MAP556 PROJECT

Automatic Control Variates for Option Pricing using Neural Networks

9 décembre 2022

Timothée SELOI and Franck SIGNE TALLA

ÉCOLE POLYTECHNIQUE

IP PARIS

1
BLACK AND SCHOLES MODEL RESULTS

1.1 First Approach : Network dimension reduction

Note that all the results that we obtained are different from the ones presented in the article because we did not use the same parameters, as many of them were not specified (interest rate, number of assets...).

The histograms in blue are the one obtained by simulating directly the random variable of the payoff without the control variate. The one in orange are the histograms obtained by using control variate.

For the first approach, the neural network has the following structure (except for the Asian option, which input contains $N_{asset} * N_{Euler}$ variables instead of N_{asset} variables) :

— Input layer : $Z = (Z_1, \ldots, Z_{N_{asset}})$ independent standard normal variables, and $N_{asset} = 100$;

— First hidden layer : $n = 10$ neurons, no bias, no activation function

— Second and third hidden layer : $n = 10$ neurons with bias and ReLu activation function.

— Output layer : scalar output

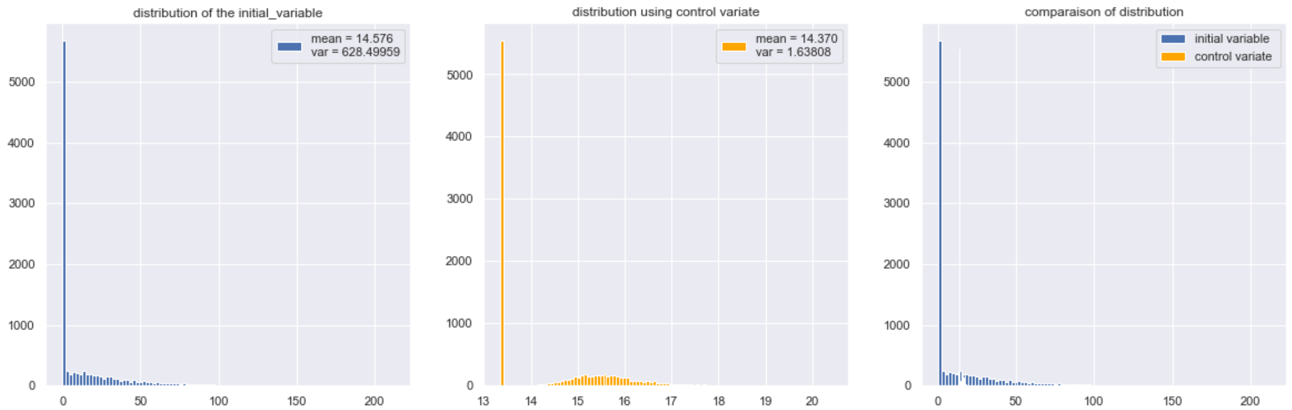

Figure 1 – Basket call option

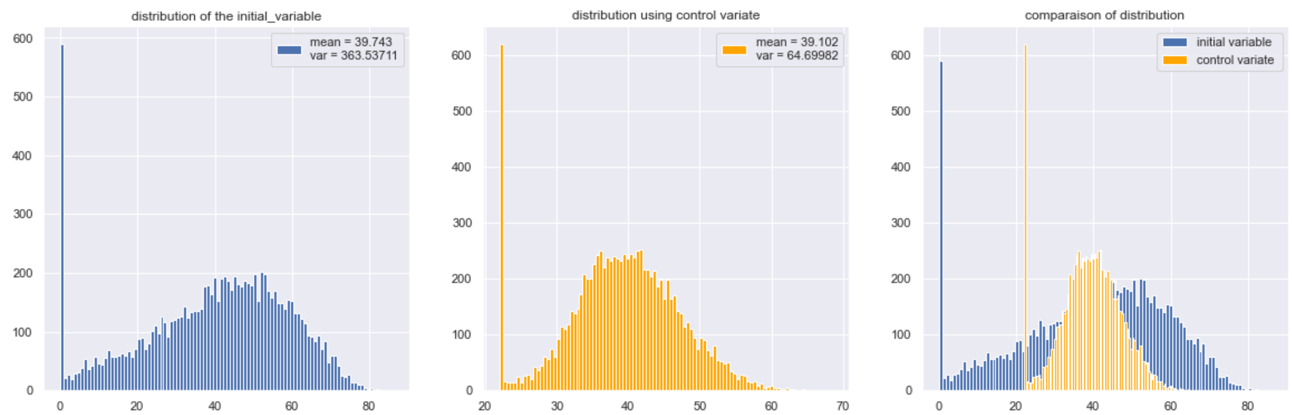

Figure 2 – Put worst of option

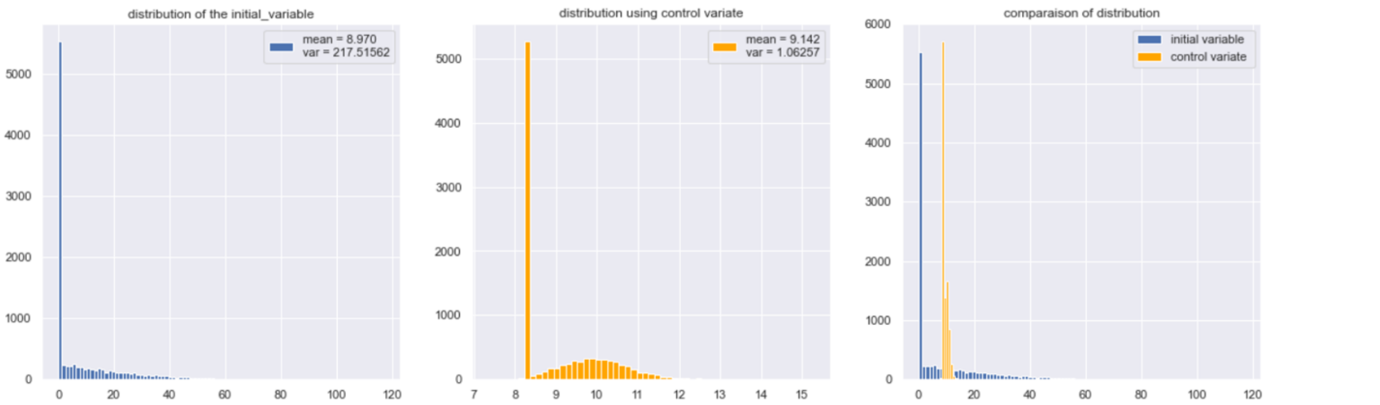

Figure 3 – Arithmetic Asian option

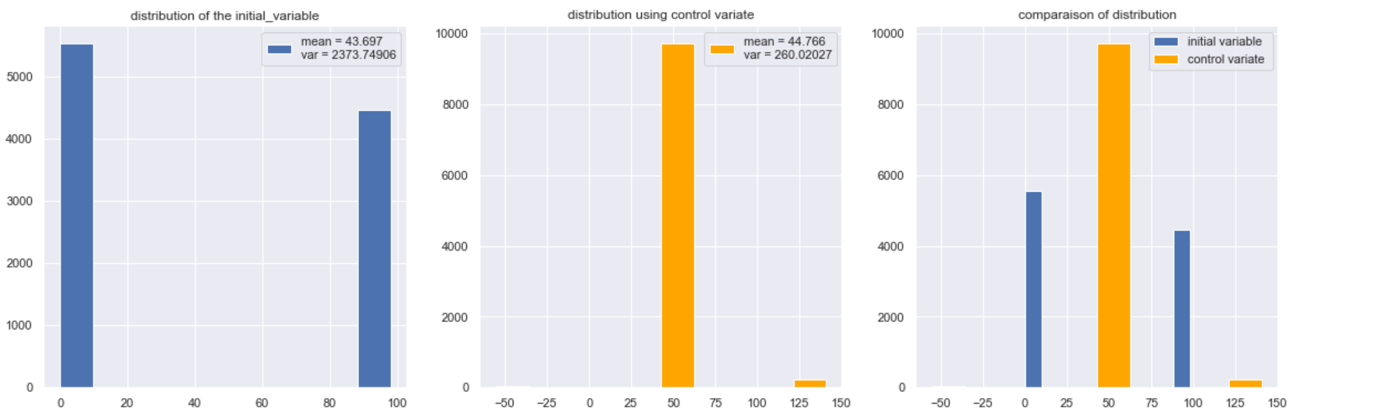

Figure 4 – Binary option

1.2 Second Approach : Automatic control variate with Numerical integration

For the second approach, the neural network has the following structure (except for the Asian option, which input contains $N_{asset} * N_{Euler}$ variables instead of N_{asset} variables) :

- Input layer : $Z = (Z_1, \dots, Z_{N_{asset}})$ independent standard normal variables, and $N_{asset} = 100$;
- First hidden layer : $n = 10$ neurons, no bias, no activation function
- Second and third hidden layer : $n' = 64$ neurons with bias and ReLu activation function.
- Output layer : scalar output

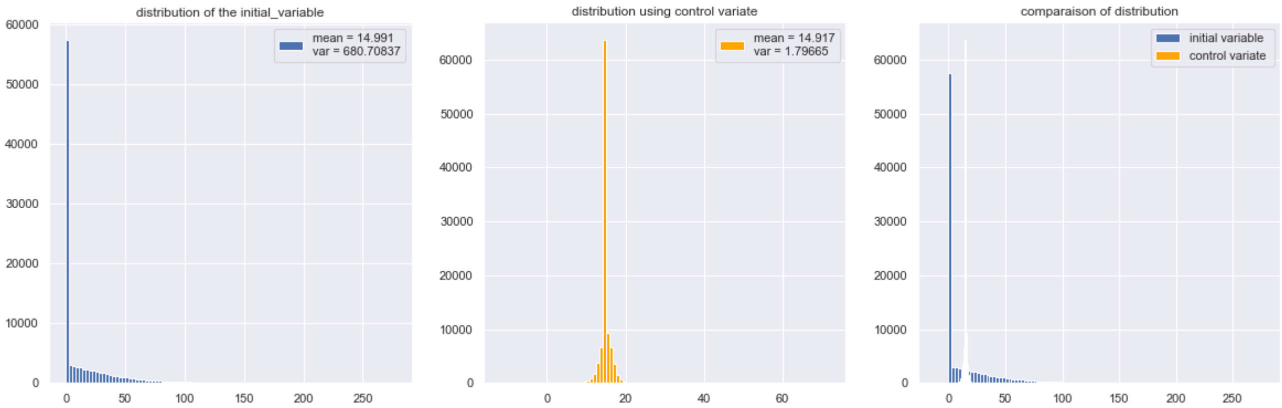

Figure 5 – Basket call option

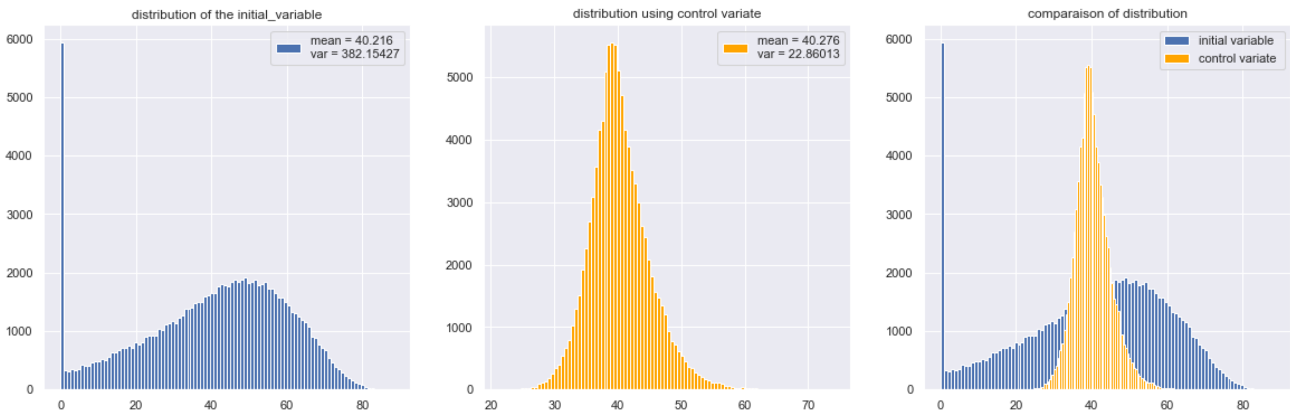

Figure 6 – Put worst of option

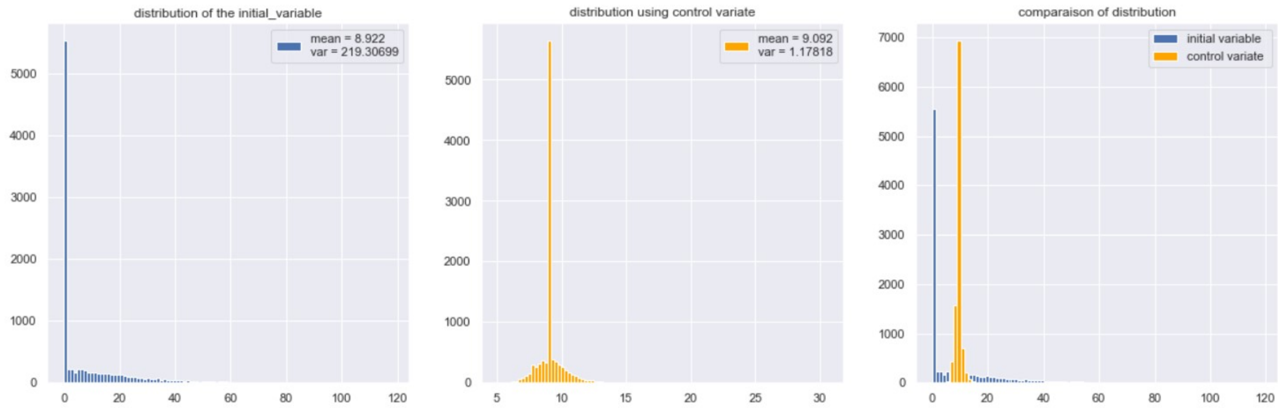

Figure 7 – Arithmetic Asian option

1.3 Third Approach : Automatic control variate with Analytic integration

For the third approach, the neural network has the following structure (except for the Asian option, which input contains $N_{asset} * N_{Euler}$ variables instead of N_{asset} variables) :

- Input layer : $Z = (Z_1, \ldots, Z_{N_{asset}})$ independent standard normal variables, and $N_{asset} = 100$;
- One hidden layer : $n' = 64$ neurons with bias andReLu activation function ;
- Output layer : scalar output with bias, no activation function.

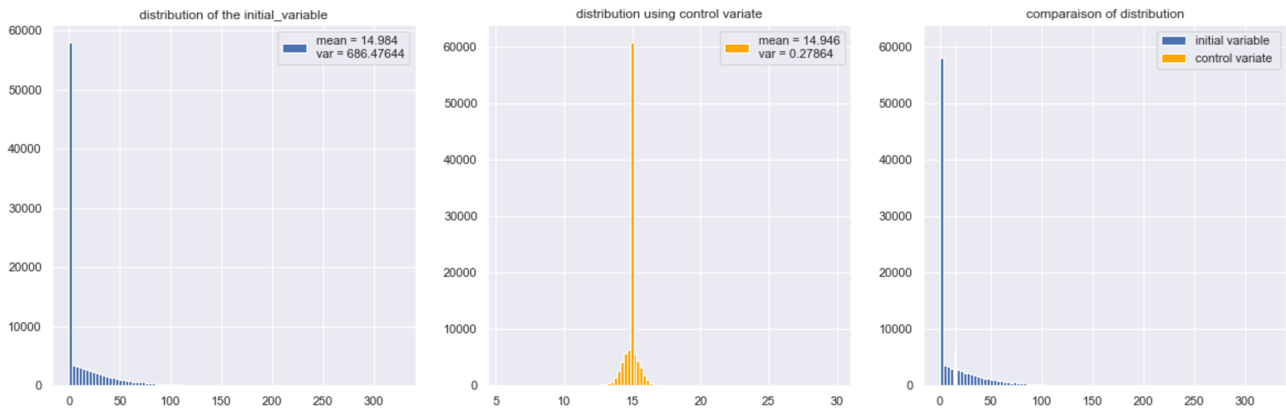

Figure 8 – Basket call option

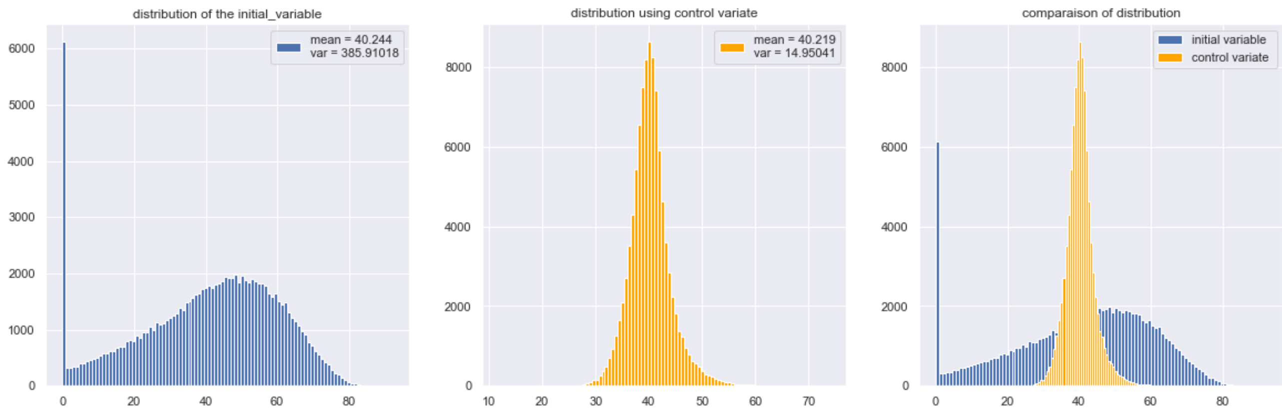

Figure 9 – Put worst of option

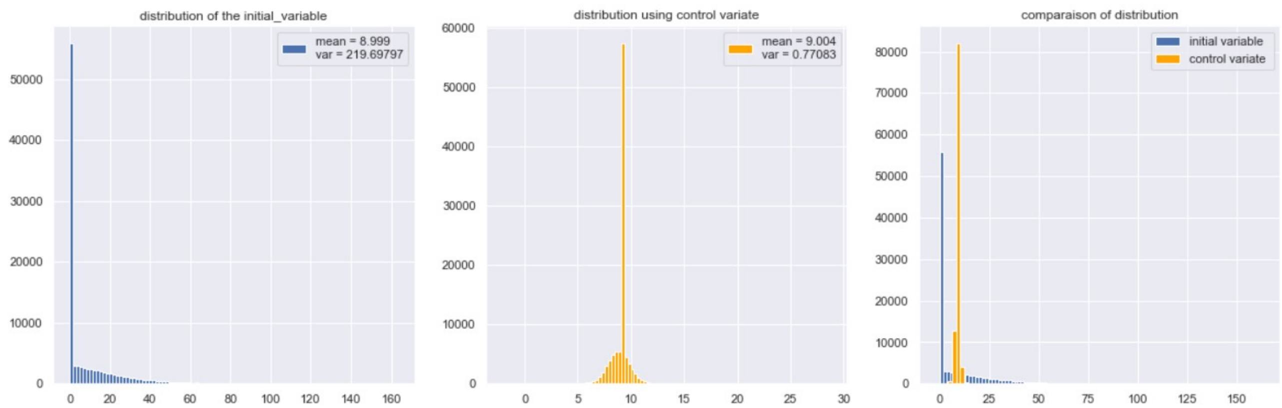

Figure 10 – Arithmetic Asian option

2
LOCAL VOLATILITY MODEL RESULTS

We only used the first approach in this part (network dimension reduction).
In this part, the neural network has the following structure :
— Input layer : $Z = (Z_1, \dots, Z_{N_{asset}})$ independent standard normal variables, and $N_{asset} * N_{Euler} = 10 * 100 = 1000$;
— First hidden layer : $n = 3$ neurons, no bias, no activation function
— Second and third hidden layer : $n = 10$ neurons with bias and ReLu activation function.
— Output layer : scalar output

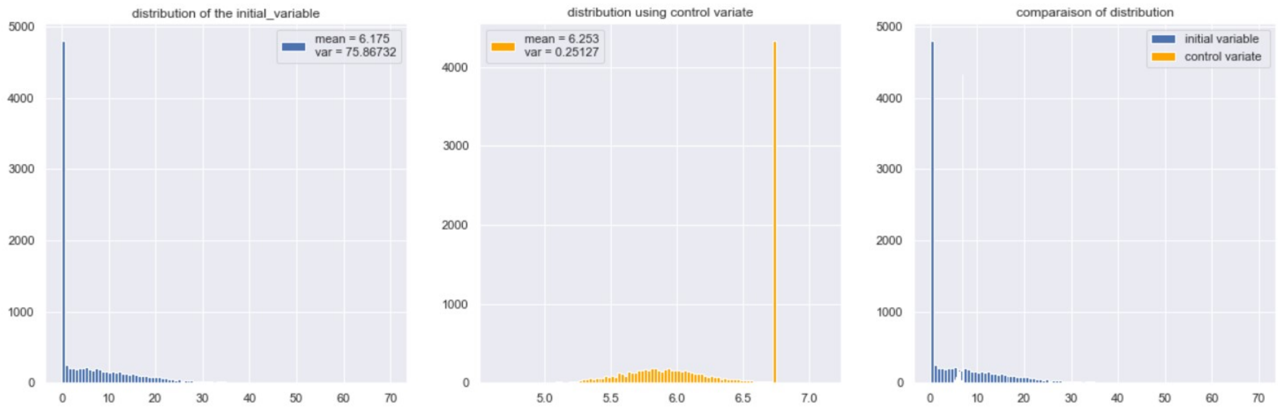

Figure 11 – Basket call option

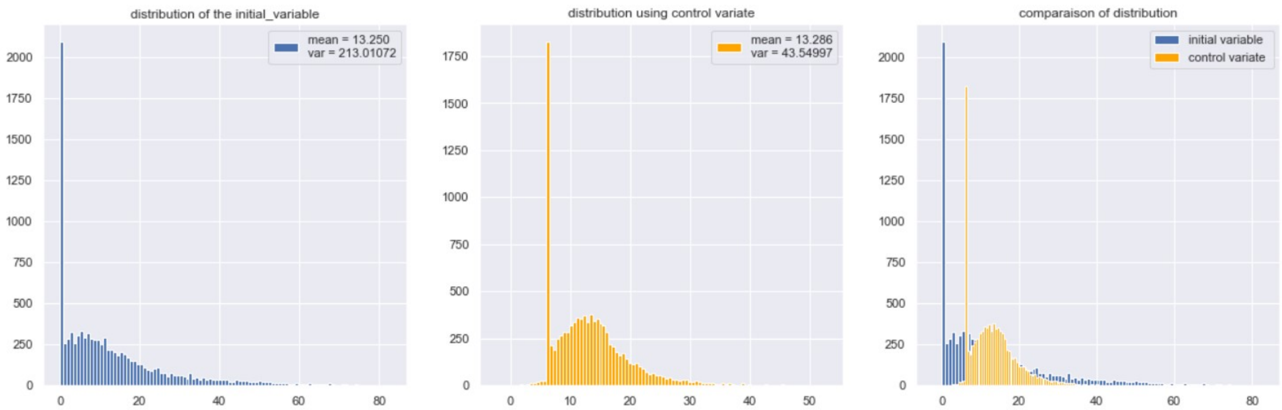

Figure 12 – Put worst of option

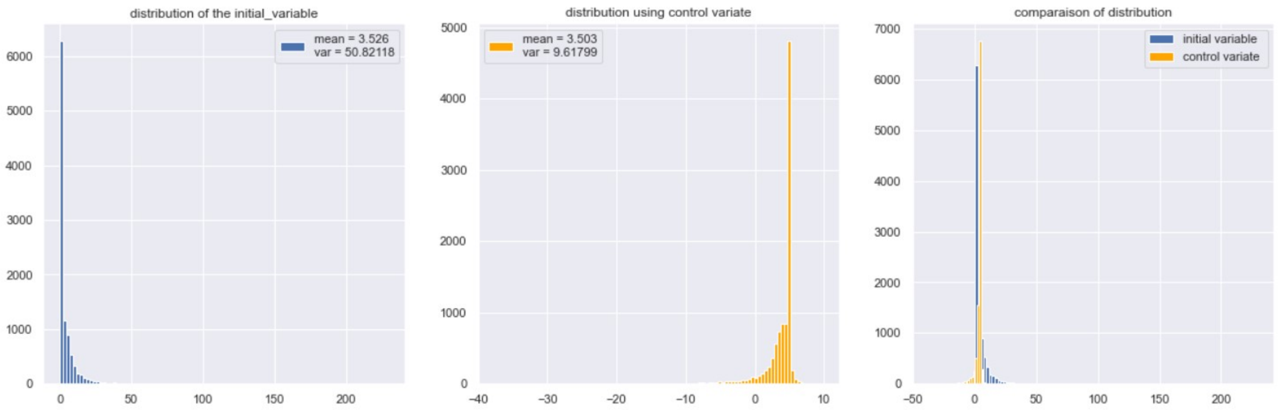

Figure 13 – Arithmetic Asian option

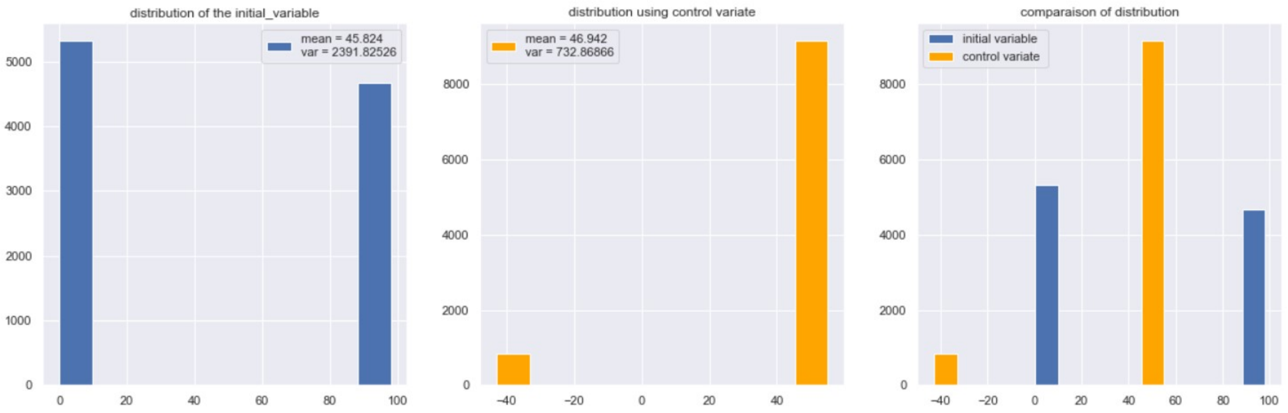

Figure 14 – Binary option

3
HESTON MODEL RESULTS

We only used the first approach in this part.
In this part, the neural network has the following structure :
- Input layer : $Z = (Z_1, \ldots, Z_{N_{asset}})$ independent standard normal variables, and $2 * N_{asset} * N_{Euler} = 2 * 10 * 100 = 2000$;
- First hidden layer : $n = 3$ neurons, no bias, no activation function
- Second and third hidden layer : $n = 10$ neurons with bias and ReLu activation function.
- Output layer : scalar output

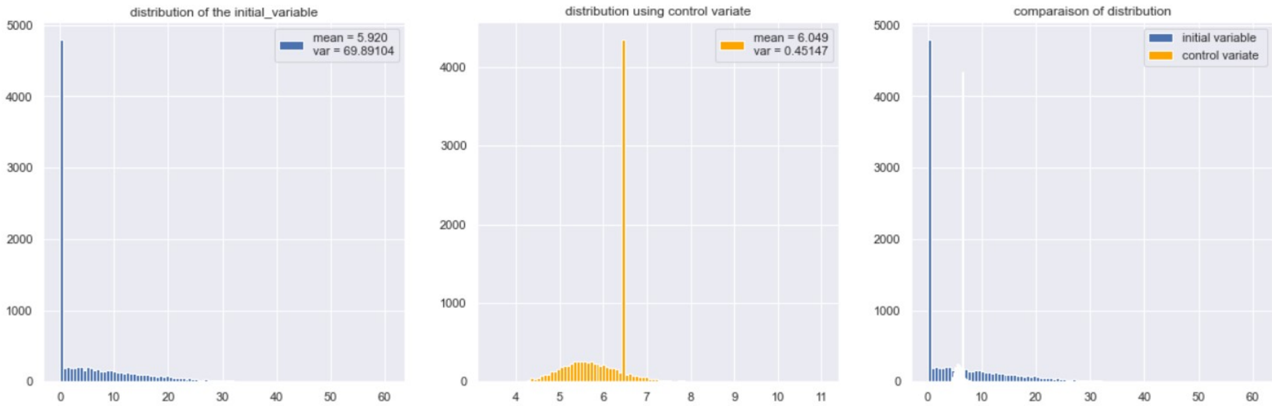

Figure 15 – Basket call option

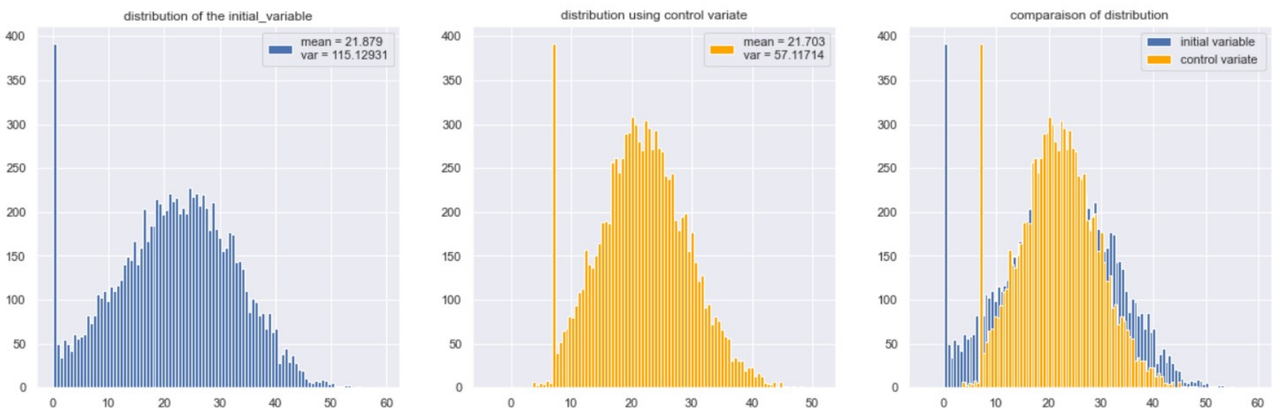

Figure 16 – Put worst of option

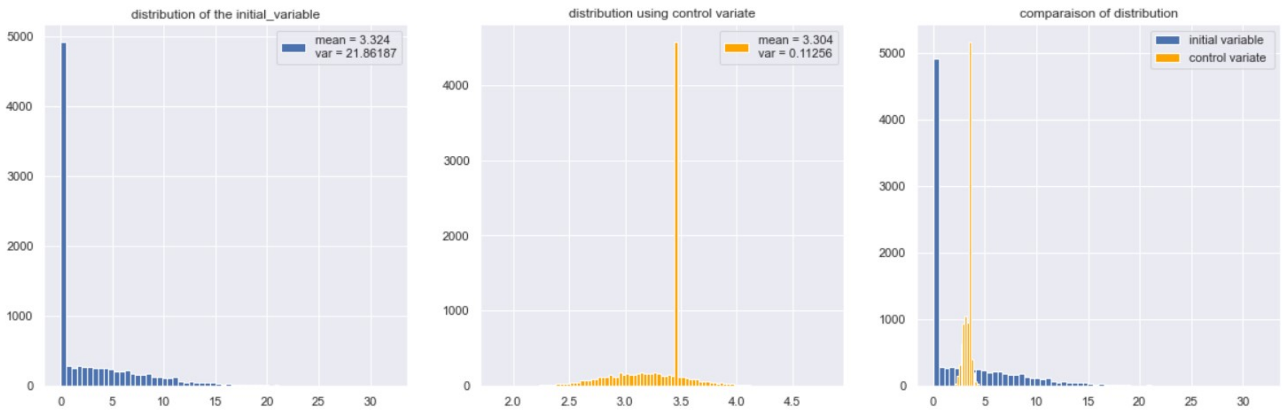

Figure 17 – Arithmetic Asian option

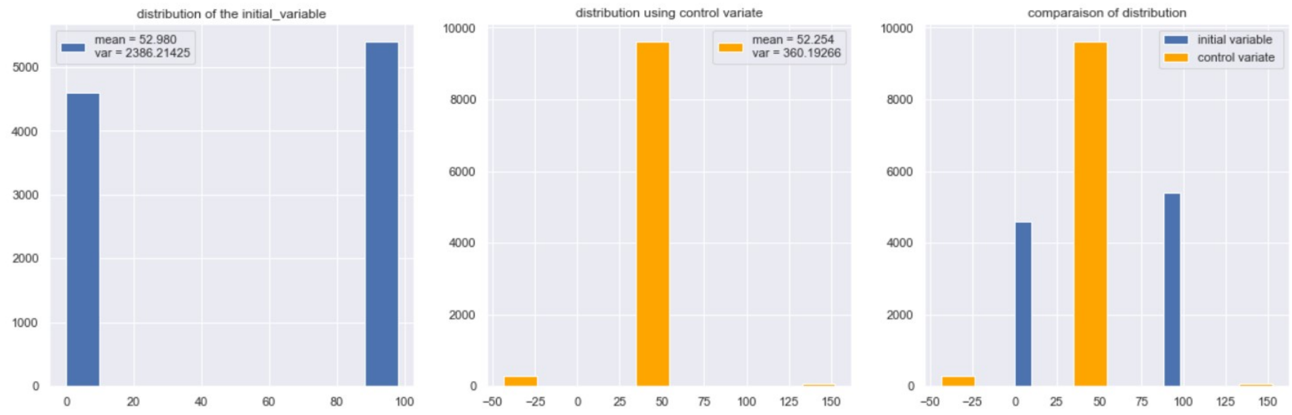

Figure 18 – Binary option

4 ROBUSTNESS

4.1 Asian option pricing robustness

This part only presents the main results about robustness. For more detail, please look at our jupyter notebooks.

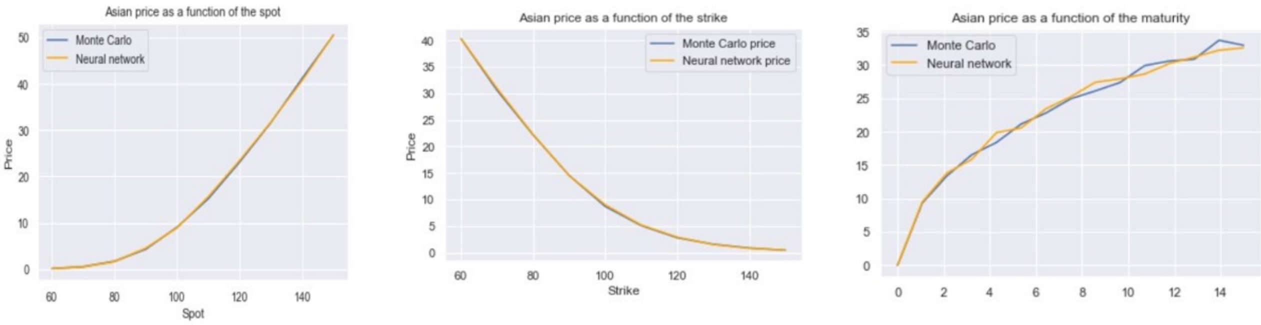

Figure 19 – Asian price as a function of the maturity, strike and spot

4.2 AutoCall pricing robustness

The AutoCall has very industrial payoff that is explained in [1].

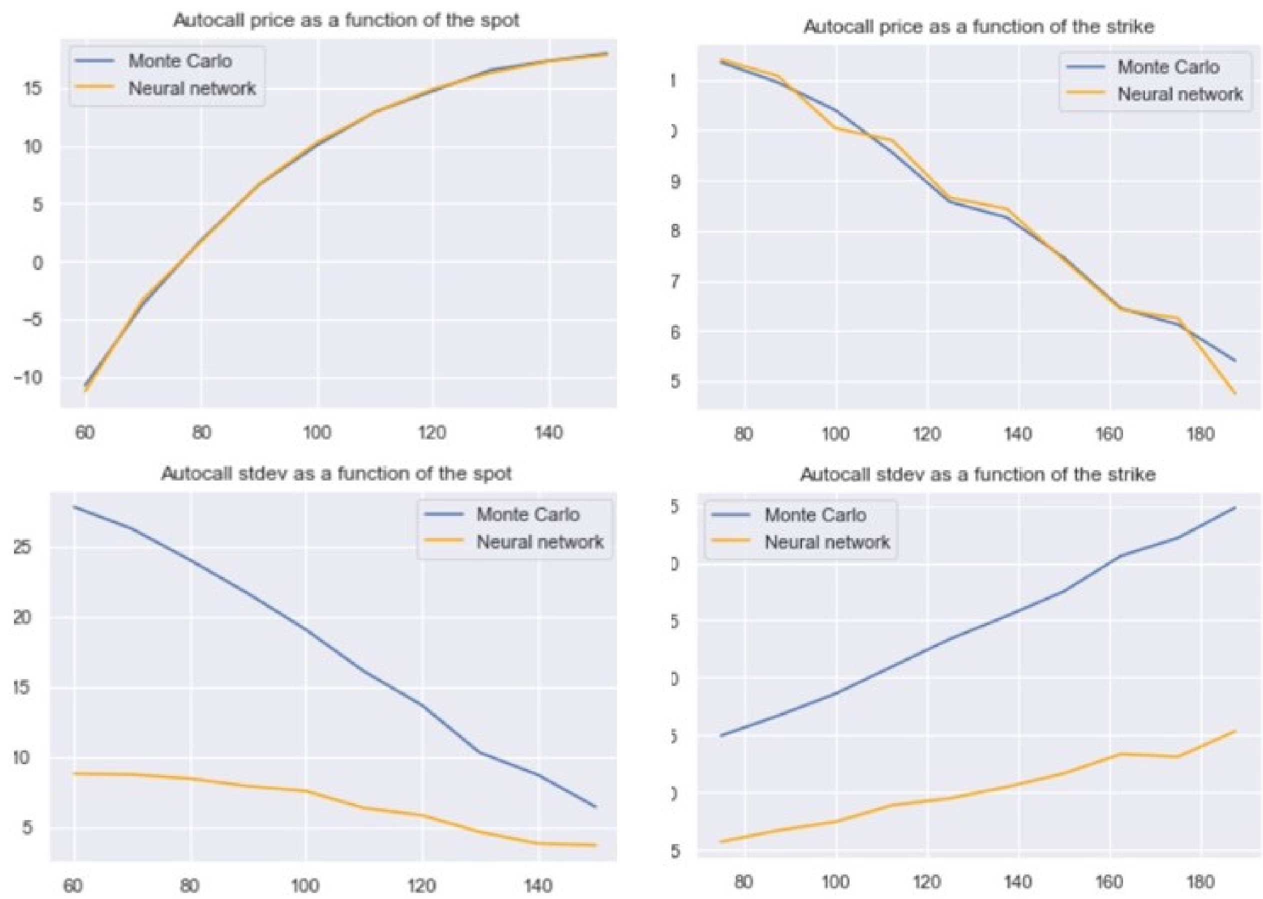

Figure 20 – AutoCall price and standard deviation as a function of the spot (left) and strike (right)

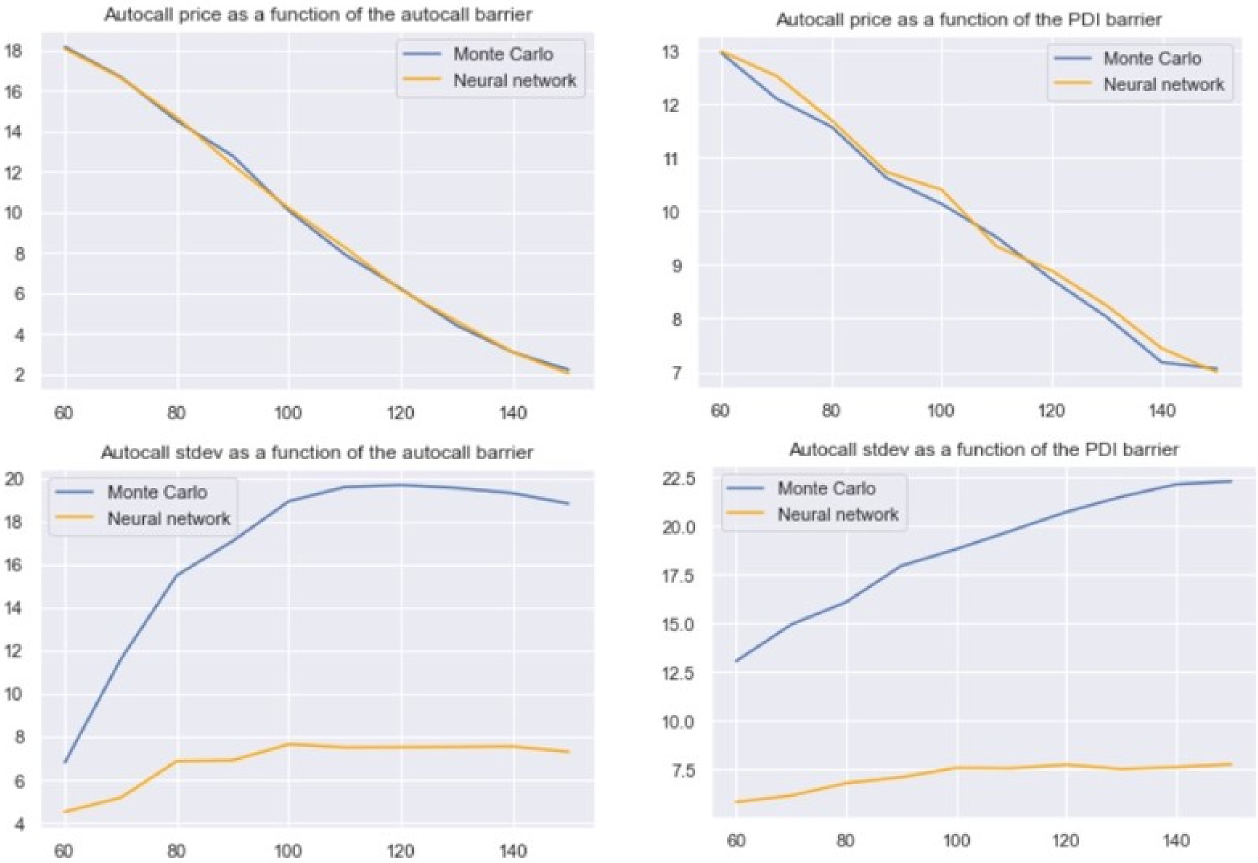

Figure 21 – AutoCall price and standard deviation as a function of the AutoCall barrier (left) and PDI barrier (right)

The interest of the dimension reduction method lies in its resistance to parameters change. In fact, this is due to the fact that we learn the directions where the variance of the payoff is concentrated rather than the actual payoff/model. The previous examples show that by learning a certain payoff given fixed parameters, the obtained network can be used to price the same payoff with different parameters.

Note that for some spots, the price of the AutoCall is negative because the payoff of the AutoCall can be negative. If the basket performance $P_t := \frac{1}{N_{asset}} \sum_{i=1}^{N_{asset}} w_i S_t^i$ doesn't reach the AutoCall barrier AB and if the final performance P_T is less than the Down and in Barrier DB, then the client pays a put on the basket with strike K, which payoff is $(K - \sum_{i=1}^{N_{asset}} w_i S_T^i)^+$.

REFERENCES

[1] Jérôme Lelong, Zineb El Filali Ech-Chafiq et Adil Reghai. "Automatic Control Variates for Option Pricing using Neural Networks". In : *Monte Carlo Methods and Applications, De Gruyter* (2021). doi : https://hal.univ-grenoble-alpes.fr/hal-02891798.