Università degli studi di Padova

Stochastic Methods for Finance

Sixth report

Monte Carlo simulation for European options pricing

Author:
Lorenzo Mancini *2019098*

Professor:
prof. Martino Grasselli

April 23, 2021

1 Goal

The purpose of this report is to simulate a Geometric Brownian Motion with VBA in order to estimate the value of European Call and Put options. The method will be extended to the case of Eotic options (Asian, Lookback). For the results, let's consider **USD** as currency.

2 Introduction

In this report we are going to provide a Monte Carlo simulation where, the stock price of and underlying is supposed to follow a Geometric Brownian Motion. Once we're able to generate one trajectory, we can perform different computations The function that generates a path is the following (the parameters are explained at the beginning of the next section).

```
Function MonteCarloTraj(S As Double, K As Double, T As Double,
r As Double, v As Double, q As Double, nsteps) As Variant
Dim dt
Dim traj() As Double
ReDim traj(nsteps)

```

```
traj(0) = S
dt = T / nsteps
For i = 1 To nsteps
    Z = Application.WorksheetFunction.NormSInv(Rnd())
    dIns = (r - q - v ^ 2 / 2) * dt + v * Z * dt ^ 0.5
    traj(i) = traj(i - 1) * Exp(dIns)
Next i
MonteCarloTraj = traj
End Function
```

Listing 1: VBA code to generate trajectory.

3 Simulation

Suppose the following paramters for the option:

- **Type = Call or Put**;
- **S = 100**, the underlying price;
- **K = 100**, the strike price (At The Money);
- **T = 1 year**, the time to maturity,
- **r = 0.001**, the free-risk interest rate;
- σ **= 0.2**, the annual volatility;
- **nsteps**, the number of steps in which the path is divided.

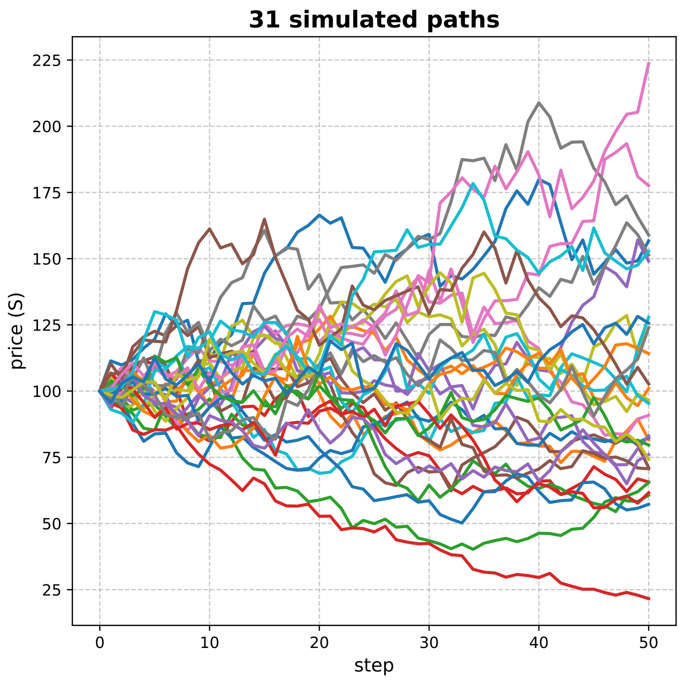

Figure 1: Simulation of 40 different dynamical trajectories.

3.1 Vanilla Options

The price of a "Vanilla" Option is given by the corresponding payoff of the final value of a trajectory, discounted with the factor e^{-rT}. An average over a large number of trajectories will give a reasonable price.

```
...
If style = "Vanilla" Then
    final_value = traj(nsteps)
    If OptType = "C" Then
        prices(i) = Application.WorksheetFunction.Max(
        (final_value - K), 0) * Exp(-r * T)
    ElseIf OptType = "P" Then
        prices(i) = Application.WorksheetFunction.Max(
        (K - final_value), 0) * Exp(-r * T)
    End If
...
```

Listing 2: VBA code to price Vanilla Options.

The procedure is done for the dynamic case (50 `nsteps`) and the static one (just one step). Furthermore, in order to see the behaviour of the algorithm as function of the number of trajectories, we repeat the computation for different `npaths`. The results are compared with the price obtained by the Black-Scholes formula (see figures).

$$\textbf{B-S(Call)} = \mathbf{8.01} \qquad \textbf{B-S(Put)} = \mathbf{7.91}$$

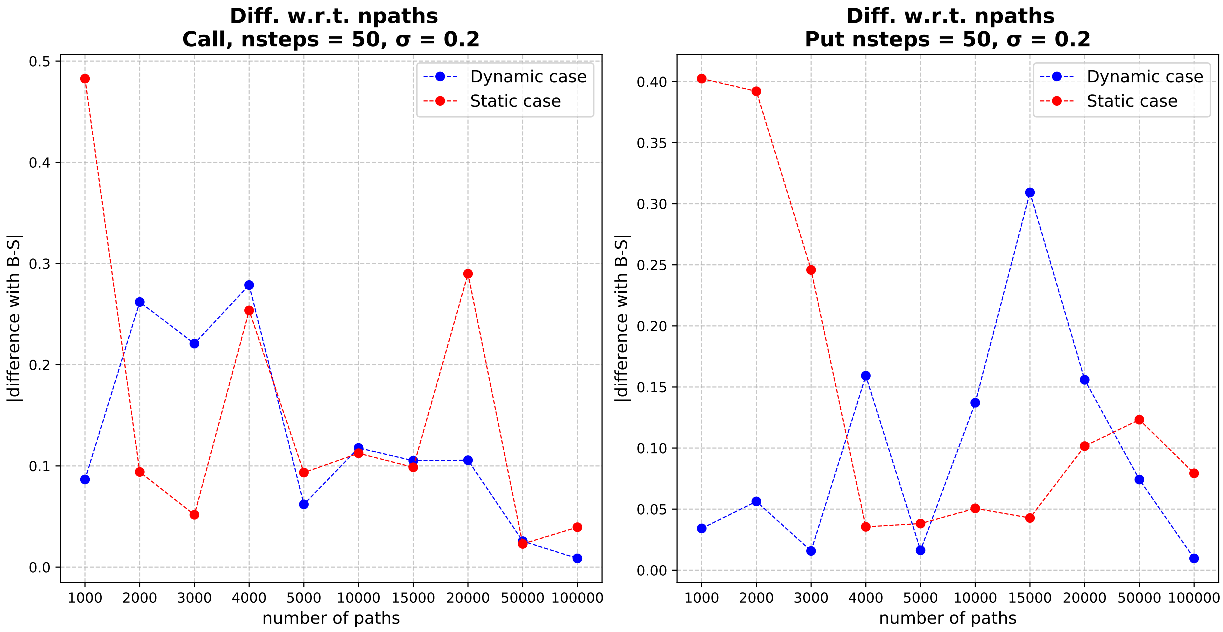

Figure 2: Absolute value of the difference between the prices obtained with the simulation (`nsteps = 50`) for different `npaths` and the Black-Scholes price.

Now, it can be useful to fix the number of trajectories and do the computation for different `nsteps`. For this new simulation we consider a new volatility $\sigma_{\text{new}} = 0.4$. The B-S prices now are:

$$\textbf{B-S(Call)} = \mathbf{15.89} \qquad \textbf{B-S(Put)} = \mathbf{15.79}.$$

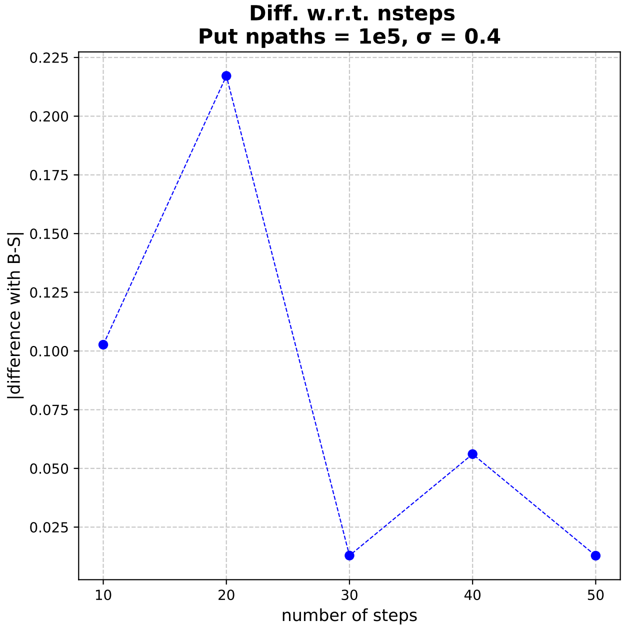

Figure 3: Absolute value of the difference between the prices obtained with the simulation (`npaths` = 10^5) for different `npaths` and the B-S price.

Also in this case we can see that the difference decreases as the `nsteps` increase.

3.2 Asian Options

In the case of Asian options, instead of considering the final values of the trajectories, we put in the pricing formula the arithmetic average of the price during the whole trajectory.

```
...
ElseIf style = "Asian" Then
        Z_t = Application.WorksheetFunction.Average(traj)
        If OptType = "C" Then
            prices(i) = Application.WorksheetFunction.Max(
            (Z_t - K), 0) * Exp(-r * T)
        ElseIf OptType = "P" Then
            prices(i) = Application.WorksheetFunction.Max(
            (K - Z_t), 0) * Exp(-r * T)
        End If
...
```

Listing 3: VBA code to price Asian Options.

With 10^5 trajectories, we obtain the following results:

strike	nsteps	npaths	Call	Put
100	50	10^5	4.62	4.54
110	50	10^5	1.41	11.34

Table 1: Prices of Asian options with different strikes.

In order to check the correct behaviour of the algorithm we can observe from the table above that, increasing the strike, the Call price decreases and the Put one becomes bigger, as expected from theory. This is just a simple way to see if it works and it is not rigorous.

3.3 Lookback Options

In the case of Lookback options the payoff of a Call is given by the difference between the final value of the trajectory and the minimum value achieved during the whole path.

```
...
ElseIf style = "Lookback" Then
        Minimum = Application.WorksheetFunction.Min(traj)
        final_value = traj(nsteps)
         If OptType = "C" Then
            prices(i) = Application.WorksheetFunction.Max(
            (Maximum - K), 0) * Exp(-r * T)
        End If
...
```

Listing 4: VBA code to price Lookback options.

Now, the comparison is done for different volatilities:

σ	**nsteps**	**npaths**	**Call**
0.2	50	10^5	13.71
0.4	50	10^5	26.03

Table 2: Prices of Lookback options with different strikes.

4 Conclusion

In this report we managed to provide a Monte Carlo simulation with the aim of pricing European options. As we showed, we obtained very reasonable results compared with the one given by the Black-Scholes formula. Nevertheless, the computations can be improved by increasing the number of trajectories generated (this should be $\sim 10^6$). The discussion has been extended to the case of Exotic options where however there was no benchmark.