

A FAST MONTE CARLO ALGORITHM FOR PRICING AMERICAN OPTIONS

Andrés D. Fundia*

Departamento de Finanzas, Tecnológico de Monterrey, Campus Ciudad de México

(Received 12 July 2001, accepted 14 January 2002)

Abstract

This paper develops a Monte Carlo algorithm for pricing American Options, that has a quadratic running time on N , the number of times at which the option can be exercised. The computational time is $O(A(B/\epsilon)^{d-1}\epsilon^{-2}N^2\ln(1/\delta)\ln(B/\epsilon))$, where β is the maximum price used, ϵ is the precision required, δ is the probability that the algorithm fails, and A and d depend on the dimension of the probabilistic model used. If \hat{f} is the option value obtained, $[\hat{f} - \epsilon, \hat{f} + \epsilon]$ constitutes a $100(1 - \delta)\%$ confidence interval. An explicit relationship between these parameters and the running time of the algorithm is also provided. The algorithm belongs, then, to the class of *Fully Polynomial Randomized Approximation Scheme for American Options* which is defined here and proposed as a criteria for establishing the efficiency of such an algorithm. This is the first algorithm for pricing several American options that provides an explicit relationship between the precision desired and the running time, and that is not exponential in the number of stopping times. It can be applied to American puts, American calls and Puts that pay dividends, and to several barrier and lookback options.

Resumen

En este trabajo se desarrolla un algoritmo del tipo Monte Carlo, para valorar opciones americanas que presenta un tiempo de ejecución cuadrático en N , el número de veces en la cual la opción puede ser ejercida. El tiempo calculado es $O(A(B/\epsilon)^{d-1}\epsilon^{-2}N^2\ln(1/\delta)\ln(B/\epsilon))$, donde β es el máximo precio utilizado, ϵ es la precisión requerida, δ es la probabilidad de que el algoritmo no funcione, y A y d dependen de la dimensión del modelo probabilístico utilizado. Si \hat{f} es el valor de la opción obtenido, $[\hat{f} - \epsilon, \hat{f} + \epsilon]$ representa un intervalo de confianza de $100(1 - \delta)\%$. Una relación explícita entre estos parámetros y el algoritmo a través del tiempo también se obtienen. El algoritmo pertenece, entonces a la clase de *Esquemas de Aproximaciones Aleatorias Completamente Polinomiales para Opciones Americanas*, la cual es definida aquí y es propuesta como un criterio para establecer la eficiencia de dicho algoritmo. Este es el primer algoritmo para valorar diversas opciones americanas que proporciona una relación explícita entre la precisión deseada y el tiempo de corrida, y que no es exponencial en el número de tiempos de paro. Este algoritmo puede ser aplicado a opciones americanas de compra y venta, opciones de venta que pagan dividendos y a opciones con barreras y del tipo *lookback*.

JEL classification: C15, G13

Keywords: Monte Carlo, Simulation, American option, Contingent claim prices

* Departamento de Finanzas, Tecnológico de Monterrey, Campus Ciudad de México. Calle del Puente 222, Col. Ejidos de Huipulco, México, D.F. Telephone 52(55)54832020, E-mail: Andres.Fundia@itesm.mx

The author is very grateful to the anonymous referees for their comments.

1. Introduction

The valuation of European options has followed a fast pace of development. When closed form solutions could not be found, analytic approximations and numerical methods were developed. For derivatives with complex path dependent payoff functions or with several random sources, lattice methods do not have the recombining property (see Jarrow and Turnbull, 1916), yielding algorithms that are often too slow. In such cases Monte Carlo methods became a valid alternative (see Boyle *et al.*, 1997), for a lucid survey). Slower than lattice methods in the simpler cases, their rate of approximation is not affected by the increased complexity of the process.

For the valuation of American options, the development of Monte Carlo methods has followed a slower pace. Some experimental successes were reported as well. For instance, Boyle *et al.*, (1997) offer a comprehensive account of them. An important breakthrough was achieved by Broadie and Glasserman (1997a), where an algorithm is proposed, and it is proved to converge to the correct solution. The algorithm is only efficient when the number of times at which the option can be exercised is relatively small, since its running time depends exponentially on this number.

In this paper, I propose a theoretical framework to compare various option pricing algorithms. A general Monte Carlo algorithm is also presented, and it is shown to converge fast, even if the number of times at which the option can be exercised is large. The algorithm is exponential, though, on the dimension of the stochastic model.

The algorithm is only quadratic on the number of stopping times (the number of paths simulated is linear on the number of stopping times), allowing for arbitrarily good discrete approximations. I also provide an explicit relationship between the precision desired B/ε , the error probability δ , the number of times when exercise is possible N , and the running time of the algorithm. This allows to construct confidence intervals of arbitrarily small width and probability of error.

An algorithm that produces confidence intervals for high dimensional options is shown in Broadie and Glasserman (1997), but the precision still remains uncontrolled. Only after running the algorithm and looking at the interval produced can be decided if the results are sharp. The algorithm depends on recursively computing the *exercise boundary*, *i.e.* a set that splits the space of states, between those for which the optimal decision is to *exercise* and those for which the optimal decision is to *continue*.

Valuing options by estimating the exercise boundary is a very appealing idea, it has been used several times and with different methodologies. Successful implementations with Monte Carlo methods can be found in Barraquand and Martineau (1995); Grant Vora and Weeks (1995); Grant, Vora and Weeks (1997); Longstaff and Schwartz (2001); Raymar and Zwecher (1997). In the following paragraph I offer the first theoretical justification of this methodology.

When a discrete model is used to estimate the value of a continuous model, two sources of error arise: the *discretization error* and the *discrete model error*. The former is the difference between the continuous model value and the exact discrete model value. The latter is the difference between the exact discrete model value and the value obtained by the approximation method being used. In the case of lattice methods for complex American options, the discrete model error is usually the reason for slow convergence since, at every discrete time, the lattice branches out, requiring a computational effort which is exponential on the number of subintervals. In the method proposed here, the discrete model error is handled with a realistic computational effort since the running time of the algorithm is only quadratic in the number of subintervals. The discretization error can then be reduced just by using a enough large number of subintervals. I will only analyze the discrete model error. Since the discretization error is independent of the discrete approximation method used, analysis done for lattice algorithms can also be applied here. This has been studied for some American options in Amin and Khana (1994); see Lamberton (1993); see Lamberton (1999) and see Leisen (1998).

Moreover, since I can manage quite small subintervals, the discrete model is closer to reality than a continuous model. The approximation variables can be considered as inputs of the problem, and an optimal solution is found under such inputs. The length of the time subintervals, dt , represents the length of the periods at which the holder of the option checks his position to decide if he wants to exercise or not. The precision parameter ε represents the accuracy required by the pricer of the option.

2. The theoretical framework

The design and analysis of numerical algorithms has a well established theory (see Aho, 1974). The relationship between the real performance of algorithms and the understanding of the complexity of the problems implemented has prove to be very rich in many fields. In pricing European options it is usually possible to find a closed form solution or a fast algorithm. For American options, the situation is quite different, the possibility of early exercise combined with barrier conditions or multidimensional problems create very challenging problems.

The most important notion in Theoretical Computer Science is that of a *polynomial time algorithm*: an algorithm that has a running time bounded by a polynomial applied to the size of the input. For option pricing, the notion of input size¹ is captured by the parameters:

- i) B , an upper bound of all input and computed prices

¹ Since numbers are usually coded in computers as binary, the size of the input is the logarithm of the numbers used. Number used in financial applications are not large, thus I deal with them as if they were coded as unary numbers; in such a case the input size equals the value of the number. Accordingly, the absolute error used in Aho (1974) becomes the natural measure of accuracy, instead of the weaker relative error often used.

- ii) ε , the precision
- iii) R , the maximum range

Usually, $R = B/\varepsilon$, require more precision than the prices. For clarity, we use B/ε as the range parameter. For example, if all the prices involved are bounded by 100,000, it is required to have two correct decimals on prices, volatilities are kept with four decimals, and no other numbers are used, we have that $R = \max\{B/\varepsilon, 10^4\} = B/\varepsilon = 10^7$. Or, if the algorithm is implemented with, say, 32 bit numbers, set $R = 2^{32}$.

Assume that we have a particular kind of option and an algorithm whose inputs are the needed parameters. Let us call f the free arbitrage price of this option, and \hat{f} the price obtained by the algorithm. An *exact algorithm* is one that satisfies $\hat{f} = f$, a notion that makes little sense if f is a real number. The following definition is more meaningful in the current context.

We will say that an algorithm is a *fully polynomial approximation scheme* if it also takes as inputs the precision parameters B , ε , runs in polynomial time in B/ε , and satisfies that

$$|f - \hat{f}| < \varepsilon. \quad (1)$$

To analyze the complexity of Monte Carlo methods for European options pricing I adapt, from Karp (1987), the following concept. The algorithm takes as input the necessary parameters of the option: the precision parameters B , ε , and the error probability $\delta > 0$. It constitutes a *fully polynomial randomized approximation scheme* if it runs in polynomial time B/ε , $\ln(1/\delta)$, and satisfies that

$$\Pr\left(|f - \hat{f}| > \varepsilon\right) < \delta, \quad (2)$$

where the probability is taken over the random steps taken by the algorithm. Here $\delta = \delta(\varepsilon)$. The absolute error used in the previous definitions allows us to say that, in the language of statistics, $[\hat{f} - \varepsilon, \hat{f} + \varepsilon]$ constitutes a $100(1 - \delta)\%$ confidence interval.

For American options, another parameter becomes important: the number of times at which the option can be exercised, let us call it N . A *fully polynomial randomized approximation scheme for American options* is defined, then, as an algorithm that also takes as input the times at which exercising is allowed, and runs in polynomial time B/ε , $\ln(1/\delta)$, N , and satisfies (2). The notion of a *closed form formula* (like "Black and Scholes") can be included in this context as an algorithm that obtains solutions satisfying (1) in linear time.

3. The model

The option can be exercised at N different times, $0 \leq t_1 < \dots < t_{N-1} < t_N = T$, in $[0, T]$. Let $dt = \text{Max}\{t_N - t_{N-1}, \dots, t_2 - t_1, t_1 - t_0\}$. For simplicity, I will say time k referring to time t_k .

I use the following stochastic model. Let $\{S_k, k = 0, 1, 2, \dots, N\}$ be a vector valued Markov chain having as state space the vector space S .

On the other hand, let r_k be the continuous rates of interest occurring in the interval $[t_k, t_{k+1}]$, and let

$$D_{k,l} = e^{-[r_k(t_{k+1}-t_k)+r_{k+1}(t_{k+2}-t_{k+1})+\dots+r_{l-1}(t_l-t_{l-1})]}$$

be the discount factor between times k and l , $k < l$; the rates can be state dependent, which in turns become part of the state variables.

Let $h_k(s)$ be the payoff function if the option is exercised at time k , in state s . Let $G_k(s)$ be the random variable that gives the time k value of the option under a random path that starts at time k in state s , if provided the option is not exercised at time k and the optimal policy is used afterwards. Let $g_k(s)$ be the value of the option in this case, *i.e.* $g_k(s) = E[G_k(s)]$, where the expectation is taken under the risk free probability measure (see Lamberton and Lapeyere, 1997). Let V and M be two numbers such that, $\forall k, \forall s$,

$$\text{Variance}(G_k(s)) \leq V. \tag{3}$$

$$E \left[e^{G_k(s)} - 1 - G_k(s) - \frac{1}{2}(G_k(s))^2 \right] \leq M. \tag{4}$$

These two variables need to be defined in order to control the growth of the continuation variable. Existence of the variable V means that the process has second moments and existence of M means that the process does not have an explosive behavior.² Let $f_k(s)$ be the option value at time t_k , and state s . From the theory of optimal stopping time (see Lamberton and Lapeyere, 1997) we know that

$$f_k(s) = \max\{h_k(s), g_k(s)\}.$$

In the vector state space \mathcal{F} , it is convenient to distinguish between the indicator variables (for example, a variable that keeps track of whether a barrier was hit or not) and quantitative variables, (like current, average or maximum prices of assets). Let us say that $\mathcal{F} = \mathcal{F}_1 \oplus \mathcal{F}_2$, with \mathcal{F}_1 the subspace of indicator variables, and \mathcal{F}_2 the subspace of quantitative variables. To find the exercise boundary, every vector of \mathcal{F}_1 has to be combined with the vectors of \mathcal{F}_2 . Let us call A , the number of indicator states, *i.e.* $A = |\mathcal{F}_1|$. Let d be the dimension

² Since we are assuming that all prices are bounded by B , it is enough to take $M = e^B - 1 - B - B^2/2$. Anyway, M is only used in the proof of convergence; V , on the other hand, is one of the parameters that determines the running time of the algorithm.

of \mathcal{F}_2 , so \mathcal{F}_2 can be seen as the space $[0, B]^d$. Consider a fixed vector in \mathcal{F}_1 , we assume that the corresponding exercise boundary forms a $d - 1$ dimensional surface that splits \mathcal{F}_2 in two parts: the exercising region and the continuation region.

For $s, v \in \mathcal{F}$ we say that $s \geq v$ if every coordinate of s is greater than or equal to the corresponding coordinate of v , and we say that $s < v$ if at least one coordinate of s is strictly smaller than the corresponding coordinate of v . By changing the natural sign of some coordinates, the assumption made in the previous paragraph can be stated as saying that at every time k there is a subset of \mathcal{F} , of size $A(1/\varepsilon)^{d-1}$, called the exercise boundary, such that all of its vectors v satisfy that $\forall s \in \mathcal{F}$, if $s \geq v$ then, $f_k(s) = h_k(s)$ (i.e., it is optimal to exercise), and if $s < v$ then, $f_k(s) = g_k(s)$ (i.e., it is optimal not to exercise). This condition is met by many types of options.

4. The algorithm

The exercise boundary vectors are computed backwards. At time N , they are determined by the strike price and easily computed. To find them at time k , we need have stored all the values of future times, but no time k value is needed.

By fixing the coordinates corresponding to \mathcal{F}_1 . Then, with ε -precision numbers, missing action the number of vectors in the exercise boundary that need to be considered is $(B/\varepsilon)^{d-1}$. So, I can fix also the first $d - 1$ coordinates of \mathcal{F}_2 , there is a unique value for the last coordinate that makes the vector be on the exercise boundary. It can be found by the following binary search procedure.

Let x be the fixed part, so we know that considering (x, u) , with $u \in [0, B]$, there is a unique value u^* such that (x, u^*) is an exercise boundary vector. Start with lower bound 0 and upper bound B . Let y be the middle point between these bounds. We ask the question is $y \geq u^*$? According to the answer, y becomes the new upper or lower bound of u^* . Repeat this with the new upper and lower bounds until the difference between these two becomes no greater than ε .

To answer these questions, as defined before, let $g_k(x, y)$ be the value of the option if the holder decides not to exercise. Simulate paths $P = (S_k, S_{k+1}, \dots, S_N)$ starting at $S_k = (x, y)$. For each path, a sample value of $G_k(x, y)$ is obtained, let us call it $\widehat{G}_k(x, y)$, or just \widehat{G}_k . To define it, let us first define,

$$\tau = \text{Min}\{l \in \{k + 1, \dots, N\} \mid S_l \geq v$$

for some v that belongs to the exercise boundary or

$$\tau = N \text{ if the condition is never met.}$$

Then, define $\widehat{G}_k = D_{k, \tau} h(S_\tau)$. In fact, n of such paths are simulated obtaining n sample values of $G_k(x, y)$, say \widehat{G}_k^i with $i = 1, \dots, n$. Now, let $\widehat{g} = \widehat{g}_k(x, y) =$

$\frac{1}{n} \sum_{i=1}^n \widehat{G}_k^i$ be the estimator of the continuation value. The answer to the question is, then:

- i) yes (or $y \geq u^*$) if $h(x, y) \geq \widehat{g}$ (i.e., exercise in state (x, y) if its payoff is greater than or equal to the estimated continuation value).
- ii) no (or $y < u^*$) if $h(x, y) < \widehat{g}$ (i.e., do not exercise in state (x, y) if its payoff is smaller than the estimated continuation value).

This is repeated until the upper and lower bound coincide, i.e., they differ by less than ε . If for all the questions asked, we obtain correct answers, the values of u^* obtained are the best approximations to the exact numbers that can be obtained with the prescribed precision. In the next section, we analyze the probability of obtaining correct answers for all the questions that occur in one execution of the algorithm.

Once we have all the values of u^* , the option value at time t_k and state s , $f_k(s)$, is obtained in a similar way. With precision ε , $\widehat{g}_k(s)$ is calculated as before (just start all paths in state s), and the estimated option value is

$$\widehat{f}_k(s) = \text{Max}\{h_k(s), \widehat{g}_k(s)\}. \tag{5}$$

5. Analysis of the algorithm

Here, we state the theorem that shows that the algorithm is a *fully polynomial randomized approximation scheme*. Moreover, explicit and asymptotically sharp bounds for its running time are provided. It is not important to know the exact values of all the constants that appear in the theorem below, since their overall effect will be visible when the algorithm is executed, and they are not big. The number of paths simulated per question n , on the other hand, is decided by the programmer in advance, so sharp estimations of the required value can save computer time. The proof of the theorem (below) requires the inequalities (6) and (8) to be satisfied.

$$n \geq \frac{4V}{\varepsilon^2} [\ln(1/\delta) + \ln(N) + \ln(A) + (d-1)\ln(B/\varepsilon) + \ln[2\log_2(B/\varepsilon)]] \tag{6}$$

For practical purposes it is sufficient that the following simplified inequality is satisfied.

$$n \geq \frac{4V}{\varepsilon^2} \ln(1/\delta). \tag{7}$$

This can be used to balance the effect of the precision ε , and the confidence $1 - \delta$ against the running time of the algorithm. The effect of V responds to the variance of the stochastic process, which can not be reduced. If the running time becomes critical, obtaining sharper bounds should prove useful.

We also need technical upper bounds on the precision parameter ε , the following will be sufficient:

$$\varepsilon \leq \min \left(\frac{V^2}{4M}, V \right). \tag{8}$$

Theorem: Assume that equations (3), (4), (6), and (8) are satisfied. Then, with probability $1 - \delta$, an execution of the algorithm will find the value of the option, $f_0(S)$, with an error of at most ε , in a time no greater than:

$$\begin{aligned} & C \frac{N(N-1)}{2} A [\log_2(B/\varepsilon)] \frac{4V}{\varepsilon^2} \left(\frac{B}{\varepsilon} \right)^{d-1} \times \\ & ([\ln(1/\delta) + \ln(N) + \ln(A) + \ln(2\log_2(B/\varepsilon))]) \\ & O(A(B/\varepsilon)^{d-1} \varepsilon^{-2} N^2 \ln(1/\delta) \ln(B/\varepsilon)) \end{aligned}$$

where C is the average time per operation, which depends on the computer used, and the implementation, including the generation of random numbers.

Proof: Consider, for now, one single question, and let W be the event that the wrong answer is obtained.

$$\begin{aligned} P(W) & \leq P((\bar{g} \leq y, g > y) \cup (\bar{g} > y, g \leq y)) \\ & \leq P(|\bar{g} - g| > \varepsilon) \\ & \leq 2e^{-\frac{\varepsilon^2 n}{4V}}. \end{aligned} \tag{9}$$

The last inequality is proved in the appendix. Its leading factors can be deduced from the central limit theorem, and estimations of the tail probabilities of the normal distribution (see, for example, Feller 1973). We need a specific bound on the error to obtain an explicit formula for the running time of the algorithm, so a large deviation inequality is used instead.

Let Q be the total number of questions asked in one execution of the algorithm, and q the number of questions asked when one exercise boundary value is determined. We have that $Q = A (B/\varepsilon)^{d-1} N q$. Also, since the initial range of possible values is at most (B/ε) , and a question is asked it gets divided by 2, the number of asked questions q must satisfy, $\frac{B/\varepsilon}{2^q} \leq 1$. This can be accomplished defining $q = \lceil \log_2(B/\varepsilon) \rceil$. So, the total number of asked questions Q satisfies

$$Q = NA (B/\varepsilon)^{d-1} \lceil \log_2(B/\varepsilon) \rceil. \tag{10}$$

For $j = 1, \dots, Q$; let W_j be the event that the j^{th} question is wrong, and W the

event that at least one question is wrong. Then,

$$\begin{aligned}
 P(W) &= P\left(\bigcup_{j=1}^Q W_j\right) \\
 &\leq \sum_{j=1}^Q P(W_j) \\
 &\leq Q P(W_1) \\
 &\leq NA(B/\varepsilon)^{d-1}[\log_2(B/\varepsilon)] 2e^{-\frac{\varepsilon^2 n}{4V}} \\
 &\leq \delta,
 \end{aligned}$$

where, the next to last inequality follows from (10) and (9), and the last inequality is true iff inequality (6) is satisfied.

When, at time k , a question is asked, n paths of length at most $N - k$ have to be simulated. So the total number of states that have to be simulated is no more than $q A (B/\varepsilon)^{d-1} n (0 + 1 + 2 + \dots + N - 1) = q A (B/\varepsilon)^{d-1} n \frac{(N-1)N}{2}$. This determines the running time of the algorithm; replacing n by the right-hand side of (6), and q for the mentioned value, the stated bound is obtained.

Appendix A

Proof of Inequality (9)

For simplicity, let $X = G_k(s) - g_k(s)$ be, so that $E[X] = 0$, and let σ^2 be the variance of X . According to (3) and (4) let V and M be numbers such that $V \geq \sigma^2$ and $E[e^X - 1 - X - X^2/2] \leq M$.

Let $X_j, j = 1, \dots, n$ be n independent replications of X , and let $S = X_1 + X_2 + \dots + X_n$. So (9) is equivalent to $P(|\frac{S}{n}| > \varepsilon) \leq 2e^{-\frac{\varepsilon^2 n}{4V}}$. We have that $P(|\frac{S}{n}| > \varepsilon) = P(S > n\varepsilon) + P(S < -n\varepsilon)$. We bound the first term, the second term's bound follows from it. We have that $\forall 0 \leq t \leq 1$:

$$\begin{aligned}
P(S \geq n\varepsilon) &= P(e^{St} \geq e^{n\varepsilon t}) \leq^I E[e^{St}]e^{-n\varepsilon t} \\
&\leq \prod_{j=1}^n E[e^{X_j t}]e^{-n\varepsilon t} = (E[e^{Xt}])^n e^{-n\varepsilon t} \\
&= \left(E \left[1 + Xt + \frac{1}{2!}X^2t^2 + \frac{1}{3!}X^3t^3 + \frac{1}{4!}X^4t^4 + \dots \right] \right)^n e^{-n\varepsilon t} \\
&= \left(1 + \frac{1}{2}\sigma^2t^2 + E \left[\frac{1}{3!}X^3t^3 + \frac{1}{4!}X^4t^4 + \dots \right] \right)^n e^{-n\varepsilon t} \\
&\leq \left(1 + \frac{1}{2}Vt^2 + t^3E[e^X - 1 - X - X^2/2] \right)^n e^{-n\varepsilon t} \\
&\leq \left(1 + \frac{1}{2}Vt^2 + t^3M \right)^n e^{-n\varepsilon t} \leq^{II} e^{-n(\varepsilon t - \frac{1}{2}Vt^2 - t^3M)} \\
&= \leq^{III} e^{-n(\frac{\varepsilon^2}{V} - \frac{1}{2}V\frac{\varepsilon^2}{V^2} - \frac{\varepsilon^3}{V^3}M)} \leq e^{-n(\frac{\varepsilon^2}{2V} - \frac{\varepsilon^3}{V^3}M)} \\
&= e^{-n\frac{\varepsilon^2}{2V}(1 - \frac{2\varepsilon}{V^2}M)} \leq^{IV} e^{-\frac{\varepsilon^2n}{4V}}
\end{aligned}$$

I) follows from Markov's inequality.

II) follows from the inequality $1 + x \leq e^x, x \geq 0$.

III) follows taking $t = \frac{\varepsilon}{V}$, which is no greater than 1 from assumption (8).

IV) follows from (8).

References

- Aho, A. (1974). *The Design and Analysis of Computer Algorithms*. Addison Wesley.
- Amin, K., and A. Khana (1994). Convergence of American Option Values from Discrete Continuous-Time Financial Models, *Mathematical Finance*, 4(4), pp. 289-304.
- Barraquand, J., and D. Martineau (1995). Numerical Valuation of High Dimensional Multivariate American Securities. *Journal of Financial and Quantitative Analysis*, 30(3), pp. 383-405.
- Boyle, P., P. Broad, and P. Glasserman (1997). Monte Carlo Methods for Security Pricing. *Journal of Economic Dynamics and Control*, 21.
- Broadie, M., and P. Glasserman (1997). A Stochastic Mesh Method for Pricing High-Dimensional American Options. Unpublished Manuscript.
- Broadie, M., and P. Glasserman (1997). Pricing American-Style Securities using Simulation. *Journal of Economic Dynamics and Control*, 21, pp. 1323-1352.
- Feller, W. (1973). *Introduction to Probability Theory*. Wiley, New York.
- Grant, D., G. Vora, and D. Weeks (1995). Simulation and the Early Exercise Option Problem. *The Journal of Financial Engineering*, 4, pp. 42-55.
- Grant, D., G. Vora, and D. Weeks (1997). Path-Dependent Options: Extending the Monte Carlo Simulation Approach. *Management Science*, 43(11), pp. 1589-1602.
- Jarrow, R., and S. Turnbull (1996). *Derivative Securities*, South-Western.
- Karp, R. (1987). An Introduction to Randomized Algorithms, *Discrete Applied Mathematics*, 34, pp. 249-260.
- Lamberton, D. (1993). Convergence of the Critical Price in the Approximation of American Options. *Mathematical Finance*, 3, pp. 179-190.

- Lamberton, D. (1999). Brownian Optimal Stopping and Random Walks, Unpublished Manuscript.
- Lamberton, D., and B. Lapeyere (1997). Introduction to Stochastic Calculus Applied to Finance, Chapman & Hall.
- Leisen, D. (1998). Pricing the American Put Option: a Detailed Convergence Analysis for Binomial Models. *Journal of Economic Dynamics and Control*, 22, pp. 1419-1444.
- Longstaff, F., and E. Schwartz (2001). Valuing American Options by Simulation: a Simple Least-Squares Approach. 14, pp. 113-147.
- Raymar, S., and T. Zwecher (1997). Monte Carlo Estimation of American Call Options on the Maximum of several Stocks. *The Journal of Derivatives*, 5(1), pp. 7-23.
- Seng, T., and K. Vetzal (1995). Early Exercise Regions for Exotic Options. *The Journal of Derivatives*, 4, pp. 42-55.