

Técnicas de reducción de varianza para el método Monte Carlo aplicado a opciones financieras

Marisol Santana Anaya



Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias Físico Matemáticas

Tesis presentada al Colegio de Matemáticas para la obtención de la licenciatura en Matemáticas

Director de Tesis
Hugo Adán Cruz Suárez

Agosto 2013

Agradecimientos

“Mientras el río corra, los montes hagan sombra y en el cielo haya estrellas, debe durar la memoria del beneficio recibido en la mente del hombre agradecido” Virgilio.

Índice general

1. Procesos Estocásticos	4
1.1. Preliminares	4
1.2. Procesos Estocásticos	6
1.2.1. Cadenas de Markov	6
1.2.2. Movimiento Browniano	9
2. Programación de Algoritmos	10
2.1. Simulación	10
2.1.1. Números Aleatorios	10
2.1.2. Algoritmos	13
2.1.3. Complejidad de Algoritmos	13
2.2. Simulaciones de Procesos Estocásticos	20
2.2.1. Cadena de Markov en <i>'The R Project for Statistical Computing'</i>	20
2.2.2. Proceso Wiener en <i>'The R Project for Statistical Computing'</i>	21
3. Método Monte Carlo	22
3.1. Método Monte Carlo	22
3.2. Reducción de Varianza	29
3.2.1. Variables Aleatorias Antitéticas	29
3.2.2. Variables de Control	31
4. Finanzas	33
4.0.3. Contratos de Opción	33
4.0.4. Valuación de Opciones	35
4.0.5. Algoritmos Financieros	39
4.0.6. Simulaciones Financieras	44
4.0.7. Técnicas de Reducción de Varianza	49
4.0.8. Tablas y Gráficas	53
5. Conclusiones	59

Introducción

La creación del método Monte Carlo suele ligarse a los nombres de los matemáticos norteamericanos J. von Neumann y S. Ulam. Se considera a 1949 como el año del nacimiento del método Monte Carlo. El nombre de *Monte Carlo* se debe al de una población del principado de Mónaco, célebre por su casa de juego [8].

El método Monte Carlo es un método numérico que permite resolver problemas matemáticos mediante la simulación de variables aleatorias. Muchos problemas numéricos en ciencias, ingeniería, estadística y finanzas son resueltos hoy en día a través del método Monte Carlo, esto es, a través de experimentos aleatorios en una computadora [7],[10].

Como Monte Carlo proporciona una estimación, tiene asociado un estimador, en esta tesis se utilizará la media muestral, \bar{X} , cuyo error estándar es σ/\sqrt{N} , donde σ^2 es la varianza y N el tamaño de la muestra. Así que para reducir el error de estimación, se puede aumentar la muestra o reducir la varianza. En muchos casos, aumentar el tamaño de la muestra no es factible computacionalmente, por esta razón nos enfocamos en reducir la varianza.

En esta tesis logramos la reducción de varianza utilizando variables aleatorias antitéticas y variables aleatorias de control, que implementamos en el contexto financiero.

El método de reducción de varianza con variables aleatorias antitéticas construye un estimador de menor varianza que el del método Monte Carlo, con el par antitético (Y, Y^*) , mediante la simulación de la mitad de las variables aleatorias con distribución Y , la otra mitad con distribución Y^* y obteniendo el promedio de los resultados.

El método de reducción de varianza con variables aleatorias de control construye un estimador de menor varianza que el método Monte Carlo mediante la simulación de Y y Y^* , con Y y Y^* correlacionadas, obteniendo el promedio del total de simulaciones de Y y Y^* menos la esperanza de Y^* .

Se hicieron simulaciones en “*The R Project for Statistical Computing*” del costo de un activo financiero con riesgo y sin riesgo, el valor de una opción de tipo Americana y de tipo Europea, suponiendo que el costo de los activos siguen el *movimiento Browniano Geométrico*. Con ayuda del algoritmo que obtiene el valor de la opción de tipo Europea, se probó la efectividad de las simulaciones comparando el valor obtenido con el valor exacto de la Fórmula Black-Scholes [5] para el valor de una opción Europea. Además, en este algoritmo se implementaron las técnicas de reducción de varianza desarrolladas.

La tesis está dispuesta de la siguiente manera: En el Capítulo 1 se proporcionan los conocimientos preliminares para la lectura de esta tesis. En el Capítulo 2 se desarrolla toda la teoría necesaria para la programación de algoritmos porque el corazón de Monte

Carlo son los números aleatorios. En el Capítulo 3 se define y se ejemplifica el método Monte Carlo y se da una descripción de las técnicas de reducción de varianza aquí estudiadas. Por último, en el Capítulo 4 se definen los conceptos financieros y se exhiben las simulaciones financieras.

Capítulo 1

Procesos Estocásticos

La teoría de los procesos estocásticos se centra en el estudio y modelación de sistemas que evolucionan a lo largo del tiempo de acuerdo a leyes no determinísticas, esto es, de carácter aleatorio. Su estudio se ha extendido a diversas áreas del conocimiento porque es posible describir a la naturaleza como un proceso estocástico.

El objetivo de este capítulo es el de contextualizar al lector con los conceptos básicos necesarios para la lectura de esta tesis. En éste, se proporcionan definiciones y teoremas relacionados con procesos estocásticos, cadenas de Markov y el movimiento Browniano.

1.1. Preliminares

Definición 1.1 Un espacio de probabilidad es una terna $(\Omega, \mathcal{F}, \mathbb{P})$, donde Ω es el espacio muestral conformado por el conjunto de resultados de un experimento, \mathcal{F} una sigma álgebra con $\mathcal{F} \subseteq \text{Pot}(\Omega) := \{A \mid A \subset \Omega\}$, \mathcal{F} es llamado el espacio de eventos, $\mathbb{P}: \mathcal{F} \rightarrow [0,1]$ una medida de probabilidad, la cual cumple con:

1. Para todo $A \in \mathcal{F}$, $0 \leq \mathbb{P}(A) \leq 1$.
2. $\mathbb{P}(\Omega) = 1$.
3. Para eventos mutuamente excluyentes, $\{A_i, i \geq 1\}$, $\mathbb{P}(\bigcup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} \mathbb{P}(A_i)$.

Definición 1.2 Sean $A, B \in \mathcal{F}$, tal que $\mathbb{P}(B) > 0$, la probabilidad condicional del evento A dado B es

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}.$$

Definición 1.3 Sean $A, B \in \mathcal{F}$. A es independiente de B si $\mathbb{P}(A|B) = \mathbb{P}(A)$ y $\mathbb{P}(B|A) = \mathbb{P}(B)$.

Observación 1.1 Equivalentemente si A es independiente de B se tiene que $\mathbb{P}(A \cap B) = \mathbb{P}(A)\mathbb{P}(B)$.

Definición 1.4 Sea $X: \Omega \rightarrow \mathbb{R}$, X es una variable aleatoria si para todo $A \in \mathcal{B}(\mathbb{R})$ se tiene que $\{\omega \in \Omega \mid X(\omega) \in A\} \in \mathcal{F}$, donde $\mathcal{B}(\mathbb{R})$ denota la σ -álgebra de Borel.

Observación 1.2

1. En forma simplificada el conjunto anterior se puede representar por $\{X \in A\}$.
2. Debido a que la sigma álgebra de Borel se puede ver como la mínima sigma álgebra generada por la clase $\mathcal{C} = \{(-\infty, x] \mid x \in \mathbb{R}\}$ la Definición 1.4 puede escribirse de forma equivalente como: " X es una variable aleatoria si para todo x en \mathbb{R} , $\{X \leq x\} \in \mathcal{F}$ ".

Ejemplo 1.1 Sea X una variable aleatoria. Si $\text{Ran}X := \{y \in \mathbb{R} \mid X(\omega) = y \text{ para algún } \omega \in \Omega\}$ es finito o infinito numerable, llamamos a X una **variable aleatoria discreta**. En este caso podemos describir al rango de X como $\text{Ran}X = \{x_1, x_2, \dots\}$.

Para todo resultado posible x_i de la variable aleatoria asociamos un número p_i , donde $p_i := \mathbb{P}(X = x_i)$.

Observemos que la sucesión $\{p_i, i=1, 2, 3, \dots\}$ satisface las siguientes propiedades:

1. $p_i \geq 0$, para todo $i=1, 2, 3, \dots$.
2. $\sum_1^\infty p_i = 1$.

Al valor p_i se le denomina **función de probabilidad puntual** de la variable aleatoria X en el punto x_i .

Ejemplo 1.2 Sea X una variable aleatoria. Si el número de valores posibles de X es no numerable, llamamos a X una **variable aleatoria continua**. Para X existe una función no negativa f , definida para todo $x \in \mathbb{R}$, que cumple la propiedad de que para todo conjunto $B \in \mathcal{B}(\mathbb{R})$:

$$\mathbb{P}\{X \in B\} = \int_B f(x) dx.$$

La función f es llamada la **función de densidad de la variable aleatoria X** .

Las variables aleatorias son funciones matemáticas que permiten asignar valores numéricos a cada uno de los posibles resultados obtenidos en un experimento de naturaleza aleatoria. Las distribuciones de probabilidad son funciones numéricas asociadas a la variable aleatoria que describen la asignación de probabilidad para cada uno de los elementos del espacio muestral.

1.2. Procesos Estocásticos

Definición 1.5 Sea $(\Omega, \mathcal{F}, \mathbb{P})$ un espacio de probabilidad y S un conjunto no vacío. Un proceso estocástico es una colección de variables aleatorias, $\{X_t : \Omega \rightarrow S, t \in T\}$, donde T es un conjunto no vacío que denota el conjunto de índices.

El conjunto S es el espacio de estados. Si S es numerable se dice que el proceso tiene espacio de estados discreto mientras que si S es continuo decimos que tiene espacio de estados continuo. Existen cuatro tipos de procesos estocásticos: tiempo discreto con espacio de estados discreto, tiempo discreto con espacio de estados continuo, tiempo continuo con espacio de estados continuo y tiempo continuo con espacio de estados discreto.

Un proceso estocástico puede considerarse como una función de dos variables $X: T \times \Omega \rightarrow S$ tal que a la pareja (t, ω) se le asocia el estado $X(t, \omega)$, lo cual también puede escribirse como $X_t(\omega)$. Para cada valor de t en T , el mapeo $\omega \mapsto X_t(\omega)$ es llamada una trayectoria o realización del proceso. Es decir, a cada ω del espacio muestral le corresponde una trayectoria del proceso.

A continuación se mostrarán algunos ejemplos de procesos estocásticos importantes para el desarrollo de la tesis.

1.2.1. Cadenas de Markov

Definición 1.6 Sea $(\Omega, \mathcal{F}, \mathbb{P})$ un espacio de probabilidad y S un conjunto no vacío, finito o numerable. Un proceso estocástico o sucesión de variables aleatorias $\{X_n: \Omega \rightarrow S, n \in \mathbb{N}\}$, se llama cadena de Markov con espacio de estados S si satisface la propiedad de Markov, es decir, si para todo $n \geq 1$ y toda sucesión $x_0, x_1, \dots, x_{n-1}, x_n, x_{n+1} \in S$ se cumple:

$$\mathbb{P}(X_{n+1} = x_{n+1} | X_0 = x_0, \dots, X_{n-1} = x_{n-1}, X_n = x_n) = \mathbb{P}(X_{n+1} = x_{n+1} | X_n = x_n).$$

La distribución de X_0 se llama distribución inicial de la cadena y usualmente la denotaremos por $\pi_0(x) = \mathbb{P}(X_0 = x)$, $x \in S$.

Si el sistema pasa del estado x al estado y en un período se dice que ha ocurrido una transición de x a y . Las probabilidades de transición de la cadena las denotaremos por:

$$p_{xy} := \mathbb{P}(X_{n+1} = y | X_n = x), \text{ con } x, y \in S.$$

Las cuales satisfacen:

1. $\sum_{y \in S} p_{xy} = 1$, $x \in S$.
2. $0 \leq p_{xy} \leq 1$, $x, y \in S$.

Cuando el espacio de estados es finito, es común representar a las probabilidades de transición en forma matricial, para ello supongamos que $S = \{1, 2, \dots, s\}$, de esta manera las probabilidades de transición se disponen en la matriz P :

$$P = \begin{pmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,s} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,s} \\ \vdots & \vdots & \ddots & \vdots \\ p_{s,1} & p_{s,2} & \cdots & p_{s,s} \end{pmatrix}$$

Observación 1.3 En el desarrollo de la tesis se usarán probabilidades de transición en n -etapas, $n \geq 1$ las cuales se definen de la siguiente forma:

$$\mathbb{P}(X_n = y | X_0 = x) = P_{xy}^n, x, y \in S.$$

En el caso matricial la probabilidad de transición de n -etapas de un estado se puede obtener elevando la matriz de transición a la n .

Esto se deriva de

$$P^n(x, y) = \sum_{y_1} \cdots \sum_{y_{m-1}} P(x, y_1)P(y_1, y_2) \cdots P(y_{m-2}, y_{m-1})P(y_{m-1}, y),$$

ver([14]).

Una cadena de Markov está caracterizada por su distribución inicial y su matriz de transición. La distribución inicial es necesaria para conocer el estado en donde inicia la cadena de Markov y con la matriz de transición, por la Observación 1.3 se obtienen todas las probabilidades de transición de la cadena.

Distribución estacionaria

Definición 1.7 Sea $\{X_n, n \geq 0\}$, una cadena de Markov con espacio de estados S y una matriz de transición $P = P_{xy}$. Si $\pi(x)$ con $x \in S$ cumple:

1. $\sum_{i=1}^{\infty} \pi(x_i) = 1, x_i \in S, \pi(x_i) \geq 0,$
2. $\sum_x \pi(x)P_{xy} = \pi(y), y \in S,$

se le llama distribución estacionaria.

Definición 1.8 Sea $\{X_n\}$ una cadena de Markov, con matriz de transición P . Si definimos $Y_n = X_{-n}$, se dice que $M' = \{Y_n\}$ es también una cadena de Markov y le llamamos la cadena revertida de M .

Definición 1.9 Una cadena de Markov M es reversible si la cadena de Markov revertida coincide con M , i.e.

$$\mathbb{P}(X_{n+1} = j | X_n = i) = \mathbb{P}(X_n = j | X_{n+1} = i), \quad i, j \in S.$$

Teorema 1.1 *Supóngase que $M = \{X_n\}$ es una cadena de Markov con matriz de transición P , con distribución estacionaria única π . M es reversible si y sólo si:*

$$\pi_i P_{ij} = \pi_j P_{ji} \quad \text{con } i, j \text{ en } S.$$

Demostración:

Veamos que si M reversible entonces $\pi_i P_{ij} = \pi_j P_{ji}$ con i, j en S .

Como M es reversible, existe $P = (P_{ji})$ matriz de transición de la cadena revertida $M' = \{Y_n\}$ donde $Y_n = X_{-n}$, i.e.

$$\begin{aligned} P_{ji} &= \mathbb{P}(Y_{n+1} = i | Y_n = j), \\ &= \mathbb{P}(X_{-n-1} = i | X_{-n} = j), \\ &= \frac{\mathbb{P}(X_{-n-1} = i, X_{-n} = j)}{\mathbb{P}(X_{-n} = j)}, \\ &= \frac{\mathbb{P}(X_{-n} = j | X_{-n-1} = i) \mathbb{P}(X_{-n-1} = i)}{\mathbb{P}(X_{-n} = j)}. \end{aligned}$$

Como X_n tiene distribución π para todo n ,

$$\begin{aligned} P_{ji} &= \frac{\mathbb{P}(X_{-n} = j | X_{-n-1} = i) \pi_i}{\pi_j}, \\ &= \frac{P_{ij} \pi_i}{\pi_j}, \\ \pi_i P_{ij} &= \pi_j P_{ji}. \end{aligned}$$

Para demostrar que si se cumple que $\pi_i P_{ij} = \pi_j P_{ji}$, con i, j en S , entonces M es una cadena reversible, se realizan los pasos invertidos de la demostración anterior partiendo de $\pi_i P_{ij} = \pi_j P_{ji}$ con i, j en S hasta llegar a $P_{ji} = \mathbb{P}(X_{-n-1} = i | X_{-n} = j)$.

Por definición de P_{ji} se concluye que

$$P_{ji} = \mathbb{P}(X_{-n-1} = i | X_{-n} = j) = \mathbb{P}(X_{-n} = i | X_{-n-1} = j),$$

y por Definición 1.9, M es reversible ■.

1.2.2. Movimiento Browniano

En 1827, el médico y botánico escocés Robert Brown (1773-1858) mientras examinaba partículas de polen (de *Clarckia Pulchella*) en el microscopio, observó que cuando éstas se encontraban suspendidas en agua y en otros líquidos se movían sin cesar en forma errática [1]. A principios del siglo XX se demostró que el movimiento irregular de las partículas de polen se debía al golpeteo constante de las moléculas invisibles de agua sobre las moléculas visibles de las partículas de polen. En 1900, el matemático francés Louis Bachelier (1870-1946) en su tesis “Théorie de la spéculation” hizo la formulación matemática del Movimiento Browniano. También el físico Albert Einstein (1879-1955) hizo una formulación del mismo en su artículo de mecánica estadística [3], [4].

Definición 1.10 Sea $(\Omega, \mathcal{F}, \mathbb{P})$ un espacio de probabilidad, el movimiento Browniano o proceso de Wiener es un proceso estocástico $\{W(t) : t \geq 0\}$, que cumple las propiedades siguientes:

1. $W(0)=0$,
2. Para $0 \leq s < t \leq T$, la variable aleatoria dada por el incremento $W(t)-W(s) \sim N(0, t-s)$; i.e., $W(t)-W(s) \sim \sqrt{t-s}N(0, 1)$.
3. Para $0 \leq s < t < u < v \leq T$, los incrementos $W(t)-W(s)$ y $W(v)-W(u)$ son independientes.
4. Las trayectorias $t \rightarrow W(t)$ son continuas.

Definición 1.11 Sea $(\Omega, \mathcal{F}, \mathbb{P})$ un espacio de probabilidad, $\{W(t) : t \geq 0\}$, el movimiento Browniano Geométrico se define como,

$$X_t = \exp(\sigma W_t + \mu t), t \geq 0,$$

con $\sigma > 0$ y $\mu \in \mathbb{R}$ constantes.

Capítulo 2

Programación de Algoritmos

La simulación es una indispensable metodología para la descripción y análisis de una amplia variedad de problemas reales. Usada apropiadamente, proporciona considerables beneficios según el contexto en la que se use: ahorro de tiempo; ahorro de recursos económicos; permite analizar la ocurrencia de ciertos fenómenos a través de la reconstrucción de escenas y un minucioso análisis, que no podría llevarse a cabo en una situación real; una vez desarrollado un modelo de simulación válido, se pueden explorar nuevas políticas, procedimientos operativos, o métodos sin necesidad de afectar al sistema. Para realizar una simulación es necesario elaborar un algoritmo. Un algoritmo es un conjunto de instrucciones que especifican la secuencia de operaciones a realizar para resolver un sistema específico o clase de problema. En otras palabras, un algoritmo es una fórmula para la resolución de un problema.

En nuestro contexto, es indispensable el estudio de algoritmos para implementar el método Monte Carlo. En este capítulo se estudia la generación de números aleatorios, los cuales son necesarios para la implementación del método Monte Carlo en la computadora, propiedades importantes de los algoritmos, estimación del tiempo de ejecución y se establece un orden en los algoritmos para poder compararlos y decidir cuál es el mejor.

2.1. Simulación

Una simulación es un experimento usualmente realizado en una computadora que involucra el uso de números aleatorios [11].

En general, la simulación puede ser apropiada cuando tenemos un problema que es muy difícil de resolver analíticamente.

2.1.1. Números Aleatorios

Suelen distinguirse tres procedimientos para obtener variables aleatorias: tablas de números aleatorios, generadores de números aleatorios y el método de números pseudo-

aleatorios.

1. Tablas de números aleatorios: Tomemos 10 tarjetas iguales y escribamos los números del 0 al 9. Al seleccionar las tarjetas al azar con reemplazo estamos generando números aleatorios.

La ruleta es uno de los aparatos mecánicos más sencillos que permite obtener variables aleatorias. En el caso de usarla, el disco giratorio es parado en seco y se toma la cifra que señala la flecha.

Las tablas de números aleatorios se emplean sólo cuando los cálculos correspondientes al método de Monte Carlo se realizan a mano.

2. Generadores de números aleatorios: Suele emplearse los ruidos en las lámparas electrónicas, si durante un intervalo fijo de tiempo el nivel del ruido sobrepasa el umbral escogido un número par de veces, se inscribe el cero; si esto ocurre un número impar de veces se inscribe el uno.
3. Números pseudoaleatorios: Los números que se obtienen a partir de una fórmula y que imitan los valores de la variable aleatoria se denominan números pseudoaleatorios. Por ejemplo, $x_n \equiv (ax_{n-1} + c)$, modulo m , con $a, b \in \mathbb{Z}$ y $m \in \mathbb{N}$.

El primer algoritmo destinado a la construcción de números pseudoaleatorios fue propuesto por J. von Neumann y se conoce como el **método de centros de cuadrados**, el cual se ejemplifica a continuación.

Ejemplo 2.1 Consideremos el número $a_0 = 0.9876$ formado por cuatro cifras. Al elevarlo al cuadrado obtendremos el número $a_0^2 = 0.97535376$ de ocho cifras. Tomemos las cuatro cifras que aparecen en el centro de este número y pongamos $a_1 = 0.5353$. Elevando ahora a_1 al cuadrado ($a_1^2 = 0.28654609$), escogamos de nuevo las cuatro cifras del centro y pongamos $a_2 = 0.6546$. Procediendo de la misma forma, tendremos $a_3 = 0.8501$, etc. (Algoritmo 2.1).

Algoritmo 2.1 Simulación del Algoritmo de Neumann en 'C'.

```
#include "milib.h"
#include <stdio.h>
#include <string.h>
int main()
{
char digitos[100];char aleatorio[100];float numero1;float definitivos[100];
double n1;
int n,i,j,l;
float l1;
scanf("%f%d",&numero1,&n);
l1=numero1;
for(j=1;j<=n;j++)
{
l1=l1*l1;
sprintf(digitos,"%f",l1);
digitos[1]='0';
for(i=4;i<=7;i++)
{
aleatorio[i-4]=digitos[i];
}
l1=atoi(aleatorio);
l1=(float)l1/10000.0;
definitivos[j]=l1;
printf("%.8f\n",definitivos[j]);
}}
```

2.1.2. Algoritmos

La solución a un problema por medio de una computadora, requiere un conjunto de instrucciones precisas. Informalmente, a tal conjunto de instrucciones precisas se le llama algoritmo. Un algoritmo es un conjunto finito de instrucciones que tiene las siguientes características [15]:

1. Precisión.
2. Unicidad. Los resultados intermedios de cada paso de la ejecución se definen únicamente y sólo dependen de las entradas y los resultados de los pasos anteriores.
3. Finitud. El algoritmo se detiene después de un número finito de instrucciones que han sido ejecutadas.
4. Entradas. El algoritmo recibe entradas.
5. Salidas. El algoritmo produce salidas.
6. Generalidad. El algoritmo se aplica a un conjunto de entradas.

2.1.3. Complejidad de Algoritmos

Muchas preguntas pueden hacerse acerca de un algoritmo en particular:

1. ¿Es correcto?.
2. ¿Hace lo que tiene que hacer cuando se introducen entradas válidas?.
3. ¿Cuál es el tiempo que tarda en ejecutarse?.
4. ¿Cuál es el espacio de memoria que se necesita para ejecutar el algoritmo?.

En un programa computacional, aunque sea derivado de un algoritmo correcto, puede ser inútil para ciertos tipos de entradas ya sea porque la memoria requerida es demasiada o porque el tiempo necesario para ejecutar el programa es demasiado grande. Por ello es importante estimar el tiempo y espacio requerido para la ejecución de un algoritmo. Determinar el rendimiento de los parámetros de un programa es una tarea difícil y depende de un número de factores tales como el equipo utilizado, la forma de representar los datos y el compilador usado. Aunque las estimaciones precisas del tiempo de ejecución de un programa deben tener en cuenta esos factores, es posible obtener información útil si se analiza el tiempo del algoritmo subyacente.

El tiempo necesario para ejecutar un algoritmo está en función del número de datos de entrada. Denotemos por n al tamaño de la entrada. Debemos decidir como medir el tiempo y espacio requerido por el algoritmo. Podemos medir los requerimientos de espacio

determinando la cantidad de memoria requerida por las variables, contando el número de instrucciones ejecutadas. Para los fines de estimar el tiempo y espacio requerido de un algoritmo y para comparar algoritmos que realizan la misma función, usualmente es útil tener estimaciones más altas del rendimiento de los parámetros que el valor exacto. Estas estimaciones se han escrito utilizando la notación “O grande”.

Definición 2.1 Sean f y g funciones con dominio $\{1,2,3,\dots\}$.

a) Se denota

$$f(n)=O(g(n))$$

y se lee que $f(n)$ es de orden a lo más $g(n)$, si existe una constante positiva C tal que

$$|f(n)|\leq C|g(n)|,$$

para todos los enteros positivos n , excepto un número finito de ellos.

b) Se denota

$$f(n)=\Omega(g(n))$$

y se lee que $f(n)$ es de orden al menos $g(n)$ si existe una constante positiva C_1 tal que

$$|f(n)|\geq C_1|g(n)|$$

para todos los enteros positivos n , excepto un número finito de ellos.

c) Sean f y g funciones con dominio $\{1,2,3,\dots\}$. Se denota

$$f(n)=\Theta(g(n))$$

Decimos que $f(n)$ es del orden $g(n)$ o $f(n)$ es theta de $g(n)$ si $f(n)=O(g(n))$ y $f(n)=\Omega(g(n))$ para todos los enteros positivos n , excepto un número finito de ellos.

Ejemplo 2.2 Si $f(n)=n$ y $g(n)=2^n$, $n \in \mathbb{N}$, entonces $f(n)=O(g(n))$, en este caso g crece mucho más rápido que f .

Ejemplo 2.3 Observe que con $f(n)=n^2 + n + 3$, $g(n)=n^2$, $n \in \mathbb{N}$, se cumple que $f(n) = O(g(n))$. Como $n^2 \leq 3n^2$, $n \leq 3n^2$ y $3 \leq 3n^2$, se tiene que $f(n) = n^2 + n + 3 \leq 9g(n) = 3n^2 + 3n^2 + 3n^2$, esto quiere decir que $|f(n)| \leq C|g(n)|$, con $C=9$.

Ejemplo 2.4 Si $f(n)=60n^2 + 5n + 1$ y $g(n)=66n^2$ con $n \in \mathbb{N}$, entonces $f(n)=\Theta(g(n))$.

Por un lado, si $60n^2 \leq 60n^2$, $5n \leq 5n^2$ y $1 \leq n^2$ se tiene que $f(n) = 60n^2 + 5n + 1 \leq 60n^2 + 5n^2 + n^2 = 66n^2 = g(n)$, esto quiere decir que $|f(n)| \leq C|g(n)|$, $C=1$ y así, $f(n) = O(g(n))$.

Por otro lado, si $60n^2 \leq 60n^2$, $0 \leq 5n$, $0 \leq 1$ se tiene que $60n^2 \leq f(n) = 60n^2 + 5n + 1$. Si $g(n)=60n^2$, entonces $|g(n)| \leq C|f(n)|$ con $C=1$ y así, $f(n)=\Omega(g(n))$.

Teorema 2.1 Sea $a_k n^k + a_{k-1} n^{k-1} + \dots + a_1 n + a_0$ un polinomio de grado k con $k \in \mathbb{N} \cup \{0\}$, $n \in \mathbb{N}$, $a_k \geq 0$, se tiene que $f(n)=a_k n^k + a_{k-1} n^{k-1} + \dots + a_1 n + a_0 = \Theta(n^k)$.

Demostración:

Sea $f(n)=a_k n^k + a_{k-1} n^{k-1} + \dots + a_1 n + a_0$ un polinomio de grado k con $k \in \mathbb{N} \cup \{0\}$, $a_k \neq 0 \forall k \in \mathbb{N} \cup \{0\}$. Tomemos $C:=\max\{|a_k|, |a_{k-1}|, \dots, |a_1|, |a_0|\}$. Por la desigualdad del triángulo tenemos que:

$$\begin{aligned} |a_k n^k + a_{k-1} n^{k-1} + \dots + a_1 n + a_0| &\leq |a_k n^k| + |a_{k-1} n^{k-1}| + \dots + |a_1 n| + |a_0|, n \in \mathbb{N}, \\ |a_k n^k + a_{k-1} n^{k-1} + \dots + a_1 n + a_0| &\leq |a_k| n^k + |a_{k-1}| n^{k-1} + \dots + |a_1| n + |a_0|, \\ &\leq C n^k + C n^{k-1} + \dots + C n + C, \\ &\leq C n^k + C n^k + \dots + C n^k + C n^k, \\ &= (k + 1) C n^k. \end{aligned}$$

Si $g(n)=n^k$ se tiene que $|f(n)| \leq C_1 |g(n)|$ con $C_1 = (k + 1)C$. Por lo tanto, $f(n)=O(n^k)$.

Por otra parte, si $C:=\min\{a_k, a_{k-1}, \dots, a_1, a_0\}$ tenemos que:

$$\begin{aligned} C n^k &\leq a_k n^k \\ C n^{k-1} &\leq a_{k-1} n^{k-1} \\ &\vdots \\ C &\leq a_0, \\ C n^k + C n^{k-1} + \dots + C &\leq a_k n^k + a_{k-1} n^{k-1} + \dots + a_0, \\ C n^k + C n^{k-1} + \dots + C &\leq |a_k n^k + a_{k-1} n^{k-1} + \dots + a_0|, \\ C [n^k + n^{k-1} + \dots + 1] &\leq |a_k n^k + a_{k-1} n^{k-1} + \dots + a_0|, \\ C n^k &\leq |a_k n^k + a_{k-1} n^{k-1} + \dots + a_0|. \end{aligned}$$

Si $g(n)=n^k$ se tiene que: $C|g(n)| \leq |f(n)|$, esto es, $f(n)=\Omega(n^k)$. Por lo tanto, $f(n)=\Theta(n^k)$, lo que significa que todo polinomio de grado k con coeficientes positivos es $\Theta(n^k)$ ■.

Si un algoritmo requiere $f(n)$ unidades de tiempo para ejecutarse con una entrada de tamaño n y

$$f(n)=O(g(n)),$$

decimos que el tiempo requerido por el algoritmo es de orden a lo más $g(n)$. Similarmente, si un algoritmo requiere $f(n)$ unidades de memoria durante su ejecución para una entrada de tamaño n y

$$f(n)=O(g(n)),$$

decimos que el espacio requerido por el algoritmo es de orden a lo más $g(n)$.

Ejemplo 2.5 Considerar el problema de determinar el mejor, el peor y el caso promedio del tiempo requerido para ejecutar el siguiente algoritmo:

Dada una secuencia $S(1), S(2), \dots, S(N)$ y un valor KEY (KEY representa algo de interés en la secuencia). El Algoritmo 2.2 encuentra la primera ocurrencia de KEY y devuelve su ubicación en I . Si KEY no es encontrada, el algoritmo devuelve 0.

1. [Inicializar] $I:=1$.
2. [No encontrado] si $I>N$, entonces regresa 0 y termina.
3. [Prueba] Si $KEY=S(i)$, con $i \in \{1, \dots, N\}$ entonces termina y regresa el valor de i .
4. [Continúa la búsqueda] $I:=I+1$ vaya al paso 2.

Algoritmo 2.2

```
(1) for(i=1;i<=N;i++)
(2)   {
(3)   if(S[i]==key)
(4)   {printf("%d\n",i);
(5)   return 0; }
(6)   if(i==N)
(7)   {printf("%d\n",0); }
(8)   }
```

En el mejor caso KEY será encontrado cuando $i=1$, i.e., que el paso 3 sólo se ejecutó una vez. Así que el mejor caso del algoritmo tiene un tiempo de ejecución $O(1)$. En el peor caso, KEY no es encontrado en la secuencia y esto quiere decir que (3) se ejecutó N veces. Así

que el peor caso del algoritmo tiene un tiempo de ejecución $O(N)$.

Finalmente, en el caso promedio, *KEY* puede ser encontrado en la posición uno hasta la N ;

El promedio de pasos que pueden ser ejecutados es:

$$\frac{(1 + \dots + N) + N}{N + 1} \leq \frac{N^2 + N}{N + 1} = N,$$

Así que el caso promedio tiene un tiempo de ejecución $O(N)$.

A continuación se presenta el Algoritmo 2.3, cuyo objetivo es, ejemplificar la determinación del orden de un Algoritmo.

Ejemplo 2.6

Algoritmo 2.3

```
(1)exponencial <- function(n)
(2){
(3)  X<-c()
(4)  R<-c()
(5)  F<-c()
(6)  M<-c()
(7)  for(i in 1:n)
(8)  {
(9)  R[i]<-abs(rnorm(1,0))
(10) }
(11) for(i in 1:n)
(12) {
(13) X[i]<-(-log(R[i],2.71828183))
(14) }
(15) M<-sort(X)
(16) for(i in 1:n)
(17) {
(18) F[i]<-exp(-M[i])
(19)
(20) }
(21) plot(F,M,type="l",col='orange')
(22) }
```

El resultado de ejecutar el algoritmo anterior es la gráfica de la función e^{-x} .

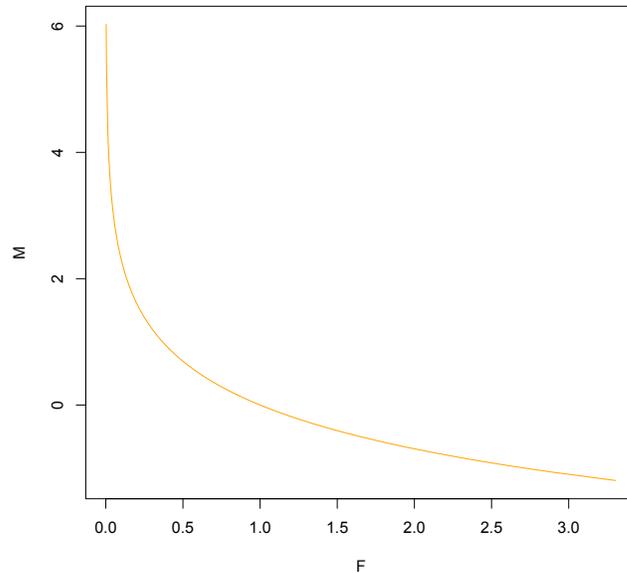


Figura 2.1: Función e^{-x} para $n=1000$.

El número de pasos que el algoritmo realiza son $6+3n$:

1. 4 pasos de las líneas (3-6).
2. n pasos de las líneas (7-10).
3. n pasos de las líneas (11-14).
4. 1 paso en la línea (15).
5. n pasos de las líneas (16-20).
6. 1 paso en la línea (21).

Así que $f(n)=6+3n$. Por lo tanto, por el Teorema 2.1 se tiene que $f(n)=\Theta(n)$.

Funciones de crecimiento comunes:

Forma theta	Nombre
$\Theta(1)$	Constante
$\Theta(\lg n)$	Logarítmica
$\Theta(n)$	Lineal
$\Theta(n^2)$	Cuadrática
$\Theta(n^m)$	Polinomial
$\Theta(n!)$	Factorial
$\Theta(m^n),$ $m \geq 2$	Exponencial

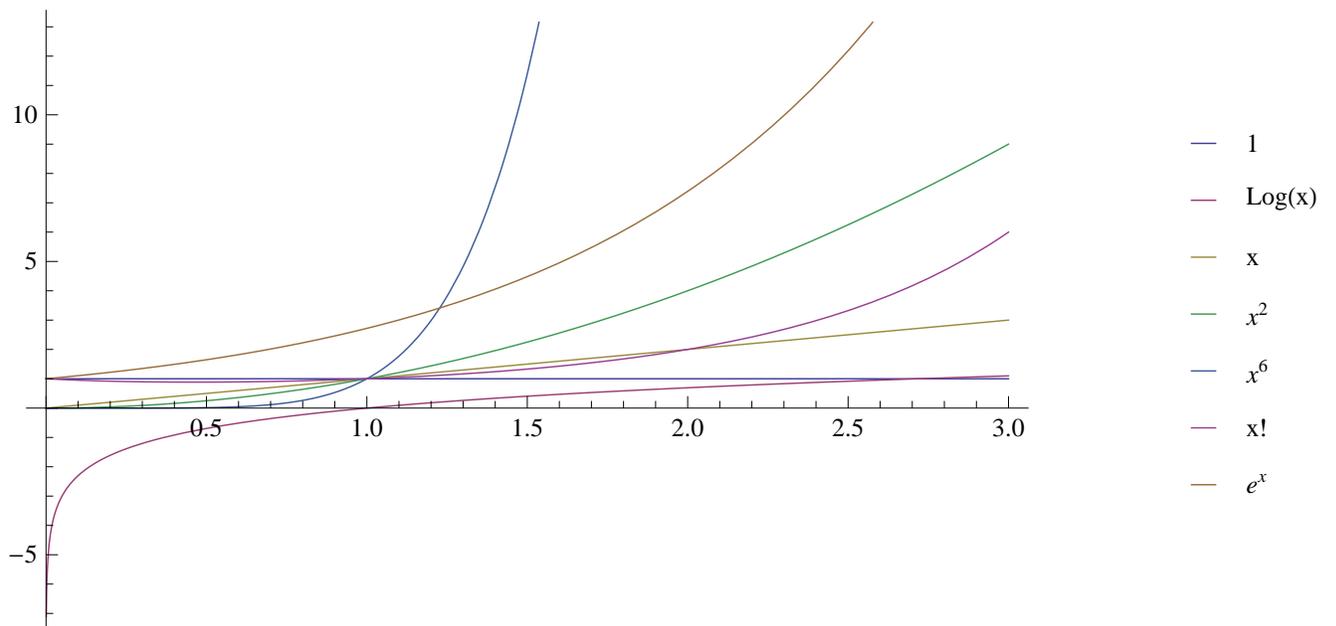


Figura 2.2: Funciones de Crecimiento.

2.2. Simulaciones de Procesos Estocásticos

2.2.1. Cadena de Markov en ‘The R Project for Statistical Computing’.

El siguiente algoritmo computacional genera valores de una cadena de Markov con una distribución inicial y matriz de transición dada. Dicho algoritmo se elaboró en el lenguaje de programación “The R Project for Statistical Computing”.

De acuerdo al Teorema 2.1, el Algoritmo 2.4 es de orden $\theta(n^2)$.

Sea $\langle X_n \rangle$ una cadena de Markov definida en el espacio de estados $S = \{0, 1, 2\}$ con distribución inicial $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ y matriz de transición:

$$\begin{pmatrix} 1/2 & 1/2 & 0 \\ 1/3 & 1/3 & 1/3 \\ 2/5 & 3/5 & 0 \end{pmatrix}$$

Algoritmo 2.4

```
Markov<-function(n)
{
Y<-c();j<-0;k<-0;m<-0;Y[1]<-sample(c(0,1,2),1, prob=c(1/3,1/3,1/3))
for (i in 1:n)
{
if (Y[i]==0)
{Y[i+1]<-sample(c(0, 1, 2),1, prob=c(1/2, 1/2, 0))
j<- j+1}
else if (Y[i]==1)
{Y[i+1]<-sample(c(0,1, 2),1, prob=c(1/3, 1/3, 1/3))
k<- k+1}
else
{Y[i+1]<-sample(c(0, 1, 2),1, prob=c(2/5, 3/5, 0))
m<- m+1}} Y}
```

2.2.2. Proceso Wiener en 'The R Project for Statistical Computing'.

El siguiente algoritmo computacional genera n valores del movimiento browniano con una partición aleatoria en el lenguaje de programación 'The R Project for Statistical Computing'.

De acuerdo al Teorema 2.1, el Algoritmo 2.5 es de orden $\theta(n)$.

Algoritmo 2.5

Browniano<-function(n)

{

t<-c() ;w<-c();t[1]=0;w[1]=0

for(i in 1:(n-1))

 {t[i+1]<-t[i]+runif(1)}

 for(i in 1:(n-1))

 {w[i+1]<-w[i]+rnorm(1,0,t[i+1]-t[i])}

plot(w,type="l")

}

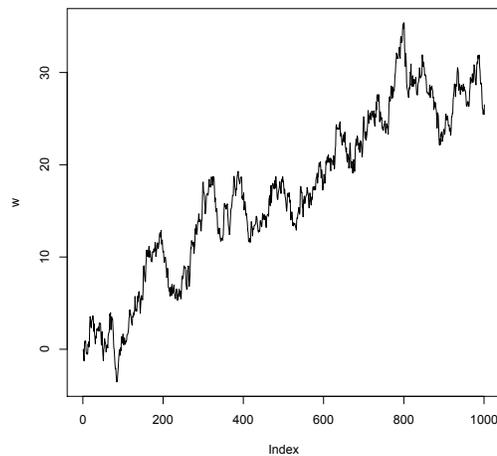


Figura 2.3: Simulación del Proceso Browniano hasta el tiempo $t[1000]$.

Capítulo 3

Método Monte Carlo

La creación de este método suele ligarse a los nombres de los matemáticos norteamericanos J. von Neumann y S. Ulam. Se considera a 1949 como el año del nacimiento del método de Monte Carlo. El nombre de *Monte Carlo* se debe al de una población del principado de Mónaco, célebre por su casa de juego [8].

El método de Monte Carlo es un método numérico que permite resolver problemas matemáticos mediante la simulación de variables aleatorias.

Es una técnica de resolución de problemas que se utiliza para aproximar la probabilidad de ciertos resultados mediante la realización de varias ejecuciones de una prueba, llamada simulaciones, usando variables aleatorias.

3.1. Método Monte Carlo

Muchos problemas numéricos en ciencias, ingeniería, estadística y finanzas son resueltos hoy en día a través del método de Monte Carlo [12]; esto es, a través de simulación de experimentos aleatorios en una computadora. Monte Carlo permite estimar valores que son difíciles de obtener analíticamente por medio de la repetición de un algoritmo computacional y el cómputo de los resultados obtenidos.

El siguiente procedimiento es una descripción del Método Monte Carlo en 5 pasos.

- P1) Crear un modelo que dependa de X_1, \dots, X_n variables aleatorias.
- P2) Generar una realización de X_1, \dots, X_n variables aleatorias.
- P3) Evaluar el modelo con los resultados obtenidos en P2).
- P4) Repetir los pasos P1), P2) y P3) N-veces.

P5) Analizar los resultados por medio de operaciones matemáticas y evaluar los resultados.

Ejemplo 3.1 Metropolis-Hastings

El método de Metropolis-Hastings fue propuesto en 1953 por Nicholas Constantine Metropolis y extendido en 1970 por Keith Hastings [18]. El algoritmo produce una cadena de Markov cuya distribución límite es la densidad de una distribución difícil de simular. Las principales características de estos métodos para el muestreo de una distribución de densidad $p(x)$ son:

1. Los cálculos sólo dependen de $p(x)$ a través de razones de la forma $p(x')/p(x)$, donde x' y x son puntos muestrales. Así, la constante de normalización no tiene que ser conocida y ninguna factorización de $p(x)$ es necesaria.
2. Una secuencia de muestras se obtiene por la simulación de una cadena de Markov. Las muestras resultantes están correlacionadas, esto es, que son dependientes.

Como ya sabemos, una cadena de Markov está caracterizada por su distribución inicial y su matriz de transición. Así que para simular una cadena de Markov necesitamos construir una matriz de transición.

Construcción de la Matriz de Transición

Sea π una distribución estacionaria y S el espacio de estados, $S \subset \mathbb{Z}$. Vamos a construir una matriz de transición P cuya distribución estacionaria es π . Ahora describiremos un procedimiento general para determinar la matriz de transición P , el cual contiene como casos especiales los métodos que han sido usados para problemas de mecánica estadística, en aquellos casos donde la matriz P fue hecha para satisfacer la condición de reversibilidad, Teorema 1.1.

La irreducibilidad de P se debe comprobar en cada aplicación específica. Es solamente necesario verificar que existe una probabilidad positiva de ir del estado i al estado j en un número finito de transiciones, para cada par de estados i y j en S .

Supondremos que $p_{ij} = q_{ij}\alpha_{ij}$, ($i \neq j$), con $p_{ii} = 1 - \sum_{j \neq i} p_{ij}$, donde $Q = \{q_{ij}\}$ es la matriz de transición de una cadena de Markov arbitraria con espacio de estados $0, 1, \dots, s$ y α_{ij} está dado por:

$$\alpha_{ij} = \frac{S_{ij}}{1 + \frac{\pi_i q_{ij}}{p_j q_{ji}}},$$

donde S_{ij} es una función simétrica de i y j escogida de tal forma que $0 \leq \alpha_{ij} \leq 1$, $\forall i$ y j . Con esta forma de p_{ij} se verifica que $\pi_i p_{ij} = \pi_j p_{ji}$, como se requiere.

Con el fin de simular el proceso, llevamos a cabo los pasos siguientes para cada tiempo t .

1. Asumamos que $X(t) = i$ y seleccionemos un estado j usando la distribución dada por la fila i -ésima de Q .

2. Tomamos $X(t+1) = j$ con probabilidad α_{ij} y $X(t+1) = i$ con probabilidad $1-\alpha_{ij}$.

Para la selección de S_{ij} consideraremos, sólo la cantidad $(\pi_j q_{ji})/(\pi_i q_{ij})$ en la simulación y de ahora en adelante nos referiremos a ella como la prueba de la razón. Dos opciones son dadas para S_{ij} , para todo i y j :

$$S_{ij}^{(M)} = \begin{cases} 1 + \frac{\pi_i q_{ij}}{\pi_j q_{ji}} & \text{para } \frac{\pi_j q_{ji}}{\pi_i q_{ij}} \geq 1, \\ 1 + \frac{\pi_j q_{ji}}{\pi_i q_{ij}} & \text{para } \frac{\pi_j q_{ji}}{\pi_i q_{ij}} \leq 1. \end{cases}$$

$S_{ij}^{(B)} = 1$ con $q_{ij}=q_{ji}$ y $s_{ij}=s_{ij}^{(M)}$ tenemos el método ideado por Metropolis et. al. (1953) con $q_{ij}=q_{ji}$ y $s_{ij}=s_{ij}^{(B)}$ tenemos el método de Barker's (1965). Observemos que en la Dinámica de Metropolis se tiene que $\alpha_{ij} = \min\{1, \frac{\pi_j q_{ji}}{\pi_i q_{ij}}\}$ cuando $\pi_{ij} \neq \pi_{ji}$ ya que:

$$S_{ij}^{(M)} = \begin{cases} 1 + \frac{\pi_i q_{ij}}{\pi_j q_{ji}} & \text{para } \frac{\pi_j q_{ji}}{\pi_i q_{ij}} \geq 1, \\ 1 + \frac{\pi_j q_{ji}}{\pi_i q_{ij}} & \text{para } \frac{\pi_j q_{ji}}{\pi_i q_{ij}} \leq 1. \end{cases}$$

Si $\frac{\pi_j q_{ji}}{\pi_i q_{ij}} \geq 1$, $S_{ij}^{(M)} = 1 + \frac{\pi_i q_{ij}}{\pi_j q_{ji}}$ entonces por definición, $\alpha_{ij}^{(M)} = 1$.

Si $\frac{\pi_j q_{ji}}{\pi_i q_{ij}} \leq 1$, $S_{ij}^{(M)} = 1 + \frac{\pi_j q_{ji}}{\pi_i q_{ij}}$ entonces $\alpha_{ij}^{(M)} = \frac{1 + \frac{\pi_j q_{ji}}{\pi_i q_{ij}}}{1 + \frac{\pi_j q_{ji}}{\pi_i q_{ij}}} = \frac{\pi_j q_{ji}}{\pi_i q_{ij}}$, i.e., $\alpha_{ij} = \min\{1, \frac{\pi_j q_{ji}}{\pi_i q_{ij}}\}$.

Cuando $q_{ij}=q_{ji}$ nosotros tenemos:

$$\alpha_{ij}^{(M)} = \begin{cases} 1 & \text{para } \frac{\pi_j}{\pi_i} \geq 1 \\ \frac{\pi_j}{\pi_i} & \text{para } \frac{\pi_j}{\pi_i} < 1 \end{cases}$$

$$\alpha_{ij}^{(B)} = \pi_j / (\pi_i + \pi_j).$$

Así, cuando $\pi_j = \pi_i$, nosotros tomaremos $X(t+1) = j$ con probabilidad 1 con el método de Metropolis y con probabilidad 1/2 con el método de Barker. Esto sugiere que el método de Metropolis puede ser preferible, ya que da un mejor muestreo de los estados. Mas general, escogemos:

$$S_{ij} = g[\min\{(\pi_i q_{ij})/(\pi_j q_{ji}), (\pi_j q_{ji})/(\pi_i q_{ij})\}]$$

donde la función $g(x)$ es escogida tal que $0 \leq g(x) \leq 1+x$ para $0 \leq x \leq 1$, y $g(x)$ puede ser simétrica en j e i . Por simplicidad nosotros consideramos sólo cadenas de Markov reversibles.

Observación 3.1 a) El algoritmo en términos computacionales es [17]:

- *Repetir for $j = 1, 2, \dots, N$.
- *Generar y con $q(x^{(j)}, \cdot)$ y u con $U(0, 1)$.
- *if $u \leq \alpha(x^{(j)}, y)$
- $x^{(j+1)} = y$.
- *else
- $x^{(j+1)} = x^{(j)}$.
- *Regresar los valores $\{x^{(1)}, x^{(2)}, \dots, x^{(N)}\}$.

b) Los resultados son considerados como una muestra de la densidad objetivo sólo después de que la cadena ha pasado la etapa transitoria y el efecto del valor inicial fijo se ha vuelto tan pequeño que puede ser ignorado.

Ejemplo:

Usando el método de Metropolis-Hastings, construir una cadena de Markov reversible y ergódica en el espacio de estados $S=\{1,2,3\}$ con distribución estacionaria,

$$\pi = \left(\frac{5}{18}, \frac{11}{18}, \frac{2}{18} \right),$$

y $q_{ij} = 1/3, \forall i,j \in S$.

Tenemos que,

$$\alpha_{ij} = \begin{pmatrix} 1 & 1 & 2/5 \\ 5/11 & 1 & 2/11 \\ 1 & 1 & 1 \end{pmatrix}$$

y

$$P_{ij} = \begin{pmatrix} 8/15 & 1/3 & 2/15 \\ 5/33 & 26/33 & 2/33 \\ 1/3 & 1/3 & 1/3 \end{pmatrix}.$$

El siguiente algoritmo, realizado en R, obtiene una muestra de una cadena de Markov con las características del ejemplo

Algoritmo 3.1

```
metropolis<-function(x0,pi1,pi2,pi3,n)
{
  PI<-c(pi1,pi2,pi3);Q <-matrix(c(1/3),nrow = 3, ncol = 3)
  P<-matrix(nrow=3,ncol=3); M<-c(); A<-matrix(nrow=3,ncol=3)
  for(j in 1:3)
  {
    for(i in 1:3)
    {
      A[i,j]<-min(1, PI[j]/PI[i])}
    }
  for(j in 1:3)
  {
    for(i in 1:3)
    {
      if(i>j || i<j)
```

```

P[i,j]<-Q[i,j]*A[i,j]
else
P[i,j]<-0
}
}
W<-c()
W<-rowSums(P)
for(j in 1:3)
{
for(i in 1:3)
{
if(i>j || i<j)
{
P[i,j]<-Q[i,j]*A[i,j]
}
else
{
P[i,j]<- (1-W[i])
}
}
}
M[1]<-x0
for (i in 1:(n-1))
{
if (M[i]==1)
{
M[i+1]<-sample(c(1, 2, 3),1, prob=c(P[1,1], P[1,2], P[1,3]))
}
else if (M[i]==2)
{
M[i+1]<-sample(c(1,2, 3),1, prob=c(P[2,1], P[2,2], P[2,3]))
}
else
{
M[i+1]<-sample(c(1, 2, 3),1, prob=c(P[3,1], P[3,2], P[3,3]))
}
}
}
M }

```

Ejecución del programa anterior con **metropolis(1,5/18,11/18,2/18,10)**

La matriz de aceptación :

	[,1]	[,2]	[,3]
[1,]	1.00000000	1 0.40000000	
[2,]	0.4545455	1 0.1818182	
[3,]	1.00000000	1 1.00000000	

La matriz de transición de la nueva cadena de Markov:

	[,1]	[,2]	[,3]
[1,]	0.53333333	0.33333333	0.13333333
[2,]	0.1515152	0.7878788	0.06060606
[3,]	0.33333333	0.33333333	0.33333333

Una realización:

[1] 1 3 3 2 2 2 1 2 2 2

Ejemplo 3.2 Una aplicación común para el método Monte Carlo es el cálculo del valor de una integral.

Supóngase que se desea calcular la siguiente integral,

$$M = \int_0^{\infty} xf(x)dx.$$

Sea f una función de densidad de una variable aleatoria X concentrada en $[0, +\infty]$

$$M = E_f[X].$$

Por la Ley de los Grandes Números, se tiene que el mejor estimador de la media poblacional es la media muestral, i.e., un buen estimador para M es la media aritmética de una muestra finita de variables aleatorias independientes e idénticamente distribuidas X_1, \dots, X_N , es decir:

$$\bar{M} := \frac{1}{N} \sum_{i=1}^N X_i \approx M,$$

\bar{M} es un estimador insesgado ya que $E[\bar{M}] = M$ y su varianza esta dada por

$$\text{Var}(\bar{M}) = \frac{\text{Var}(M)}{N}.$$

En lo que resta de este trabajo el paso P5) del Método Monte Carlo será promediar los resultados obtenidos en los pasos anteriores.

Teorema 3.1 Sea X una población normal con media desconocida μ o difícil de calcular y varianza finita σ^2 . Un estimador para μ es \bar{X} , el cual cumple que,

1. Es insesgado.
2. Es suficiente.
3. Es consistente.

Demostración:

1. Es insesgado porque

$$E(\bar{X}) = E\left[\frac{1}{N} \sum_{k=1}^N X_k\right] = \frac{1}{N} \sum_{k=1}^N E(X_k) = \frac{1}{N} \sum_{k=1}^N \mu = \mu \blacksquare$$

2. Sean X_1, \dots, X_n i.i.d $N(\mu, \sigma^2)$, suponiendo que la población es normal, se tiene que $T(x_1, \dots, x_n) = \bar{X}$, es una estadística suficiente para μ porque:

$$\begin{aligned} f(x|T = t) &= \frac{f(x, t)}{f_T(t)} = \frac{(2\pi\sigma^2)^{-n/2} \exp\{-[\sum_{i=1}^n (x_i - \bar{x})^2 + n(\bar{x} - \mu)^2]/2\sigma^2\}}{(2\pi\sigma^2/n)^{-1/2} \exp\{-n(\bar{x} - \mu)^2/2\sigma^2\}} \\ &= n^{-1/2} (2\pi\sigma^2)^{-(n-1)/2} \exp\left\{-\sum_{i=1}^n (x_i - \bar{x})^2 / (2\sigma^2)\right\}, \end{aligned}$$

la cual no depende de μ ■

3. Por la desigualdad de Chebyshev, para cualquier número dado $\epsilon > 0$,

$$P(|\bar{X} - \mu| \geq \epsilon) \leq \frac{V(\bar{X})}{\epsilon^2},$$

$$P(|\bar{X} - \mu| < \epsilon) \geq 1 - \frac{\sigma^2}{n\epsilon^2},$$

Por lo tanto,

$$\lim_{n \rightarrow \infty} P(|\bar{X} - \mu| < \epsilon) = 1 \blacksquare$$

Observación 3.2 En este estudio se considera al **estimador de Monte Carlo** como \bar{X} , el cual tiene un error estándar:

$$D(\bar{X}) = \sqrt{V(\bar{X})} = \sqrt{\frac{\sigma^2}{N}} = \frac{\sigma}{\sqrt{N}}.$$

Una forma de reducir el error, es reducir la varianza o aumentar el tamaño de la muestra. En este trabajo nos enfocaremos a reducir la varianza usando dos técnicas básicas de reducción de varianza, variables aleatorias antitéticas y variables aleatorias de control, las cuales serán implementadas en el siguiente capítulo en el contexto de opciones financieras [2].

3.2. Reducción de Varianza

El valor esperado y la varianza son los parámetros por medio de los cuales es posible caracterizar el comportamiento de los datos reunidos en el desarrollo de una situación experimental; el valor esperado permite establecer el valor sobre el cual se centra la distribución de la probabilidad, mientras que la varianza proporciona información acerca de la manera como se distribuyen los datos obtenidos.

Las simulaciones que proveen una estimación de un parámetro de interés siempre lo hacen con un error estándar proporcional a $1/\sqrt{N}$ donde N es el tamaño de la muestra. Las técnicas de reducción de varianza discutidas en este capítulo son:

1. Variables aleatorias antitéticas.
2. Variables de Control.

3.2.1. Variables Aleatorias Antitéticas

Un par de variables aleatorias reales (Y, Y^*) es llamado un par antitético si Y y Y^* tienen la misma distribución y están negativamente correlacionadas. La principal aplicación de variables aleatorias antitéticas en la estimación de Monte Carlo está basada en el siguiente teorema.

Teorema 3.2 (Estimador antitético). Sea Y una variable aleatoria, N un número par y sean $(Y_1, Y_1^*), \dots, (Y_{N/2}, Y_{N/2}^*)$ pares de variables aleatorias antitéticas, donde cada Y_k y Y_k^* está distribuida como Y . El estimador antitético:

$$\hat{I}^{(a)} = \frac{1}{N} \sum_{k=1}^{N/2} \{Y_k + Y_k^*\}.$$

Es un estimador insesgado de $l = \mathbb{E}Y$, con varianza,

$$\begin{aligned}
\text{Var}(\hat{l}^{(a)}) &= \text{Var}\left(\frac{1}{N} \sum_{k=1}^{N/2} \{Y_k + Y_k^*\}\right) \\
&= \frac{1}{N^2} \text{Var}\left(\sum_{k=1}^{N/2} \{Y_k + Y_k^*\}\right) \\
&= \frac{1}{N^2} \text{Var}\left(\sum_{k=1}^{N/2} \{Y_k\}\right) + \text{Var}\left(\sum_{k=1}^{N/2} \{Y_k^*\}\right) + 2\text{Cov}\left(\sum_{k=1}^{N/2} \{Y_k\}, \sum_{k=1}^{N/2} \{Y_k^*\}\right) \\
&= \frac{1}{N^2} \frac{N}{2} \text{Var}(Y) + \frac{N}{2} \text{Var}(Y^*) + 2\text{Cov}\left(\sum_{k=1}^{N/2} \{Y_k\}, \sum_{k=1}^{N/2} \{Y_k^*\}\right) \\
&= \frac{1}{N^2} \frac{N}{2} \text{Var}(Y) + \frac{N}{2} \text{Var}(Y^*) + 2\text{Cov}\left(\sum_{k=2}^{N/2} \{Y_k + Y_1\}, \sum_{k=2}^{N/2} \{Y_k^* + Y_1^*\}\right) \\
&= \frac{1}{2N} \text{Var}(Y) + \frac{N}{2} \text{Var}(Y^*) + 2[\text{Cov}\left(\sum_{k=2}^{N/2} Y_k, \sum_{k=2}^{N/2} Y_k^*\right) + \text{Cov}\left(\sum_{k=2}^{N/2} Y_k, Y_1^*\right) + \text{Cov}(Y_1, \sum_{k=2}^{N/2} Y_k^*) \\
&\quad + \text{Cov}(Y_1, Y_1^*)] \\
&= \frac{1}{N^2} \frac{N}{2} \text{Var}(Y) + \frac{N}{2} \text{Var}(Y^*) + 2[\text{Cov}\left(\sum_{k=2}^{N/2} Y_k, \sum_{k=2}^{N/2} Y_k^*\right) + \text{Cov}(Y_1, Y_1^*)] \\
&= \frac{1}{N^2} \frac{N}{2} \text{Var}(Y) + \frac{N}{2} \text{Var}(Y^*) + 2\frac{N}{2} [\text{Cov}(Y, Y^*)] \\
&= \frac{N/2}{N^2} [\text{Var}(Y) + \text{Var}(Y^*) + 2\text{Cov}(Y, Y^*)] \\
&= \frac{N/2}{N^2} [2\text{Var}(Y) + 2\text{Cov}(Y, Y^*)] \\
&= \frac{\text{Var}(Y) + \text{Cov}(Y, Y^*)}{N} \\
&= \frac{\text{Var}(Y)}{N} + \frac{\text{Cov}(Y, Y^*)}{N} \\
&= \frac{\text{Var}(Y)}{N} + \frac{\text{Cov}(Y, Y^*)}{N} \frac{\text{Var}(Y)}{\text{Var}(Y)} \\
&= \frac{\text{Var}(Y)}{N} + \rho_{YY^*} \frac{\text{Var}(Y)}{N} \\
&= \frac{\text{Var}(Y)}{N} (1 + \rho_{YY^*}).
\end{aligned}$$

Donde ρ_{YY^*} es el coeficiente de correlación entre Y y Y^* .

Como la varianza del estimador de Monte Carlo es:

$$\text{Var}(\hat{l}) = \text{Var}\left(\frac{\sum_{k=1}^N Y_i}{N}\right) = \text{Var}(Y)/N,$$

y $-1 \leq \rho_{YY^*} < 0$, se tiene que la varianza del estimador antitético, $\frac{\text{Var}(Y)}{N} * (1 + \rho_{YY^*})$, es menor que la varianza del estimador de Monte Carlo.

Esto muestra que el uso de variables aleatorias antitéticas conduce a una menor varianza del estimador. La cantidad de reducción depende crucialmente de la cantidad de correlación negativa entre las variables antitéticas.

3.2.2. Variables de Control

Sea Y una variable aleatoria que representa la salida de una ejecución de una simulación. Una variable Y^* , obtenida de la misma ejecución de una simulación, es llamada variable de control de Y si Y y Y^* están correlacionadas (negativamente o positivamente) y la esperanza de Y^* es conocida. El uso de variables de control para la reducción de varianza está basado en el siguiente:

Teorema 3.3 (*Estimador de variable de control*) Sea Y_1, \dots, Y_N la salida de N ejecuciones independientes de una simulación y sean Y_1^*, \dots, Y_N^* sus correspondientes variables de control, con $E(Y_k^*) = l^*, k = 1, \dots, n$, conocida. Sea $\rho_{Y_k Y_k^*}$ el coeficiente de correlación entre cada Y_k y Y_k^* , Para cada $\alpha \in \mathbb{R}$ el estimador (lineal) se define como

$$\bar{l} = \frac{1}{N} \sum_{k=1}^N (Y_k - \alpha(Y_k^* - l^*)).$$

Tomemos $\alpha = \text{Cov}(Y, Y^*) / \text{Var} Y^* = \rho_{Y Y^*}$,

$$\begin{aligned} \text{Var}(\bar{l}) &= \text{Var}\left(\frac{1}{N} \sum_{k=1}^N (Y_k - \alpha(Y_k^* - l^*))\right), \\ &= \frac{1}{N^2} [\text{Var}\left(\sum_{k=1}^N (Y_k - \alpha(Y_k^* - l^*))\right)], \\ &= \frac{1}{N^2} [\text{Var}\left(\sum_{k=1}^N Y_k - \sum_{k=1}^N \alpha Y_k^*\right) + \text{Var}(\alpha l^*) + 2\text{Cov}\left(\sum_{k=1}^N Y_k - \sum_{k=1}^N \alpha Y_k^*, \alpha l^*\right)], \\ &= \frac{1}{N^2} [\text{Var}\left(\sum_{k=1}^N Y_k - \sum_{k=1}^N \alpha Y_k^*\right)], \\ &= \frac{1}{N^2} [\text{Var}\left(\sum_{k=1}^N Y_k\right) + \text{Var}\left(\sum_{k=1}^N \alpha Y_k^*\right) - 2\text{Cov}\left(\sum_{k=1}^N Y_k, \sum_{k=1}^N \alpha Y_k^*\right)], \end{aligned}$$

$$\begin{aligned}
&= \frac{1}{N^2} \left[\text{Var} \left(\sum_{k=1}^N Y_k \right) + \text{Var} \left(\sum_{k=1}^N \alpha Y_k^* \right) - 2\alpha \text{Cov} \left(\sum_{k=1}^N Y_k, \sum_{k=1}^N Y_k^* \right) \right], \\
&= \frac{1}{N^2} \left[\text{Var} \left(\sum_{k=1}^N Y_k \right) + \text{Var} \left(\sum_{k=1}^N \alpha Y_k^* \right) - 2\alpha N \text{Cov}(Y, Y^*) \right], \\
&= \frac{1}{N^2} \left[N \text{Var}(Y) + \alpha^2 N \text{Var}(Y) - 2\alpha N \text{Cov}(Y, Y^*) \right], \\
&= \frac{\text{Var}(Y) + \alpha^2 \text{Var}(Y) - 2\alpha \text{Cov}(Y, Y^*)}{N}, \\
&= \frac{\text{Var}(Y) [1 + \alpha^2 - 2\alpha \text{Cov}(Y, Y^*) / \text{Var}(Y)]}{N}, \\
&= \frac{\text{Var}(Y) [1 - \rho_{YY^*}^2]}{N}.
\end{aligned}$$

Como la varianza del estimador de Monte Carlo es:

$$\text{Var}(\hat{l}) = \text{Var} \left(\frac{\sum_{k=1}^N Y_i}{N} \right) = \text{Var}(Y)/N,$$

y $0 < \rho_{YY^*}^2 \leq 1$, se tiene que la varianza del estimador de control, $\frac{\text{Var}(Y)[1 - \rho_{YY^*}^2]}{N}$, es menor que la varianza del estimador de Monte Carlo.

La cantidad de reducción de varianza depende crucialmente de la cantidad de correlación entre las variables de control.

Capítulo 4

Finanzas

La historia de los mercados de futuros inicia en la Edad Media con la necesidad de satisfacer las demandas de los agricultores y comerciantes. El agricultor en años de escasez es posible que obtenga precios relativamente altos para su cosecha y en años de abundancia precios bajos. En el caso de un comerciante, la situación de exceso de oferta puede que le aporte precios favorables, en otros casos precios exorbitantes. Así que para mediar esa situación, es conveniente que meses antes de tener la cosecha físicamente, el agricultor y el comerciante pacten un precio para la cosecha esperada. Uno de los mercados financieros creados para servir de enlace entre agricultores y comerciantes es el 'Chicago Board of Trade' fundado en 1848. Al cabo de pocos años se produjo el primer tipo de contrato de futuros llamado "to-arrive"[9].

Existen básicamente dos tipos de mercados financieros: el mercado organizado y el mercado sobre mostrador, también llamado mercado OTC (por las iniciales en inglés de "over-the-counter" markets) [6]. La ventaja clave del mercado "over-the-counter" es que el objeto y términos de la contratación no tienen porqué ser los que especifica el mercado organizado. Una desventaja que presenta es que en las transacciones "over-the-counter" normalmente se dará cierto riesgo de crédito (es decir, hay un pequeño riesgo de que el contrato no sea satisfecho por alguna de las dos partes).

Definición 4.1 *Un acuerdo para comprar o vender un activo en una fecha futura a un precio determinado en un mercado organizado es un **Contrato de futuros**.*

Definición 4.2 *Un acuerdo entre dos partes (reforzado legalmente) que obliga a una de las partes a comprar y a la otra a vender un activo (financiero) a un precio preestablecido en una fecha futura es un **Contrato a plazo o "forward"**. Estos contratos son exclusivos de los mercados "over-the-counter".*

4.0.3. Contratos de Opción

Existen básicamente dos tipos de opciones: de compra y de venta (call y put).

Definición 4.3 Una **opción** es un contrato financiero que otorga a su comprador (o propietario) el derecho, pero no la obligación, de comprar o vender un bien subyacente (o una cierta cantidad de un bien) a un precio pactado de antemano, llamado precio de ejercicio.

Esta compra o venta se efectúa en alguna fecha futura o lapso de tiempo establecido al inicio del contrato. El vendedor de dicha opción se obliga a vender o comprar el activo subyacente al precio convenido.

Las opciones se clasifican en dos tipos, dependiendo del derecho que adquiere el comprador,

1. Una **Opción de compra** el propietario adquiere el derecho pero no la obligación de comprar el bien subyacente al precio de ejercicio y en una fecha futura establecida al inicio del contrato. En este caso el vendedor de la opción se obliga a vender el activo subyacente al precio convenido.
2. Una **Opción de venta** el propietario adquiere el derecho pero no la obligación de vender el bien subyacente al precio de ejercicio y en una fecha futura establecida al inicio del contrato. En este caso el vendedor de la opción se obliga a comprar el activo subyacente al precio convenido.

El estilo de la opción lo define el periodo o fecha en que ésta puede ser ejercida. En esta tesis sólo nos enfocaremos a las siguientes:

1. Una **Opción Europea** otorga a su comprador el derecho, pero no la obligación, de comprar o vender un bien subyacente al precio de ejercicio en una fecha futura convenida al inicio del contrato, llamada fecha de vencimiento.
2. Una **Opción Americana** otorga a su comprador el derecho, pero no la obligación, de comprar o vender un bien subyacente al precio de ejercicio durante el lapso de tiempo comprendido entre la emisión del contrato y la fecha de vencimiento.

En la negociación de un opción están involucrados el comprador o poseedor de la opción y el vendedor o emisor de la opción, para estas dos partes el riesgo asumido por la compra-venta del contrato es diferente, pues el poseedor siempre tiene el derecho y ninguna obligación, mientras que el emisor siempre tiene la obligación de satisfacer la demanda del poseedor de la opción. El vendedor o emisor recibirá un pago por parte del poseedor de la opción, a este pago se le conoce comúnmente como la prima del contrato (**costo de la opción en $t=0$**), i.e., la prima es la compensación al vendedor de la opción por el riesgo que asume. El cálculo de este precio es el problema conocido como **valuación de opciones** (*option pricing*).

4.0.4. Valuación de Opciones

La prima de una opción o costo de una opción se negocia en función de la ley de oferta y demanda que establece el mercado. No obstante, existen modelos teóricos que tratan de determinar el precio de la opción en función de una serie de parámetros. Sea $S(t)$ el precio del activo en el tiempo $t \in [0, T]$ y k el precio de ejecución.

La ganancia que da lugar en el momento del ejercicio por parte del comprador se denomina “pay-off”.

El valor de la opción (“pay-off”) para una opción **Europea tipo call** es una función $V: [0, \infty) \rightarrow [0, \infty)$, definida como:

$$V(S(t), t) = \text{máx}\{S(t) - k, 0\}.$$

Para el caso de una opción **Europea tipo put**,

$$V(S(t), t) = \text{máx}\{k - S(t), 0\}.$$

El valor de la opción (“pay-off”) para una opción **Americana tipo call** es una función $G: [0, \infty) \rightarrow [0, \infty)$, definida como:

$$G(S(t), t) = \text{máx}\{S(t) - k, 0\}.$$

Para el caso de una opción **Americana tipo put**,

$$G(S(t), t) = \text{máx}\{k - S(t), 0\}.$$

Para obtener el valor del “pay-off” al día de hoy, los multiplicamos por el factor de descuento e^{-rt} con r la tasa de interés instantánea.

Existen diferentes modelos para simular el precio de un activo con riesgo y sin riesgo en un período de tiempo. En este trabajo, se tomaron en cuenta los siguientes modelos:

Definición 4.4 Sea $S(t)$, $t \in [0, T]$, decimos que es un **activo sin riesgo**, que paga una tasa de interés instantánea $r \geq 0$, si evoluciona de acuerdo a la siguiente ley:

$$\frac{dS(t)}{S(t)} = rdt, s(0) = 1,$$

cuya solución es:

$$S(t) = e^{rt}.$$

Definición 4.5 Sea $S(t)$, $t \in [0, T]$, decimos que es un **activo con riesgo**, si evoluciona aleatoriamente, según la ley,

$$\frac{dS(t)}{S(t)} = \mu dt + \sigma dW_t,$$

donde,

1. μ es la tendencia o rendimiento medio de un activo.
2. σ la volatilidad.
3. W_t es un movimiento Browniano estándar.

Una solución a la ecuación diferencial estocástica anterior fue propuesta por Samuelson en [16]:

$$S(t) = S(0)\exp\left[\sigma W_t + \left(\mu - \frac{1}{2}\sigma^2\right)t\right],$$

donde $S(t)$ representa al movimiento Browniano Geométrico.

Debido a la consideración de un rendimiento esperado μ , la Definición 4.5 no es independiente de las preferencias al riesgo de los agentes que participan en el mercado del subyacente. En efecto, entre mayor sea la aversión al riesgo de un agente, mayor tiene que ser el rendimiento medio esperado, μ , a fin de que el premio $v = \mu - r$ le sea atractivo al agente. Si se supone que todos los agentes son neutrales al riesgo, es decir, no requieren de un premio para inducirlos a participar en el mercado, entonces $v = 0$, así, $\mu = r$ y de esta manera el rendimiento medio esperado de cualquier activo es la tasa de interés libre de riesgo, r . Otra forma de medir el premio al riesgo, de uso más frecuente, consiste en estandarizar v por unidad de varianza (más precisamente por unidad de desviación estándar), es decir, $\lambda = v/\sigma$. Como antes, si los agentes no requieren de un premio para inducirlos a participar en el mercado, entonces $\lambda = 0$, lo cual implica, a su vez, que $\mu = r$. En este caso se dice que el **movimiento Browniano Geométrico** está definido sobre una medida de probabilidad **neutral al riesgo**. El teorema que formaliza esto fue demostrado por Girsanov en [6].

El teorema de Girsanov (1960) construye explícitamente una medida de probabilidad equivalente, definida sobre el espacio original, que permite transformar un movimiento Browniano Geométrico con tendencia igual al rendimiento medio esperado del activo en un movimiento Browniano Geométrico, neutral al riesgo, con tendencia igual a la tasa de interés libre de riesgo. Esto quiere decir que si dos agentes tienen diferentes expectativas sobre el rendimiento promedio del activo subyacente, ellos están dispuestos a omitirlas en sus decisiones de inversión siempre y cuando la volatilidad del activo subyacente se mantenga constante [6].

Black, Scholes y Merton

A principios de la década de los setentas, Fisher Black y Myron Scholes publicaron su artículo en “The Pricing of Options and Corporate Liabilities, The Journal of Political Economy” [5].

En su investigación, bajo condiciones de equilibrio (condiciones de no arbitraje), desarrollaron un modelo para valorar opciones europeas sobre una acción que no paga dividendos. Los supuestos básicos (o condiciones ideales de los mercados de acciones y opciones) del modelo de Black-Scholes son:

- i) El activo subyacente es una acción que no paga dividendos durante la vida del contrato.
- ii) El precio del activo subyacente es conducido por el movimiento Browniano Geométrico con riesgo neutral.
- iii) La volatilidad del precio del activo subyacente se mantiene constante a través del tiempo.
- iv) Las ventas en corto del subyacente en cuestión son permitidas.
- v) El mercado del subyacente es líquido y divisible, es decir, el subyacente siempre se puede comprar y vender en cualquier fracción del título.
- vi) No hay costos de transacción (comisiones e impuestos).
- vii) El mercado opera en forma continua, es decir, no hay sábados, domingos ni días festivos.
- viii) Existe un mercado de crédito, un sistema bancario, en el que los agentes pueden prestar y pedir prestado a una tasa de interés constante para todos los plazos, y libre de riesgo (tasa de interés pasiva igual a la activa).
- ix) Todos los agentes comparten exactamente la misma información, es decir, la información es simétrica.
- x) Los mercados están en equilibrio, es decir, no existen oportunidades de arbitraje.
- xi) La tasa de interés libre de riesgo es igual a la tendencia e igual a la tasa de descuento e igual a la tasa de rendimiento.

El modelo de Black-Scholes describe la evolución de los precios de los activos a través de la ecuación diferencial

$$\frac{dS(t)}{S(t)} = rdt + \sigma dW_t,$$

donde:

1. r es la tasa de interés libre de riesgo [13].
2. σ la volatilidad.
3. W_t es un movimiento Browniano estándar.

Cuya solución es el movimiento Browniano Geométrico con riesgo neutro.

La formulación de la valuación de una opción europea de Black, Sholes y Merton es:

Opción tipo Europea call,

$$C = S \phi \left(\frac{\ln \left(\frac{s}{k} \right) + \left(r + \frac{1}{2} \sigma^2 \right) T}{\sigma \sqrt{T}} \right) - k e^{-rT} \phi \left(\frac{\ln \left(\frac{s}{k} \right) + \left(r - \frac{1}{2} \sigma^2 \right) T}{\sigma \sqrt{T}} \right)$$

Opción Europea tipo put,

$$P = k e^{-rT} \phi \left(\frac{\ln \left(\frac{k}{s} \right) + \left(r + \frac{1}{2} \sigma^2 \right) T}{\sigma \sqrt{T}} \right) - S \phi \left(\frac{\ln \left(\frac{k}{s} \right) + \left(r - \frac{1}{2} \sigma^2 \right) T}{\sigma \sqrt{T}} \right)$$

donde:

1. T es el tiempo de ejecución.
2. S el precio inicial del activo.
3. σ^2 la tasa de variación en el precio del activo.
4. r tasa de interés libre de riesgo.
5. k es el precio de ejercicio de la opción.

4.0.5. Algoritmos Financieros

En esta sección se dará una descripción de los algoritmos financieros. Los Algoritmos desarrollados en esta sección son:

1. Costo de un activo sin riesgo.
2. Costo de un activo con riesgo.
3. Costo de una opción Americana y Europea tipo call.
4. Costo de una opción Americana y Europea tipo call al riesgo neutral.
5. Simulación MC del ejercicio de una opción Americana tipo call.
6. Variables Antitéticas: Costo de una opción Europea tipo call al riesgo neutral.
7. Variables de Control: Costo de una opción Europea tipo call al riesgo neutral.

Costo de un activo sin riesgo (Algoritmo 4.1)

El algoritmo requiere de los parámetros r y t , donde r es la tasa de interés instantánea y t es el tiempo.

Paso 1. Generar los precios S_1, S_2, \dots, S_t de acuerdo a la Definición 4.4 con simulaciones independientes.

Paso 3. Imprimir y graficar los precios S_1, S_2, \dots, S_t .

Costo de un activo con riesgo (Algoritmo 4.2)

El algoritmo requiere de los parámetros n , S_0 , v y m , donde n es el tiempo, S_0 el precio de un activo al día de hoy, v la volatilidad y m la tendencia.

Paso 1. Generar t_1, \dots, t_{nt} con $t_{i+1} - t_i \stackrel{i.i.d}{\sim} U(0, 1)$, $i = 1, \dots, nt$, donde nt es el número de tiempos generados en el intervalo $(0, n)$.

Paso 2. Generar w_1, \dots, w_{nt} con $w_{i+1} - w_i \stackrel{i.i.d}{\sim} N(0, t_{i+1} - t_i)$, $i = 1, \dots, nt$, con los tiempos generados en el Paso 1.

Paso 3. Obtener y_1, \dots, y_{nt} de acuerdo a la Definición 4.5 con los valores generados en los Pasos 1 y 2.

Paso 4. Graficar los valores obtenidos en el Paso 3.

Costo de una opción Americana y Europea tipo call. (Algoritmo 4.3)

El algoritmo requiere de los parámetros $n1$, n , S_0 , v , m , k y r donde $n1$ es el número de simulaciones del precio de un contrato al tiempo n , S_0 el precio al día de hoy de un activo, v la volatilidad, m la tendencia, k el precio de ejercicio y r es la tasa de descuento.

Paso 1. Generar los precios y_1, y_2, \dots, y_{nt} de un activo de acuerdo al Algoritmo 4.2.

Paso 2. Para cada y_1, y_2, \dots, y_{nt} obtener el $\max\{y_i - k, 0\}$ con $i = 1, 2, 3, \dots, nt$.

Paso 3. Sólo guardar $\max\{y_{nt} - k, 0\}$ en nuevo vector, M , de longitud $n1$.

Paso 4. Obtener el promedio de M al día de hoy que es el precio de la opción.

Observación 4.1 *El (Algoritmo 4.3) se puede convertir al algoritmo que genera el costo de una opción Americana y Europea tipo call al riesgo neutral, sólo haciendo la tendencia igual a la tasa de interés.*

En esta tesis no se utilizó esa modificación. Se utilizó una partición unitaria del tiempo.

Costo de una opción Americana y Europea tipo call al riesgo neutral. (Algoritmo 4.4)

El algoritmo requiere de los parámetros $n1$, n , S_0 , v , k y r donde $n1$ es el número de simulaciones del precio de un contrato al tiempo n , S_0 el precio al día de hoy de un activo, v la volatilidad, k el precio de ejercicio y r es la tasa de descuento y la tendencia.

Paso 1. Generar w_1, \dots, w_n con $w_{i+1} - w_i \stackrel{i.i.d}{\sim} N(0, 1)$, $i = 1, \dots, n$.

Paso 2. Obtener y_1, \dots, y_n de acuerdo a la Definición 4.5 con los valores generados en el Paso 1.

Paso 3. Para cada y_1, y_2, \dots, y_n obtener el $\max\{y_i - k, 0\}$ con $i = 1, 2, 3, \dots, n$.

Paso 4. Sólo guardar $\max\{y_n - k, 0\}$ en nuevo vector, M , de longitud $n1$.

Paso 5. Obtener el promedio de M al día de hoy que es el precio de la opción.

Simulación MC del ejercicio de una opción Americana tipo call (Algoritmo 4.5)

El algoritmo requiere de los parámetros $n1$, n , S_0 , v y m , donde $n1$ es el número de simulaciones, n es el tiempo, S_0 el precio de un activo al día de hoy, v la volatilidad, m la tendencia, precio de ejercicio y r factor de descuento.

Paso 1. Generar los precios y_1, y_2, \dots, y_{n1} de un activo de acuerdo al Algoritmo 4.2.

Paso 2. Para cada y_1, y_2, \dots, y_{n1} obtener el $\max\{y_i - k, 0\}$ con $i = 1, 2, 3, \dots, n$.

Paso 3. Si el $\max\{y_i - k, 0\}$ con $i = 1, 2, 3, \dots, n$ es mayor que cero detener el programa e imprimir el precio de la opción al día de hoy.

Técnicas de reducción de varianza: Variables antitéticas (Algoritmo 4.6)

Paso 1. Generar w_1, \dots, w_n con $w_{i+1} - w_i \stackrel{i.i.d}{\sim} N(0, 1)$, $i = 1, \dots, n$.

Paso 2. Obtener w_1, \dots, w_n con $w_{i+1} - w_i \stackrel{i.i.d}{\sim} -N(0, 1)$, $i = 1, \dots, n$, con los inversos aditivos de $N(0, 1)$ del Paso 1.

Paso 3. Obtener y_1, \dots, y_n de acuerdo a la Definición 4.5 con los valores generados en el Paso 1 y para cada y_1, y_2, \dots, y_n obtener el $\max\{y_i - k, 0\}$ con $i = 1, 2, 3, \dots, n$.

Paso 4. Obtener y_1, \dots, y_n de acuerdo a la Definición 4.5 con los valores generados en el Paso 2 y para cada y_1, y_2, \dots, y_n obtener el $\max\{y_i - k, 0\}$ con $i = 1, 2, 3, \dots, n$.

Paso 5. Guardar el promedio del $\max\{y_n - k, 0\}$ obtenido en el Paso 3 con el $\max\{y_n - k, 0\}$ obtenido en el Paso 4 en el vector M.

Paso 6. Obtener el promedio de M al día de hoy, el cual es el precio de la opción con reducción de varianza.

Observación 4.2 a) El par antitético aquí utilizado es $(N(0, 1), -N(0, 1))$.

b) El estimador es:

$$\frac{\sum_{k=1}^{n1} \left\{ \frac{Y_k + Y_{k^*}}{2} \right\}}{n1} = \frac{\sum_{k=1}^{n1} \{Y_k + Y_{k^*}\}}{2n1}.$$

si $n1 = N/2$, tenemos el estimador antitético:

$$\frac{\sum_{k=1}^{N/2} \{Y_k + Y_{k^*}\}}{N}.$$

Técnicas de reducción de varianza: Variables de control (Algoritmo 4.7)

Paso 1. Obtener el costo de la opción con el Algoritmo 4.3.

Paso 2. Generar w_1, \dots, w_n con $w_{i+1} - w_i \stackrel{i.i.d}{\sim} N(0, 1)$, $i = 1, \dots, n$.

Paso 3. Obtener y_1, \dots, y_n de acuerdo a la Definición 4.5 con los valores generados en el Paso 2 y para cada y_1, y_2, \dots, y_n obtener el $\max\{y_i - k, 0\}$ con $i = 1, 2, 3, \dots, n$.

Paso 4. Obtener w_1, \dots, w_n con $w_{i+1} - w_i \stackrel{i.i.d}{\sim} -N(0, 1)$, $i = 1, \dots, n$, con los inversos aditivos de $N(0, 1)$ del Paso 2.

Paso 5. Obtener y_1, \dots, y_n de acuerdo a la Definición 4.5 con los valores generados en el Paso 4 y para cada y_1, y_2, \dots, y_n obtener el $\max\{y_i - k, 0\}$ con $i = 1, 2, 3, \dots, n$.

Paso 6. Sumar el $\max\{y_n - k, 0\}$ obtenido en el Paso 3, con el obtenido en el Paso 5 menos el costo de la opción obtenido en el Paso 1. Guardarlo en un vector, M, y ejecutar todo el algoritmo n1 veces.

Paso 7. Obtener el promedio de M al día de hoy, el cual es el precio de la opción con reducción de varianza.

Observación 4.3 a) *En este caso, la variable de control es obtenida en el Paso 1 con $E(Y_k^*) = l^*$.*

b) *Las variables están perfectamente correlacionadas negativamente.*

c) *El estimador de control es:*

$$\frac{\sum_{k=1}^{n1} (Y_k + Y_k^* - l^*)}{n1}.$$

4.0.6. Simulaciones Financieras

Las siguientes simulaciones fueron realizadas en el lenguaje de programación “The R Project for Statistical Computing”.

Simulación de un activo sin riesgo

El siguiente programa genera $S(0), \dots, S(t)$ precios de un activo sin riesgo. De acuerdo al Teorema 2.1, el Algoritmo es de orden $\theta(n)$.

Algoritmo 4.1

```
activo<-function(r,t)
{
S<-c()
S[1]<-1
for(i in 2:t)
{
S[i]<-(exp(r*i))
}
print(S,digits=10)
matplot(S,type="l",xlab = "TIEMPO", ylab = "COSTO DEL ACTIVO SIN RIESGO")
}
```

Simulación de un activo con riesgo

De acuerdo al Teorema 2.1, el Algoritmo es de orden $\theta(n)$.

Algoritmo 4.2

```
activo<-function(n,So,v,m)
{
t<-c()
w<-c()
y<-c()
nt<-0
t[1]=0
w[1]=0
i<-1
while(t[i]<n)
{
t[i+1]<-t[i]+runif(1)
i<-i+1
}
t[i]<-n
nt<-i
for(i in 1:(nt-1))
{
w[i+1]<-w[i]+rnorm(1,0,t[i+1]-t[i])
}
for(i in 1:nt)
{
y[i]<-(So*exp(v*w[i]+(m-(.5*v*v))*t[i]))
}
matplot(t,y,type="l",xlab = "TIEMPO", ylab = "COSTO DEL ACTIVO")
}
```

Simulación MC del costo de una opción Americana y Europea tipo call

De acuerdo al Teorema 2.1, el Algoritmo es de orden $\theta(n^2)$.

Algoritmo 4.3

```
activo<-function(n1,n,So,v,m,k,r)
{
t<-c() ;w<-c();y<-c();nt<-0;t[1]=0
w[1]=0
i<-1
co<-c()
col<-c()
for(j in 1:n1)
{
while(t[i]<n)
{
t[i+1]<-t[i]+runif(1)
i<-i+1
}
t[i]<-n
nt<-i

for(i in 1:(nt-1))
{
w[i+1]<-w[i]+rnorm(1,0,t[i+1]-t[i])
}
for(i in 1:nt)
{
y[i]<-(So*exp(v*w[i]+(m-(.5*v*v))*t[i]))
co[i]<-max(y[i]-k,0)
}
col[j]<-co[i]
}
print(mean(col)*exp(-r*n))
matplot(t,y,type="l",xlab = "TIEMPO", ylab = "COSTO DE LA OPCION")
}
```

Simulación MC del costo de una opción Americana y Europea tipo call al riesgo neutral

De acuerdo al Teorema 2.1, el Algoritmo es de orden $\theta(n^2)$.

Algoritmo 4.4

```
activo<-function(n1,n,So,v,k,r)
{

w<-c()
y<-c()
co<-c()
con<-c()
y[1]<-So
w[1]=0
m<-r

for(j in 1:n1)
{
for(i in 1:(n-1))
{
w[i+1]<-w[i]+rnorm(1,0,1)
}

for(i in 1:n)
{
y[i]<-(So*exp(v*w[i]+(m-(.5*v*v))*i))
co[i]<-max(y[i]-k,0)
}
con[j]<-co[i]
}
print(mean(con)*exp(-r*n))

}
```

Simulación MC del ejercicio de una Opción Americana tipo call

De acuerdo al Teorema 2.1, el Algoritmo es de orden $\theta(n^2)$.

```
Algoritmo 4.5 activo<-function(n1,n,So,v,m,k,r)
{
t<-c();w<-c();y<-c();nt<-0;t[1]=0;w[1]=0;i<-1;co<-c();co1<-c();te<-c()
for(j in 1:n1)
{
while(t[i]<n)
{
t[i+1]<-t[i]+runif(1)
i<-i+1
}
t[i]<-n
nt<-i
for(i in 1:(nt-1))
{
w[i+1]<-w[i]+rnorm(1,0,t[i+1]-t[i])
}
for(i in 1:nt)
{
y[i]<-(So*exp(v*w[i]+(m-(.5*v*v))*t[i]))
if(max(y[i]-k,0)>0)
{
co[j]<-max(y[i]-k,0)*exp(-r*n)
te[j]<-i
break
}
}
i<-1
}
print("Precio de la Opci'on Americana al d'ia de hoy ")
print(mean(co))
print("Tiempo promedio de ejecuci'on de la Opci'on Americana")
print(mean(te))
}
```

4.0.7. Técnicas de Reducción de Varianza

Variables Antitéticas

Se presenta una simulación del costo de una Opción Europea tipo call al riesgo neutral. De acuerdo al Teorema 2.1, el Algoritmo es de orden $\theta(n^2)$.

Algoritmo 4.6

```
activo<-function(n1,n,So,v,k,r)
{

w<-c();
w1<-c();
y<-c();
co<-c();
col<-c()
con<-c()
nor<-c()
y[1]<-So
w[1]=0
w1[1]=0
m<-r
for(j in 1:n1)
{
for(i in 1:(n-1))
{
nor[i]<-rnorm(1,0,1)
w[i+1]<-w[i]+nor[i]
}

for(i in 1:(n-1))
{
w1[i+1]<-w1[i]-nor[i]
}

for(i in 1:n)
{
y[i]<-(So*exp(v*w[i]+(m-(.5*v*v))*i))
col[i]<-max(y[i]-k,0)
}
}
```

```
for(i in 1:n)
{
y[i]<-(So*exp(v*w1[i]+(m-(.5*v*v))*i))
co[i]<-max(y[i]-k,0)
}
con[j]<-(co[i]+co1[i])/2
}

print(mean(con)*exp(-r*n))
}
```

Variables de Control

Es una simulación del costo de una Opción Europea tipo call con riesgo neutro. De acuerdo al Teorema 2.1, el Algoritmo es de orden $\theta(n^2)$.

Algoritmo 4.7

```
activo<-function(n1,n,So,v,k,r)
{

w<-c()
w1<-c()
y<-c()
co<-c()
co1<-c()
con<-c()
co11<-c()
c1<-c()
c2<-c()
nor<-c()
y[1]<-So
w[1]=0
w1[1]=0
est<-c()
m<-r

for(j in 1:n1)
{
for(i in 1:(n-1))
{
nor[i]<-rnorm(1,0,1)
w[i+1]<-w[i]-rnorm(1,0,1)
}

for(i in 1:n)
{
y[i]<-(So*exp(v*w[i]+(m-(.5*v*v))*i))
co[i]<-max(y[i]-k,0)
}
con[j]<-co[i]
```

```

}
est[1]<-mean(con)*exp(-r*n)
print("Costo de la Opci\'on Monte Carlo")
print(est[1])

for(j in 1:n1)
{

for(i in 1:(n-1))
{
nor[i]<-rnorm(1,0,1)
w[i+1]<-w[i]+nor[i]
}
for(i in 1:n)
{
y[i]<-(So*exp(v*w[i]+(m-(.5*v*v))*i))
col[i]<-max(y[i]-k,0)*exp(-r*n)
}
col1[1]<-col[i]

for(i in 1:(n-1))
{
w1[i+1]<-w1[i]-nor[i]
}
for(i in 1:n)
{
y[i]<-(So*exp(v*w1[i]+(m-(.5*v*v))*i))
co[i]<-max(y[i]-k,0)*exp(-r*n)
}
c2[1]<-co[i]

con[j]<-(col1[1]+c2[1]-est[1])
}

print("costo de la opcion aplicando reduccion de varianza")
print(mean(con))

}

```

4.0.8. Tablas y Gráficas

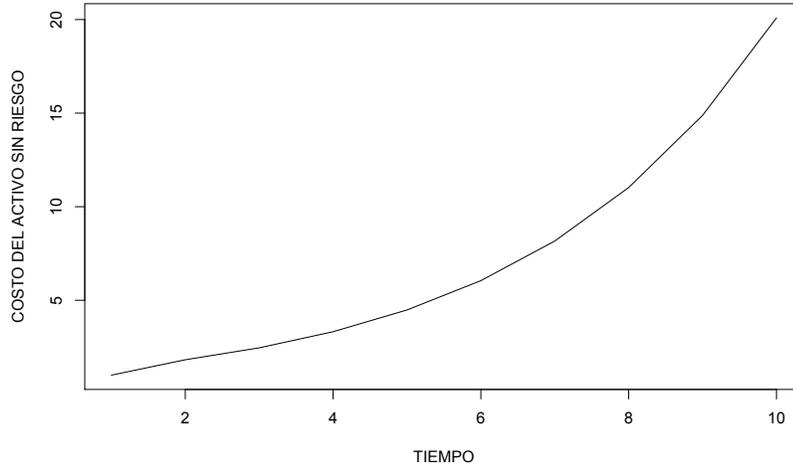


Figura 4.1: Simulación del costo de un activo sin riesgo con $r=0.3$ y $n=10$.

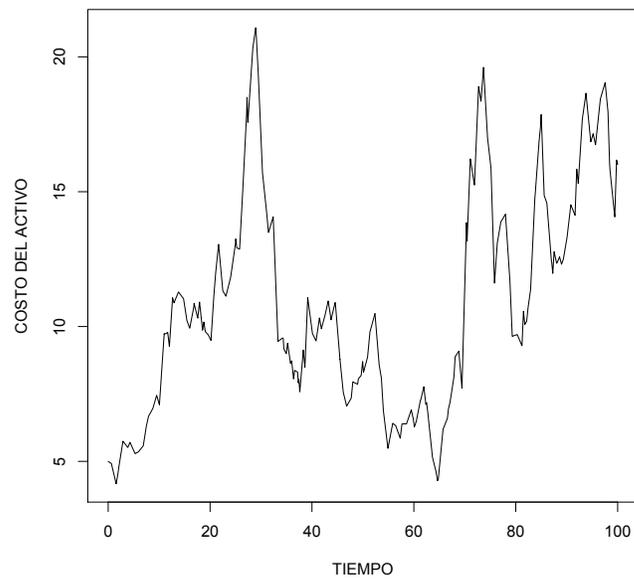


Figura 4.2: Simulación del costo de un activo con riesgo con $n=100$, $S_0=5$, $v=0.2$ y $m=0.008$.

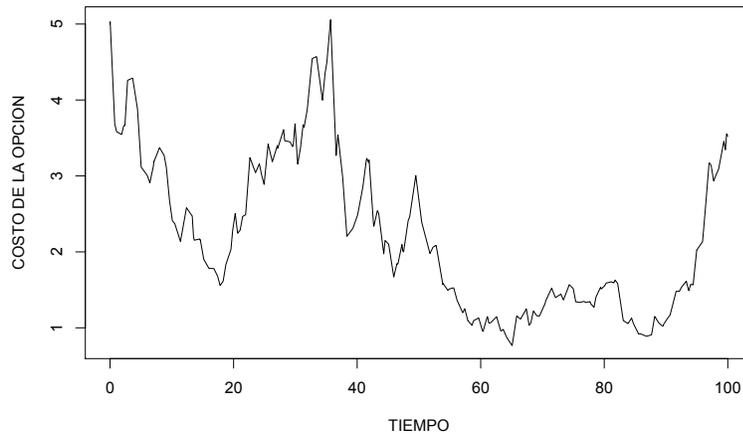


Figura 4.3: Simulación del costo de una Opción Europea tipo call con $n_1=100$, $n=10$, $S_0=5$, $v=0.2$, $m=0.008$, $k=14$ y $r=0.02$.

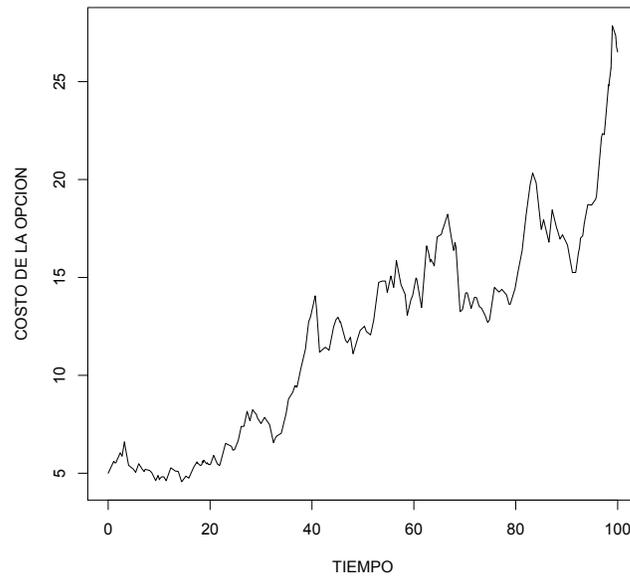


Figura 4.4: Simulación del costo de una Opción Europea tipo call al riesgo neutral con $n_1=100$, $n=100$, $S_0=5$, $v=0.1$, $k=14$ y $r=0.02$.

El objetivo de la siguiente tabla es mostrar el funcionamiento de los siguientes algoritmos:

1. Simulación MC del costo de una opción Americana y Europea tipo call al riesgo neutral.
2. Técnica de reducción de varianza con variables antitéticas.

Se compara los valores obtenidos en los algoritmos con la Fórmula de Black-Scholes para el costo de una Opción Americana tipo call al riesgo neutral.

Monte Carlo Vs. Black-Sholes

No.S	T	So	V	k	R	BS	MC	VMC	EMC	MCA	VMCA	EMCA
100	6	5	0.1	7	0.2	2.891	2.655	12.546	0.236	2.702	0.199	0.189
100	10	12	0.01	15	0.02	0.051	0.036	0.017	0.015	0.038	0.005	0.013
100	2	21	0.02	15	0.03	6.588	6.898	0.200	0.309	6.860	0	0.272
1000	4	8	0.03	10	0.04	0.037	0.021	0.010	0.016	0.023	0.005	0.014
1000	20	3	0.1	5	0.04	0.932	0.671	5.148	0.261	0.703	1.552	0.229

Donde:

1. No.S: Es el número de simulaciones del algoritmo.
2. T es el tiempo.
3. So: Es el valor inicial del precio del activo.
4. V: Es la volatilidad del activo.
5. k: Es el precio de ejercicio.
6. R: Es la tasa libre de riesgo.
7. BS: Es el valor de la Opción Europea tipo call obtenido con la Fórmula de Black-Scholes.
8. MC: Es el valor de la Opción Europea tipo call obtenido con la simulación 1.
9. VMC: Es el valor de la varianza del estimador de Monte Carlo en la simulación 1.
10. EMC: Es el error de Monte Carlo.
11. MCA: Es el valor de la Opción Europea tipo call obtenido con la simulación 2.
12. VMCA: Es el valor de la varianza del estimador de Monte Carlo Antitético.

13. EMCA: Es el error de Monte Carlo Antitético.

En la siguiente tabla se muestra el funcionamiento de los siguientes algoritmos:

- a) Simulación MC del costo de una opción Americana y Europea tipo call al riesgo neutral.
- b) Técnica de reducción de varianza con variables de control.

Se compara los valores obtenidos en los algoritmos con la Fórmula de Black-Scholes para el costo de una Opción americana tipo call al riesgo neutral.

Variable de Control

No.S	T	So	V	k	R	BS	MC	EMC	MCC	EMCC
100	50	10	0.001	21	0.4	10	9.993	0.007	10.005	0.005
100	2	10	0.001	8	0.02	2.313	2.312	0.001	2.314	0.001
100	5	10	0.01	10	0.02	0.951	0.944	0.007	0.952	0.001
100	22	10	0.01	10	0.02	3.559	3.506	0.053	3.588	0.029
1000	22	10	0.01	10	0.02	3.559	3.525	0.034	3.570	0.011

Donde:

- 1. No.S: Es el número de simulaciones del algoritmo.
- 2. T es el tiempo.
- 3. So: Es el valor inicial del precio del activo.
- 4. V: Es la volatilidad del activo.
- 5. k: Es el precio de ejercicio.
- 6. R: Es la tasa libre de riesgo.
- 7. BS: Es el valor de la Opción Europea tipo call obtenido con la Fórmula de Black-Scholes.
- 8. MC: Es el valor de la Opción Europea tipo call obtenido con la simulación a).
- 9. EMC: Es el error de Monte Carlo.
- 10. MCC: Es el valor de la Opción Europea tipo call obtenido con la simulación b).

11. EMCC: Es el error de Monte Carlo de Control.

Observación 4.4 *La simulación (Algoritmo 4.4) del costo de una opción Americana y Europea tipo call al riesgo neutral, se convierte en la simulación del costo de una opción Americana y Europea tipo call con riesgo, sólo quitando la tendencia igual a la tasa de interés libre de riesgo.*

Para aplicar la técnica de reducción de varianza antitética (Algoritmo 4.6) y la técnica de reducción de varianza de control (Algoritmo 4.7) al (Algoritmo 4.4) modificado, se utiliza nuevamente que la tendencia es diferente a la tasa de interés libre de riesgo en los correspondientes algoritmos.

A continuación se ejemplifican los Algoritmos 4.4, 4.6 y 4.7 con tendencia diferente a la tasa de interés.

Costo de una opción Americana y Europea tipo call con riesgo

No.S	T	So	V	M	k	R	MC	VMC	MCA	VMCA	MCC	VMCC
100	10	15	0.001	0.2	20	0.3	4.53	0.11	4.52	0	4.523	0
100	10	15	0.001	0.2	20	0.3	301.94	308.59	300.287	0.001	300.35	0
100	10	50	0.001	0.8	20	0.3	7477	214445	7419	0.989	7423.5	0.006
100	10	33	0.01	0.3	20	0.1	242.46	306.75	236.4	0.303	236.62	0.128
100	10	10	0.4	0.3	20	0.2	80.40	1171496	22	54372	18	1403

Donde:

1. No.S: Es el número de simulaciones del algoritmo.
2. T es el tiempo.
3. So: Es el valor inicial del precio del activo.
4. V: Es la volatilidad del activo.
5. M: La tendencia.
6. k: Es el precio de ejercicio.
7. R: Es la tasa de descuento.
8. MC: Es el valor de la Opción Europea tipo call obtenido con método Monte Carlo.
9. VMC: Varianza Monte Carlo.
10. MCA: Valor de la Opción Europea tipo call obtenido con método Monte Carlo Antitético.

11. VMCA: Varianza Monte Carlo Antitético.
12. MCC: Valor de la Opción Europea tipo call obtenido con método Monte Carlo de Control.
13. VMCC: Varianza Monte Carlo de Control.

Capítulo 5

Conclusiones

Las simulaciones son útiles cuando no podemos obtener valores analíticamente o son difíciles de calcular. Modelar fenómenos ayuda a la solución y prevención de todo tipo de problemas como los incendios forestales, pandemias, escasez de alimentos, etc. En este trabajo nos enfocamos al problema de simular activos financieros, que es una solución al problema de la estabilidad económica.

Muchos investigadores han propuesto modelos para los activos financieros, como Samuelson, Robert C. Merton, Fisher Black y Myron Scholes, con los cuales se intenta modelar los activos financieros. Nosotros siguiendo la línea de Samuelson, consideramos que los precios de los activos siguen un movimiento Browniano Geométrico, con esa hipótesis, simulamos el costo y tiempo de ejecución de una opción americana tipo call en *“The R Project for Statistical Computing”* utilizando el método de Monte Carlo. Cabe hacer mención que hasta el momento no hay una fórmula para el costo de una opción americana como la propuesta por Black y Scholes para el costo de una Opción Europea.

Además, mejoramos el valor del costo de una Opción Americana y Europea tipo call obtenido con el método de Monte Carlo, proponiendo dos algoritmos apegados a las técnicas de reducción de varianza antitéticas y de control. Corroboramos su efectividad al comparar nuestros algoritmos, en el caso particular, donde la tendencia es igual a la tasa de interés, con el valor obtenido con la fórmula de Black-Scholes.

Es importante mencionar que con las técnicas de reducción de varianza se gana en cuanto a precisión pero se pierde en cuanto a tiempo de ejecución. El Algoritmo 4.4 se ejecuta en $3n^2 + 9$ pasos y el Algoritmo 4.6 en $7n^2 + n + 12$ pasos, esto es, que el algoritmo antitético tarda más en ejecutarse que el algoritmo de Monte Carlo. Lo mismo con el algoritmo de control, $11n^2 + 4n + 20$ pasos, y el de Monte Carlo. Podemos concluir que en cuanto a tiempo de ejecución, el algoritmo de reducción de varianza antitético, es mejor que el algoritmo de reducción de varianza de control aquí propuesto.

Bibliografía

- [1] Brown, R., A brief account of microscopical observations made in the months of June, July and August, 1827, on the particles contained in the pollen of plants; and on the general existence of active molecules in organic and inorganic bodies, *Phil. Mag.* 4, 161-173, 1828.
- [2] Dirk P. Kroese, Thomas Taimre, Zdravko I. Botev, *Handbook of Monte Carlo Methods*, Wiley, March 15, 2011.
- [3] Einstein, A., Über die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen, *Ann. Phys.* 17, 549, 1905.
- [4] Einstein, A., *Investigations on the Theory of Brownian Movement*, New York: Dover, 1956.
- [5] Fischer Black and Myron Scholes, The Pricing of Options and Corporate Liabilities, *The Journal of Political Economy*, Vol. 81, No. 3, 1973.
- [6] Francisco Venegas Martínez, *Riesgos financieros y económicos: Productos derivados y decisiones económicas bajo incertidumbre*, CENGAGE Learning, 2008.
- [7] H. Niederreiter, Quasi-Monte Carlo methods and pseudorandom numbers, *Bulletin of the American Mathematical Society*, 84(6):957, 1978.
- [8] I.M. Sóbol, *Método de Monte-Carlo*, Mir Moscú, segunda edición, 1983.
- [9] John C. Hull, *Introducción a los Mercados de Futuros y Opciones*, Pearson, 2009.
- [10] M. Chung, M. Buro, and J. Schaeffer, Monte Carlo planning in RTS games, In *CIG 2005*, Colchester, UK, 2005.
- [11] Ngai Hang Chan, Hoi-Ying Wong, *Simulation Techniques in Financial Risk Management*, John Wiley-Sons, 2006.
- [12] N. Metropolis, The Beginning of the Monte Carlo Method, *Los Alamos Science*, No. 15, p. 125.1987.

- [13] Paul Glasserman, Monte Carlo Methods in Financial Engineering (Stochastic Modelling and Applied Probability), Springer, New York, 2004.
- [14] Paul G. Hoel, Sidney C. Port, Charles J. Stone, Introduction to stochastic processes, Houghton Mifflin Company, 1972.
- [15] Richard Johnsonbaugh, Matemáticas Discretas, Pearson Educación, sexta edición, 2005.
- [16] Samuelson, Paul A., Rational Theory of Warrant Pricing, Industrial Management Review, p.13, 1965.
- [17] Siddhartha Chib and Edward Greenberg, Understanding the Metropolis-Hastings Algorithm, The American Statistician, Vol.49, No. 4, 1995.
- [18] W. K. Hastings, Monte Carlo Sampling Methods Using Markov Chains and Their Applications, Biometrika, Vol. 57, No. 1, pp. 97-109, 1970.