Pricing path-dependent options: Monte Carlo simulation and variance reduction techniques

Student ID: 200828554
Candidate Exam Number 76912
FM442: Quantitative Methods for Finance and Risk Analysis
London School of Economics

January 2009

1 Introduction

European option is a contract that gives the option holder the right to buy or sell one unit of underlying assets at a prescribed price, known as exercise price X, at a prescribed time, known as expiration date T. This kind of options are sometimes called *vanilla* or *plain vanilla* options because their payoffs depend only on the final value of the underlying asset. However, there exist some over-the-counter (OTC) options whose payoffs depend on historical values of the underlying asset over a given time period as well as its current price. The latter are known as *path-dependent* or *exotic* options.

Exotics options have been developed for a number of reasons, such as, hedging, taxes, accounting, legal or regulatory reasons, funds managers, etc. Sometimes they are design to reflect a view on potential future movements in particular market variables. They are really important for investment banks because they are generally much more profitable than vanilla products and occasionally they are designed to appear more attractive than they are to an unwary corporate treasurer or fund manager. Therefore it is quite important to develop accurate and efficient methods to evaluate exotic option prices in financial derivative markets.

This project considers three types of exotic options: barrier options, lookback options and Asian arithmetic/ geometric average options. There are others well-known examples of exotics, such as packages, forward options, compound options, chooser options, binary options, shout options, etc. Nevertheless this paper focuses on the first ones because they have explicit pricing formula. This allows to compare the resulting prices through Monte Carlo simulation with the ones obtained from de formulas. Such formula for Asian options is only known for the case of geometric average. However, for the arithmetic ones there exist some approximation formulas.

This project considers the setting of the classical Black&Scholes option pricing model. The evolution of the underlying asset price can be written as a stochastic process $S(t), 0 \leq t \leq T$ defined on a suitable probability space $(\Omega; \mathcal{F}; \mathcal{P})$. The underlying asset is assumed to follow a Geometric Brownian Motion $dS = \mu S dt + \sigma S dW$, where μ is the expected instantaneous rate of return on the underlying asset, σ is the instantaneous volatility of the rate of return, and dW is a Wiener process. The volatility and the risk-free rate r are assumed to be constant throughout the life of the option. Most of the formulas are written on a general form including a cost of carry term[1] b, which makes possible to use the same formula to price options on a wide range of underlying assets, such as stocks or indexes paying a dividend yield $d = r - b$, currencies or futures. For simplicity the empirical part of this project considers the situation where $d = 0$, and hence the cost of carry reduces to the interest rate r.

The reminder of this project is organized as follows. Section 2 explains the exotic options considered in this paper. A brief descriptions of the implementation is described in section 3, while in section 4 are presented the main results.

[1] The cost of carry is the cost of interest plus any additional cost

Finally section 5 draws some conclusions. Additionally there are some appendix that contain,

2 Exotic options

2.1 Barrier options

Since 1967, barrier options have been traded in the OTC market and nowadays are one of the most popular class of exotic options. The Chicago Board Option Exchange and the American Option Exchange have lists of barrier options on stock indexes. Several barrier options on currencies, interest-rate, and commodities are traded actively in the OTC market. They are attractive to some market participants because they are less expensive than the correspondence vanilla options. They have been used extensively to manage risks related to commodities, FX and interest rate exposures.

Barrier options are similar to standard vanilla options except that the payoff depends on whether the underlying asset price hits the barrier price, H, before expiration date and, therefore, they are path-dependent options. Their key characteristic is that these options are either initiated, knocked in, or exterminated, knocked out, once the underlying price hits H. Eventhough there are many types of barrier options, this project considers the most basic type of barrier option, the single barrier: Up&In, Up&Out, Down&In and Down&Out options. Each type can take the form of a call or a put, giving a total of 8 single barrier types. An In (Out) barrier means that the option becomes active (inactive) once the underlying crosses a particular barrier level. An Up (Down) barrier means that the option becomes active or inactive when the underlying price hits the barrier from below (above).

Among the literature one can find three approaches to evaluate barrier option prices: Black and Scholes partial differential equation[2], Monte Carlo simulation and close form solutions. Merton [19] provided the first analytical formula for a Down&Out call option which was extended for all 8 types of barriers by Reiner and Rubinstein [22]. Haug [13] gives a generalization.

Following Reiner and Rubinstein [22], this project considers the 8 single barrier types under Black and Scholes framework.

$$
\begin{aligned}
A &= \phi S e^{(b-r)T} N(\phi x_1) - \phi X e^{-rT} N(\phi x_1 - \phi\sigma\sqrt{T}), &(1)\\
B &= \phi S e^{(b-r)T} N(\phi x_2) - \phi X e^{-rT} N(\phi x_2 - \phi\sigma\sqrt{T}), &(2)\\
C &= \phi S e^{(b-r)T} (H/S)^{2(\mu+1)} N(\eta y_1) - \phi X e^{-rT} (H/S)^{2\mu} N(\eta y_1 - \eta\sigma\sqrt{T}), &(3)\\
D &= \phi S e^{(b-r)T} (H/S)^{2(\mu+1)} N(\eta y_2) - \phi X e^{-rT} (H/S)^{2\mu} N(\eta y_2 - \eta\sigma\sqrt{T}), &(4)
\end{aligned}
$$

where

[2] See [1], [19].

$$\phi = \begin{cases} 1 & call \\ -1 & put \end{cases}, \qquad \eta = \begin{cases} 1 & down \\ -1 & up \end{cases}, \tag{5}$$

$$x_1 = \frac{Ln(S/X)}{\sigma\sqrt{T}} + (1+\mu)\sigma\sqrt{T}, \qquad x_2 = \frac{Ln(S/H)}{\sigma\sqrt{T}} + (1+\mu)\sigma\sqrt{T}, \tag{6}$$

$$y_1 = \frac{Ln(H^2/SX)}{\sigma\sqrt{T}} + (1+\mu)\sigma\sqrt{T}, \qquad y_2 = \frac{Ln(H/S)}{\sigma\sqrt{T}} + (1+\mu)\sigma\sqrt{T}, \tag{7}$$

$$z = \frac{Ln(H/S)}{\sigma\sqrt{T}} + \lambda\sigma\sqrt{T}, \quad \mu = \frac{b - \sigma^2/2}{\sigma^2}, \quad \lambda = \sqrt{\mu^2 + \frac{2r}{\sigma^2}}. \tag{8}$$

2.1.1 Down&In Barriers

For $S(0) > H$ call and put prices can be obtained by the following formulas:

$$c_{di(X>H)} = C, \qquad c_{di(X<H)} = A - B + D, \tag{9}$$

$$p_{di(X>H)} = B - C + D, \qquad p_{di(X<H)} = A. \tag{10}$$

2.1.2 Up&In Barriers

For $S(0) < H$ call and put prices can be obtained by the following formulas:

$$c_{ui(X>H)} = A, \qquad c_{ui(X<H)} = B - C + D, \tag{11}$$

$$p_{ui(X>H)} = A - B + D, \qquad p_{ui(X<H)} = C. \tag{12}$$

2.1.3 Down&Out Barriers

For $S(0) > H$ call and put prices can be obtained by the following formulas:

$$c_{do(X>H)} = A - C, \qquad c_{do(X<H)} = B - D, \tag{13}$$

$$p_{do(X>H)} = A - B + C - D, \qquad p_{do(X<H)} = 0. \tag{14}$$

2.1.4 Up&Out Barriers

For $S(0) < H$ call and put prices can be obtained by the following formulas:

$$c_{uo(X>H)} = 0 \qquad c_{uo(X<H)} = A - B + C - D, \tag{15}$$

$$p_{uo(X>H)} = B - D, \qquad p_{uo(X<H)} = A - C. \tag{16}$$

2.2 Lookback options

Lookback options are a type of path-dependent option where the payoff dependents on the maximum or minimum price that the undelying reaches over the life of the option. The holder of the lookback can 'look back' over time to determine the payoff, this is where the name comes from. There are two main types of lookback options: fixed and floating strike.

2.2.1 Fixed Strike

This type of lookback option is only settled in cash. They have the strike predetermined. The payoff for calls is $(S_{\max}-X,0)^+$and for puts $(X-S_{\min},0)^+$, where $S_{\max}$ and $S_{\min}$ are the maximum and minimum observed prices of the underlying. These options can be priced using the Conze and Viswanathan [5] formula,

$$
\begin{aligned}
c &= Se^{(b-r)T}N(d_1) - Xe^{-rT}N(d_2) \\
&\quad +Se^{-rT}\frac{\sigma^2}{2b}\left[-(\frac{S}{X})^{-\frac{2b}{\sigma^2}}N(d_1-\frac{2b}{\sigma}\sqrt{T})+e^{bT}N(d_1)\right], \qquad (17) \\
p &= Xe^{-rT}N(-d_2) - Se^{(b-r)T}N(-d_1) \\
&\quad +Se^{-rT}\frac{\sigma^2}{2b}\left[(\frac{S}{X})^{-\frac{2b}{\sigma^2}}N(-d_1+\frac{2b}{\sigma}\sqrt{T})-e^{bT}N(-d_1)\right], \qquad (18) \\
d_1 &= \frac{Ln(S/X)+(b+\sigma^2/2)T}{\sigma\sqrt{T}}, \qquad d_2 = d_1-\sigma\sqrt{T}, \qquad (19)
\end{aligned}
$$

when $X \leq S_{\max}$

$$
\begin{aligned}
c &= e^{-rT}(S_{\max}-X) + Se^{(b-r)T}N(e_1) - S_{\max}e^{-rT}N(e_2) \\
&\quad +Se^{-rT}\frac{\sigma^2}{2b}\left[-(\frac{S}{S_{\max}})^{-\frac{2b}{\sigma^2}}N(e_1-\frac{2b}{\sigma}\sqrt{T})+e^{bT}N(e_1)\right], \qquad (20) \\
e_1 &= \frac{Ln(S/S_{\max})+(b+\sigma^2/2)T}{\sigma\sqrt{T}}, \qquad e_2 = e_1-\sigma\sqrt{T}, \qquad (21)
\end{aligned}
$$

and when $X \geq S_{\min}$

$$
\begin{aligned}
p &= e^{-rT}(X-S_{\min}) + S_{\min}e^{-rT}N(-f_2) - Se^{(b-r)T}N(-f_1) \\
&\quad +Se^{-rT}\frac{\sigma^2}{2b}\left[(\frac{S}{S_{\min}})^{-\frac{2b}{\sigma^2}}N(-f_1+\frac{2b}{\sigma}\sqrt{T})-e^{bT}N(-f_1)\right] \qquad (22) \\
f_1 &= \frac{Ln(S/S_{\min})+(b+\sigma^2/2)T}{\sigma\sqrt{T}}, \qquad f_2 = f_1-\sigma\sqrt{T}, \qquad (23)
\end{aligned}
$$

2.2.2 Floating Strike

Introduced in 1979, these options have payoffs which are either cash or asset settled. A floating strike lookback call gives the holder the right to buy the underlying at the lowest price observed, $S_{\min}$, during the life of the option. Similarly, a put gives the option holder the right to sell the underlying at the highest price observed, $S_{\max,}$ during the options's life. Hence the strike is given as the optimal value of the underlying asset.

It can be noted that although floating strike options are called options, they are not actually options as they are always exercised, see Yu, Kwok & Wu [25]. An attractive benefit of these options is that they are never out-of-the-money, on the other hand they are always more expensive than similar vanilla style options.

Floating strike lookback options can be priced within the Black&Scholes framework following Goldman, Sosin & Satto [11]:

$$\begin{aligned}
c &= Se^{(b-r)T}N(a_1) - S_{\min}e^{-rT}N(a_2) \\
&\quad + Se^{-rT}\frac{\sigma^2}{2b}\left[(\frac{S}{S_{\min}})^{-\frac{2b}{\sigma^2}} N(-a_1 + \frac{2b}{\sigma}\sqrt{T}) - e^{bT}N(-a_1) \right], \quad (24) \\
p &= S_{\max}e^{-rT}N(-a_2) - Se^{(b-r)T}N(-b_1) \\
&\quad + Se^{-rT}\frac{\sigma^2}{2b}\left[-(\frac{S}{S_{\max}})^{-\frac{2b}{\sigma^2}} N(b_1 - \frac{2b}{\sigma}\sqrt{T}) + e^{bT}N(b_1) \right], \quad (25) \\
a_1 &= \frac{Ln(S/S_{\min}) + (b + \sigma^2/2)T}{\sigma\sqrt{T}}, \qquad a_2 = a_1 - \sigma\sqrt{T}, \quad (26) \\
b_1 &= \frac{Ln(S_{\max}/S) + (b + \sigma^2/2)T}{\sigma\sqrt{T}}, \qquad b_2 = b_1 - \sigma\sqrt{T}. \quad (27)
\end{aligned}$$

Note that, since this project works in the Black&Scholes framework where there is continuous monitoring of the asset price, the above formulas are used when closed forms are needed. However, in the Monte Carlo simulations there is a discrete monitoring where the asset price is sampled at particular points in time, as investments banks do as well. These are often referred to as finite sampling lookbacks and can be priced by making the adjustment given by Broadie, Glasserman & Kou [3] or by Levy & Mantion [18]. Nevertheless, for simplicity, this project ignores this fact.

2.3 Asian options

Asian options are options where the payoff depends on the average price of the underlying asset during at least some part of the life of the option. The payoff from an Asian call is $(\bar{S} - X, 0)^+$ and likewise the payoff of an Asian put is $(X - \bar{S}, 0)^+$, where $\bar{S}$ is the considered the average value of the underlying asset. Call options have positive vega, which means that the options price depends positively on the volatility. Since averages are less volatile than the stock price himself, it makes, *ceteris paribus*, Asian options less valuable than the vanilla ones.

They were originally developed in 1987 when Banker's Trust Tokyo office, hence the name of Asian, used them for pricing average options on crude oil contracts.

They are commonly traded on currencies and commodities products which have low trading volumes. Asian options provide payoffs that may match the risk exposure of firms and thus possess desirable risk management characteristics. For example, a firm receiving or paying foreign exchange on a regular basis may be concerned with the risk of the average value of its receipts and/or payments over some period of time. Likewise, the revenues of a firm, producing and selling natural gas on an essentially continuous basis, vary with the average price of natural gas during some period of time. These patterns of cash flows create hedging potential for options written on the average prices of an asset. The idea of such path-dependent options has been around for quite some time and trading in them has increased rapidly in recent years. Moreover, this kind of options increases the difficulty to manipulate prices near to maturity since the final payoff not only depends on the final price but on the average over a predetermined period of time.

The are broadly segregated into two categories: arithmetic average Asians and geometric average Asians.

2.3.1 Geometric average Asians

Geometric averaging options can be priced via closed form analytic solution since the geometric average of the underlying prices follows a lognormal distribution as well. Kemna & Vorst [16] derived a closed form pricing formula for geometric averaging options by altering the volatility and cost of carry term in the Black&Scholes formula for vanilla options:

$$c = Se^{(b_A - r)T}N(d_1) - Xe^{-rT}N(d_2), \tag{28}$$

$$p = Xe^{-rT}N(-d_2) - Se^{(b_A - r)T}N(-d_1), \tag{29}$$

$$d_1 = \frac{Ln(S/X) + (b_A + \sigma^2/2)T}{\sigma_A\sqrt{T}}, \qquad d_2 = d_1 - \sigma_A\sqrt{T}, \tag{30}$$

$$\sigma_A = \frac{\sigma}{\sqrt{3}}, \qquad b_A = \frac{1}{2}(b - \frac{\sigma^2}{6}). \tag{31}$$

2.3.2 Arithmetic average Asians

Up today, there are no known analytical closed form solution for arithmetic options. The main reason is that the sum of the lognormal random variables will not itself has a lognormal distribution. Therefore, these options have to be priced by analytical approximations. A number of approximations have emerged in literature[3]. This paper uses the approximation suggested by Turnbull and Wakeman [24]. They argue that the arithmetic averaging distribution is approximately lognormal and using its first and second moments an approximated analytic solution can be found.

$$
\begin{aligned}
c &\approx Se^{(b_A-r)T_2}N(d_1) - Xe^{-rT_2}N(d_2), && (32)\\
p &\approx Xe^{-rT_2}N(-d_2) - Se^{(b_A-r)T_2}N(-d_1), && (33)\\
d_1 &= \frac{Ln(S/X)+(b_A+\sigma^2/2)T_2}{\sigma_A\sqrt{T_2}}, \qquad d_2 = d_1 - \sigma_A\sqrt{T_2}, && (34)\\
\sigma_A &= \sqrt{\frac{Ln(M_2)}{T} - 2b_A}, \qquad b_A = \frac{Ln(M_1)}{T}, && (35)
\end{aligned}
$$

where T_2 is the remaining time to maturity. The exact first and second moments of the arithmetic average are

$$
\begin{aligned}
M_1 &= \frac{e^{bT}-e^{b\tau}}{b(T-\tau)}, && (36)\\
M_2 &= \frac{2e^{(2b+\sigma^2)T}}{(b+\sigma^2)(2b+\sigma^2)(T-\tau)^2} + \frac{2e^{(2b+\sigma^2)T}}{b(T-\tau)^2}\left[\frac{1}{2b+\sigma^2} - \frac{e^{b(T-\tau)}}{b+\sigma^2}\right], && (37)
\end{aligned}
$$

where T is the original time to maturity, and τ is the time to the beginning of the average period. If the option is into the average period, the option value must be multiplied by T_2/T and the strike price must be replaced by $\hat{X} = \frac{T}{T_2}X - \frac{T_1}{T_2}S_A$, where S_A is the average asset price during the realized or observed time period $T_1 = T - T_2$.

3 Implementation

As explained before, this paper uses two approaches to compute path-dependent option prices: Closed form solutions and Monte Carlo simulation. Monte Carlo simulation is a very popular and robust numerical method that has been used to value options since Boyle [2] introduced the method in a seminal paper. In appendix A there is brief discussion.

One of main drawback of the Monte Carlo method is the slow convergence. The statistical error of the Monte Carlo method is of order $O(1/\sqrt{N})$ with N simulations, i.e. if ω is the standard deviation of the discounted payoffs, then standard error of the estimate is $\omega/\sqrt{N}$. This shows that the uncertainty about the value of the derivative is inversely proportional to the squared root of the number of simulations[4]. Therefore, a very large number of trials is necessary.

[3] See for example Levy [17], Curran [6], Gema and Yor [9], Haykov [14], Curran [7], Bouaziz, Briys and Grouhy [4], Zhang [26], Geman and Eydeland [8] and Zhang [27].

[4] i.e. to double the accuracy of a simulation the number of trials must be multiplied by 4; likewise to incease the accuracy by a factor of k the number of trials must increase by a factor of k^2.

This is very expensive in terms of computational time. To solve this problem this paper uses different approaches based in the two broad strategies shown by the literature: taking advantage of tractable features of a model to adjust or correct simulation outputs, and reducing the variability in simulation inputs. These methods reduce the variance of the simulation without increasing the number of estimations and therefore reducing significantly the computational cost.

The reminder of this section will briefly show the implementation of the techniques, while the theoretical details can be found in appendix B.

3.1 Antithetic Variables

In this method, starting from the matrix of simulated independent normal shocks, the antithetic one is created. Then, using both matrices, $2N$ antithetic stock price paths are generated, in the sense that they are pairwise reflective paths as shown in figure 1.

3.2 Control Variate

In this project the control variate technique is used in two ways. First, the control variate estimator is specified as

$$\frac{1}{N}\sum_{i=1}^{N}\left(Y_i - \hat{b}^* \left[S_i(T) - e^{rT}S(0)\right]\right).$$

And second, when it is applied to Asian options the geometric average option is used as control to price the arithmetic one.

3.3 Matching Underlying Assets: Moment Matching

This method is implemented in two directions.

First, through error adjustments, i.e. given a simulated matrix of shock and its antithetic one, both are standarized dividing by the standard error of the shocks in each time step. However as shown in figure 2 there is no huge differences between both paths, the one using the initial errors and the one using the standarized errors. This suggests that, the main gain using this method will come from the antithetic component and not from matching the variance.

Second, this paper uses moment matching through path adjustments, transforming stock price paths in the following way:

$$\tilde{S}_i(t) = S_i(t)\frac{E\left[S(t)\right]}{\bar{S}(t)} = S_i(t)\frac{\exp(rt)S(0)}{\bar{S}(t)}, \qquad i = 1, ..., N,$$

Figure 3 shows that moment matching through path adjustments does not introduce important differences in the path.

4 Results

Table 1 shows prices of barrier call options for different strike prices X. As it was expected, in all cases these options are cheaper than the equivalent vanilla ones. As it is shown, there always exists discrepancy between the closed form price and the ones obtained by simualtion. In the cases of Down&Out and Up&Out calls the simulations tends to overestimate the value of the call, while for Down&In and Up&In calls the values are underestimated. However, in all cases, the higher are the strike prices the smaller are the errors and the standard deviations of the estimates. It is also important to note that the variance reduction techniques indeed reduce the standard deviations. Moreover, as previously stated, the moment matching through error adjustments gives very similar results to the antithetic variables method [5].

Table 2 contains call prices for fixed strike lookback options for different values of T and σ. First of all, as theory suggests, this option is more valuable than the vanilla one. The price also increases with T and σ, since in both cases the optimal value can be higher, hence the option pays more. The simulations are, in general, underestimating the closed form solution price. The techniques for reducing the variance are effectively cutting down the absolute error as well as the variance. Nonetheless the moment matching through path adjustments technique seem not be very useful, which was expected from figure 3. Finally, note that the underlying asset control variate technique has not been used since the payoff has a very little dependence with the final stock price.

In table 3 there are the prices for floating strike lookback call options for different values of T and σ. As in the previous case, these options are more valuable than the vanilla calls, and it price increases with T and σ. Again, the results using simulated values are underestimating the theoretical prices. Even so, the variance reduction techniques make a really good job, decreasing the absolute error and the standard error of the estimates. Furthermore, the control variate technique is very efficient reducing the standard error considerably, especially in the case of $T = 1$ and $\sigma = 0.35$[6]. This happens because the smaller is $S_{\min}$ the higher is the correlation between the payoff $(S_T - S_{\min})$ and the final stock price S_T, as clearly illustrated in figure 5 versus 6 . As a result of this higher correlation the control variate technique works better. Finally, on figure 4 the gain of using control variates in this case is shown. This figure contains the histogram for the simulated prices using Monte Carlo and the ones using the control variable technique. Clearly, this technique reduces the variance considerably, increasing the mass around the true value and producing thinner tails.

Table 4 has the prices for geometric Asian call options for different values of X and σ. As explained in section 2.3 , the value of this options is smaller than the vanilla one. The values obtained by Monte Carlo overestimate the value of

[5] In all the tables with the same number of trials the complete set of estimators has been computed from the same simulated paths.

[6] As were remarked before, the higher are volatility and time to maturity the higher is $S_{\max}$ and the lower is $S_{\min}$.

these options; while the variance reduction techniques reduce the standard error. Furthermore, the moment matching through path adjustments give a value much more closer to the one obtained by the closed form solution. Finally note that, as for lookback options with fixed strike, the underlying asset control variate technique has not been used because the payoff has a very little dependence on the final stock price.

Table 5 shows prices of arithmetic Asian call options for the same values of X and σ than in table 4. This allows to compare both. As expected, *ceteris paribus*, the arithmetic Asian price is bigger than the geometric one, this is due to the mathematical result that the arithmetic average is bigger than the geometric average, as can be seen in figure 9. Just looking to this table, can be seen again that the prices are lower to the vanilla ones, the reason is the same as in the previous case. Once more the simulations overestimates the value obtained by the approximated closed form. The variance reduction techniques in general reduce the standard error. The control variate technique, which uses geometric average options as control variates, gives a much better result, reducing the error considerably.

Figures 7, 8 and 9 show scatter plots of the Monte Carlo simulated values for the discounted payoff of the arithmetic asian option $e^{-rT}(\bar{S}_A - K)^+$ against the final stock price $S(T)$, against the vanilla call discounted payoff $e^{-rT}(S_T - K)^+$, and against the geometric discounted payoff $e^{-rT}(\bar{S}_G - K)^+$ respectively. Figure 7 shows a weak correlation between $e^{-rT}(\bar{S}_A - K)^+$ and $S(T)$. The next figure shows as using the vanilla call discounted payoff the correlation increases. Finally, figure 9 shows the extremely strong correlation between the discounted payoffs of the arithmetic and geometric asian call options, which explains why the control variate technique works so good.

Tables 6, 7, 8, 9 and 10 show the slow convergence rate of Monte Carlo simulation for the different types of options. Between the first and second panel the number of trials increases by a factor of 10, therefore the accuracy increases by a factor of $\sqrt{10}$. The third panel has doubled the accuracy with respect the second, as the number of trials has been quadrupled. Finally can be observed an increase in the efficiency when using the reduction variance techniques due to the reduction in the variance without increasing the number of trials.

5 Conclusion

Boyle [2] introduced the simple and flexible Monte Carlo simulation for valuing options. Although widely recognized as a feasible method for pricing options, in the literature the use of Monte Carlo simulation in a relatively limited. One of the main reasons is probably the slow speed of solution compared with other numerical techniques. Hence, Monte Carlo simulation tends to be used only when other methods cannot effectively deal with the problem. A second reason is the belief that it cannot be used for complex, or path-dependent, options.

This project shows that it is not only possible but also useful, the use of Monte Carlo simulation to price path-dependent options. Moreover, the speed of convergence can be increased using variance reduction techniques. Furthermore, this paper shows that depending on the payoff form of the option, and sometimes on the value of the parameters, some techniques work better.

Finally, note that Monte Carlo method can easily incorporate additional complexities, such as time-varying parameters, multiple sources of uncertainty, and different distributions for random variates. This inherent capability makes Monte Carlo simulation an extremely powerful tool for valuing complex path-dependent options.

References

[1] F. Black and M. Scholes, The pricing of options and corporate liabilities, J. Political Economy 81 (1973), no. 3, 637–659.

[2] Boyle, P. Options: A Monte Carlo Approach. Journal of Financial Economics, 4 (May 1977), 323–338.

[3] Broadie, M., P. Glasserman, and S. Kou. A continuity correction for discrete barrier options, Mathematical Finance, 7:4, pp. 325-349 (October 1997).

[4] Bouaziz, L., E. Briys and M. Grouhy. The Pricing of Forward Starting Asian Options. Journal of Banking and Finance, 18, 823-839 (1994)

[5] Conze, A., & Viswanathan, R., Path Dependent Options: The Case of Lookback Options, Journal of Finance, S. 1893-1907 (1991)

[6] Curran, M., Beyond Average Intelligence, Risk Magazine 5, 10, 60 (1992)

[7] Curran, M., Valuing Asian and Portfolio Options by Conditioning on the Geometric Mean Price, Management Science, 40, 12, pp 1705-1711 (1994)

[8] H. Geman and A. Eydeland. Domino effect. Risk, pages 65–67, April 1995.

[9] Geman, H. and Yor, M. Bessel processes, Asian option and perpetuities. Mathematical Finance, 3 (4), 349-375 (1993)

[10] Glasserman, P. Monte Carlo Methods in Financial Engineering. Springer (2004)

[11] Goldman, B., Sosin, H., Gatto, M. A., Path Dependent Options: Buy at the Low, Sell at the High, Journal of Finance, S.1111-1127 (1979)

[12] Hammersley, J. M., and D. C. Handscomb. Monte Carlo Simulation. Metheun, London, (1964).

[13] E. G. Haug The Complete Guide to Option Pricing Formulas, MaGraw-Hill, (1997).

[14] Haykov, J.M. A Better Control Variate for Pricing Standard Asian Options . Journal of Financial Engineering, 2, 3, pp. 207-216 (1993)

[15] J. C. Hull, Options, Futures and Others, Prentice Hall, (2003).

[16] Kemna, A. G. Z. & Vorst, A. C. F., A Pricing Method for Options Based on Average Asset Values, Journal of Banking and Finance, 14, 113-129, (1990)

[17] Levy, E., Pricing European Average Rate Currency Options, Journal of International Money and Finance, 14, 474-491. (1992)

[18] Levy, E., & Mantion, F., Approximate Valuation of Discrete Lookback & Barrier Options. Working Paper, (1998)

[19] R. C. Merton, Theory of rational option pricing, Bell J. Econ. Manag. Sci. 4 , no. 1, 141–183 (1973)

[20] Morgan, B. J. T. Elements of Simulation. Chapman & Hall, London, (1984).

[21] Naylor, T. J.; J. L. Balintfy; D. S. Burdick; and K. Chu. Computer Simulation Techniques. Wiley, NY, (1966).

[22] E. Reiner and M. Rubinstein, Breaking down the barriers, Risk 4 , no. 8, 28–35 (1991)

[23] Rubinstein, R. Y. Simulation and the Monte Carlo Method. Wiley, NY, (1981).

[24] Turnbull, S. M. & Wakeman, L. M., A Quick Algorithm for Pricing European Average Options, Journal of Financial and Quantitative Finance, 26, 377-389. (1991)

[25] Yu, H., Kwok, Y.K., Wu, L., Early Exercise Policies of American Floating Strike & Fixed Strike Lookback Options, Nonlinear Analysis (2001)

[26] P.G. Zhang, Flexible Asian options, Journal of Financial Engineering 3 (1) 65-83 (1994)

[27] P.G. Zhang, Correlation digital options Journal of Financial Engineering 4 75-96 (1995)

A Monte Carlo simulation

This is a necessarily brief discussion because excellent detailed expositions are available[7]. Von Neumann and Ulam coined the term Monte Carlo during World War II and applied the technique to problems related to the development of the atom bomb. Soon Monte Carlo methods were employed to calculate complex multidimensional integrals as well as integral and differential equations that were not solvable by analytical methods. Another area of intense use of the Monte Carlo method has been sampling of random variables from different probability distributions. Most finance and economic applications use sampling from probability distributions to replicate uncertainty of the "real world," e.g., prices of risky assets. Thus, the Monte Carlo method is a numerical method which can be used to approximate the expected value of the price of a risky asset in the future.

When applied to value an option, Monte Carlo simulation uses the risk-neutral valuation result. The numerical procedure consists in sampling a random path for the asset price from 0 to T in a risk-neutral world. Under the Black&Scholes economy the underlying asset in a risk-neutral world is assumed to follow a Geometric Brownian Motion

$$dS = rSdt + \sigma Sd\hat{W}. \tag{38}$$

To simulate the path followed by S, the life of the derivative is divided in N short intervals of length Δt and equation 38 is approximated as

$$S(t+\Delta t) - S(t) = rS(t)\Delta t + \sigma S(t)\sqrt{\Delta t}\epsilon, \tag{39}$$

where $S(t)$ denotes the value of S at time t, ϵ is a random sample from a standard normal distribution. However, in practice it is usually more accurate to simulate LnS instead of S because it follows a Wiener process. So the following equation is used to construct a path for S

$$S(t+\Delta t) = S(t)\exp\left[(r - \frac{\sigma^2}{2})\Delta t + \sigma\sqrt{\Delta t}\epsilon\right]. \tag{40}$$

Once this point is reached one have to calculate the discounted payoff from the derivative and repeat the previous procedure a large number of times. Finally the mean of all the sample discounted payoffs is calculated. Appealing to the law of large numbers, the average is taken as the estimate of the expected value of the option.

[7] See, for example, Hammersley and Handscomb [12], Naylor, Balintfy, Burdick and Chu [?], Rubinstein [23], and Morgan [20]. Boyle's [2] brief discussion, introducing the Monte Carlo method for option valuation, remains the most accessible.

B Variance Reduction Techniques

B.1 Antithetic Variables

This method attempts to reduce variance by introducing negative dependence between pairs of replications. The method can take various forms, however, the paper focuses in the standard normal case. Since the normal distribution is symmetric with respect the origin, if ε is standard normal distributed then $-\varepsilon$ it is also. Hence, if we generate a path using as inputs $\varepsilon_1, ..., \varepsilon_N$, we can generate a second path using $-\varepsilon_1, ..., -\varepsilon_N$. The variables ε_i and $-\varepsilon_i$ are antithetics pairs in the sense that the have opposite sign, hence, an unusually large or small output computed from the first path may be balanced by the value computed from the antithetic path, resulting in a reduction in variance.

This works well because, if $\varepsilon_1, ..., \varepsilon_N$, is used to simulate the increments of a Brownian path, then $-\varepsilon_1, ..., -\varepsilon_N$ simulate the increments of the reflection path about the origin. Hence we use them to compute two values of the derivative, when one value is above the true value, the other tends to be below, and vice versa. This may result in lower variance.

B.2 Control Variates

This method exploits information about the errors when estimating known quantities to reduce the error when estimating unknown quantities.

To describe this method, let $Y_1, ..., Y_n$ be the outputs from n replications of a simulation, i.e. in our case Y_i is the discounted payoff of an option from its ith simulated path. Suppose that the Y_i are independent and identically distributed and that the objective is to estimate $E[Y_i]$. The usual estimator is the sample mean $\bar{Y} = (Y_1 + ... + Y_n)/n$ which is unbiased and converges with probability 1 as $n \to \infty$.

Suppose now that, on each replication can be calculated another output X_i in addition to Y_i. Suppose that the pairs $(X_i, Y_i) : i = 1, ..., n$ are i.i.d. and that $E[X_i]$ is known. Then, for any fixed b, from the ith replication can be obtained

$$Y_i(b) = Y_i - b(X_i - E[X]), \tag{41}$$

and then the sample mean computed

$$\bar{Y}(b) = \bar{Y} - b(\bar{X} - E[X]), \tag{42}$$

this is the control variate estimator. The observed error $\bar{X} - E[X]$ serves as a control when estimating $E[Y]$.

As an estimator of $E[Y]$, the control variate estimator 42 is unbiased and consistent. Moreover, if $2b\sigma_X\sigma_Y\rho_{XY} > b\sigma_X^2$, the control variate estimator's variance, $\sigma_Y^2 - 2b\sigma_X\sigma_Y\rho_{XY} + b\sigma_X^2/n$, is smaller than the standard estimator variance, σ_Y^2/n.

The optimal coefficient b^* that minimize the control variate estimator's variance is given by $b^* = \frac{\sigma_Y}{\sigma_X}\rho_{XY} = \frac{Cov[X,Y]}{Var[X]}$[8]. With this specification the ratio between both variances is

$$\frac{Var\left[\bar{Y} - b^*(\bar{X} - E\left[X\right])\right]}{Var\left[\bar{Y}\right]} = 1 - \rho_{XY}^2. \tag{43}$$

We can see that, if the computational effort per replication is roughly the same with and without a control variate, then 43 measures the computational speed-up resulting from the use of a control, i.e. is required $n/(1-\rho_{XY}^2)$ replications of the Y_i to achieve the same variance as n replications of the control variate estimator[9]. On the other hand, a rather high degree of correlation is needed to obtain substantial benefits.

B.2.1 Application I: Underlying assets

The idea is to use the underlying asset as a control variate. The absence of arbitrage is equivalent to the condition that discounted asset prices are martingales under the risk neutral measure, which gives a potential control variate because its expectation at any future time must be equal to its initial value, i.e. $E\left[e^{-rt}S(t)\right] = S(0) \quad \text{for all } 0 \leq t \leq T$. Hence, when pricing an option with discounted payoff Y, the next control variate estimator can be used:

$$\frac{1}{N}\sum_{i=1}^{N}\left(Y_i - \hat{b}^*\left[S_i(T) - e^{rT}S(0)\right]\right). \tag{44}$$

The effectiveness of this method depends on the correlation between the payoff Y and $S(T)$, hence it is affected by the payoff form and the strike X. For deep out-of-money options the correlation could be quite low.

B.2.2 Application II: Tractable options

The idea behind it is to apply the control variate technique when there are two similar derivatives but only one of them has closed form solution. Hence we can price the more complex option using the tractable one as a more effective control.

B.3 Matching Underlying Assets: Moment Matching

There exists some techniques that try to ensure that certain sample moments produced in a simulation exactly coincide with their population values, i.e. with the values that would be attained in the limit of infinitely many replications.

[8] b^* involves unknow objetcs, hence, in practice we will replace the population parameters with their sample counterparts with yields to estimate b^* using the slope of the least-squares regresion between (X, Y).

[9] This equation apply when the optimal b^* is known. Replacing b^* with its sample conterparty introduces some bias that we will not take into account for this proyect.

The goal of derivative pricing is to determine the value of a derivative security relative to its underlying asset. Hence, the correct pricing of the underlying asset is a prerequisite for accurate valuation of derivatives. This project develops two applications of the moment matching, i.e., for transforming the simulated paths[10].

B.3.1 Application I: Moment matching through error adjustments

Given that the process to simulate the stock price process, summarized in **??**, involves sampling from a standard normal, along the next lines this method will be explained in this particular context.

The key idea is to adjust the samples taken from the standard normal distribution to make sure that, at least, the first and second moments are equal to the population ones. Suppose that the samples are $\epsilon_i : i = 0, ...n$, in order to match the first two moments the samples can be adjusted as $\epsilon_i^* = \frac{\epsilon_i - \mu}{\sigma}$, where μ and σ are the sample mean and standard deviation of the original samples respectively.

This method can be used in conjunction with the antithetic variable technique, because the latter matches all odd moments, hence only left to be matched the second moment.

B.3.2 Application II: Moment matching through path adjustments

As remarked before, under the risk neutral measure, discounted prices are martingales, i.e. $E\left[\exp(-rt)S(t)\right] = S(0) \quad \text{for all } 0 \leq t \leq T$. The idea of this method is to transform paths to match this moment. Suppose N independent copies $S_1, ..., S_N$ of the process are simulated. Then for finite N the sample mean $\bar{S}(t)$ will not coincide with the population mean $E\left[S(t)\right] = e^{rt}S(0)$. A possible solution is to transform the simulated paths by setting

$$\tilde{S}_i(t) = S_i(t)\frac{E\left[S(t)\right]}{\bar{S}(t)} = S_i(t)\frac{\exp(rt)S(0)}{\bar{S}(t)}, i = 1, ..., N, \tag{45}$$

or

$$\tilde{S}_i(t) = S_i(t) + E\left[S(t)\right] - \bar{S}(t) = S_i(t) + \exp(rt)S(0) - \bar{S}(t), i = 1, ..., N, \tag{46}$$

and then use $\tilde{S}(t)$ rather than $S(t)$ to price derivatives. Both, multiplicative adjustment or additive adjustment, ensures that the sample mean of $\tilde{S}_1(t), ..., \tilde{S}_N(t)$ is equal to $E\left[S(t)\right]$. However, for the geometric brownian motion the multiplicative transform 45 seems to be more natural.

[10] There exists other methods that weight, but do not transfor, paths in order to march moments. However we are not going to study them.

C Tables

Table 1: Barrier call options: Prices using diferents methods. In brackets, the standard error of the estimate. MM refers to Moment Matching. c_{do} refers to the Down&Out call, c_{di} to the Down&In call, c_{uo} to the Up&Out call and finally c_{ui} refers to the Up&In call.

					Barrier call option prices				
Type	X	H	Closed form	Monte Carlo	Antithetic variables	Control variates	MM error adj.	MM path adj.	B&S Vanilla
c_{do}	35	38	2.87	3.0527 (0.21)	3.3253 (0.1536)	3.1791 (.1604)	3.3036 (0.1538)	3.1959 (0.2158)	6.8785
c_{do}	40	38	1.9327	1.9979 (0.1575)	2.1934 (0.1152)	2.0931 (0.1199)	2.1861(0.1155)	2.1091(0.1626)	3.842
c_{do}	45	38	1.1173	1.1226(0.1119)	1.2405(0.0815)	1.1869(0.0882)	1.2427 (0.0819)	1.2011(0.1163)	1.9229
c_{uo}	35	42	0.1291	0.1841(0.0247)	0.1944(0.0186)	0.1828(0.0246)	0.1913(0.0184)	0.1815 (0.0244)	6.8785
c_{uo}	40	42	0.003	0.0044 (0.0023)	0.0072 (0.0020)	0.0044(0.0023)	0.0074 (0.0021)	0.0036 (0.0017)	3.842
c_{uo}	45	42	0	0	0	0	0	0	1.9229
c_{di}	35	38	4.0085	3.621 (0.1737)	3.5345 (0.1205)	3.7011 (0.151)	3.5584(0.1203)	3.6671(0.1749)	6.8785
c_{di}	40	38	1.9093	1.6919(0.1242)	1.6308 (0.0854)	1.7513 (0.1066)	1.6378 (0.0851)	1.7195 (0.1254)	3.842
c_{di}	45	38	0.8056	0.6955(0.0820)	0.6630 (0.0557)	0.7306(0.0728)	0.6607(0.0553)	0.7137 (0.0828)	1.9229
c_{ui}	35	42	6.7493	6.4896 (0.2322)	6.6654 (0.1652)	6.6974(0.0655)	6.6706 (0.1652)	6.6815 (0.2356)	6.8785
c_{ui}	40	42	3.839	3.6855 (0.1829)	3.8171(0.1304)	3.8400 (0.0776)	3.8165 (0.1305)	3.8251 (0.1869)	3.842
c_{ui}	45	42	1.9229	1.8181 (0.1329)	1.9036 (0.0945)	1.9175(0.0796)	1.9035 (0.0946)	1.9148 (0.1367)	1.9229
					S=40, T=0.5, r=ln(1.05), d=0, sigma=0.3, N=1000				

Table 2: Fixed strike lookback call options: Prices using diferents methods. In brackets, the standard error of the estimate. MM refers to Moment Matching.

Fixed strike lookback call option prices								
T	Sigma	X	Closed form	Monte Carlo	Antithetic variables	MM error adj.	MM path adj.	B&S Vanilla
0.5	0.2	35	10.0438	9.6085 (0.1254)	9.7610 (0.0887)	9.7690 (0.0888)	9.6970 (0.1268)	6.1899
0.5	0.2	40	5.1643	4.7290 (0.1254)	4.8815 (0.0887)	4.8895 (0.0888)	4.8175 (0.1268)	2.7427
0.5	0.2	45	1.7255	1.4905 (0.0880)	1.5597 (0.0629)	1.5633 (0.0630)	1.5443 (0.0897)	0.904
0.5	0.3	35	12.5218	11.8428 (0.1954)	12.0722 (0.1385)	12.0834 (0.1385)	11.9796 (0.1978)	6.8785
0.5	0.3	40	7.6423	6.9633 (0.1954)	7.1927 (0.1385)	7.2039 (0.1385)	7.1001 (0.1978)	3.842
0.5	0.3	45	3.8282	3.3646 (0.1631)	3.5119 (0.1165)	3.5183 (0.1166)	3.4700 (0.1661)	1.9229
1	0.2	35	12.4068	11.8867 (0.1863)	12.0940 (0.1350)	12.1022 (0.1351)	12.0728 (0.1907)	7.3824
1	0.2	40	7.6449	7.1248 (0.1863)	7.3321 (0.1350)	7.3403 (0.1351)	7.3108 (0.1907)	4.1545
1	0.2	45	3.8519	3.4448 (0.1574)	3.6254 (0.1148)	3.6331 (0.1149)	3.6057 (0.1622)	2.0651
1	0.3	35	16.0141	15.1972 (0.2965)	15.5159 (0.2150)	15.5309 (0.2151)	15.4826 (0.3044)	8.4802
1	0.3	40	11.2522	10.4353 (0.2965)	10.7540 (0.2150)	10.7689 (0.2151)	10.7207 (0.3044)	5.6692
1	0.3	45	7.2315	6.5351 (0.2727)	6.8287 (0.1985)	6.8410 (0.1987)	6.7992 (0.2811)	3.6402

S=S_{min}=S_{max}=40, r=ln(1.05),d=0, N=1000

Table 3: Floating strike lookback call options: Prices using diferents methods. In brackets, the standard error of the estimate. MM refers to Moment Matching.

Fixed strike lookback call option prices								
T	Sigma	Closed form	Monte Carlo	Antithetic variables	Control variates	MM error adj.	MM path adj.	B&S Vanilla
0.5	0.2	4.7692	4.5001 (0.1220)	4.5496 (0.0862)	4.6016 (0.0533)	4.5542 (0.0863)	4.5772 (0.1236)	0.904
0.5	0.3	6.7532	6.3664 (0.1865)	6.4379 (0.1319)	6.5249 (0.0771)	6.4462 (0.1320)	6.4903 (0.1894)	1.9229
0.5	0.35	7.7126	7.2686 (0.2206)	7.3521 (0.1560)	7.4579 (0.0883)	7.3624 (0.1560)	7.4182 (0.2242)	2.4633
1	0.2	6.8641	6.4313 (0.1760)	6.6523 (0.1281)	6.6536 (0.0710)	6.6612 (0.1282)	6.6649 (0.1800)	2.0651
1	0.3	9.4954	8.8396 (0.2723)	9.1783 (0.1983)	9.1935 (0.1023)	9.1893 (0.1985)	9.2055 (0.2797)	3.6402
1	0.35	10.7504	9.9835 (0.3246)	10.3829 (0.2364)	10.4109 (0.1166)	10.3953 (0.2366)	10.4185 (0.3341)	4.437
S=S_{min}=S_{max}=40, r=ln(1.05),d=0, N=1000								

Table 4: Geometric average asian call options: Prices using diferents methods. In brackets, the standard error of the estimate. MM refers to Moment Matching.

Geometric average asian call option prices							
sigma	X	Closed form	Monte Carlo	Antithetic variables	MM error adj.	MM path adj.	B&S Vanilla
0.2	35	4.7055	5.2587 (0.0982)	5.3408 (0.0694)	5.3396 (0.0693)	4.5767 (0.0940)	6.1899
0.2	40	1.1458	1.4249 (0.0663)	1.4715 (0.0471)	1.4685 (0.0470)	1.0535 (0.0572)	2.7427
0.2	45	0.094	0.1522 (0.0215)	0.1535 (0.0152)	0.1517 (0.0152)	0.0936 (0.0160)	0.904
0.3	35	4.8347	5.3659 (0.1385)	5.4791 (0.0981)	5.4755 (0.0980)	4.7751 (0.1319)	6.8785
0.3	40	1.6969	1.9818 (0.0987)	2.0471 (0.0701)	2.0432 (0.0700)	1.6291 (0.0892)	3.842
0.3	45	0.3824	0.4980 (0.0509)	0.5147 (0.0362)	0.5126 (0.0360)	0.3739 (0.0434)	1.9229
S=S_{min}=S_{max}=40, r=ln(1.05),d=0, N=1000							

Table 5: Arithmetic average asian call options: Prices using diferents methods. In brackets, the standard error of the estimate. MM refers to Moment Matching.

Arithmetic average asian call option prices								
sigma	X	Aprox closed form	Monte Carlo	Antithetic variables	Control variates	MM error adj.	MM path adj.	B&S Vanilla
0.2	35	5.4049	6.4827 (0.1015)	6.5657 (0.0718)	5.9115 (0.0057)	6.5653 (0.0716)	6.3523 (0.1009)	6.1899
0.2	40	1.5382	2.1897 (0.0798)	2.2562 (0.0565)	1.8594 (0.0144)	2.2540 (0.0564)	2.0945 (0.0783)	2.7427
0.2	45	0.1594	0.3061 (0.0321)	0.3162 (0.0227)	0.2231 (0.0094)	0.3144 (0.0226)	0.2823 (0.0307)	0.904
0.3	35	5.6212	6.5612 (0.1470)	6.6824 (0.1039)	5.9988 (0.0104)	6.6798 (0.1038)	6.4775 (0.1462)	6.8785
0.3	40	2.1768	2.7292 (0.1147)	2.8134 (0.0813)	2.4010 (0.0153)	2.8093 (0.0812)	2.6694 (0.1134)	3.842
0.3	45	0.5555	0.7926 (0.0656)	0.8099 (0.0466)	0.6465 (0.0129)	0.8066 (0.0465)	0.7659 (0.0644)	1.9229
S=Smin=Smax=40, sigma=0.3, r=ln(1.05),d=0, N=1000								

Table 6: Barrier call options: Prices using diferents methods and varing the number of trials. In brackets, the standard error of the estimate. MM refers to Moment Matching. c_{do} refers to the Down&Out call, c_{di} to the Down&In call, c_{uo} to the Up&Out call and finally c_{ui} refers to the Up&In call.

Barrier call option prices									
N=1000									
Type	X	H	Closed form	Monte Carlo	Antithetic variables	Control variates	MM error adj.	MM path adj.	B&S Vanilla
c_{do}	40	38	1.9327	1.9979 (0.1575)	2.1934 (0.1152)	2.0931 (0.1199)	2.1861(0.1155)	2.1091(0.1626)	3.842
c_{uo}	40	42	0.003	0.0044 (0.0023)	0.0072 (0.0020)	0.0044(0.0023)	0.0074 (0.0021)	0.0036 (0.0017)	3.842
c_{di}	40	38	1.9093	1.6919(0.1242)	1.6308 (0.0854)	1.7513 (0.1066)	1.6378 (0.0851)	1.7195 (0.1254)	3.842
c_{ui}	40	42	3.839	3.6855 (0.1829)	3.8171(0.1304)	3.8400 (0.0776)	3.8165 (0.1305)	3.8251 (0.1869)	3.842
N=10000									
Type	X	H	Closed form	Monte Carlo	Antithetic variables	Control variates	MM error adj.	MM path adj.	B&S Vanilla
c_{do}	40	38	1.9327	2.1645 (0.0526)	2.1339 (0.0367)	2.1399 (0.0389)	2.1389 (0.0367)	2.1573 (0.0524)	3.842
c_{uo}	40	42	0.003	0.0047 (0.0007)	0.0058 (0.0005)	0.0047 (0.0007)	0.0054 (0.0005)	0.0044 (0.0007)	3.842
c_{di}	40	38	1.9093	1.6785 (0.0382)	1.6605 (0.0268)	1.6657 (0.0335)	1.6557 (0.0267)	1.6544 (0.0379)	3.842
c_{ui}	40	42	3.839	3.8383 (0.0592)	3.7886 (0.0414)	3.8009 (0.0245)	3.7891 (0.0414)	3.8073 (0.0590)	3.842
N=40000									
Type	X	H	Closed form	Monte Carlo	Antithetic variables	Control variates	MM error adj.	MM path adj.	B&S Vanilla
c_{do}	40	38	1.9327	2.1555 (0.0263)	2.1693 (0.0186)	2.1618 (0.0195)	2.1717 (0.0186)	2.1713 (0.0263)	3.842
c_{uo}	40	42	0.003	0.0060 (0.0004)	0.0060 (0.0003)	0.0060 (0.0004)	0.0061 (0.0003)	0.0060 (0.0004)	3.842
c_{di}	40	38	1.9093	1.6846 (0.0193)	1.6792 (0.0136)	1.6880 (0.0169)	1.6797 (0.0136)	1.6811 (0.0193)	3.842
c_{ui}	40	42	3.839	3.8342 (0.0297)	3.8425 (0.0210)	3.8438 (0.0124)	3.8454 (0.0210)	3.8464 (0.0297)	3.842
S=40, T=0.5, r=ln(1.05), d=0, sigma=0.3									

Table 7: Fixed strike lookback call options: Prices using diferents methods and varing the number of trials. In brackets, the standard error of the estimate. MM refers to Moment Matching.

Fixed strike lookback call option prices					
N=1000					
Closed form	Monte Carlo	Antithetic variables	MM error adj.	MM path adj.	B&S Vanilla
7.6423	6.9633 (0.1954)	7.1927 (0.1385)	7.2039 (0.1385)	7.1001 (0.1978)	3.842
N=10000					
Closed form	Monte Carlo	Antithetic variables	MM error adj.	MM path adj.	B&S Vanilla
7.6423	7.2357 (0.06290)	7.1838 (0.0443)	7.1821 (0.0443)	7.2155 (0.0627)	3.842
N=40000					
Closed form	Monte Carlo	Antithetic variables	MM error adj.	MM path adj.	B&S Vanilla
7.6423	7.1998 (0.0317)	7.2158 (0.0224)	7.2213 (0.0224)	7.2238 (0.0317)	3.842
S=S_{min}=S_{max}=40,X=40, T=0.5, sigma=0.3, r=ln(1.05),d=0					

Table 8: Floating strike lookback call options: Prices using diferents methods and varing the number of trials. In brackets, the standard error of the estimate. MM refers to Moment Matching.

Floating strike lookback call option prices						
N=1000						
Closed form	Monte Carlo	Antithetic variables	Control variates	MM error adj.	MM path adj.	B&S Vanilla
6.7532	6.3664 (0.1865)	6.4379 (0.1319)	6.5249 (0.0771)	6.4462 (0.1320)	6.4903 (0.1894)	1.9229
N=10000						
Closed form	Monte Carlo	Antithetic variables	Control variates	MM error adj.	MM path adj.	B&S Vanilla
6.7532	6.4615 (0.0598)	6.4165 (0.0419)	6.4239 (0.0251)	6.4151 (0.0418)	6.4187 (0.0596)	1.9229
N=40000						
Closed form	Monte Carlo	Antithetic variables	Control variates	MM error adj.	MM path adj.	B&S Vanilla
6.7532	6.4465 (0.0300)	6.4505 (0.0212)	6.4562 (0.0125)	6.4550 (0.0212)	6.4510 (0.0300)	1.9229
S=S_{min}=S_{max}=40, T=.05, sigma=0.3, r=ln(1.05),d=0, N=1000						

Table 9: Geometric average asian call options: Prices using diferents methods and varing the number of trials. In brackets, the standard error of the estimate. MM refers to Moment Matching.

Geometric average asian call option prices					
N=1000					
Closed form	Monte Carlo	Antithetic variables	MM error adj.	MM path adj.	B&S Vanilla
1.6969	1.9818 (0.0987)	2.0471 (0.0701)	2.0432 (0.0700)	1.6291 (0.0892)	3.842
N=10000					
Closed form	Monte Carlo	Antithetic variables	MM error adj.	MM path adj.	B&S Vanilla
1.6969	2.0645 (0.0309)	2.0511 (0.0218)	2.0500 (0.0218)	2.0580 (0.0308)	3.842
N=40000					
Closed form	Monte Carlo	Antithetic variables	MM error adj.	MM path adj.	B&S Vanilla
1.6969	2.0673 (0.0155)	2.0741 (0.0110)	2.0751 (0.0110)	2.0790 (0.0156)	3.842
S=S_{min}=S_{max}=40, X=40, sigma=0.3,T=0.5, r=ln(1.05),d=0					

Table 10: Arithmetic average asian call options: Prices using diferents methods and varing the number of trials. In brackets, the standard error of the estimate. MM refers to Moment Matching.

Arithmetic average asian call option prices						
N=1000						
Aprox closed form	Monte Carlo	Antithetic variables	Control variates	MM error adj.	MM path adj.	B&S Vanilla
2.1768	2.7292 (0.1147)	2.8134 (0.0813)	2.4010 (0.0153)	2.8093 (0.0812)	2.6694 (0.1134)	3.842
N=10000						
Aprox closed form	Monte Carlo	Antithetic variables	Control variates	MM error adj.	MM path adj.	B&S Vanilla
2.1768	2.8402 (0.0360)	2.8200 (0.0254)	0.0047 (0.0047)	2.8185 (0.0254)	2.8318 (0.0359)	3.842
N=40000						
Aprox closed form	Monte Carlo	Antithetic variables	Control variates	MM error adj.	MM path adj.	B&S Vanilla
2.1768	2.8313 (0.0181)	2.8396 (0.0128)	0.0024 (0.0024)	2.8407 (0.0128)	2.8455 (0.0181)	3.842
S=S_{min}=S_{max}=40, X=40,T=0.5, sigma=0.3, r=ln(1.05),d=0						

D Figures

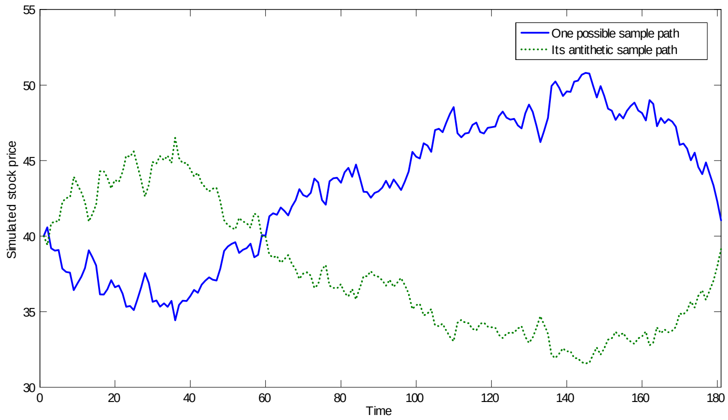

Figure 1: One draw of antithetic paths

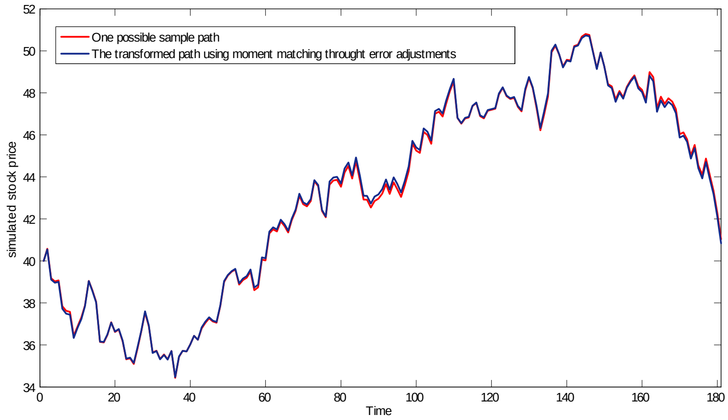

Figure 2: One path and its modification using moment matching throught error adjustments.

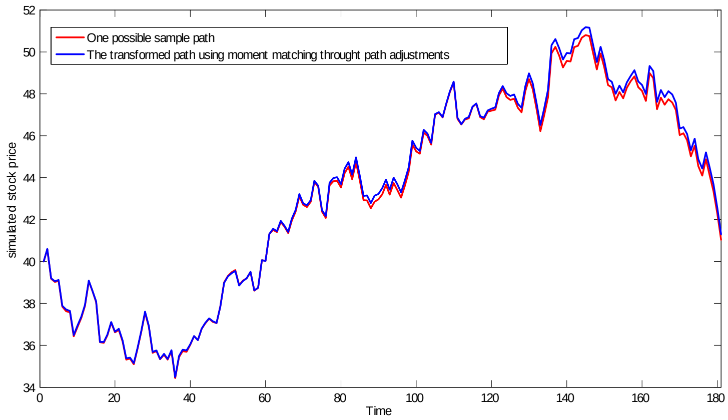

Figure 3: One path and its modification using moment matching throught path adjustments.

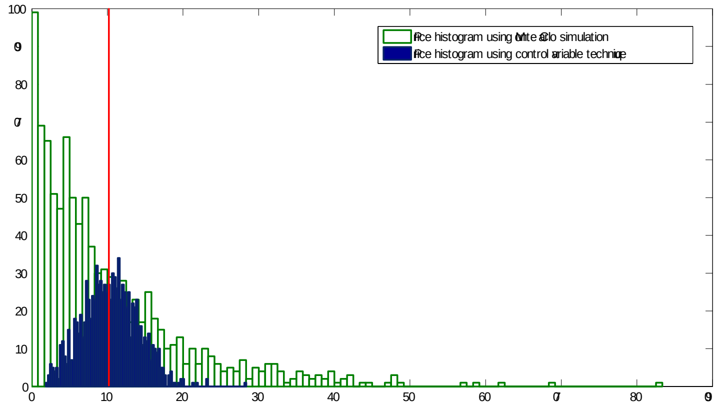

Figure 4: The red line is the true price, computed using the closed form solution.

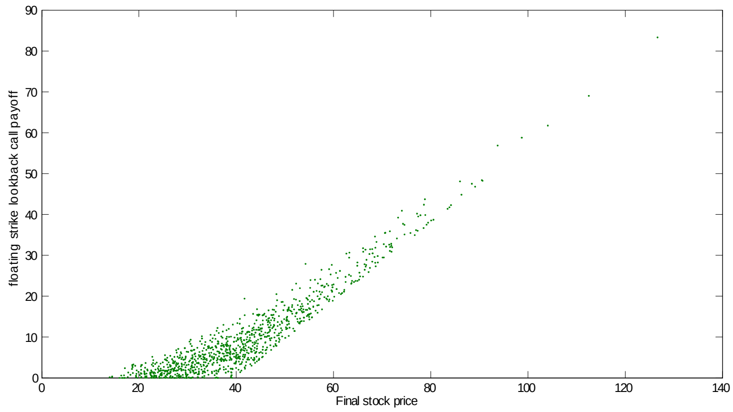

Figure 5: Floating strike lookback call: Final stock price vs payoff for T=1 and sigma=0.35

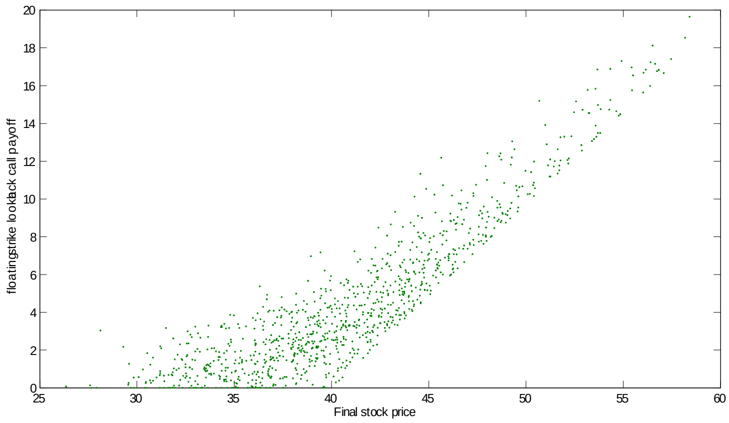

Figure 6: Floating strike lookback call: Final stock price vs payoff for T=0.5 and sigma=0.2

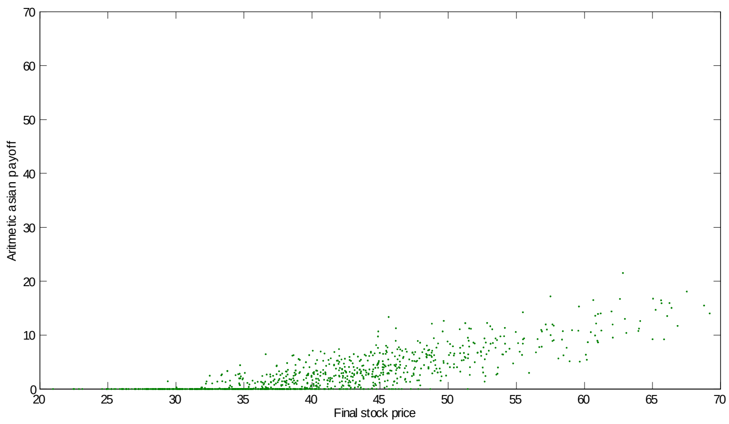

Figure 7: Aritmetic average call asian option: Final stock price vs payoff

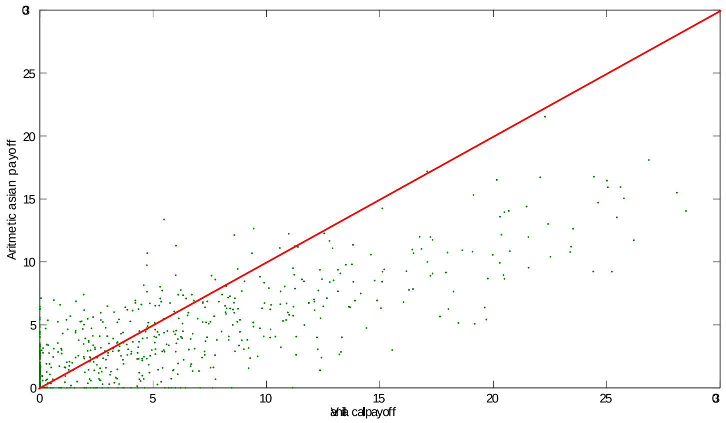

Figure 8: Aritmetic average call asian option payoff vs vanilla call payoff

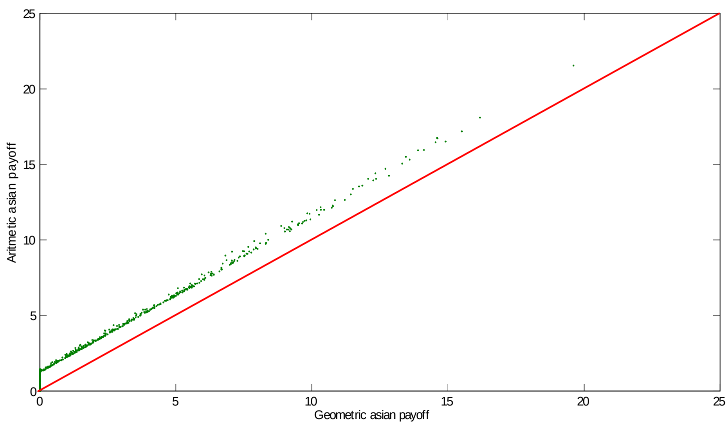

Figure 9: Aritmetic average call asian option payoff vs geometric average call payoff

```
%London School of Economics
%FM442 - MT 2008/9
%Proyect

clc; clear all;

%Inputs we will use
S      = 40;
X      = 40;
r     = log(1.05);
sigma  = 0.3;
T      = 0.5;
d      = 0;

%First of all I generate N whole paths for the stock price
N=1000;
%To simulate the whole path of the stock price I use Euler discretization
%I simulate lnS instead of S because it follows a Wiener process
r_daily=r/360;
delta_t=1/360; %time steps of size of 1 day, hence t(j+1)-t(j)=1/365
No_step= T*360; %if T is 6 months (T=0.5) I generate paths of 180 days
randn('state',100);shocks=randn(No_step,N);%the one I have used for N=1000
%randn('state',10);shocks=randn(No_step,N); %for N=10000
%randn('state',15);shocks=randn(No_step,N); %for N=40000


%Simulating stock price paths
S_T= NaN*ones(No_step+1,N); %each simulated path will be save in one column
S_T(1,:)=S;
for i=2:No_step+1
    S_T(i,:)=S_T(i-1,:).*exp((r-0.5*sigma^2)*delta_t+sigma*sqrt(delta_t).*shocks(i-↙
1,:));
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%BARRIER OPTIONS: 4 types (Up & In, Up & Out, Down & In, Down & Out),
%each type can take the form of a call or a put,giving us a total of 8 types.

%down-and-out call option
H=38; %lower barried
%Price using the closed form solution
%function y=barrier(cp,oi,ud,S,X,H,r,d,T,sigma);
do_call_f=barrier(1,-1,1,S,X,H,r,d,T,sigma)
%Price via Monte Carlo
%I generate a dummy with value 1 if the stock price does not hit the barrier
dummy_barrier=(min(S_T)>H);
payoff=dummy_barrier.*(max(0,S_T(end,:)-X));
payoff_d=payoff*exp(-r*T);
do_call_mc=mean(payoff_d)
do_call_mc_SD=sqrt(var(payoff_d))/(N^0.5)



%down-and-in call option
H=38; %lower barried
```

```
%Price using the closed form solution
%function y=barrier(cp,oi,ud,S,X,H,r,d,T,sigma);
di_call_f=barrier(1,1,1,S,X,H,r,d,T,sigma)
%Price via Monte Carlo
%I generate a dummy with value 1 if the stock price hits the barrier
dummy_barrier=(min(S_T)<H);
payoff=dummy_barrier.*(max(0,S_T(end,:)-X));
payoff_d=payoff*exp(-r*T);
di_call_mc=mean(payoff_d)
di_call_mc_SD=sqrt(var(payoff_d))/(N^0.5)



%up-and-out call option
H=42; %upper barried
%Price using the closed form solution
%function y=barrier(cp,oi,ud,S,X,H,r,d,T,sigma);
uo_call_f=barrier(1,-1,-1,S,X,H,r,d,T,sigma)
%Price via Monte Carlo
%I generate a dummy with value 1 if the stock price does not hit the barrier
dummy_barrier=(max(S_T)<H);
payoff=dummy_barrier.*(max(0,S_T(end,:)-X));
payoff_d=payoff*exp(-r*T);
uo_call_mc=mean(payoff_d)
uo_call_mc_SD=sqrt(var(payoff_d))/(N^0.5)


%up-and-in call option
H=42; %upper barried
%Price using the closed form solution
%function y=barrier(cp,oi,ud,S,X,H,r,d,T,sigma);
ui_call_f=barrier(1,1,-1,S,X,H,r,d,T,sigma)
%Price via Monte Carlo
%I generate a dummy with value 1 if the stock price hits the barrier
dummy_barrier=(max(S_T)>H);
payoff=dummy_barrier.*(max(0,S_T(end,:)-X));
payoff_d=payoff*exp(-r*T);
ui_call_mc=mean(payoff_d)
ui_call_mc_SD=sqrt(var(payoff_d))/(N^0.5)



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%LOOKBACK OPTION

%Fixed strike lookback call
%Price using the closed form solution
%function y=lookback_fixedstrike(cp,S,Smax,Smin,X,r,d,T,sigma);
lookb_call_fx_f=lookback_fixedstrike(1,S,S,S,X,r,d,T,sigma)
%Price by Monte Carlo
S_T_max= max(S_T);
payoff=max(0,S_T_max-X);
payoff_d=payoff*exp(-r*T);
lookb_call_fx_mc=mean(payoff_d)
lookb_call_fx_mc_SD=sqrt(var(payoff_d))/(N^0.5)
```

```
%Fixed strike lookback put
%Price using the closed form solution
%function y=lookback_fixedstrike(cp,S,Smax,Smin,X,r,d,T,sigma);
lookb_put_fx_f=lookback_fixedstrike(-1,S,S,S,X,r,d,T,sigma)
%Price by Monte Carlo
S_T_min= min(S_T);
payoff=max(0,X-S_T_min);
payoff_d=payoff*exp(-r*T);
lookb_put_fx_mc=mean(payoff_d)
lookb_put_fx_mc_SD=sqrt(var(payoff_d))/(N^0.5)

%Floating strike lookback call
%Price using the closed form solution
%function y=lookback_floatingstrike(cp,S,Smax,Smin,r,d,T,sigma);
lookb_call_fl_f=lookback_floatingstrike(1,S,S,S,r,d,T,sigma)
%Price by Monte Carlo
payoff=S_T(end,:)-S_T_min;
payoff_d=payoff*exp(-r*T);
lookb_call_fl_mc=mean(payoff_d)
lookb_call_fl_mc_SD=sqrt(var(payoff_d))/(N^0.5)

%Floating strike lookback put
%Price using the closed form solution
%function y=lookback_floatingstrike(cp,S,Smax,Smin,r,d,T,sigma);
lookb_put_fl_f=lookback_floatingstrike(-1,S,S,S,r,d,T,sigma)
%Price by Monte Carlo
payoff=S_T_max-S_T(end,:);
payoff_d=payoff*exp(-r*T);
lookb_put_fl_mc=mean(payoff_d)
lookb_put_fl_mc_SD=sqrt(var(payoff_d))/(N^0.5)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%ASIAN OPTION

%Geometric average call asian option
%Price using the closed form solution
%function y=asian_geometric(cp,S,X,r,d,T,sigma);
g_asian_call_f=asian_geometric(1,S,X,r,d,T,sigma)
%Price by Monte Carlo
S_T_g_average=prod(S_T.^(1/(No_step+1)));
payoff=max(0,S_T_g_average-X);
payoff_d=payoff*exp(-r*T);
g_asian_call_mc=mean(payoff_d)
g_asian_call_mc_SD=sqrt(var(payoff_d))/(N^0.5)

%Geometric average put asian option
%Price using the closed form solution
%function y=asian_geometric(cp,S,X,r,d,T,sigma);
g_asian_put_f=asian_geometric(-1,S,X,r,d,T,sigma)
%Price by Monte Carlo
payoff=max(0,X-S_T_g_average);
payoff_d=payoff*exp(-r*T);
g_asian_put_mc=mean(payoff_d)
g_asian_put_mc_SD=sqrt(var(payoff_d))/(N^0.5)
```

```
%Aritmetic average call asian option
%Price using the aproximation formula of Turnbull and Wakeman (1991)
%function y=asian_arithmetic_TW(cp,S,SA,X,r,d,T,T2,tau,sigma);
a_asian_call_f_tw=asian_arithmetic_TW(1,S,S,X,r,d,T,T,0,sigma)
%Price by Monte Carlo
S_T_a_average=sum(S_T)./No_step+1;
payoff=max(0,S_T_a_average-X);
payoff_d=payoff*exp(-r*T);
a_asian_call_mc=mean(payoff_d)
a_asian_call_mc_SD=sqrt(var(payoff_d))/(N^0.5)


%Aritmetic average put asian option
%Price using the aproximation formula of Turnbull and Wakeman (1991)
%function y=asian_arithmetic_TW(cp,S,SA,X,r,d,T,T2,tau,sigma);
a_asian_put_f_tw=asian_arithmetic_TW(-1,S,S,X,r,d,T,T,0,sigma)
%Price by Monte Carlo
payoff=max(0,X-S_T_a_average);
payoff_d=payoff*exp(-r*T);
a_asian_put_mc=mean(payoff_d)
a_asian_put_mc_SD=sqrt(var(payoff_d))/(N^0.5)


clear payoff;
clear payoff_d;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%VARIANCE REDUCTION THECHNIQUES

%Antithetic variables
shocks2=[shocks, -shocks];
S_T2= NaN*ones(No_step+1,N*2);
S_T2(1,:)=S;
for i=2:No_step+1
    S_T2(i,:)=S_T2(i-1,:).*exp((r-0.5*sigma^2)*delta_t+sigma*sqrt(delta_t).*shocks2↙
(i-1,:));
end

%Antithetic variables: down-and-out call option
H=38; %lower barried
payoff=(min(S_T2)>H).*(max(0,S_T2(end,:)-X))*exp(-r*T);
do_call_av=mean(payoff)
do_call_av_SD=sqrt(var(payoff))/((2*N)^0.5)

%Antithetic variables:down-and-in call option
H=38; %lower barried
payoff=(min(S_T2)<=H).*(max(0,S_T2(end,:)-X))*exp(-r*T);
di_call_av=mean(payoff)
di_call_av_SD=sqrt(var(payoff))/((2*N)^0.5)

%Antithetic variables:up-and-out call option
H=42; %upper barried
payoff=(max(S_T2)<H).*(max(0,S_T2(end,:)-X))*exp(-r*T);
uo_call_av=mean(payoff)
```

```
uo_call_av_SD=sqrt(var(payoff))/((2*N)^0.5)

%Antithetic variables:up-and-in call option
H=42; %upper barried
payoff=(max(S_T2)>=H).*(max(0,S_T2(end,:)-X))*exp(-r*T);
ui_call_av=mean(payoff)
ui_call_av_SD=sqrt(var(payoff))/((2*N)^0.5)

%Antithetic variables: Fixed strike lookback call
payoff=max(0,max(S_T2)-X)*exp(-r*T);
lookb_call_fx_av=mean(payoff)
lookb_call_fx_av_SD=sqrt(var(payoff))/((2*N)^0.5)

%Antithetic variables: Fixed strike lookback put
payoff=max(0,X-min(S_T2))*exp(-r*T);
lookb_put_fx_av=mean(payoff)
lookb_put_fx_av_SD=sqrt(var(payoff))/((2*N)^0.5)


%Antithetic variables: Floating strike lookback call
%Price by Monte Carlo
payoff=(S_T2(end,:)-min(S_T2))*exp(-r*T);
lookb_call_fl_av=mean(payoff)
lookb_call_fl_av_SD=sqrt(var(payoff))/((2*N)^0.5)

%Antithetic variables:Floating strike lookback put
payoff=(max(S_T2)-S_T2(end,:))*exp(-r*T);
lookb_put_fl_av=mean(payoff)
lookb_put_fl_av_SD=sqrt(var(payoff))/((2*N)^0.5)

%Antithetic variables:Geometric average call asian option
S_T_g_average2=prod(S_T2.^(1/(No_step+1)));
payoff=max(0,S_T_g_average2-X)*exp(-r*T);
g_asian_call_av=mean(payoff)
g_asian_call_av_SD=sqrt(var(payoff))/((2*N)^0.5)

%Antithetic variables:Geometric average put asian option
payoff=max(0,X-S_T_g_average2)*exp(-r*T);
g_asian_put_av=mean(payoff)
g_asian_put_av_SD=sqrt(var(payoff))/((2*N)^0.5)

%Antithetic variables:Aritmetic average call asian option
S_T_a_average2=sum(S_T2)./No_step+1;
payoff=max(0,S_T_a_average2-X)*exp(-r*T);
a_asian_call_av=mean(payoff)
a_asian_call_av_SD=sqrt(var(payoff))/((2*N)^0.5)

%Antithetic variables: Aritmetic average put asian option
payoff=max(0,X-S_T_a_average2)*exp(-r*T);
a_asian_put_av=mean(payoff)
a_asian_put_av_SD=sqrt(var(payoff))/((2*N)^0.5)

clear payoff
```

```
%Control variates: Underlying asset

%down-and-out call option
H=38; %lower barried
payoff=(min(S_T)>H).*(max(0,S_T(end,:)-X))*exp(-r*T);
c=cov(payoff, S_T(end,:));
beta=c(2,1)/c(2,2);
payoff=payoff-beta*(S_T(end,:)-exp(r*T)*S);
do_call_cv=mean(payoff)
do_call_cv_SD=sqrt(var(payoff))/(N^0.5)

%down-and-in call option
H=38; %lower barried
payoff=(min(S_T)<H).*(max(0,S_T(end,:)-X))*exp(-r*T);
c=cov(payoff, S_T(end,:));
payoff=payoff-(c(2,1)/c(2,2))*(S_T(end,:)-exp(r*T)*S);
di_call_cv=mean(payoff)
di_call_cv_SD=sqrt(var(payoff))/(N^0.5)

%up-and-out call option
H=42; %upper barried
payoff=(max(S_T)<H).*(max(0,S_T(end,:)-X))*exp(-r*T);
c=cov(payoff, S_T(end,:));
payoff=payoff-(c(2,1)/c(2,2))*(S_T(end,:)-exp(r*T)*S);
uo_call_cv=mean(payoff)
uo_call_cv_SD=sqrt(var(payoff))/(N^0.5)

%up-and-in call option
H=42; %upper barried
payoff=(max(S_T)>H).*(max(0,S_T(end,:)-X))*exp(-r*T);
c=cov(payoff, S_T(end,:));
payoff=payoff-(c(2,1)/c(2,2))*(S_T(end,:)-exp(r*T)*S);
ui_call_cv=mean(payoff)
ui_call_cv_SD=sqrt(var(payoff))/(N^0.5)

%Floating strike lookback call
payoff=(S_T(end,:)-S_T_min)*exp(-r*T);
c=cov(payoff, S_T(end,:));
payoff=payoff-(c(2,1)/c(2,2))*(S_T(end,:)-exp(r*T)*S);
lookb_call_fl_cv=mean(payoff)
lookb_call_fl_cv_SD=sqrt(var(payoff))/(N^0.5)


%Control variates: Tractable options: Asian option

%Aritmetic average call asian option
payoff=max(0,S_T_a_average-X)*exp(-r*T);
payoff_c=max(0,S_T_g_average-X)*exp(-r*T);
c=cov(payoff,payoff_c);
payoff=payoff-(c(2,1)/c(2,2))*(payoff_c-g_asian_call_f);
a_asian_call_cv=mean(payoff)
a_asian_call_cv_SD=sqrt(var(payoff))/(N^0.5)

%Aritmetic average put asian option
payoff=max(0,X-S_T_a_average)*exp(-r*T);
```

```
payoff_c=max(0,X-S_T_g_average)*exp(-r*T);
c=cov(payoff,payoff_c);
payoff=payoff-(c(2,1)/c(2,2))*(payoff_c-g_asian_put_f);
a_asian_put_cv=mean(payoff)
a_asian_put_cv_SD=sqrt(var(payoff))/(N^0.5)

clear payoff;
clear payoff_c;
clear c;



%Moment matching throught error adjustments
%I use the errors using form the antithetic variable technique, hence,
%I only have match the second moment
%for each time step I will compute the sample SD of the shocks
v=var(shocks2');
for i=1:No_step
    shocks3(i,:)=[shocks2(i,:)./v(i)];
end
clear v;
S_T3= NaN*ones(No_step+1,N*2);
S_T3(1,:)=S;
for i=2:No_step+1
    S_T3(i,:)=S_T3(i-1,:).*exp((r-0.5*sigma^2)*delta_t+sigma*sqrt(delta_t).*shocks3↙
(i-1,:));
end

%Moment matching throught error adjustments: down-and-out call option
H=38; %lower barried
payoff=(min(S_T3)>H).*(max(0,S_T3(end,:)-X))*exp(-r*T);
do_call_mm=mean(payoff)
do_call_mm_SD=sqrt(var(payoff))/((2*N)^0.5)

%Moment matching throught error adjustments:down-and-in call option
H=38; %lower barried
payoff=(min(S_T3)<=H).*(max(0,S_T3(end,:)-X))*exp(-r*T);
di_call_mm=mean(payoff)
di_call_mm_SD=sqrt(var(payoff))/((2*N)^0.5)

%Moment matching throught error adjustments:up-and-out call option
H=42; %upper barried
payoff=(max(S_T3)<H).*(max(0,S_T3(end,:)-X))*exp(-r*T);
uo_call_mm=mean(payoff)
uo_call_mm_SD=sqrt(var(payoff))/((2*N)^0.5)

%Moment matching throught error adjustments:up-and-in call option
H=42; %upper barried
payoff=(max(S_T3)>=H).*(max(0,S_T3(end,:)-X))*exp(-r*T);
ui_call_mm=mean(payoff)
ui_call_mm_SD=sqrt(var(payoff))/((2*N)^0.5)

%Moment matching throught error adjustments: Fixed strike lookback call
payoff=max(0,max(S_T3)-X)*exp(-r*T);
lookb_call_fx_mm=mean(payoff)
lookb_call_fx_mm_SD=sqrt(var(payoff))/((2*N)^0.5)
```

```
%Moment matching throught error adjustments: Fixed strike lookback put
payoff=max(0,X-min(S_T3))*exp(-r*T);
lookb_put_fx_mm=mean(payoff)
lookb_put_fx_mm_SD=sqrt(var(payoff))/((2*N)^0.5)

%Moment matching throught error adjustments: Floating strike lookback call
payoff=(S_T3(end,:)-min(S_T3))*exp(-r*T);
lookb_call_fl_mm=mean(payoff)
lookb_call_fl_mm_SD=sqrt(var(payoff))/((2*N)^0.5)

%Moment matching throught error adjustments:Floating strike lookback put
payoff=(max(S_T3)-S_T3(end,:))*exp(-r*T);
lookb_put_fl_mm=mean(payoff)
lookb_put_fl_mm_SD=sqrt(var(payoff))/((2*N)^0.5)

%Moment matching throught error adjustments:Geometric average call asian
S_T_g_average3=prod(S_T3.^(1/(No_step+1)));
payoff=max(0,S_T_g_average3-X)*exp(-r*T);
g_asian_call_mm=mean(payoff)
g_asian_call_mm_SD=sqrt(var(payoff))/((2*N)^0.5)

%Moment matching throught error adjustments:Geometric average put asian
payoff=max(0,X-S_T_g_average3)*exp(-r*T);
g_asian_put_mm=mean(payoff)
g_asian_put_mm_SD=sqrt(var(payoff))/((2*N)^0.5)

%Moment matching throught error adjustments:Aritmetic average call asian option
S_T_a_average3=sum(S_T3)./No_step+1;
payoff=max(0,S_T_a_average3-X)*exp(-r*T);
a_asian_call_mm=mean(payoff)
a_asian_call_mm_SD=sqrt(var(payoff))/((2*N)^0.5)

%Moment matching throught error adjustments: Aritmetic average put asian option
payoff=max(0,X-S_T_a_average3)*exp(-r*T);
a_asian_put_mm=mean(payoff)
a_asian_put_mm_SD=sqrt(var(payoff))/((2*N)^0.5)

clear payoff;


%Moment matching throught path adjustments
%for each time step I will compute the sample mean of the stock prices
%and transform the simulated paths by the multiplicative adjustment
mu=mean(S_T');
for i=1:No_step+1
    S_T4(i,:)=[S_T(i,:).*(S*exp(r*i/360)/mu(i))];
end
clear mu;

%Moment matching throught path adjustments: down-and-out call option
H=38; %lower barried
payoff=(min(S_T4)>H).*(max(0,S_T4(end,:)-X))*exp(-r*T);
do_call_mmp=mean(payoff)
do_call_mmp_SD=sqrt(var(payoff))/(N^0.5)
```

```
%Moment matching throught path adjustments:down-and-in call option
H=38; %lower barried
payoff=(min(S_T4)<=H).*(max(0,S_T4(end,:)-X))*exp(-r*T);
di_call_mmp=mean(payoff)
di_call_mmp_SD=sqrt(var(payoff))/(N^0.5)

%Moment matching throught path adjustments:up-and-out call option
H=42; %upper barried
payoff=(max(S_T4)<H).*(max(0,S_T4(end,:)-X))*exp(-r*T);
uo_call_mmp=mean(payoff)
uo_call_mmp_SD=sqrt(var(payoff))/(N^0.5)

%Moment matching throught path adjustments:up-and-in call option
H=42; %upper barried
payoff=(max(S_T4)>=H).*(max(0,S_T4(end,:)-X))*exp(-r*T);
ui_call_mmp=mean(payoff)
ui_call_mmp_SD=sqrt(var(payoff))/(N^0.5)

%Moment matching throught path adjustments: Fixed strike lookback call
payoff=max(0,max(S_T4)-X)*exp(-r*T);
lookb_call_fx_mmp=mean(payoff)
lookb_call_fx_mmp_SD=sqrt(var(payoff))/(N^0.5)

%Moment matching throught path adjustments: Fixed strike lookback put
payoff=max(0,X-min(S_T4))*exp(-r*T);
lookb_put_fx_mmp=mean(payoff)
lookb_put_fx_mmp_SD=sqrt(var(payoff))/(N^0.5)

%Moment matching throught path adjustments: Floating strike lookback call
payoff=(S_T4(end,:)-min(S_T4))*exp(-r*T);
lookb_call_fl_mmp=mean(payoff)
lookb_call_fl_mmp_SD=sqrt(var(payoff))/(N^0.5)

%Moment matching throught path adjustments:Floating strike lookback put
payoff=(max(S_T4)-S_T4(end,:))*exp(-r*T);
lookb_put_fl_mmp=mean(payoff)
lookb_put_fl_mmp_SD=sqrt(var(payoff))/(N^0.5)

%Moment matching throught path adjustments:Geometric average call asian
S_T_g_average4=prod(S_T4.^(1/(No_step+1)));
payoff=max(0,S_T_g_average4-X)*exp(-r*T);
g_asian_call_mmp=mean(payoff)
g_asian_call_mmp_SD=sqrt(var(payoff))/(N^0.5)

%Moment matching throught path adjustments:Geometric average put asian
payoff=max(0,X-S_T_g_average4)*exp(-r*T);
g_asian_put_mmp=mean(payoff)
g_asian_put_mmp_SD=sqrt(var(payoff))/(N^0.5)

%Moment matching throught path adjustments:Aritmetic average call asian option
S_T_a_average4=sum(S_T4)./No_step+1;
payoff=max(0,S_T_a_average4-X)*exp(-r*T);
a_asian_call_mmp=mean(payoff)
a_asian_call_mmp_SD=sqrt(var(payoff))/(N^0.5)
```

```
%Moment matching throught path adjustments: Aritmetic average put asian option
payoff=max(0,X-S_T_a_average4)*exp(-r*T);
a_asian_put_mmp=mean(payoff)
a_asian_put_mmp_SD=sqrt(var(payoff))/(N^0.5)

clear payoff;

%finally to compare prices we compute the price of the vanilla options
%using the Black & Scholes formula
[call_bs, put_bs] = blsprice(S,X,r,T,sigma,d)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%To make the tables is easier to construct vectors with the prices and SD

do_call=[do_call_f,do_call_mc,do_call_mc_SD,do_call_av, do_call_av_SD,do_call_cv,↙
do_call_cv_SD, do_call_mm ,do_call_mm_SD, do_call_mmp ,do_call_mmp_SD]

di_call=[di_call_f,di_call_mc,di_call_mc_SD,di_call_av, di_call_av_SD,di_call_cv,↙
di_call_cv_SD, di_call_mm ,di_call_mm_SD, di_call_mmp ,di_call_mmp_SD]

uo_call=[uo_call_f,uo_call_mc,uo_call_mc_SD,uo_call_av, uo_call_av_SD,uo_call_cv,↙
uo_call_cv_SD, uo_call_mm ,uo_call_mm_SD, uo_call_mmp ,uo_call_mmp_SD]

ui_call=[ui_call_f,ui_call_mc,ui_call_mc_SD,ui_call_av, ui_call_av_SD,ui_call_cv,↙
ui_call_cv_SD, ui_call_mm ,ui_call_mm_SD, ui_call_mmp ,ui_call_mmp_SD]

lookb_call_fx=[lookb_call_fx_f,lookb_call_fx_mc,lookb_call_fx_mc_SD,lookb_call_fx_av,↙
lookb_call_fx_av_SD, lookb_call_fx_mm ,lookb_call_fx_mm_SD, lookb_call_fx_mmp ,↙
lookb_call_fx_mmp_SD]

lookb_call_fl=[lookb_call_fl_f,lookb_call_fl_mc,lookb_call_fl_mc_SD,lookb_call_fl_av,↙
lookb_call_fl_av_SD, lookb_call_fl_cv, lookb_call_fl_cv_SD, lookb_call_fl_mm ,↙
lookb_call_fl_mm_SD, lookb_call_fl_mmp ,lookb_call_fl_mmp_SD]

g_asian_call=[g_asian_call_f,g_asian_call_mc,g_asian_call_mc_SD,g_asian_call_av,↙
g_asian_call_av_SD, g_asian_call_mm ,g_asian_call_mm_SD, g_asian_call_mmp ,↙
g_asian_call_mmp_SD]

a_asian_call=[a_asian_call_f_tw,a_asian_call_mc,a_asian_call_mc_SD,a_asian_call_av,↙
a_asian_call_av_SD, a_asian_call_cv_SD, a_asian_call_cv_SD, a_asian_call_mm ,↙
a_asian_call_mm_SD, a_asian_call_mmp ,a_asian_call_mmp_SD]

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Plot one possible sample path and his Antithetic sample path
plot(S_T2(:,1))
hold on
plot(S_T2(:,1001),'g')
hold off
xlabel('Time')
ylabel('simulated stock price')
title('One draw of antithetic sample paths')

%Plot one possible sample path and equivalent obtained using moment
%matching throught error adjustments
```

```
plot(S_T2(:,1))
hold on
plot(S_T3(:,1),'g')
hold off
xlabel('Time')
ylabel('simulated stock price')
title('Moment matching throught error adjustments')

%Plot one possible sample path and equivalent obtained using moment
%matching throught error adjustments
plot(S_T2(:,1))
hold on
plot(S_T4(:,1),'g')
hold off
xlabel('Time')
ylabel('simulated stock price')
title('Moment matching throught path adjustments')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Histogram for the price of a floating strike lookback call

%Price by Monte Carlo
payoff_mc=(S_T(end,:)-S_T_min)*exp(-r*T);

%Antithetic variables
payoff_av(1,1:(N/2))=(S_T2(end,(N/2)+1:N)-min(S_T2(:,(N/2)+1:N)))*exp(-r*T);
payoff_av(1,((N/2)+1):N)=(S_T2(end,(3*N/2)+1:end)-min(S_T2(:,(3*N/2)+1:end)))*exp(-↙
r*T);
% I do it because I want to have the same number of simulations, ie N
%Also I want antithetic simulations

%Control variates: Underlying asset
payoff_cv=(S_T(end,:)-S_T_min)*exp(-r*T);
c=cov(payoff_cv, S_T(end,:));
payoff_cv=payoff_cv-(c(2,1)/c(2,2))*(S_T(end,:)-exp(r*T)*S);

%Moment matching throught error adjustments
payoff_mm(1,1:(N/2))=(S_T3(end,(N/2)+1:N)-min(S_T3(:,(N/2)+1:N)))*exp(-r*T);
payoff_mm(1,((N/2)+1):N)=(S_T3(end,(3*N/2)+1:end)-min(S_T3(:,(3*N/2)+1:end)))*exp(-↙
r*T);

%Moment matching throught path adjustments
payoff_mmp=(S_T4(end,:)-min(S_T4))*exp(-r*T);

hist(payoff_mc, 100);h = findobj(gca,'Type','patch');
set(h,'FaceColor','w','EdgeColor','g');
hold on
hist(payoff_cv,100);
hold off

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%plot bias and standard error versus sample size
%Case of floating strike lookback call
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%floating strike lookback call
%correlation between the payoff and the final price
plot(S_T(end,:) ,payoff_mc,'.g' )
xlabel('Final stock price')
ylabel('floating strike lookback call payoff')
title('Final stock price vs payoff')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Arithmetic asian option
%correlation between the payoff and differents objects

%Aritmetic average call asian option
payoff=max(0,S_T_a_average-X).*exp(-r*T);
payoff_vanilla=max(0,S_T(end,:)-X).*exp(-r*T);
payoff_g=max(0,S_T_g_average-X).*exp(-r*T);

plot(S_T(end,:) ,payoff,'.g' )
xlabel('Final stock price')
ylabel('Aritmetic asian payoff')
title('Aritmetic average call asian option: Final stock price vs payoff')

plot(payoff_vanilla ,payoff,'.g' )
xlabel('Vanilla call payoff')
ylabel('Aritmetic asian payoff')
title('Aritmetic average call asian option payoff vs vanilla call payoff')

plot(payoff_g ,payoff,'.g' )
xlabel('Geometric asian payoff')
ylabel('Aritmetic asian payoff')
title('Aritmetic average call asian option payoff vs geometric average call payoff')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
function y=barrier(cp,oi,ud,S,X,H,r,d,T,sigma);

%function y=barrier(cp,oi,ud,S,X,H,r,d,T,sigma);
%
%This function gives us the price of the single barrier.
%This option comes in 4 types (Up & In, Up & Out, Down & In, Down & Out),
%each type can take the form of a call or a put,giving us a total of 8 types.
%The price of the option depends on the relationship between the barrier
%level and the strike price.
%
% inputs:
% cp - callputflag = 1 if call, -1 if put
% oi - outinflag = 1 if in, -1 if out
% ud - updownflag = -1 if up, 1 if down
% S - current spot
% X - strike
% H - Barrier
% r - risk-free rate
% d - dividend rate
% T - time to expiry
% sigma - volatility
%
% outputs:
% y - current price of the option
%
mu=((r-d)-sigma^2/2)/sigma^2;
x1=(log(S/X)/(sigma*(T)^0.5))+(1+mu)*sigma*(T)^0.5;
x2=(log(S/H)/(sigma*(T)^0.5))+(1+mu)*sigma*(T)^0.5;
y1=(log(H^2/(S*X))/(sigma*(T)^0.5))+(1+mu)*sigma*(T)^0.5;
y2=(log(H/S)/(sigma*(T)^0.5))+(1+mu)*sigma*(T)^0.5;

A=cp*S*exp(-d*T)*normcdf(cp*x1)-cp*X*exp(-r*T)*normcdf(cp*x1-cp*sigma*T^0.5);
B=cp*S*exp(-d*T)*normcdf(cp*x2)-cp*X*exp(-r*T)*normcdf(cp*x2-cp*sigma*T^0.5);
C=cp*S*exp(-d*T)*(H/S)^(2*(mu+1))*normcdf(ud*y1)-cp*X*exp(-r*T)*(H/S)^(2*mu)*normcdf↙
(ud*y1-ud*sigma*(T)^0.5);
D=cp*S*exp(-d*T)*(H/S)^(2*(mu+1))*normcdf(ud*y2)-cp*X*exp(-r*T)*(H/S)^(2*mu)*normcdf↙
(ud*y2-ud*sigma*(T)^0.5);

if cp==1 & oi==1 & X>H & ud==1;
    y=C;
elseif cp==1 & oi==1 & X<H & ud==1;
    y=A-B+D;
elseif cp==1 & oi==1 & X>H & ud==-1;
    y=A;
elseif cp==1 & oi==1 & X<H & ud==-1;
    y=B-C+D;
elseif cp==-1 & oi==1 & X>H & ud==1;
    y=B-C+D;
elseif cp==-1 & oi==1 & X<H & ud==1;
    y=A;
elseif cp==-1 & oi==1 & X>H & ud==-1;
    y=A-B+D;
elseif cp==-1 & oi==1 & X<H & ud==-1;
    y=C;
elseif cp==1 & oi==-1 & X>H & ud==1;
```

```
    y=A-C;
elseif cp==1 & oi==-1 & X<H & ud==1;
    y=B-D;
elseif cp==1 & oi==-1 & X>H & ud==-1;
    y=0;
elseif cp==1 & oi==-1 & X<H & ud==-1;
    y=A-B+C-D;
elseif cp==-1 & oi==-1 & X>H & ud==1;
    y=A-B+C-D;
elseif cp==-1 & oi==-1 & X<H & ud==1;
    y=0;
elseif cp==-1 & oi==-1 & X>H & ud==-1;
    y=B-D;
elseif cp==-1 & oi==-1 & X<H & ud==-1;
    y=A-C;
else;
    y=NaN;
end
```

```
function y=lookback_fixedstrike(cp,S,Smax,Smin,X,r,d,T,sigma);

%function y=lookback_fixedstrike(cp,S,Smax,Smin,X,r,d,T,sigma);
%
%This function gives us the price of a fixed strike lookback option
%For a call the price has a particular form if X<=Smax
%For a put the price has a particular form if X>=Smin
%
% inputs:
% cp - callputflag = 1 if call, -1 if put
% S - current spot
% Smax - maximum stock price observed so far
% Smin - minimum stock price observed so far
% r - risk-free rate
% d - dividend rate
% T - time to expiry
% sigma - volatility
%
% outputs:
% y - current price of the option
%
b=r-d;
d1=(log(S/X)+(((r-d)+sigma^2/2)*T))/(sigma*(T)^0.5);
d2=d1-(sigma*(T)^0.5);
e1=(log(S/Smax)+(((r-d)+sigma^2/2)*T))/(sigma*(T)^0.5);
e2=e1-(sigma*(T)^0.5);
f1=(log(S/Smin)+(((r-d)+sigma^2/2)*T))/(sigma*(T)^0.5);
f2=f1-(sigma*(T)^0.5);

if cp==1 & X>Smax;
    y=S*exp(-d*T)*normcdf(d1)-X*exp(-r*T)*normcdf(d2)+S*exp(-r*T)*(sigma^2/(2*(r-d)))↙
*(-1*((S/X)^(-2*(r-d)/sigma^2))*normcdf(d1-(2*b/sigma)*T^0.5)+exp((r-d)*T)*normcdf↙
(d1));
elseif cp==1 & X<=Smax;
    y=exp(-r*T)*(Smax-X)+S*exp(-d*T)*normcdf(e1)-Smax*exp(-r*T)*normcdf(e2)+S*exp(-↙
r*T)*(sigma^2/(2*(r-d)))*(-1*((S/Smax)^(-2*(r-d)/sigma^2))*normcdf(e1-(2*b/sigma)↙
*T^0.5)+exp((r-d)*T)*normcdf(e1));
elseif cp==-1 & X<Smin;
    y=X*exp(-r*T)*normcdf(-d2)-S*exp(-d*T)*normcdf(-d1)+S*exp(-r*T)*(sigma^2/(2*(r-↙
d)))*(((S/X)^(-2*(r-d)/sigma^2))*normcdf(-d1+(2*b/sigma)*T^0.5)-exp((r-d)*T)*normcdf↙
(-d1));
elseif cp==-1 & X>=Smin;
    y=exp(-r*T)*(X-Smin)-S*exp(-d*T)*normcdf(-f1)+Smin*exp(-r*T)*normcdf(-f2)+S*exp(-↙
r*T)*(sigma^2/(2*(r-d)))*(((S/Smin)^(-2*(r-d)/sigma^2))*normcdf(-f1+(2*b/sigma)*T^0.↙
5)-exp((r-d)*T)*normcdf(-f1));
else;
    y=NaN;
end
```

```
function y=lookback_floatingstrike(cp,S,Smax,Smin,r,d,T,sigma);

%function y=lookback_floatingstrike(cp,S,Smax,Smin,r,d,T,sigma);
%
%This function gives us the price of a floating strike lookback option
%
% inputs:
% cp - callputflag = 1 if call, -1 if put
% S - current spot
% Smax - maximum stock price observed so far
% Smin - minimum stock price observed so far
% r - risk-free rate
% d - dividend rate
% T - time to expiry
% sigma - volatility
%
% outputs:
% y - current price of the option
%
b=r-d;
a1=(log(S/Smin)+(((r-d)+sigma^2/2)*T))/(sigma*(T)^0.5);
a2=a1-(sigma*(T)^0.5);
b1=(log(S/Smax)+(((r-d)+sigma^2/2)*T))/(sigma*(T)^0.5);
b2=b1-(sigma*(T)^0.5);

if cp==1;
    y=S*exp(-d*T)*normcdf(a1)-Smin*exp(-r*T)*normcdf(a2)+S*exp(-r*T)*(sigma^2/(2*(r-
d)))*(((S/Smin)^(-2*(r-d)/sigma^2))*normcdf(-a1+(2*b/sigma)*T^0.5)-exp((r-d)*T)
*normcdf(-a1));
elseif cp==-1;
    y=Smax*exp(-r*T)*normcdf(-b2)-S*exp(-d*T)*normcdf(-b1)+S*exp(-r*T)*(sigma^2/(2*
(r-d)))*(-1*((S/Smax)^(-2*(r-d)/sigma^2))*normcdf(b1-(2*b/sigma)*T^0.5)+exp((r-d)*T)
*normcdf(b1));
else;
    y=NaN;
end
```

```
function y=asian_geometric(cp,S,X,r,d,T,sigma);

%function y=asian_geometric(cp,S,X,r,d,T,sigma);
%
%This function gives us the price of a geometric average-rate asian option
%This option can be price as a standard option by changing the volatility
%and cost-of-carry term (b=r-d)
%
% inputs:
% cp - callputflag = 1 if call, -1 if put
% S - current spot
% X - strike price
% r - risk-free rate
% d - dividend rate
% T - time to expiry
% sigma - volatility
%
% outputs:
% y - current price of the option
%
sigmaA=sigma/3^0.5;
b=r-d;
bA=(b-((sigma^0.5)/6))/2;
d1=(log(S/X)+((bA+sigmaA^2/2)*T))/(sigmaA*(T)^0.5);
d2=d1-(sigmaA*(T)^0.5);

if cp==1
    y=S*exp((bA-r)*T)*normcdf(d1)-X*exp(-r*T)*normcdf(d2);
elseif cp==-1;
    y=X*exp(-r*T)*normcdf(-d2)-S*exp((bA-r)*T)*normcdf(-d1);
else;
    y=NaN;
end
```

```
function y=asian_arithmetic_TW(cp,S,SA,X,r,d,T,T2,tau,sigma);

%function y=asian_arithmetic_TW(cp,S,SA,X,r,d,T,T2,tau,sigma);
%
%This function gives us the price of a arithmetic average-rate asian option
%using the aproximation formula of Turnbull and Wakeman (1991)
%
% inputs:
% cp - callputflag = 1 if call, -1 if put
% S - current spot
% SA - arithmetic average asset price during the observed period T1=T-T2
% X - strike price
% r - risk-free rate
% d - dividend rate
% T - original time to maturity
% T2 - remaining time to maturity
% tau - time to the beginning of the average period
% sigma - volatility
%
% outputs:
% y - current price of the option
%
b=r-d;
M1=(exp(b*T)-exp(b*tau))/(b*(T-tau));
M2=2*exp((2*b+sigma^2)*T)/((b+sigma^2)*(2*b+sigma^2)*(T-tau)^2)+2*exp((2*b+sigma^2)↙
*tau)/(b*(T-tau)^2)*(1/(2*b+sigma^2)-exp(b*(T-tau))/(b+sigma^2));
bA=log(M1)/T;
sigmaA=sqrt(log(M2)/T-2*bA);
T1=T-T2;
X=(T/T2)*X-(T1/T2)*SA;
d1=(log(S/X)+((bA+sigmaA^2/2)*T2))/(sigmaA*(T2)^0.5);
d2=d1-(sigmaA*(T2)^0.5);

if cp==1
    y=(S*exp((bA-r)*T2)*normcdf(d1)-X*exp(-r*T2)*normcdf(d2))*(T2/T);
elseif cp==-1;
    y=(X*exp(-r*T2)*normcdf(-d2)-S*exp((bA-r)*T2)*normcdf(-d1))*(T2/T);
else;
    y=NaN;
end
```