of changes in σ is quite clear: the option is Vega positive with respect to both underlyings, and in order to see it even better we coloured the points in the graph according to the value of the z-axis which represents the price (the same will apply also for future graphs). The finding makes a lot of sense, since our loss is capped at zero, and hence high volatility implies higher possible profits, both coming from high values of S_1 or from low values of S_2. Another interesting thing to nitice is that the two graphs are indeed similar but not identical. When we have negative correlation the growth in value happens in an almost linear way with respect to both volatilities. Also the highest values are reached when both volatilities are high. On the contrary when correlation is positive the function becomes more convex. This is because even though the overall value of the option decreases (for the reason explained in the previous paragraph), it becomes more sensitive to changes in the single volatility. The highest values are indeed reached when at least one volatility is high. This is probably due to the fact that the more the two stocks move together, the more we want just one volatility to be high, in order to create differences in the values that can increase the payoff.

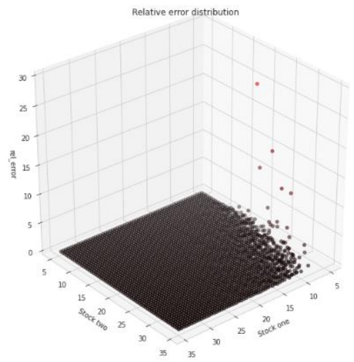


To see if the Monte Carlo simulation is in any way impacted by changes in volatilities, we compute the relative error for each point and again we display it in the graph. We are not able to identify any clear pattern, meaning that the performance of the simulation is independent also from the values of volatilities. It is important, however, to comment on this result. Indeed we might expect that higher fluctuations of the underlyings will cause higher fluctuations of the option price, making in turn the approximation less reliable. This seems confirmed if we look at the absolute differences of the simulating price from the true price in the table from which we extract the above graphs. This difference should however be compared again with the true value of the option since even though the approximation becomes worse in absolute term, an error of 5 for example over a price of 50 can not be compared directly with an error of 0.5 over a true price of 1. That is why we compute the relative errors, which we believe to be a more suitable measure to evaluate the performance of an approximation procedure like the Montecarlo simulation.

Spot prices of the underlyings

Now we take into consideration the fluctuation of $S_{1,2}(0)$, hence the price of the two stocks at inception. The procedure is exactly the same described in the last section, with the difference that now $S_{1,2}(0)$ are the varying variables, ranging from 5 to 35 with steps of 0.5. Correlation is set again to -0.5 and volatilities both to 20% in order to isolate the sensitivity to $S_{1,2}(0)$. Here follow the results.

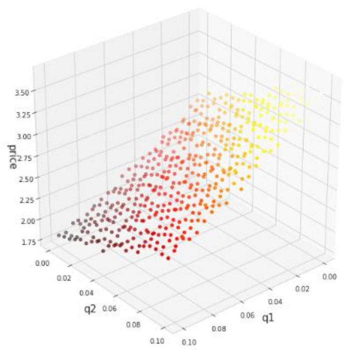


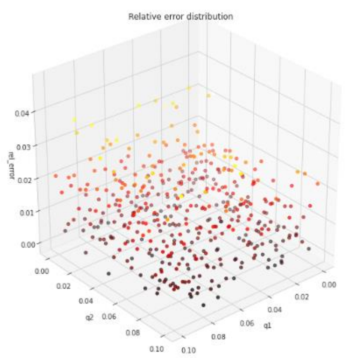


If we look at the first graph it is clear that for high values of $S_1(0)$ and low values of $S_2(0)$ the option is ITM hence with a higher value. We can also see that it is better to have simultaneously both high values than low values of the two underlyings, since even if the option is ATM, high values imply high range of possible payoffs while the loss is still capped at zero. The relative error is again quite uniform, even though we report some discrepancies when the option is deeply OTM. This is because the option value is so low that approaches zero, making the approximation much harder. Indeed even small differences become relevant in relative terms.

Continuous dividend yields

Finally we consider $q_{1,2,}$ and we run again the whole procedure, supposing that the option in ATM.





Since high values of q decrease the stock value, it is quite expected that the value of the option will be higher for higher values of q_2 and lower values of q_1. This is confirmed by our findings. Once again the relative error analysis does not return any important pattern, and we conclude that the performance of the simulation is independent from fluctuations in the dividend yield.

4. Conclusions

We report here the main findings and conlusions of our analysis.

- Montecarlo simulation represents a valid method to price Margrabe options, even though if we want the approximation error to be really close to zero we need a high number of simulations (e.g. 10,000).
- The Montecarlo simulation has a good performance also with regards to the replication of the sensitivity of the option price to movements in its major drivers. Indeed all the paths that we found with respect to correlation, volatilities, spot prices of the underlyings and continuous dividend yields are in accordance to what was predicted from the final payoff composition and from the structure of the closed form formula.
- The reliability of our simulation is not jeopardized by movements in the drivers. We can not find any relevant pattern in the relative errors in almost any of the scenarios we analyzed. The only exception is represented by the combination of lower values of $S_1(0)$ and high values of $S_2(0)$, which makes the option deeply OTM and in turn the approximation less accurate.