UPPSALA
UNIVERSITET

U.U.D.M. Project Report 2009:8

Pricing American Options using Monte Carlo Methods

Quiyi Jia

Examensarbete i matematik, 30 hp
Handledare och examinator: Johan Tysk
Juni 2009

Department of Mathematics
Uppsala University

Abstract

The pricing of options is a very important problem encountered in financial markets today. Many problems in mathematical finance entail the computation of a particular integral. In many cases these integrals can be valued analytically, and in still more cases they can be valued using numerical integration, or computed using a partial differential equation (PDE). The famous Black-Scholes model, for instance, provides explicit closed form solutions for the values of certain (European style) call and put options. However when the number of dimensions in the problem is large, PDEs and numerical integrals become intractable, the formulas exhibiting them are complicated and difficult to evaluate accurately by conventional methods. In these cases, Monte Carlo methods often give better results, because they have proved to be valuable and flexible computational tools to calculate the value of options with multiple sources of uncertainty or with complicated features.

A number of Monte Carlo simulation-based approaches have been proposed within the past decade to address the problem of pricing American-style derivatives. The purpose of this paper is to discuss some of the recent applications of Monte Carlo methods to American option pricing problems. Our results suggest that the Least Squares Monte Carlo method is more suitable for problems in higher dimensions than other comparable Monte Carlo methods.

Keywords: American Options, Monte Carlo Simulation, Quasi-Monte Carlo Methods, Least Squares Monte Carlo, Options Pricing, Multiple Underlying Assets

Contents

1. Introduction

1.1 Theoretical Background

Option pricing is an important area of research in the finance community. Actually, at the beginning, as a result of many problems in applying simulation, the primary methods for pricing American options are binomial trees and other lattice methods, such as trinomial trees, and finite difference methods to solve the associated boundary value partial differential equations (PDEs). Due to the complexity of the underlying dynamics, analytical models for option pricing entail many restrictive assumptions, so for real-world applications approximate numerical methods are employed, especially for American options, these include the valuation of options, the estimation of their sensitivities, risk analysis, and stress testing of portfolios. But, in recent years the complexity of numerical computation in financial theory and practice has increased enormously, putting more demands on computational speed and efficiency.

One of the most popular numerical techniques in option pricing is Monte Carlo simulation that was coined by Stanislaw Ulam in the 1940's. The Monte Carlo approach simulates paths for asset prices. For the n-dimension problem, Monte Carlo methods could converge to the solution more quickly, require less memory and are easier to program. In contrast to simpler situations, simulation is not the better solution because it is very time-consuming and computationally intensive.

Since the convergence rate of Monte Carlo methods is generally independent of the number of state variables, it is clear that they become viable as the underlying models (asset prices, volatilities and interest rates) and derivative contracts themselves (defined on path-dependent functions or multiple assets) become more complicated, Phelim Boyle was among the first to propose using Monte Carlo simulation to study option pricing in 1977 (for European options). Since then, other important examples of this literature include Hull and White (1987), Johnson and Shanno (1987), Scott (1987), and Figlewski (1992), have employed Monte Carlo simulation for analyzing options markets. The advantage of the approach is obvious, as Boyle (1977) has stated, “the Monte Carlo method should prove most valuable in situations where it is difficult if not impossible to proceed using a more accurate approach.” Its disadvantages are its computational inefficiency when compared to most other numerical methods. Later, Bossaerts (1989) solves for the exercise strategy that maximizes the simulated value of the option. With respect to using Monte Carlo simulation to perform pricing of options with early exercise features, more early work includes Tilley (1993) and Grant, Vora, and Weeks (1997). Tilley was the first person who attempt to apply simulation to American option pricing, using a bundling technique and a backward induction algorithm. His approach is a “single pass” algorithm, in that all simulations are carried out first before the algorithm is applied. Whereas the work in 1997 considers more general path-dependent options such as Asian options, the approach is sequential in its use of simulation, proceeding inductively backwards to approximate the exercise boundary at each early exercise point, finally estimating the price in a forward simulation based on the obtained boundaries. And with improvements on the basic idea of Tilley, Carriere (1996) presents a backward induction algorithm and applies it to calculate the early exercise premium. He showed that the estimation of the decision rule to exercise early should be equivalent to the estimation of a series of conditional expectations. The conditional expectations are estimated using local regressions. And in the same year, M. Broadie and P.Glasserman showed how to price Asian options by Monte Carlo. Fu and Hu (1995) treated a simple American call option with discrete dividends by casting the pricing problem as an optimization problem of maximizing the expected payoff of the contract under the risk-neutral measure with respect to the early exercise thresholds. Moreover, Averbukh (1997), Broadie and Glasserman (1997) used the likelihood ratio method alluded to earlier and perturbation analysis, and the use of perturbation analysis was restricted to infinitesimal perturbation analysis, and they considered only European and Asian options, but without early exercise features, as in American options.

Longstaff and Schwartz (1999), who used least-squares regression on polynomials to approximate the early exercise boundary, and Tsitsiklis and Van Roy (1999) studied perpetual American options and proposed a stochastic algorithm that could approximate the conditional expectations by a linear combination of basis functions. They also analyzed the convergence properties of this algorithm. In general, there is no closed-form expression for the conditional expectation function, and they selected a set of basis functions such that their weighted combination is “close” to the true function. Later, Tsitsiklis and Van Roy (2001) analyzed finite-horizon pricing problems. They presented an algorithm for pricing complex American options that involves the evaluation of value functions at a finite set of “representative” elements of the state space. A linear combination of basis functions is fitted to the data via least-squares regressions, in order to approximate the conditional expectation over the entire state space. These authors provide convergence results and error bounds for their algorithm. And also in 2001, F. A. Longstaff and E. S. Schwartz developed a practical Monte Carlo method to state some partial convergence results for pricing American options. They applied least-squares regressions in which the explanatory variables are certain polynomial functions and estimated the continuation values of a number of derivatives. They used only in-the-money path in the regressions to increase efficiency. Clément, Lamberton, and Protter (2002) analyzed it in more detail. They proved that the estimated conditional expectation approaches (with probability one), the true conditional expectation as the number of basis functions goes to infinity. They also determined convergence rates and proved that the normalized estimation error is asymptotically Gaussian.

1.2 The Feynman-Kac Theorem

This theorem by R. Feynman and M. Kac connects the solutions of a specific class of partial differential equations to an expectation which establishes the mathematical link between the PDE formulation of the diffusion problems we encounter in finance and Monte Carlo simulations.

Given the set of stochastic processes

$$dX_i = b_i dt + \sum_{j=1}^{n} a_{ij} dW_j \quad \text{for i = 1...n ,}$$

with formal solution

$$X_i(T) = X_i(t) + \int_t^T b_i dt + \int_t^T \sum_{j=1}^{n} a_{ij} dW_j \quad ,$$

any function V (t, X) with boundary conditions

$$\text{V (T, X) = f(X)}$$

that satisfies the partial differential equation

$$\frac{\partial V}{\partial t} + g + \frac{1}{2} \sum_{i,j=1}^{n} c_{ij} \frac{\partial^2 V}{\partial X_i \partial X_j} + \sum_{i=1}^{n} b_i \frac{\partial V}{\partial X_I} = kV \quad \text{with} \quad c_{ij} := \sum_{k=1}^{n} a_{ik} a_{jk}$$

can be represented as the expectation

$$\text{V (t, X)} = E\left[f(X_T) e^{-\int_t^T kdu} + \int_t^T g e^{-\int_t^s kdu} ds \right].$$

Hereby, all of the coefficients a_{ij}, b_i, k, and g can be functions both of time t and the state vector X (t).

We have reviewed the methodological development of the Monte Carlo approach. Then, the rest of this paper is organized as follows. A brief introduction to the salient features of American options is given, and various approaches to pricing American options using simulation are briefly described in the next section. Each of the approaches using Monte Carlo methods is described in more detail in Section 3. Brief numerical comparisons with Monte Carlo methods on one or several assets are given in Section 4. Summary and conclusions based on the experiments are then presented on briefly in Section 5.

2. American options

In this section we describe some of the basic features of American options. These include the Black-Scholes PDE and the risk-neutral valuation formula for option price.

2.1 American options

For American options, these are typically more common than Europeans. An American option is like a European option except that the holder may exercise at any time between the start date and the expiry date. An American call or put option is a contract such that the owner may (without obligation) buy or sell some prescribed asset (called the underlying) S from the writer at any time (expiry date) T between the start date and a prescribed expiry date in the future, not just at expiry, and at a prescribed price (exercise or strike price) K. The exercise time τ can be represented as a stopping time; so that American options are an example of optimal stopping time problems.

Without prescribed exercise time makes it much harder to evaluate these options. The holder of an American option is thus faced with the dilemma of deciding when, if at all, to exercise. If, at time t, the option is out-of-the-money then it is clearly best not to exercise. However, if the option is in-the-money it may be beneficial to wait until a later time where the payoff might be even bigger.

The chief components of options are the striking price and the expiration date. The dynamics of the underlying asset are generally described by a stochastic differential equation, usually containing diffusion processes and jump processes. Due to the complexity of these dynamics, the valuation and optimal exercise of American options remains one of the most challenging problems in derivatives finance, particularly when more than one factor affects the value of the option. One of reason is finite difference and binomial techniques become impractical in situations where there are multiple factors, and because at any exercise time, the holder of an American option optimally compares the payoff from immediate exercise with the expected payoff from continuation, and then exercises if the immediate payoff is higher. Thus the optimal exercise strategy is fundamentally determined by the conditional expectation of the payoff from continuing to keep the option alive.

2.2 Risk-neutral valuation

Consider an equity price process *S(t)* that follows an geometric Brownian motion process according to the following stochastic differential equation

$$dS = \mu S dt + \sigma S dX \tag{2.1}$$

in which μ is the measure of the average rate of growth of the asset price, also known as the drift and σ is volatility, which measures the standard deviation of the returns (both assumed to be constant) and $X= X(t)$ is standard Brownian motion.

The option payout function is $u(S, t)$. A path dependent option is one for which $u(S, t)$ depends on the entire path $\{S(t^{'}) : 0 < t^{'} < t\}$; whereas a simple (non-path dependent) option has $u(S, t) = u(S(t), t)$. For a simple European option the payout may only be collected at the final time so that it is $f(T) = u(S(T), T)$. For a simple American option, exercise may be at any time before T so that the payout is $f(\tau) = u(S(\tau), \tau)$ in which τ is an optimally chosen stopping time.

In some classic papers, Black and Scholes (1973) and Merton (1973) described two methods for valuation of derivative securities. The first is the Black-Scholes PDE which is mentioned in the next section. The second method here, which is applicable to path dependent options and other derivatives for which the PDE is either unavailable or intractable, is the risk-neutral valuation formula

$$F(S,t) = \max_{\tau} E^{'}[e^{-r(\tau-t)}u(S(\tau),\tau) \mid S(t) = S] \tag{2.2}$$

in which $E^{'}$ is the risk-neutral expectation, for which the growth rate μ in (2.1) is replaced by r. The maximum is taken over all stopping times τ with $t < \tau < T$. This is the formula to which Monte Carlo quadrature can be applied.

2.3 Black–Scholes PDE

Suppose that we have an option whose value $V(S, t)$ depends only on S and t. It is not necessary at this stage to specify whether V is a call or a put; indeed, V can be the value of a whole portfolio of different options although for simplicity we also can think of a simple call or put. Using Itô's lemma, equation (2.1),

$$df = \sigma S\frac{\partial f}{\partial S}dx + (\mu S\frac{\partial f}{\partial S} + \frac{1}{2}\sigma^2 S^2\frac{\partial^2 f}{\partial S^2} + \frac{\partial f}{\partial t})dt \tag{2.3}$$

we can write

$$dV = \sigma S\frac{\partial V}{\partial S}dX + (\mu S\frac{\partial V}{\partial S} + \frac{1}{2}\sigma^2 S^2\frac{\partial^2 V}{\partial S^2} + \frac{\partial V}{\partial t})dt \tag{2.4}$$

This gives the random walk followed by V. Note that we require V to have at least one t derivative and two S derivatives.

Now construct a portfolio consisting of one option and a number $-\Delta$ of the underlying asset. This number is as yet unspecified. The value of this portfolio is

$$\prod = V - \Delta S \,. \tag{2.5}$$

The jump in the value of this portfolio in one time-step is

$$d\prod = dV - \Delta dS \,.$$

Here Δ is held fixed during the time-step; if it were not then $d\prod$ would contain terms in $d\Delta$. Putting (2.1), (2.4) and (2.5) together, we find that $\prod$ follows the random walk

$$d\prod = \sigma S(\frac{\partial V}{\partial S} - \Delta)dX + (\mu S\frac{\partial V}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + \frac{\partial V}{\partial t} - \mu\Delta S)dt \tag{2.6}$$

We can eliminate the random component in this random walk by choosing the delta, given by

$$\Delta = \frac{\partial V}{\partial S} \tag{2.7}$$

is the rate of change of the value of our option or portfolio of options with respect to *S*. And we note that Δ is the value of $\partial V / \partial S$ at the start of the time-step *dt*.
This result in a portfolio whose increment is wholly deterministic:

$$d\prod = (\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2})dt \tag{2.8}$$

We now appeal to the concepts of arbitrage and supply and demand, with the assumption of no transaction costs. The return on an amount $\prod$ invested in riskless assets would see a growth of $r\prod dt$ in a time *dt*. If the right-hand side of (2.8) were greater than this amount, an arbitrager could make a guaranteed riskless profit by borrowing an amount $\prod$ to invest in the portfolio. The return for this risk-free strategy would be greater than the cost of borrowing. Conversely, if the right-hand side of (2.8) were less than $r\prod dt$ then the arbitrager would short the portfolio and invest $\prod$ in the bank. Either way the arbitrager would make a riskless, no cost, or instantaneous profit. The existence of such arbitragers with the ability to trade at low cost ensures that the return on the portfolio and on the riskless account are more or less equal. Thus, we have

$$r\prod dt = (\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2})dt \qquad (2.9)$$

Substituting (2.3) and (2.7) into (2.9) and dividing throughout by *dt* we arrive at

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 s^2 \frac{\partial^2 V}{\partial s^2} + rS\frac{\partial V}{\partial S} - rV = 0 \qquad (2.10)$$

This is the famous *Black-Scholes* partial differential equation (PDE). It is a relationship between *V*, *S*, *t* and certain partial derivatives of *V*. And we should note that the Black-Scholes equation (2.10) does not contain the drift parameter μ. In other words, the value of an option is independent of how rapidly or slowly an asset grows. The only parameter from the stochastic differential equation (2.1) for the asset price that affects the option price is the volatility, σ. A consequence of this is that two people may differ in their estimates for μ yet still agree on the value of an option. Another point is worth rising immediately is that we have not yet specified what type of option is being valued. The PDE must be satisfied for any option on *S* whose value can be expressed as some smooth function *V(S, t)*.

2.4 Black-Scholes Formula for European Options

For the moment we restrict our attention to a European call, with value now denoted by *C*(*S*, *t*), with exercise price *E* and expiry date *T*. The final condition, to be applied at *t* = T, the value of a call is known with certainty to be the payoff:

$$C(S, T) = \max(S - E, 0). \qquad (2.11)$$

This is the final condition for our partial differential equation.

Our “spatial” or asset-price boundary conditions are applied at zero asset price, *S* = 0, and as $S \to \infty$. We can see from (2.1) that if S is ever zero then d*S* is also zero and therefore *S* can never change. This is the only deterministic case of the stochastic differential equation (2.1). If *S* = 0 at expiry, the payoff is zero. Thus the call option is worthless on *S* = 0 even if there is a long time to expiry. Hence on *S* = 0 we have

$$\mathrm{C}(0, t) = 0. \qquad (2.12)$$

As the asset price increases without bound it becomes ever more likely that the option will be exercised and the magnitude of the exercise price becomes less and less important. Thus as $S \to \infty$ the value of the option becomes that of the asset and we write

$$C(S, t) \sim S \quad \text{as} \quad S \to \infty. \qquad (2.13)$$

For a European call option, without the possibility of early exercise, (2.10)—(2.13) can be solved exactly to give the Black-Scholes value of a call option.

For a put option, with value $P(S, t)$, the final condition is the payoff

$$P(S, T) = \max (E - S, 0). \qquad (2.14)$$

We have already mentioned that if S is ever zero then it must remain zero. In this case the final payoff for a put is known with certainty to be E. To determine $P(0, t)$ we simply have to calculate the present value of an amount E received at time T. Assuming that interest rates are constant we find the boundary condition at $S = 0$ to be

$$P(0,t) = Ee^{-r(T-t)}. \qquad (2.15)$$

More generally, for a time-dependent interest rate we have

$$P(0,t) = Ee^{-\int_t^T r(r)dr}.$$

As $S \to \infty$ the option is unlikely to be exercised and so

$$P(S,t) \to 0 \text{ as } S \to \infty. \qquad (2.16)$$

Here we quote the exact solution of the European call option problem (2.10)—(2.13) when the interest rate r and volatility σ are constant, a explicit unique solution for the European call is

$$C(S,t) = SN(d_1) - Ee^{-r(T-t)}N(d_2) \qquad (2.17)$$

where N(.) is the N(0,1) distribution function, given by

$$N(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{1}{2}y^2} dy.$$

Here $d_1 = \dfrac{\log(S/E) + (r + \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}}$ and $d_2 = \dfrac{\log(S/E) + (r - \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}}$

We may also write

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

The equation (2.17) displays the *Black–Scholes formula* for the value of a European call.

For a put, i.e. (2.10), (2.14), (2.15) and (2.16), the solution is

$$P(S,t) = Ee^{-r(T-t)}N(-d_2) - SN(-d_1)\,. \tag{2.18}$$

The equation (2.18) displays the *Black–Scholes formula* for the value of a European put.

3. Monte Carlo methods

The theoretical understanding of Monte Carlo methods draws on various branches of mathematics. In this version, the increase in the complexity of derivative securities in recent years has led to a need to evaluate high-dimensional integrals. Monte Carlo becomes increasingly attractive compared to other methods of numerical integration as the dimension of the problem increases. A great number of Monte Carlo simulation models are known and used in practice.

Monte Carlo technique for valuation of derivatives securities is a method, which is based on the probability distribution of complete histories of the underlying security process. The Monte Carlo method lends itself naturally to the evaluation of security prices represented as expectations. Generically, the approach consists of the following steps:

(1) Simulate sample paths of the underlying state variables (e.g., underlying asset prices and interest rates) over the relevant time horizon. Simulate these according to the risk-neutral measure.

(2) Evaluate the discounted cash flows of a security on each sample path, as determined by the structure of the security in question.

(3) Average the discounted cash flows over sample paths.

However, a difficulty occurs for Monte Carlo valuation of American options, Monte Carlo methods are required for options that depend on multiple underlying securities or that involve path dependent features. Since determination of the optimal exercise time depends on an average over future events, Monte Carlo simulation for an American option has a “Monte Carlo on Monte Carlo” feature that makes it computationally complex.

3.1 Mathematical Background

Monte Carlo (MC) simulation is an alternative to the numerical PDE method. Boyle (1977) is the first researcher to introduce Monte Carlo simulation into finance. The method itself is simple and easy to implement. Monte Carlo (MC) simulation is the primary method for pricing complex financial derivatives, such as contracts whose payoff depends on several correlated assets or on the entire sample path of an asset price. We can simulate as many sample paths as desired according to the underlying stochastic differential equation that describes the stock process. For each sample path, the option value is determined and the average from all paths is the estimated option price. The option price μ is written as an integral that represents the mathematical expectation of the discounted payoff under a so-called risk-neutral probability measure. This expectation is usually with respect to a nonuniform density over the real space, but with a change of variables, it can be rewritten as an integral over the s-dimensional unit hypercube

$[0,1)^t = \{\, u=(u_0,...,u_{t-1}) : 0 < u_j < 1 \;\text{ for all } j \}$:

$$\mu = \mu(f) = \int_0^1 ... \int_0^1 f(u_0,...,u_{t-1}) du_0 ... du_{t-1} = \int_{[0,1)^t} f(u) du = E[f(U)], \tag{3.1}$$

for some function $f : [0,1)^t \to \mathbb{R}$, where u represents a point in $[0,1)^t$, and $U \sim U[0,1)^t$ is a random point with the uniform distribution over the unit hypercube. In this paper, we assume that the integral is already written in the form (3.1) for a fixed positive integer t, and we want to estimate μ. This t represents the number of calls to the underlying random number generator used in our simulation. In situations where this number of calls is random and unbounded, t can be taken as infinite, with the usual assumption that with probability one, only a finite number of coordinates of U need to be explicitly generated.

For the European option, the MC method works well. In fact, we even have an analytical solution, e.g., using the Black-Scholes formula. More importantly, the value is determined only by the terminal stock price if one assumes a given starting point, time, constant interest rate and volatility. It is easy to see that Monte Carlo simulation must work in a forward fashion. However, for the American option, because of early exercise, in contrast to a partial differential equation, we would also need to know the option value at the intermediate times between the simulation start time and the option expiry time. In Monte-Carlo this information is harder to obtain, therefore, even though it is simple and capable of handling multi-factor problems, once we have to solve a problem backwards, Monte Carlo simulation becomes awkward to implement.

3.2 Least Square Monte Carlo method (LSM)

There is basically one way to value American-style options, instead of determining the exercise boundary before simulation, this approach focuses on the conditional expectation function; see e.g., Carriere (1996), Tsitsiklis and Roy (1999). Longstaff and Schwartz (2001) proposed the Least-Squares Monte Carlo (LSM) method, an easy way to implement this approach. Clement, Lamberton and Protter (2001) studied related convergence issues. Tian and Burrage (2002) discussed the accuracy of the LSM method. Moreno and Navas (2003) further discussed the robustness of LSM with regard to the choice of the basis functions.

Longstaff and Schwartz (2001) introduce the use of Monte Carlo simulation and least squares algorithm of Carriere to value American options since nothing more than simple least square is required. At each exercise time point, option holders compare the payoff for immediate exercise with the expected payoff for continuation. If the payoff for immediate exercise is higher, then they exercise the options. Otherwise, they will leave the options alive. The expected payoff for continuation is conditional on the information available at that time point. The key insight underlying this approach is that this conditional expectation can be estimated from the cross-sectional information in the simulation by using least squares. This makes this approach readily applicable in path-dependent and multifactor situations where traditional finite difference techniques cannot be used. To find out the conditional expectation function, we regress the realized payoffs from continuation on a set of basis functions in the underlying asset prices. The fitted values are chosen as the expected continuation values. We simply compare these continuation values with the immediate exercise values and make the optimal exercise decisions, then we obtain a complete specification of the optimal exercise strategy along each path. We recursively use this algorithm and discount the optimal payoffs to time zero. That is the option price.

The method starts with Nrandom paths (S^k_n, t_n) for $1 \le k \le N$ and $t_n = ndt$. Valuation is performed by rolling-back on these paths. Suppose that $F^k_{n+1} = F(S^k_{n+1}, t_{n+1})$ is known. For points (S^k_n, t_n) set X= S^k_n the current equity value and Y=e^{-rdt} $F(S^k_{n+1}, t_{n+1})$ the value of deferred exercise. Then perform regression of Y as a function of the polynomials X, X^2, …, X^m for some small value of m which is called basic function; i.e. approximate Y^k by a least squares fit of these polynomials in X. Hence we use this regressed value in deciding whether to exercise early.

3.3 Quasi-Monte Carlo Methods (low-discrepancy)

Instead of generating sample paths randomly, it is possible to systematically select points in a probability spaces so as to optimally "fill up" the space. The selection of points is a low-discrepancy sequence. Taking averages of derivative payoffs at points in a low-discrepancy sequence is often more efficient than taking averages of payoffs at random points.

The general problem for which quasi-Monte Carlo methods have been proposed as an alternative to the Monte Carlo method is multidimensional numerical integration. Hence for the remainder of this section, we assume the problem under consideration is to evaluate

$$\mu = \int_{[0,1)^t} f(u)du,$$

where f is a square-integral function. To approximate μ, both MC and QMC proceed by choosing a point set P_n= { u_0, ⋯, u_{n-1} } ∈ [0, 1) ^{t}, and then the average value of f over P_n is computed, i.e., we get the estimator

$$Q_n = \frac{1}{n}\sum_{i=0}^{n-1} f(u_i). \tag{3.2}$$

In the Monte Carlo method, the points u_0,…, u_{n-1} are independent random vectors and uniformly distributed over $[0,1)^t$ and n is the number of replications (a constant). This estimator is unbiased and has variance σ^2/n, and define $\sigma^2 = \mathrm{Var}[f(U_i)]$, where

$$\mathrm{Var}[f(U_i)] = \int_{[0,1)^t} f^2(u)du - \mu^2 .$$

If $\sigma^2 < \infty$, then Q_n obeys a central-limit theorem, and we can rely on it to compute a confidence interval on μ, whose width converges roughly as $O(\sigma n^{-1/2})$. The use of MC for pricing financial options was first proposed by Boyle as mentioned in the introduction. The techniques have evolved tremendously since then.

In practice, one uses a pseudorandom number generator to choose these points. The idea of quasi-Monte Carlo methods is to use a more regularly distributed point set, so that a better sampling of the function can be achieved. An important difference with Monte Carlo is that the set P_n is typically deterministic when a quasi-Monte Carlo method is applied. In other words, Quasi-Monte Carlo replaces the independent random points U_i in (3.2) by a set of n deterministic points, $P_n = \{u_0, \cdots, u_{n-1}\}$, which cover the unit hypercube $[0,1)^t$ more evenly (uniformly) than a typical set of random points. The point set P_n is called a design by some statisticians. Niederreiter presents these methods in detail in his book, and describes different ways of measuring the quality of the point sets P_n on which quasi-Monte Carlo methods rely. More specifically, the goal is to measure how far is the empirical distribution induced by P_n from the uniform distribution over $[0,1)^t$. Such measures can be useful for providing upper bounds on the deterministic integration error $|Q_n - \mu|$. Typically, a point set P_n is called a low-discrepancy point set if $D^*(P_n) = O(n^{-1} \log^t n)$. For a function of bounded variation in the sense of Hardy and Krause, the integration error $|Q_n - \mu|$ is in $O(n^{-1} \log^t n)$ when P_n is a low-discrepancy point set.

This type of upper bound suggests that the advantage of QMC methods over MC, which has a probabilistic error in $O(n^{-1/2})$, will eventually be lost as the dimension t increases, or more precisely, it suggests that quasi-Monte Carlo methods will require a sample size n too large, for practical purposes, to improve upon Monte Carlo when t is large.

In this context, the programming codes for QMC simulation on several assets are too much complex, then numerical results showing an improvement of QMC over MC in high dimensions and using a relatively small sample size n seem hard to explain. I will not use this method in the following numerical studies.

4. A Numerical Example and Results of LSM

In this section we report the results of some numerical studies by applying the methodologies above to price American style put options and illustrate the Monte Carlo simulation methods.

4.1 Why American put options

As usual, let *S(t)* denote the asset price at time t and let K denote the exercise price. Suppose the holder wishes to exercise the option at some time $t < T$. This is only worthwhile if $S(t) > K$, and it gives a payoff of $S(t) - K$ at time t. Instead, the holder could sell the asset short at the market price at time t and then purchase the asset at time $t = T$. With this strategy the holder has gained amount $S(t) > K$ at time t and paid out an amount less than or equal to K at time T. This is clearly better than gaining $S(t) - K$ at time t. Since it is never optimal to exercise an American call option before the expiry date, an American call option must have the same value as a European call option. Hence, it is never optimal to exercise an American call option before the expiry date, but the same is not true for put options. It is valuable to choose and analyze American put options.

4.2 An American Put on a Single Asset

We begin by pricing a standard American put option on a single asset and a share of non-dividend-paying stock whose price is governed by a geometric Brownian motion process, given by

$$S_{\mathrm{t}} = S_0 \exp(\sigma W_t + (r - \frac{\sigma^2}{2})t) \tag{4.1}$$

with r denoting as usual the riskless rate of interest, and σ denoting the constant volatility, r and σ are constants. No closed-form solution for the price is known, but various numerical methods give good approximations to the price very rapidly. Furthermore, assume the put option is exercisable 50 times per year at a strike price of K up to and including the final expiration date T of the option. As the set of basis functions, we use a constant and the first four Laguerre polynomials as given by

$$L_3(X) = \exp(-\frac{X}{2}) \frac{e^X}{3!} \frac{d^3}{dX^3}(X^3 e^{-X}), \tag{4.2}$$

assume that X is the value of the asset underlying the option and that X follows a Markov process. Thus I regress discounted realized payoffs on the above equation.

For example, the American put option is evaluated, under which no dividends are paid. The price of exercising is 100 and we have three possible dates of exercising. The compound risk-free interest in the option continuation is equal to 0.04, with $S_0,$ σ and T varying as shown in the table. The results of the simulation are presented in Table 4.1 that reports the values of the early exercise option of a simple optimization for different starting values for the stock price which is implied by LSM techniques. The value of the early exercise option is the difference between the American and European put values. The fourth column gives the Black–Scholes values for the corresponding European option. We next give the Monte Carlo values from the present method for comparison. The LSM estimates are based on 10,000 simulations using 50 exercise points per year. These paths are generated under the risk-neutral measure. The calculations were performed throughout in MATLAB.

As shown, in the sixth column, the standard error of the estimates of the price is also reported from 0.0388 to 0.1957, moreover if the volatility enlarges one time from 0.2 to 0.4, then the corresponding standard error will be at least as twice much as before, then we conclude that the smaller volatility creates much more precise simulations. We focus primarily on the early exercise value since it is the most difficult component of an American option's value to determine, and the European component of an American option's value is much easier to identify. As is seen from the following table, according to all positive early exercise values which ranges from 0.0198 to 3.2510, if we use the Monte Carlo simulation for the least squares, some values are obtained, the American put option may have a strictly higher price than the corresponding European one, and then the algorithm performs better for options that are in-the-money. This is due to the fact that the early exercise decision plays a bigger role when the derivative is in-the-money.

Table 4.1
Simulation Prices of Standard American Puts with Parameter values: K=100, r=0.04

S（0）	σ	T	Closed form European	Simulated American (MC)	Standard Error	Early exercise value
80	0.20	1	17.7845	19.9924	0.0425	2.2079
80	0.20	2	17.0495	20.3005	0.0779	3.2510
80	0.40	1	23.4092	24.3201	0.1677	0.9109
80	0.40	2	25.9366	27.6697	0.1896	1.7331
85	0.20	1	14.0575	15.4924	0.0710	1.4349
85	0.20	2	14.0446	16.3273	0.0953	2.2827
85	0.40	1	20.5499	21.2595	0.1559	0.7096
85	0.40	2	23.6037	24.9556	0.1957	1.3519
90	0.20	1	10.8414	11.6997	0.0846	0.8583
90	0.20	2	11.4484	13.0961	0.0961	1.6477
90	0.40	1	17.9818	18.4667	0.1573	0.4849
90	0.40	2	21.4763	22.6358	0.1876	1.1595
95	0.20	1	8.1618	8.8639	0.0828	0.7021
95	0.20	2	9.2425	10.3061	0.0965	1.0636
95	0.40	1	15.6903	15.9783	0.1529	0.2880
95	0.40	2	19.5394	20.4489	0.1896	0.9095
100	0.20	1	6.0040	6.4022	0.0735	0.3982
100	0.20	2	7.3963	8.1683	0.0908	0.7720
100	0.40	1	13.6572	14.0117	0.1482	0.3545
100	0.40	2	17.7780	18.5118	0.1878	0.7338
105	0.20	1	4.3213	4.5979	0.0653	0.2766
105	0.20	2	5.8722	6.3964	0.0857	0.5242
105	0.40	1	11.8625	12.2267	0.1454	0.3642
105	0.40	2	16.1778	16.5401	0.1807	0.3623
110	0.20	1	3.0476	3.1438	0.0560	0.0962
110	0.20	2	4.6292	5.0001	0.0777	0.3709
110	0.40	1	10.2849	10.6562	0.1360	0.3713
110	0.40	2	14.7251	15.2884	0.1759	0.5633
115	0.20	1	2.1095	2.1293	0.0472	0.0198
115	0.20	2	3.6263	3.8081	0.0690	0.1818
115	0.40	1	8.9036	9.0256	0.1272	0.1220
115	0.40	2	13.4069	13.6494	0.1687	0.2425
120	0.20	1	1.4354	1.5014	0.0388	0.0660
120	0.20	2	2.8248	3.1769	0.0652	0.3521
120	0.40	1	7.6979	7.7794	0.1202	0.0815
120	0.40	2	12.2111	12.6356	0.1678	0.4245

4.3 American Puts on Multiple Underlying Assets

An example of multiple stochastic factors is the case in which there is not only one but several underlying assets. Options on multiple underlying assets typically have payoffs which are made a function of the maximum of the asset prices, the minimum of these, or the average of the *n* underlying assets. This study takes *n* log-Brownian assets, given by

$$S_i(t) = S_i(0)\exp(\sigma_i W_i(t) + (r - \frac{\sigma_i^2}{2})t), \qquad i=1,\dots,n \tag{4.2}$$

where $S_i(t)$ is a standard Brownian motion, in the following, to be precise, we will consider pricing American put options on several assets with payoffs given by:

$$V(S^1,...,S^L) = \max(0, K - \max(S^1,...,S^L)), \tag{4.3}$$

where S^l , $l = 1, \dots, L$ are the prices of the underlying assets, *L* being the dimension, and *K* is the strike price as before. I consider the pricing of an American put option on the maximum of two stocks using essentially the same simulation procedure used in Section 4.2, then this option yields the payoff $V(X_t;\ t) = \max(K\text{-}\max(S_t^1,\ S_t^2))$. This type of derivative has become popular as a test of Monte Carlo methods in higher dimensions, since accurate prices can be obtained in two dimensions, and the early exercise decision is non-trivial. Theoretically, the decision of whether to exercise at date *t* or not depends on the values of the two stock prices at that date.

In this numerical example, Table 4.2 reports a range of numerical values for different parameter choices. Throughout, I use K=100, r=0.04，T=1 and take the same volatilities of both assets to be 0.2 or 0.4, with 10,000 paths and one basis function (4.2) consisting of a constant.

As shown, when we change a higher initial value of one asset, then $X=\max(S_t^1,\ S_t^2)$ will increase, moreover $(K\text{-}X)^+$ is decreasing in X, hence simulated American option prices decrease, such as one asset with the stable initial value 80, and change the initial value of the other asset as 80,90,100,110 and 120, we let volatility is 0.2, and then prices of American options by Monte Carlo are respectively 18.6287, 10.7353, 5.4678, 2.6207 and 1.2119. The standard errors of the estimations in the fifth column are reported from 0.0104 to 0.1032, and the standard errors of the max-put American option with two independent stocks at the same price by the Monte Carlo method are less than that of the one-dimensional counterpart with the same stock price. For example, when the initial values are both 80 of max-puts on two assets, the standard error is only 0.0191, but for the single asset with the same initial value, the standard error is 0.0425, hence Monte Carlo methods is better to accurately simulate high-dimensional assets. The results of volatility changes (from 0.2 to 0.4) are similar with the trend in Table 4.1, then the smaller volatility the more efficient estimator we get.

Table 4.2 Simulation Prices of Max-Puts on Two Assets
Parameter Values: K=100, T=1, r=0.04

S_1（0）	S_2（0）	σ	Simulated American (MC)	Standard Error
80	80	0.20	18.6287	0.0191
80	80	0.40	17.8526	0.0685
80	90	0.20	10.7353	0.0525
80	90	0.40	13.1697	0.0997
80	100	0.20	5.4678	0.0587
80	100	0.40	9.7858	0.1032
80	110	0.20	2.6207	0.0457
80	110	0.40	7.1118	0.0967
80	120	0.20	1.2119	0.0313
80	120	0.40	5.3074	0.0870
90	90	0.20	8.6643	0.0312
90	90	0.40	10.2973	0.0919
90	100	0.20	4.1568	0.0465
90	100	0.40	7.7107	0.0938
90	110	0.20	2.0259	0.0389
90	110	0.40	6.0064	0.0864
90	120	0.20	0.9486	0.0270
90	120	0.40	4.3126	0.0774
100	100	0.20	2.4700	0.0387
100	100	0.40	6.1517	0.0863
100	110	0.20	1.2618	0.0301
100	110	0.40	4.5964	0.0776
100	120	0.20	0.6310	0.0215
100	120	0.40	3.5081	0.0699
110	110	0.20	0.6946	0.0216
110	110	0.40	3.5568	0.0689
110	120	0.20	0.3269	0.0152
110	120	0.40	2.8608	0.0633
120	120	0.20	0.1678	0.0104
120	120	0.40	2.0375	0.0523

4.4 Two Important Articles on Higher-dimensional problems

A number of numerical and implementation issues associated with the LSM algorithm. Two of excellent paper will be introduced individually below.

In the first article, Broadie and Glasserman (1997) apply a stochastic mesh approach to place bounds on the value an American call option on the maximum of five assets, where each asset has a return volatility of 20% and each return is independent of the others. The option has a three-year life and is exercisable three times per year. The assets each pay a 10% proportional dividend and the riskless rate is assumed to be 5%. The strike price of the option is 100, and the initial values of all assets are assumed to be the same and equal to either 90, 100, or 110. Using their algorithm, they are able to estimate a 90% confidence band for the value of the option. Broadie and Glasserman report that computing these bounds requires slightly more than 20 hours.

In the second article, Longstaff and Schwartz (2001) value this American call option using essentially the same simulation procedure used both the finite difference and LSM techniques. Specifically, they use 50,000 paths and choose 19 basis functions consisting of a constant, the first five Hermite polynomials in the maximum of the values of the five assets, the four values and squares of the values of the second through fifth highest asset prices, the product of the highest and second highest, second highest and third highest, etc., and finally, the product of all five asset values. In each of these cases, the LSM value is within the tightest bounds given by the Broadie and Glasserman algorithm. They also note that computing these values by LSM requires only one to two minutes.

4.5 Convergence Rates

The convergence properties of the LSM algorithm have been studied by Clement, Lamberton, and Protter (2002). There are two types of approximations in the algorithm.
(1) Approximate the continuation value by its projection on the linear space of a finite set of basis functions.
(2) Use Monte Carlo simulations and OLS regression to estimate the coefficients of the basis functions.

Clement, Lamberton, and Protter (2002) showed that under certain regularity conditions on the continuation value, Type II approximation error decreases to zero as the number of simulation paths goes to infinity, holding the number of basis functions fixed (fixed Type I approximation error). In addition, if the Type II approximation is exact; i.e., the coefficients of the projection can be calculated without random errors, then the Type I approximation error diminishes as polynomials of the state variables of increasing degree are added to the basis functions. As the degree of the polynomials tends to infinity, Type I error tends to zero; i.e., the algorithm converges to the true price as the number of simulation paths and basis functions tends to infinity. If the linear span of the basis functions does not include the continuation value function; i.e., the Type I approximation is not exact, then there will always be an error in the overall approximation, irrespective of the number of simulation paths.

Glasserman and Yu (2004) study the convergence rate of the algorithm when the number of basis functions and the number of paths increase simultaneously. They demonstrate that in certain cases, in order to guarantee convergence, the number of paths must grow exponentially with the number of polynomial basis functions when the underlying state variable follows Brownian motion, or faster than exponential when the underlying state variable follows geometric Brownian motion.

5. Conclusion

This article presents techniques for approximating the value of American-style options by Monte Carlo simulations. The paper provides the mathematical foundation for the use of the Least Squares Monte-Carlo (LSM) method. A detailed analysis of this method is given in Longstaff and Schwartz (2001). We illustrate this technique using a number of realistic examples, including the valuation of an American put. Moreover we analyze how the LSM method fares when the number of stochastic factors is increased. Our results suggest that the Least Squares Monte Carlo method is more suitable for problems in higher dimensions than other comparable Monte Carlo methods.

Pricing American options still remains an interesting research area, particularly when Monte Carlo techniques are used. This is due mainly to the flexibility of this method when used to solve high dimensional problems. For example, the management of a firm may have the opportunity to choose between initiating a project, expanding or contracting to one of several levels, or abandoning a project. Hence, this method could be particularly useful for the valuation of complex real options. In the future, with the ability to value American options, the applicability of simulation techniques becomes much broader and more promising, particularly in markets with multiple factors.

Acknowledgements

Firstly, I appreciate the master program of Financial Mathematics at Uppsala University that sets up a solid and professional foundation for me. Specially, I would like to thank my advisor, Professor Johan Tysk who gave specific advice for improvements in this paper. I have received much help and guidance from him. Thanks also go to Erik Ekström for his valuable course materials of Financial Mathematics. I am also grateful to Hongbin Zhang for his indispensable computer assistance and valuable suggestions. Finally, thanks to all seminar participants at Mathematics Department of Uppsala University.

References

[1] Boyle, P.: Options: a Monte Carlo approach. Journal of Financial Economics 4, 1977.

[2] Boyle, P. P., Broadie, M., Glasserman, P., 1997. Monte Carlo Methods for Security Pricing. Journal of Economic Dynamics and Control 21 (8).

[3] Broadie, M., Glasserman, P., Jain, G., 1997. Enhanced Monte Carlo estimates for American option prices. Journal of Derivatives 5 (1).

[4] Cl´ement, E., Lamberton, D., Protter, P., 2002. An Analysis of a Least Squares Regression Method for American Option Pricing. Finance and Stochastics 6 (4).

[5] Caflisch, R. 1998. Monte Carlo and Quasi-Monte Carlo Methods. Acta Numerica.

[6] Glasserman, P.: Monte Carlo Methods in Financial Engineering. Springer-Verlag, New York (2004)

[7] J¨ackel, P.: Monte Carlo Methods in Finance. John Wiley, Chichester, U.K. (2002)

[8] Janse van Resenburg, E. J., Torrie, G. M., 1993. Estimation of multidimensional integrals: is Monte Carlo the best method? Journal of Physics A: Mathematics and General 26 (4).

[9] Joy, C., Boyle, P. P., Tan, K. S., Jun. 1996. Quasi-Monte Carlo methods in numerical finance. Management Science 42 (6).

[10] Longstaff, F. A., Schwartz, E. S., 2001. Valuing American Options by Simulation: a Simple Least-Squares Approach. Review of Financial Studies 14 (1).

[11] Stentoft, L., 2004a. Assessing the Least Squares Monte-Carlo Approach to American Option Valuation. Review of Derivatives Research 7 (2).

[12] Stentoft, L., 2004b. Convergence of the Least Squares Monte Carlo Approach to American Option Valuation. Management Science 50 (9).

[13] Rogers, L. C. G. 2002. Monte Carlo Valuation of American Options. *Mathematical Finance* 12.

[14] Zaremba, SK., 1968, The mathematical basis of Monte Carlo and Quasi-Monte Carlo methods. SIAM Review 10.

Appendix: Matlab codes

Section 4.2 one-dimensional option

```
close all
clear all


% =======================
% Parameters of American Options
% =======================

% Stock
sigma=.2;            % Volatility
S0=80;               % Initial price

% Interest rate and Dividend Yield
r=.04;
D=0;

% Options
T=1;           % Time to Maturity
KP=100;        % Strike Price


% ==================
% Monte Carlo Simulations
% ==================

dt =1/50;          %Lenght of the interval of time
N=T/dt;            %Number of periods to simulate the price of the stock
NSim=10000;   %Munber of simulations

dBt=sqrt(dt)*randn(NSim,N);       % Brownian motion
St=zeros(NSim,N);                    % Initialize matrix
St(:,1)=S0*ones(NSim,1);            % vector of   initial stock price per simulation

for t=2:N;
        St(:,t)=St(:,t-1).*exp((r-D-.5*sigma^2)*dt+sigma*dBt(:,t)); %simulation of
prices
end
```

```
SSit=St;                        % just change the name
NSim=size(SSit,1);          % Number of simulations


%=================================
% Computing the value of American Options
%=================================

% Work Backwards

% Initialize CashFlow Matrix
MM=NaN*ones(NSim,N);
MM(:,N)=max(KP-SSit(:,N),0);

for tt=N:-1:3;
%       disp('Time to Maturity')
%       disp(1-tt/N)

    % Step 1: Select the path in the money at time tt-1
    I=find(KP-SSit(:,tt-1)>0);
    ISize=length(I);

    % Step 2: Project CashFlow at time tt onto basis function at time tt-1
    if tt==N
    YY=(ones(ISize,1)*exp(-r*[1:N-tt+1]*dt)).*MM(I,tt:N);
          else
        YY=sum(((ones(ISize,1)*exp(-r*[1:N-tt+1]*dt)).*MM(I,tt:N))')';
    end


    SSb=SSit(I,tt-1);
    XX=[ones(ISize,1),SSb,SSb.^2,SSb.^3];
    BB=inv(XX'*XX)*XX'*YY;

    SSb2=SSit(:,tt-1);
    XX2=[ones(NSim,1),SSb2,SSb2.^2,SSb2.^3];
    % Find when the option is exercised:
    IStop=find(KP-SSit(:,tt-1)>=max(XX2*BB,0));
    % Find when the option is not exercised:
    ICon=setdiff([1:NSim],IStop);
    % Replace the payoff function with the value of the option (zeros when
    % not exercised and values when exercised):
    MM(IStop,tt-1)=KP-SSit(IStop,tt-1);
    MM(IStop,tt:N)=zeros(length(IStop),N-tt+1);
```

```
    MM(ICon,tt-1)=zeros(length(ICon),1);


  end

  YY=sum(((ones(NSim,1)*exp(-r*[1:N-1]*dt)).*MM(:,2:N))')';

  Value=mean(YY)
  sterr=std(YY)/sqrt(NSim)


% =================
% B-S European Options
% =================

d1 = (log(S0/KP)+(r+sigma^2/2)*T)/(sigma*sqrt(T));
d2 = (log(S0/KP)+(r-sigma^2/2)*T)/(sigma*sqrt(T));
P_bseu= KP*exp(-r*T)*normcdf(-d2)-S0*normcdf(-d1)
```

Section 4.3 high-dimensional option

```
close all
clear all


% ========================
% Parameters of American Options
% ========================

% Stock
sigma= .2;            % Volatility
S01=80;              % Initial price
S02=80;

% Interest rate and Dividend Yield
r=.04;
D=0;

% Options
T=1;       % Time to Maturity
KP=100; % Strike Price
```

```
% ==================
% Monte Carlo Simulations
% ==================

dt =1/50;      %Lenght of the interval of time
N=T/dt;          %Number of periods to simulate the price of the stock
NSim=10000;    %Munber of simulations

dBt1=sqrt(dt)*randn(NSim,N);
dBt2=sqrt(dt)*randn(NSim,N);        % Brownian motion
St1=zeros(NSim,N);
St2=zeros(NSim,N);                            % Initialize matrix
St1(:,1)=S01*ones(NSim,1);              % vector of    initial stock price per simulation
St2(:,1)=S02*ones(NSim,1);

for t=2:N;
       St1(:,t)=St1(:,t-1).*exp((r-D-.5*sigma^2)*dt+sigma*dBt1(:,t));
       St2(:,t)=St2(:,t-1).*exp((r-D-.5*sigma^2)*dt+sigma*dBt2(:,t));
         %simulation of prices
end


SSit1=St1;
SSit2=St2;                         % just change the name
NSim=size(SSit1,1);         % Number of simulations
NSim=size(SSit2,1);

%===============================
% Computing the value of American Options
%===============================

% Work Backwards

% Initialize CashFlow Matrix
MM=NaN*ones(NSim,N);
MM(:,N)=max(KP-max(SSit1(:,N), SSit2(:,N)),0);

for tt=N:-1:3;

     % Step 1: Select the path in the money at time tt-1
     I=find(KP-max(SSit1(:,tt-1), SSit2(:,tt-1))>0);
     ISize=length(I);

     % Step 2: Project CashFlow at time tt onto basis function at time tt-1
```

```
    if tt==N
    YY=(ones(ISize,1)*exp(-r*[1:N-tt+1]*dt)).*MM(I,tt:N);
           else
         YY=sum(((ones(ISize,1)*exp(-r*[1:N-tt+1]*dt)).*MM(I,tt:N))')';
    end

    SSb=max(SSit1(I,tt-1), SSit2(I,tt-1));
    XX=[ones(ISize,1),SSb,SSb.^2,SSb.^3];
    BB=inv(XX'*XX)*XX'*YY;


    SSb2=max(SSit1(:,tt-1), SSit2(:,tt-1));
    XX2=[ones(NSim,1),SSb2,SSb2.^2,SSb2.^3];
    % Find when the option is exercised:
    IStop=find(KP- max(SSit1(:,tt-1), SSit2(:,tt-1))>=max(XX2*BB,0));
    % Find when the option is not exercised:
    ICon=setdiff([1:NSim],IStop);
    % Replace the payoff function with the value of the option (zeros when
    % not exercised and values when exercised):
    MM(IStop,tt-1)=KP- max(SSit1(IStop,tt-1), SSit2(IStop,tt-1));
    MM(IStop,tt:N)=zeros(length(IStop),N-tt+1);
    MM(ICon,tt-1)=zeros(length(ICon),1);


end

YY=sum(((ones(NSim,1)*exp(-r*[1:N-1]*dt)).*MM(:,2:N))')';

Value=mean(YY)
sterr=std(YY)/sqrt(NSim)
```