**Name:** Elena Zoppellari
**MSc:** Physics of Data
**Student ID:** 2076715

**Course:** Stochastic Methods for Finance
**Professor:** Martino Grasselli
**Date:** May 12, 2023

# Monte Carlo Simulation of Traditional and Exotic Options

## 1 Introduction

The objective of this report is to explore the performance of Monte Carlo methods in calculating option prices for a range of options, including Vanilla, Asian, Lookback and Barrier Options. Section 2 provides a brief introduction to the Monte Carlo Option model, including an overview of the Euler scheme for Vanilla and path-dependent Exotic Options. In Section 3, we present the methodology employed and the results obtained from our analysis. The report concludes with a discussion of the findings in Section 4. The Appendix includes VBA scripts used to simulate all the results presented in this report.

## 2 Monte Carlo Option Model

Monte Carlo methods consist of simulating $M$ trajectories, each evolving in $N$ steps from $t = 0$ to $t = T$ (the maturity time), through a Geometric Brownian Motion (GBM). The payoffs of an option are then evaluated for each trajectory, and the expected call and put prices are calculated based on the average payoffs of the $M$ trajectories. This method is particularly useful for path-dependent options, such as Asian Options, Lookback Options and Barrier Options, because these options depend on the path of the underlying asset over the $N$ time steps. While the Monte Carlo simulation method is typically used for path-dependent options, it can also be extended to Vanilla Options using the Euler scheme. However, it is important to note that this may not be the most efficient solution for Vanilla Options because they are not time-dependent, and the payoffs are only determined by the final price of the underlying asset, $S_T$, which is simulated at the maturity date.

### 2.1 Euler Scheme for Vanilla Options

The Geometric Brownian Motion (GBM) dynamic is described by the following differential equation:

$$dS = \mu S dt + \sigma S dz, \tag{1}$$

where $dz$ is a Wiener process, $\mu$ is the expected return in a risk-neutral world and $\sigma$ is the volatility. This differential equation follows an Itô process, therefore its lemma can be applied, which is defined as:

$$dG = \left( \frac{\partial G}{\partial S} \mu S + \frac{1}{2} \frac{\partial^2 G}{\partial S^2} \sigma^2 S^2 + \frac{\partial G}{\partial t} \right) + \frac{\partial G}{\partial S} \sigma S dz \tag{2}$$

In order to simplify the computation, we consider $G = \ln S$ rather than $G = S$:

$$\frac{\partial G}{\partial S} = \frac{1}{S}, \qquad \frac{\partial^2 G}{\partial S^2} = -\frac{1}{S^2}, \qquad \frac{\partial G}{\partial t} = 0, \tag{3}$$

therefore:

$$d\left(lnS\right) = \left(\mu - \frac{\sigma^2}{2}\right) dt + \sigma dz \tag{4}$$

In order to evaluate the value of $lnS$ in a discrete time interval $\Delta t$, the above equation can be approximated into:

$$lnS(t + \Delta t) - lnS(t) = \left(\mu - \frac{\sigma^2}{2}\right) \Delta t + \sigma \epsilon \sqrt{\Delta t}, \tag{5}$$

where $\epsilon$ is a random sample from a normal distribution with mean zero and standard deviation of 1.0. Equivalently, the above equation can be rewritten as:

$$S(t + \Delta t) = S(t) \cdot e^{\left(\mu - \frac{\sigma^2}{2}\right)\Delta t + \sigma \epsilon \sqrt{\Delta t}} \tag{6}$$

Therefore, for Vanilla Options whose payoffs are:

$$p(call) = (S_T - K)^+, \qquad p(put) = (K - S_T)^+, \tag{7}$$

the value of the underlying asset at maturity time $S_T$ is given by the final value of equation 6 after $N + 1$ time steps.

## 2.2 Asian Options

Asian Options are a type of exotic option that derive their payoff based on the average price of the underlying asset over a specific time period $S_{avg}$, rather than the price of the asset at a specific point in time. There are two main types of Asian Options: Fixed Strike Asian Options and Floating Strike Asian Options.

### 2.2.1 Fixed Strike Asian Options

In this case the strike price is predetermined and remains fixed throughout the life of the option. The prices for call and put options are:

$$p(call) = (S_{avg} - K)^+, \qquad p(put) = (K - S_{avg})^+, \tag{8}$$

where $S_{avg} = \frac{1}{N}\int_0^T S_t dt$ for continous time and $S_{avg} = \frac{1}{N}\sum_t S_t$ for discrete time steps.

### 2.2.2 Floating Strike Asian Options

In contrast to Fixed Strike Options, Floating Strike ones have a strike price that is calculated based on the average price of the underlying asset over the life of the option. This means that the strike price of a floating strike Asian Option can fluctuate over time, depending on the performance of the underlying asset, therefore the resulting prices for call and puts are:

$$p(call) = (S_T - \frac{k}{N}\int_0^T S_t dt)^+, \qquad p(put) = (\frac{k}{N}\int_0^T S_t dt - S_T)^+, \tag{9}$$

where $k$ is a weight and $S_T$ is the value of the underlying asset at maturity.

| $K$ | $\mu$ | $\sigma$ | $S_0$ | $T$ | $dt$ |
|---|---|---|---|---|---|
| 99 | 0.01 | 0.2 | 100 | 1 | 1 working day |

Table 1: Reference values used for the analysis

### 2.3 Lookback Options

Lookback Options are a type of exotic option that allow the holder to buy or sell an underlying asset at its lowest or highest price over a specified period of time. Unlike traditional options that are based on the price of the underlying asset at a specific point in time, Lookback Options give the holder the benefit of hindsight by providing the ability to exercise the option based on the most favorable price over the life of the option. They are based on the lowest or highest price of the underlying asset over the life of the option, in particular call and put prices are:

$$p(call) = (S_T - S_{min})^+, \qquad p(put) = (S_{max} - S_T)^+, \tag{10}$$

### 2.4 Barrier Options

Barrier options are options that have a payout that is dependent not only on the terminal stock price, but also on whether the stock attains some "barrier" during the life of the option. There are two general kinds of Barrier Options: Knock-out and Knock-in options. For both kinds of options, the payouts are calculated for each path and then averaged over all the $M$ simulated paths.

#### 2.4.1 Knock-out Barrier Options

With this particular kind of option, the payouts are set equal to zero if the underlying asset at a certain time step $t$ reaches the barrier. This barrier could be an up or down barrier, resulting in $p(call)$ and $p(put)$ being set to zero if the barrier is reached. Otherwise, the payout is given by $p(call) = (S_T - K)^+$ and $p(put) = (K - S_T)^+$, as in Vanilla Options. When the barrier is only up we refer to the option as Up-and-Out, while when it is only down it is called Down-and-Out.

#### 2.4.2 Knock-in Barrier Options

This kind of option is the opposite of the Knock-out Barrier Option: in this case, the option price is zero until the underlying asset reaches the barrier at some $t$ value. Similary as the Knock-out options, when the barrier is only up the option is called Up-and-In, while when it is only down its name is Down-and-In.

## 3 Methodology and Results

As explained in Section 2, in order to evalue the payoffs of Options at maturity time, $M$ GBM trajectories have been simulated according to equation 6 through a VBA script 1 reported in the Appendix. The fixed parameter used are reported in Table 1, in particular the time window $[0, T]$ with $T = 1$ year has been discretized into $N = 252$ working days. In Figure 1 $M = 100$ paths generated using the reference values can be observed.

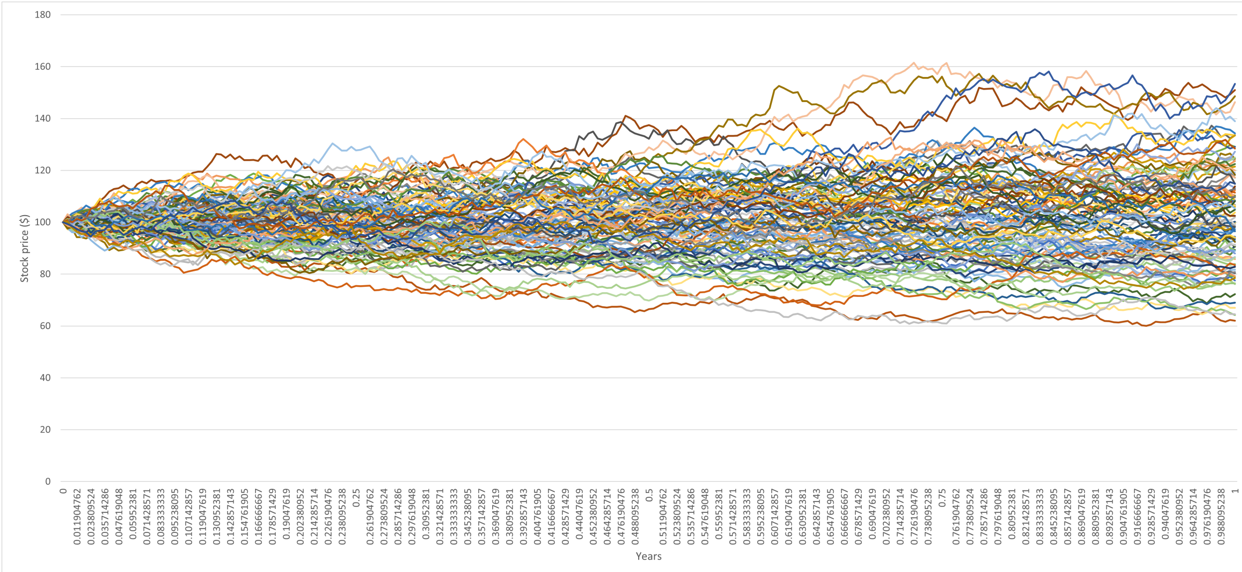


Figure 1: 100 simulated paths with parameters reported in Table 1

## 3.1 Traditional Options

To begin, the Monte Carlo method was applied to Vanilla Options with a time step of $dt = 1$ year, representing no further discretization between $t = 0$ and $t = T$. The payoffs were calculated using the script 2 in the Appendix, with $N = 1$ (number of time steps) and varying numbers of simulated paths.
Next, the number of time steps was increased to $N = 252$ (discretization of 1 day). The results of the two simulations, with varying numbers of paths, are presented in the following plots, along with a comparison to the Black-Scholes model used as a reference value. The BS values have been calculated using script 3 in the Appendix. Percentage errors were calculated separately for calls and puts.

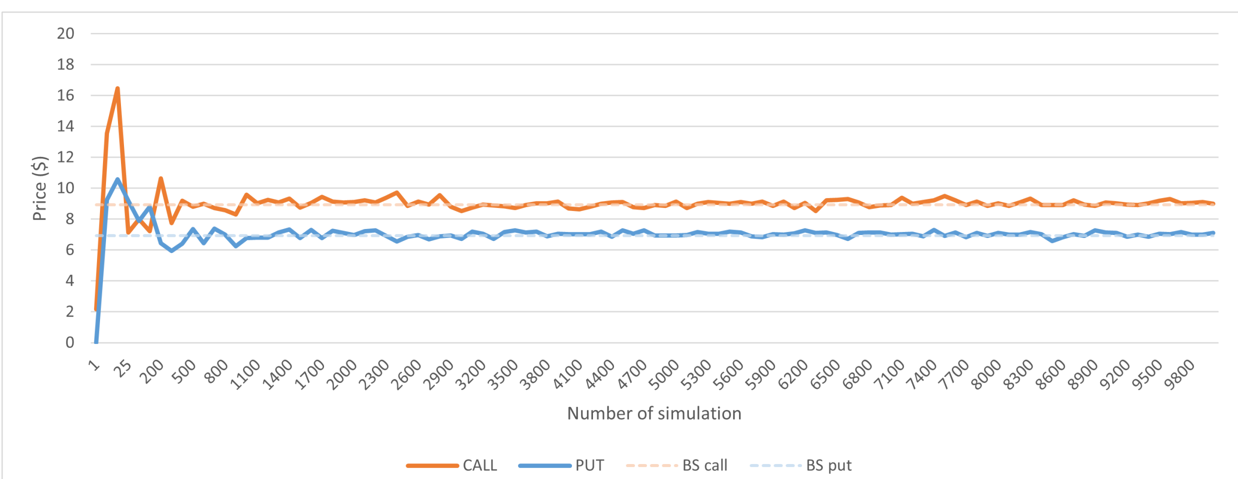


Figure 2: One-Step Vanilla Option using an increasing number of Monte Carlo simulations

| $p(call)$ (\$) | $p(put)$(\$) |
|---|---|
| 8.918 | 6.933 |

Table 2: Payoffs using Black-Scholes Model

| | $p(call)$ (\$) | $p(put)$(\$) |
|---|---|---|
| One-Step | 8.983 | 7.097 |
| Multi-Step | 8.984 | 7.082 |

Table 3: Payoffs of Vanilla Options for $M = 10k$ simulated paths

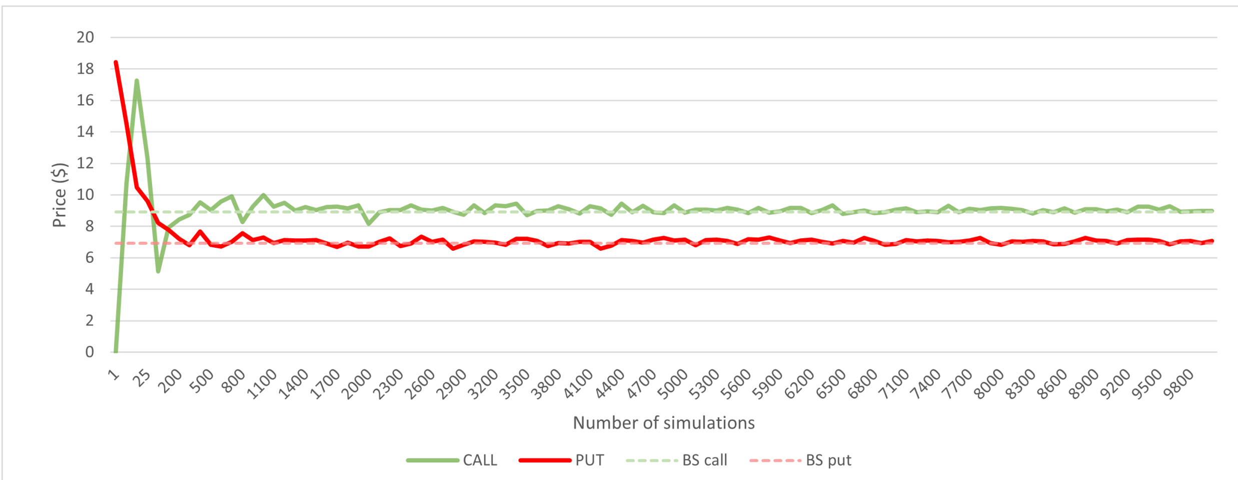


Figure 3: Multi-Step Vanilla Option using an increasing number of Monte Carlo simulations

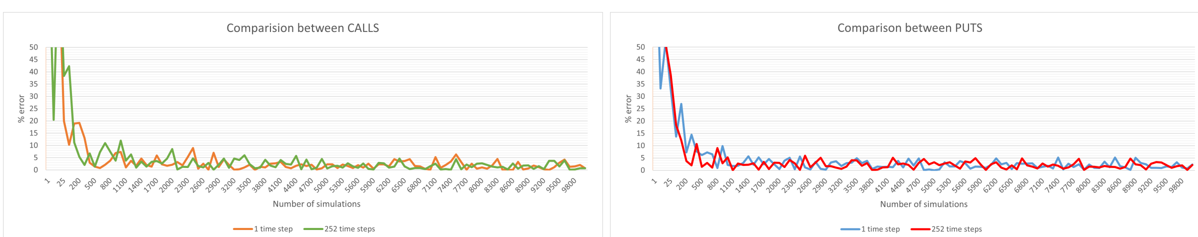


Figure 4: Percentage errors associated to Calls and Puts for different time steps and paths number

Figures 2 and 3 demonstrate that both models converge to the Black-Scholes reference values when using $M > 500$ simulated paths. Nonetheless, some oscillations for both models can make the resulting values in both cases less precise. This is because only the last price of the simulated path is considered and information about the complete evolution of the path is not used. The payoffs for $M = 10k$ simulated paths are reported in Table 3.

| | $p(call)$ (\$) | $p(put)$(\$) |
|---|---|---|
| Asian Fixed Strike | 5.400 | 4.002 |
| Asian Floating Strike | 4.370 | 4.853 |
| Lookback | 14.992 | 15.711 |
| Barrier Down-and-out | 9.200 | 1.854 |
| Barrier Up-and-out | 1.392 | 6.776 |
| Barrier Knock-Out Both | 1.338 | 1.732 |
| Barrier Up-and-in | 0.076 | 5.072 |
| Barrier Down-and-in | 7.562 | 0.191 |
| Barrier Knock-In Both | 7.539 | 5.351 |

Table 4: Payoffs of Exotic Options

### 3.2 Exotic Options

The payoffs of the Exotic Options described in Section 2 are presented in Table 4. For each of the options, $N = 252$ and $M = 10k$ have been used as simulation parameters. In particular, for Barrier Options, the up and down barrier values were set to $Bar_{UP} = 120$ and $Bar_{DOWN} = 80$, respectively, based on the obseration of Figure 1, in which most of trajectories stay between these limits. The codes used to compute the payoffs are provided in the Appendix as script 4, 5, and 6.

## 4 Conclusions

In this report, we have applied Monte Carlo methods to calculate the payoffs of both traditional (vanilla, independent-time) and exotic time-dependent options. The results for Vanilla Options indicate that both the One-step and Multi-step approaches yield similar results, converging to the Black-Scholes values when more than 500 paths are simulated. However, the One-step approach is more computationally efficient.
Overall, it can be concluded that the Monte Carlo Option model is particularly suitable for time-dependent option models such as Asian Options, Lookback Options and Barrier Options.

## 5 Appendix

```
Function GBM(S, mu, T, sigma, n, sim) As Variant
    dt = T / n ’ time step
    Dim Scoll() As Double
    ReDim Scoll(1 To (n + 1), 1 To sim) ’dimension = n. time steps x n
       . simulations
    For j = 1 To sim
        Scoll(1, j) = S
        For i = 2 To (n + 1)
            Scoll(i, j) = Scoll(i - 1, j) * Exp((mu -
               WorksheetFunction.Power(sigma, 2) / 2) * dt + sigma *
```

```
               WorksheetFunction.NormSInv(Rnd) * Sqr(dt))
        Next i
    Next j
    GBM = Scoll 'return the array
End Function
```

Listing 1: Geometric Brownian Motion

```
Function Callop(S, mu, T, sigma, n, sim, K) As Variant
    BM = GBM(S, mu, T, sigma, n, sim)
    Dim ST As Double
    ST = 0 'we will use this variable to compute the mean price of the
       "sim" generated paths
    For j = 1 To sim
        ST = ST + WorksheetFunction.Max(0, BM(n + 1, j) - K)
    Next j
    Callop = ST / sim
End Function

Function Putop(S, mu, T, sigma, n, sim, K) As Variant
    BM = GBM(S, mu, T, sigma, n, sim)
    Dim ST As Double
    ST = 0 'we will use this variable to compute the mean price of the
       "sim" generated paths
    For j = 1 To sim
        ST = ST + WorksheetFunction.Max(0, K - BM(n + 1, j))
    Next j
    Putop = ST / sim
End Function
```

Listing 2: Call and Put function for Vanilla Options

```
'd1 for BS formula
Function d1(S, K, T, r, q, sigma)
    d1 = (Log(S / K) + (r - q + sigma   2 / 2) * T) / (sigma * Sqr(T))
End Function

'd2 for BS formula
Function d2(S, K, T, r, q, sigma)
    d2 = d1(S, K, T, r, q, sigma) - sigma * Sqr(T)
End Function

'Black-Scholes Model
Function BS(S, K, T, r, q, sigma)
    Dim BScoll() As Double
    ReDim BScoll(1 To 2, 1 To 1) 'dimension = 1x2, in order to include
        both call and put
    BScall = S * WorksheetFunction.NormSDist(d1(S, K, T, r, q,
       sigma), True) - K * Exp(-r * T) * WorksheetFunction.NormSDist(
```

```
        d2(S, K, T, r, q, sigma), True)
    BSput = BScall + K * Exp(-r * T) - S
    BScoll(1, 1) = BScall
    BScoll(2, 1) = BSput
    BS = BScoll
End Function
```

Listing 3: Black-Scholes model

```
'Asian Fixed Strike Options
Function AsiaFX(S, mu, T, sigma, n, sim, K)
    BM = GBM(S, mu, T, sigma, n, sim)
    Dim Asia() As Double
    ReDim Asia(1 To 1, 1 To 2) 'dimension = 1x2, in order to include
        both call and put
    For j = 1 To sim
        Dim ST As Double
        ST = 0 'we will use this variable to compute the mean St value
            in n time steps
        For i = 1 To (n + 1)
            ST = ST + BM(i, j)
        Next i
        Savg = ST / (n + 1)
        CallFS = WorksheetFunction.Max(0, Savg - K)
        PutFS = WorksheetFunction.Max(0, K - Savg)
        Asia(1, 1) = Asia(1, 1) + CallFS
        Asia(1, 2) = Asia(1, 2) + PutFS
    Next j
    Asia(1, 1) = Asia(1, 1) / sim
    Asia(1, 2) = Asia(1, 2) / sim
    AsiaFX = Asia
End Function

'Asian Floating Strike Options
Function AsiaFL(S, mu, T, sigma, n, sim)
    BM = GBM(S, mu, T, sigma, n, sim)
    Dim Asia() As Double
    ReDim Asia(1 To 1, 1 To 2) 'dimension = 1x2, in order to include
        both call and put
    For j = 1 To sim
        Dim ST As Double
        ST = 0 'we will use this variable to compute the mean St value
            in n time steps
        For i = 1 To (n + 1)
            ST = ST + BM(i, j)
        Next i
        Savg = ST / (n + 1)
        CallFS = WorksheetFunction.Max(0, Savg - BM(n + 1, j))
```

```
        PutFS = WorksheetFunction.Max(0, BM(n + 1, j) - Savg)
        Asia(1, 1) = Asia(1, 1) + CallFS
        Asia(1, 2) = Asia(1, 2) + PutFS
    Next j
    Asia(1, 1) = Asia(1, 1) / sim
    Asia(1, 2) = Asia(1, 2) / sim
    AsiaFL = Asia
End Function
```

Listing 4: Asian Options

```
'Lookback Options
Function Lookback(S, mu, T, sigma, n, sim)
    BM = GBM(S, mu, T, sigma, n, sim)
    Dim LB() As Double
    ReDim LB(1 To 1, 1 To 2) 'dimension = 1x2, in order to include both
        call and put
    For j = 1 To sim
        Dim Smin As Double
        Dim Smax As Double
        Smin = S 'we will use this variable to collect the min value
        Smax = S 'we will use this variable to collect the max value
        For i = 1 To (n + 1)
            If BM(i, j) ¡ Smin Then Smin = BM(i, j)
            If BM(i, j) ¿ Smax Then Smax = BM(i, j)
        Next i
        CallLB = WorksheetFunction.Max(0, BM(n + 1, j) - Smin)
        PutLB = WorksheetFunction.Max(0, Smax - BM(n + 1, j))
        LB(1, 1) = LB(1, 1) + CallLB
        LB(1, 2) = LB(1, 2) + PutLB
    Next j
    LB(1, 1) = LB(1, 1) / sim
    LB(1, 2) = LB(1, 2) / sim
    Lookback = LB
End Function
```

Listing 5: Lookback Options

```
'Barrier Options: Knock-out options
Function BarrierKO(S, mu, T, sigma, n, sim, K, UP, DOWN)
    BM = GBM(S, mu, T, sigma, n, sim)
    Dim KO() As Double
    ReDim KO(1 To 1, 1 To 2) 'dimension = 1x2, in order to include both
        call and put
    For j = 1 To sim
        Dim BarrierUp As Boolean
        Dim BarrierDown As Boolean
        BarrierUp = False
        BarrierDown = False
```

```
        For i = 1 To (n + 1)
            If BM(i, j) ¿= UP Then BarrierUp = True
            If BM(i, j) ¡= DOWN Then BarrierDown = True
            If BarrierUp = True Then i = n + 1
            If BarrierDown = True Then i = n + 1
        Next i
        If BarrierUp = True Then
        CallB = 0
        PutB = 0
        ElseIf BarrierDown = True Then
        CallB = 0
        PutB = 0
        Else
        CallB = WorksheetFunction.Max(0, BM(n + 1, j) - K)
        PutB = WorksheetFunction.Max(0, K - BM(n + 1, j))
        End If
        KO(1, 1) = KO(1, 1) + CallB
        KO(1, 2) = KO(1, 2) + PutB
    Next j
    KO(1, 1) = KO(1, 1) / sim
    KO(1, 2) = KO(1, 2) / sim
    BarrierKO = KO
End Function

'Barrier Options: Knock-in options
Function BarrierKI(S, mu, T, sigma, n, sim, K, UP, DOWN)
    BM = GBM(S, mu, T, sigma, n, sim)
    Dim KI() As Double
    ReDim KI(1 To 1, 1 To 2) 'dimension = 1x2, in order to include both
        call and put
    For j = 1 To sim
        Dim BarrierUp As Boolean
        Dim BarrierDown As Boolean
        BarrierUp = False
        BarrierDown = False
        For i = 1 To (n + 1)
            If BM(i, j) ¿= UP Then BarrierUp = True
            If BM(i, j) ¡= DOWN Then BarrierDown = True
        Next i
        If BarrierUp = True Then
        CallB = WorksheetFunction.Max(0, BM(n + 1, j) - K)
        PutB = WorksheetFunction.Max(0, K - BM(n + 1, j))
        ElseIf BarrierDown = True Then
        CallB = WorksheetFunction.Max(0, BM(n + 1, j) - K)
        PutB = WorksheetFunction.Max(0, K - BM(n + 1, j))
        Else
        CallB = 0
```

```
        PutB = 0
        End If
        KI(1, 1) = KI(1, 1) + CallB
        KI(1, 2) = KI(1, 2) + PutB
        Next j
    KI(1, 1) = KI(1, 1) / sim
    KI(1, 2) = KI(1, 2) / sim
    BarrierKI = KI
End Function
```

Listing 6: Barrier Options