Universitat de Barcelona

Facultat de Matemàtiques i Informàtica

GRAU DE MATEMÀTIQUES

Treball final de grau

Valoració d'opcions financeres amb el mètode de Monte Carlo

Autor: Oriol Tubella Domingo

Director: Dr. Josep Vives i Santa-Eulàlia

Realitzat a: Departament de Matemtiques i Informtica

Barcelona, 24 de gener de 2022

Abstract

We introduce partially the Black-Scholes Model and the pricing options scheme under this model. Then we introduce the Monte Carlo method and the theory that ensures it works and we use it for option pricing. Finally we apply some variance reduction techniques and compare the results.

Resum

S'introdueix parcialment el model de Black-Scholes així com la valoració d'opcions financeres sota aquest marc teòric. A continuació s'explica el mètode de Monte Carlo i com usar-lo com a mètode numèric per aproximar el preu d'aquests derivats financers. Finalment, s'implementen diverses tècniques de reducció de la variància per millorar el mètode i es comparen els resultats obtinguts amb aquestes tècniques amb els resultats obtinguts amb el mètode de Monte Carlo estàndard.

2010 Mathematics Subject Classification: 65C05, 91G20, 91G60

Agraïments

Vull agrair al meu tutor, el Dr. Josep Vives i Santa-Eulàlia, pel seu temps i la seva ajuda així com per haver-me guiat en aquest treball.
I a la meva família i amics pel seu suport.

Índex

1 Introducció **1**

2 Opcions financeres **2**

2.1 Opcions asiàtiques 2

2.2 Opcions barrera 3

2.3 Opcions look-back 4

2.4 Valoració d'opcions financeres 4

3 Processos estocàstics **5**

3.1 Martingales 7

3.2 Moviment brownià geomètric 7

4 El model de Black-Scholes **9**

4.1 Trobar la probabilitat neutral al risc 9

4.1.1 Estratègies admissibles i replicables 11

4.2 Valoració d'opcions europees 13

5 El mètode de Monte Carlo **17**

6 Valoració d'opcions amb el metode de Monte Carlo **20**

6.1 Generació del moviment brownià 20

6.2 Generació de mostres aleatòries 20

6.2.1 Mètode de la transformada inversa 20

6.2.2 Mètode de Box-Muller 21

6.3 El pont Brownià 22

6.4 Valoració d'opcions europees 22

6.5 Valoració d'opcions que depenen de la trajectòria 23

7 Tècniques per la reducció de la variància **24**

7.1 Varietats de control 24

7.2 Variacions antitètques 25

7.3 Mostra estratificada 27

7.3.1 Estratificació terminal 29

7.4 Monte Carlo condicionat 30

8 Part pràctica **31**

8.1 Valoració d'opcions europees 31

8.2 Valoració de diferents opcions barrera 33
8.3 Opcions asiàtiques 36

9 Conclusions **40**

10 Referències **41**

11 Annexos **42**
11.1 Apèndix 42
11.2 Codi 44

1 Introducció

En les finances modernes, enmarcades dins del model de Black-Scholes, molts mètodes numèrics són usats per a diferents aplicacions, com poden ser l'anàlisi de riscos, la valoració d'opcions financeres o la teoria moderna de carteres entre d'altres. Un d'aquests mètodes és el mètode Monte Carlo, que enmarcat dins de la llei forta dels grans nombres, ha resultat molt útil en moltes d'aquestes aplicacions per la seva simplicitat. Una de les possibles aplicacions d'aquest mètode és la de la valoració d'opcions financeres. El primer en usar aquesta tècnica va ser Phelim P. Boyle l'any 1977 (veure [10]). Des d'aleshores, però, la complexitat de les finances modernes ha anat augmentant. La necessitat no només d'un mètode eficaç, sinó també eficient, ha portat a desenvolupar diferents tècniques per millorar-lo, conegudes com tècniques de reducció de la variància. Autors importants en aquest àmbit són, per exemple, Paul Glasserman, Mark Broadie o el mateix Phelim Boyle. Seguint precisament un dels llibres de referència de l'àmbit esmentat, com és [3], en aquest treball s'ha desenvolupat la valoració de diferents opcions financeres usant el mètode de Monte Carlo, i s'han implementat diferents tècniques de reducció de la variància.

Estructura de la Memòria

Primerament, s'introdueixen d'una manera bàsica les opcions financeres.
A continuació, s'introdueixen els conceptes més importants del càlcul estocàstic aplicat a les finances. La tercera secció d'aquesta memòria introdueix el model de Black-Scholes que serveix com a marc teòric per modelitzar els preus d'actius financers. S'acaba el capítol derivant les fórmules de Black-Scholes de les opcions de compra i venda europees.
En el cinquè punt de la memòria s'introdueix el mètode de Monte Carlo, un mètode numèric basat en la llei forta dels grans nombres. S'explica el marc teòric que justifica el mètode, així com els seus punts forts i els seus punts febles.
En el punt 6 de la memòria es donen eines matemàtiques per poder implementar el mètode anteriorment explicat. En particular, es defineixen mètodes per generar mostres aleatòries de variables amb distribució normal i s'explica com usar-los sota el model de preus de Black-Scholes. Per acabar, es donen els algoritmes per valorar opcions financeres.
En l'apartat 7, s'introdueixen tècniques teòriques per millorar tant l'eficàcia com l'eficiència del mètode Monte Carlo. També s'explica en què es basa la seva millora i com s'usaran en la part pràctica.
Finalment, en la penúltima secció es donen els resultats de les simulacions realitzades. En particular s'han valorat opcions europees, opcions barrera i opcions asiàtiques.
S'acaba el treball amb les conclusions, on es resumeixen les fortaleses i les febleses dels mètodes usats.

2 Opcions financeres

Una opció financera és un contracte que dona dret a comprar un actiu financer subjacent, com pot ser una acció financera, bons o altres a un preu determinat en una data prèviament establerta (anomenada data de venciment), o en qualsevol moment anterior a aquesta data. Per tenir aquest dret cal pagar una quantitat anomenada prima.
L'opció financera més simple és l'anomenada opció europea de compra (European plain Vanilla call). Aquesta opció es defineix com el dret, però no l'obligació, a comprar una unitat d'un cert producte a un cert preu K en una certa data de venciment T. Si en aquesta data T el preu del producte al mercat, que d'ara en endavant denotarem per S_T, és superior a K (preu d'exercici de l'opció), podem exercir l'opció de compra i comprar el producte a preu K. Vendre'l immediatament al mercat ens donarà doncs $S_T - K$ de benefici. Si en canvi $S_T \leq K$ en la data de venciment T, no té sentit exercir l'opció de compra; n'hi ha prou amb comprar el producte directament al mercat. Així doncs, podem reinterpretar l'opció europea de compra com el dret a rebre en la data T el perfil de beneficis $G = (S_T - K)^+ = max(S_T - K, 0)$.
De la mateixa manera podem definir una opció europea de venda (European plain vanilla put), que es pot interpretar com el dret a vendre una unitat d'una producte subjacent a preu K en la data de venciment T. El perfil de beneficis d'aquests put ve definit per $(K - S_T)^+$.
Notem que el perfil de beneficis d'un call, $(S_T - K)^+$, és potencialment infinit, mentre que el del put $(K - S_T)^+$ està afitat per K. Això últim és matemàticament important, com es veurà més endavant. D'opcions financeres n'hi ha de molts tipus i totes venen determinades pel seu perfil de beneficis. A continuació presentem algunes que més endavant sortiran.

2.1 Opcions asiàtiques

La diferència entre les opcions asiàtiques i les opcions europees esmentades anteriorment rau en el fet que el perfil de beneficis ve determinat per la mitjana dels preus de l'actiu subjacent durant el període abans de la data de venciment T. Si denotem per T la data de venciment de l'opció asiàtica, K el preu d'exercici d'aquesta, $A(0,T)$ la mitjana de l'actiu subjacent durant el període $[0,T]$ i $S(t)$ el preu de l'actiu a temps t, a grans trets hi ha dos tipus d'opcions asiàtiques:

1. Amb perfil de beneficis $G = (A(0,T) - K)^+$.
2. Amb perfil de beneficis $G = (S(T) - cA(0,T))^+$ on c és una constant.

Quan parlem de temps continu entenem per mitjana aritmètica

$$A(0,T) = \frac{1}{T}\int_0^T S(t)dt.$$

Si en canvi el temps està discretitzat amb temps $0 = t_0, t_1, ..., t_n = T, t_i = i\frac{T}{n}$, aleshores

$$A(0,T) = \frac{1}{n}\sum_{i=1}^{n} S(t_i).$$

Finalment hi ha opcions asiàtiques amb mitjana geomètrica. En el cas discret això és

$$A(0,T) = (\prod_{i=1}^{n} S(t_i))^{\frac{1}{n}}$$

i en el cas continu

$$A(0,T) = exp\left(\frac{1}{T}\int_0^T log(S(t))dt\right).$$

Definició 2.1. *Quan el perfil de beneficis d'una opció amb data de venciment T depèn no només del valor de l'actiu a preu T sinó també d'una certa evolució del valor de l'actiu en el període [0,T] diem que l'opció depèn de la trajectòria.*

2.2 Opcions barrera

Les opcions barrera són opcions molt semblants a les opcions europees però en les quals el perfil de beneficis queda activat (o extingit) si el preu de l'actiu assoleix un cert preu barrera durant el període fins la data de venciment de l'opció. Dit d'una altra manera, mentre que en les opcions europees ens és totalment igual l'evolució del preu de l'actiu en el període [0,T] (notem que en el perfil de beneficis només importa el preu de l'actiu a temps T), en les opcions barrera el perfil de beneficis queda determinat per una certa evolució del preu. En particular, són un altre tipus d'opció que depenen de la trajectòria. La següent taula resumeix diferents tipus d'opcions barrera de venda i de compra.

Opcions Barrera		
Tipus d'opció	Nom (comportament)	perfil de beneficis
Barrier Call	Up-and-Out (knock-out)	$(S_T - K)^+ \mathbb{1}_{\{\max_{0\le t\le T} S_t \le L\}}$
	Down-and-Out (knock-out)	$(S_T - K)^+ \mathbb{1}_{\{\min_{0\le t\le T} S_t \ge L\}}$
	Up-and-In (knock-in)	$(S_T - K)^+ \mathbb{1}_{\{\max_{0\le t\le T} S_t \ge L\}}$
	Down-and-In (knock-in)	$(S_T - K)^+ \mathbb{1}_{\{\min_{0\le t\le T} S_t \le L\}}$
Barrier Put	Up-and-Out (knock-out)	$(K - S_T)^+ \mathbb{1}_{\{\max_{0\le t\le T} S_t \le L\}}$
	Down-and-Out (knock-out)	$(K - S_T)^+ \mathbb{1}_{\{\min_{0\le t\le T} S_t \ge L\}}$
	Up-and-In (knock-in)	$(K - S_T)^+ \mathbb{1}_{\{\max_{0\le t\le T} S_t \ge L\}}$
	Down-and-In (knock-in)	$(K - S_T)^+ \mathbb{1}_{\{\min_{0\le t\le T} S_t \le L\}}$

Taula 1: Diferents tipus d'opcions Barrera. K és el preu d'exercici de l'opció, L el valor barrera, i S_T el preu de l'actiu subjacent en la data de venciment T.

2.3 Opcions look-back

Són un altre tipus d'opció que depenen de la trajectòria. En les opcions look-back el perfil de beneficis depèn del valor màxim i mínim que assoleixi el preu de l'actiu subjacent durant el període fins la data de venciment. Podem diferenciar dos tipus d'opcions look-back: Opcions look-back amb preu d'exercici fix i opcions look-back amb preu d'exercici variable. Com abans, sigui T la data de venciment de l'opció i $S(t)$ el preu de l'actiu a preu t. Tenim :

1- Opcions de compra i venda look-back amb preu d'exercici variable: El perfil de beneficis ve donat per

$$LCall_{float} = \max(S_T - S_{min}, 0) = S_T - S_{min}$$

$$LPut_{float} = \max(S_{max} - S_T, 0) = S_{max} - S_T$$

Observem que l'opció de compra/venda gairebé sempre serà exercida pel propietari ja que el seu perfil de beneficis gairebé sempre dona benefici (a la pràctica això comporta un preu més car per aquests derivats financers).

2- Opcions de compra i venda look-back amb preu d'exercici fix. El perfil de beneficis és

$$LCall_{fix} = (S_{max} - K)^+ = \max(S_{max} - K, 0)$$
$$LPut_{fix} = (K - S_{min})^+ = \max(K - S_{min}, 0)$$

2.4 Valoració d'opcions financeres

El problema de la valoració de derivats financers és el problema de determinar quin és el preu avui, a temps $t = 0$, del dret a guanyar el perfil de beneficis corresponent (en funció del tipus d'opció) en la data de venciment T.
Un problema estrictament relacionat amb la valoració de les opcions financeres és el problema de la cobertura d'aquestes, és a dir, què ha de fer el venedor de l'opció financera amb els diners que rep del comprador de l'opció per tal de poder respondre a la seva obligació de pagar el que correspongui en la data T. En aquest treball però ens centrarem només en el problema de la valoració.

3 Processos estocàstics

En tot aquest capítol suposem que estem en el marc d'un cert espai de probabilitat $(\Omega, \mathcal{F}, \mathbb{P})$. Denotarem indistintivament per $\mathbb{T} := \{0, 1, ...T\}$ quan parlem a temps discret o bé $\mathbb{T} = [0, T]$ quan parlem de temps continu. Referències adequades per aquest apartat són [1], [4], [6].

Definició 3.1. *Un procés estocàstic X és una aplicació*

$$\begin{aligned} X : \Omega \times \mathbb{T} &\longrightarrow \mathbb{R} \\ (\omega, t) &\longrightarrow X_t(\omega) \end{aligned}$$

mesurable, és a dir, tal que

$$X^{-1}(B) \in \mathcal{F} \otimes \mathcal{B}(\mathbb{T}) \tag{3.1}$$

per a tot $B \in \mathcal{B}(\mathbb{R})$.

Definició 3.2. *Una filtració associada a un espai de probabilitats $(\Omega, \mathcal{F}, \mathbb{P})$ és una successió de $\sigma-$àlgebres $\mathbb{F} := \{\mathcal{F}_n, n \in \mathbb{T}\}$ tals que*

1. $\mathcal{F}_n \subseteq \mathcal{F}, \forall n \in \mathbb{T}$,
2. $\mathcal{F}_{n-1} \subseteq \mathcal{F}_n, \forall n \geq 1$.

Si denotem per $\mathcal{F}_t^X = \sigma\{X_s : s \leq t\}$ per a $t \in \mathbb{T}$, la σ-àlgebra donada per un procés estocàstic X, la filtració $\mathbb{F}^X = \{\mathcal{F}_t^X, t \in \mathbb{T}\}$ és la filtració generada pel procés X i s'anomena filtració natural.

Definició 3.3. *Un espai de probabilitat amb una filtració associada $\mathbb{F}$ s'anomena espai de probabilitat filtrat. Ho denotarem per $(\Omega, \mathcal{F}, \mathbb{F}, \mathbb{P})$.*

Una filtració representa la informació disponible en un cert instant de temps t.

Definició 3.4. *Es diu que un procés estocàstic X definit sobre un espai de probabilitats filtrat és adaptat si per $n \in \mathbb{T}$, X_n és una variable aleatòria $\mathcal{F}_n-$mesurable.*

Definició 3.5. *Un procés estocàstic $\{X_t, t \in \mathbb{T}\}$ on T és un subconjunt de $\mathbb{R}$ es diu que té increments independents si per tot $t_1 < t_2 < ... < t_m$ les variables aleatòries $X_{t_2} - X_{t_1}, ..., X_m - X_{t_{m-1}}$ són independents.*

Definició 3.6. *Un moviment Brownià estàndard és un procés estocàstic $\{W_t; t \geq 0\}$ definit en un espai de probabilitat $(\Omega, \mathcal{F}, \mathbb{P})$ que satisfà:*
(i) $W_0 = 0$ quasi segurament.
(ii) La funció $t \longrightarrow W_t := W(t, \omega)$, és, amb probilitat 1, una funció contínua.[2]
(iii) Té increments independents.
(iv) Per a tot $0 \leq s \leq t, W_t - W_s \sim N(0, t - s)$.

[2]Segons convingui s'usarà la notació $W(t)$ com a sinònim de W_t

Clarament de (i) i (iv) obtenim que $W_t \sim \mathcal{N}(0,t)$ i de la definició també es desprèn que els increments són estacionaris, és a dir, que $W_t - W_s$ té la mateixa distribució que W_{t-s}. [3]
Per a constants μ i $\sigma > 0$ diem que un procés X_t és un moviment Brownià amb desplaçament μ i difusió σ^2 (ho denotarem per $X \sim BM(\mu, \sigma^2)$) si

$$\frac{X_t - \mu t}{\sigma}$$

és un moviment Brownià estàndard. Per tant, podem construir un moviment brownià X a partir d'una moviment brownià estàndard W posant $X_t = \mu t + \sigma W_t$. Notem que $X_t \sim N(\mu t, \sigma^2 t)$.

Proposició 3.7. *EL vector $W = (W_{t_1}, \ldots.., W_{t_n})$, amb $0 < t_1 < t_2 < \ldots. < t_n$ segueix una distribució normal multivariable.*

Demostració. Si denotem per f_t la llei de W_t aleshores $f_t(x) = \frac{1}{\sqrt{2\pi t}} e^{-\frac{x^2}{2t}}$ i per tant $W_{t_i} - W_{t_j}$ té per funció de densitat $f_{t_i - t_j}$. A més a més, com els increments són independents, tenim que la densitat del vector $W' = (W_{t_1}, W_{t_2} - W_{t_1}, \ldots., W_{t_n} - W_{t_{n-1}})$ és $g(y_1, y_2, \ldots.., y_n) = f_{t_1}(y_1) f_{t_2 - t_1}(y_2) f_{t_n - t_{n-1}}(y_n)$. Considerem el canvi de variables

$$\begin{aligned} h \colon \mathbb{R}^n &\to \mathbb{R}^n \\ (y_1, \ldots., y_n) &\to (y_1, y_1 + y_2, \ldots., y_1 + \ldots. + y_n) \end{aligned}$$

Aleshores $h(W') = W$, $h^{-1}(x_1, \ldots.., x_n) = (x_1, x_2 - x_1, \ldots..., x_n - x_{n-1})$ i $det(J_{h^{-1}}(z)) = 1$, i per tant, si denotem per $f_{t_1,\ldots,t_n}$ la densitat conjunta de W, tenim que $f_{t_1,\ldots,t_n} = f_{t_1}(x_1) f_{t_2 - t_1}(x_2 - x_1) f_{t_3 - t_2}(x_3 - x_2) \ldots.... f_{t_n - t_{n-1}}(x_n - x_{n-1})$ □

Proposició 3.8. *Sigui $W = \{W_t; t \geq 0\}$ un moviment brownià. El procés $W^u(t) = W(u+t) - W(u)$ és també un moviment brownià.*

Demostració. Clarament $W^u(0) = 0$. A més a més les propietats (ii) i (iii) per $W^u(t)$ es segueixen de la cancel·lació del terme $W(u)$ i del fet que W és un moviment brownià. Finalment la funció $t \longrightarrow W(u+t) - W(u)$ és una funció contínua per ser la composició de funcions contínues (quasi segurament). □

Denotem per $\mathcal{F}_t^W = \sigma\{W(u); 0 \leq u \leq t\}$.

Proposició 3.9. *Per $0 < s \leq t < u, W_u - W_t$ és independent de la σ-àlgebra $\mathcal{F}_s^W$. En particular, $W_u - W_t$ és independent de $\mathcal{F}_t^W$.*

Demostració. La σ-àlgebra $\mathcal{F}_t^W = \sigma\{W(r) : r \leq s\}$ està generada per les variables aleatòries $W(r)$ amb $r \leq s$. Notem que per qualsevol $0 < s \leq t < u$ els increments $W(u) - W(t)$ són independents de $W(r) - W(0) = W(r)$. Per tant la variable aleatòria $W(u) - W(t)$ és independent de $\mathcal{F}_s^W$ per $s \leq t$. □

[3]No es pretén aquí desenvolupar la teoria del moviment brownià. Probablement, la primera pregunta que un es podria fer és si realment existeix un procés amb les propietats descrites a la Definició 3.6. La resposta és que sí. Tot i això, la definició donada aquí del moviment brownià és una definició estàndard que serveix pels propòsits d'aquest projecte. Per exemple essent més rigorossos, en (iv) seria suficient demanar que les variables $W_t - W_s$ tinguessin esperança 0 i variància $t - s$, és a dir, que el que fet que la distribució de $W_t - W_s$ sigui una distribució normal es segueix de la continuïtat de les trajectòries (ii) i de la indepedència dels increments (iii). Un desenvolupament profund del tema es pot trobar a [4] i [10].

Anteriorment hem provat que el procés $W^u(t) = W(t+u) - W(u)$ és també un moviment brownià. En particular, la proposició anterior mostra que $W^u(t)$ és independent de $\mathcal{F}_u^W$ i per tant també son independents els seus increments $W^u(t) - W^u(s) = W(t+u) - W(s+u)$ per $0 \leq s \leq t$

3.1 Martingales

El concepte de martingala és crucial per la teoria de processos estocàstics i la teoria financera. Donem aquí els resultats més importants per al projecte.

Definició 3.10. *Un procés estocàstic $X = \{X_t; t \in \mathbb{T}\}$ és una martingala respecte una filtració $\mathbb{F}$ si*

1. *X és un procés adaptat a $\mathbb{F}$,*
2. *X_t és una variable aleatòria integrable per a tot t, és a dir , $\mathbb{E}(|X_t|) < \infty$ per a tot $t \in \mathbb{T}$,*
3. *$\mathbb{E}(X_t|\mathcal{F}_s) = X_s$ per qualsevol $s \leq t$.*

Proposició 3.11. *El moviment brownià és una martingala respecte $\mathbb{F}^W$.*

Demostració. És clar que per a tot $t \geq 0$ les variables W_t són $\mathcal{F}_t^W$-mesurables i, en particular, també hem vist que són integrables (doncs tenen esperança finita). Falta doncs comprovar la propietat de martingala, és a dir que per $0 \leq s < t$ es compleix que $E(W_t|\mathcal{F}_s^W) = W_s$. Tenim que:

$$E(W_t|\mathcal{F}_s^W) = E(W_t + W_s - W_s|\mathcal{F}_s^W) = E(W_t - W_s|\mathcal{F}_s^W) + E(W_s|\mathcal{F}_s^W) = E(W_t - W_s) + W_s = W_s.$$

on hem usat que W_s és $\mathcal{F}_s^W$-mesurable, que $W_t - W_s$ és independent de $\mathcal{F}_s^W$ i té esperança zero i que $E(W_t - W_s) = 0$. □

Proposició 3.12. *El procés $exp(\sigma W_t - \frac{1}{2}\sigma^2 t)$ és una martingala respecte $\mathbb{F}^W$.*

Demostració. Recordem que si $Y \sim \mathcal{N}(0, \sigma^2)$ aleshores $\mathbb{E}(e^Y) = e^{\frac{1}{2}\sigma^2}$. Així doncs tenim:

$$\begin{aligned}
\mathbb{E}(exp(\sigma W_t - \frac{1}{2}\sigma^2 t)|\mathcal{F}_s^W) &= \mathbb{E}(exp(\sigma W_t - \frac{1}{2}\sigma^2 t + \sigma W_s - \sigma W_s)|\mathcal{F}_s^W) \\
&= exp(\sigma W_s - \frac{1}{2}\sigma^2 t)\mathbb{E}(exp(\sigma(W_t - W_s))|\mathcal{F}_s^W) \\
&= exp(\sigma W_s - \frac{1}{2}\sigma^2 t)\mathbb{E}(exp(\sigma(W_t - W_s))) \\
&= exp(\sigma W_s - \frac{1}{2}\sigma^2 s)
\end{aligned} \tag{3.2}$$

on a la segona igualtat hem usat que W_s és $\mathcal{F}_s^W$-mesurable. □

3.2 Moviment brownià geomètric

Un moviment brownià pot pendre valors negatius, i és per això que no té gaire sentit usar-lo per modelitzar el preu dels actius financers, doncs aquests només poden pendre valors positius. Podem introduir però una variació no negativa del moviment Brownià anomenat moviment brownià geomètric definit per $S(t) = S_0 e^{X_t}$ on $X_t = \sigma W_t + at$ és un moviment brownià i $S(0) = S_0 > 0$. Observem que al prendre el logaritme al moviment geomètric brownià, retornem al moviment brownià: $X_t = ln(S(t)/S_0) = ln(S(t)) - ln(S_0)$ i per tant $ln(S(t)) = ln(S_0) + X_t$ segueix una distribució normal amb mitjana $at + ln(S_0)$ i variància $\sigma^2 t$. Es diu que S(t) segueix una distribució lognormal.

Distribució Lognormal

Si $Y \sim \mathcal{N}(\mu, \sigma^2)$ aleshores $X = e^Y$ és una variable aleatòria no negativa que segueix una distribució lognormal, en el sentit que $ln(X)$ ens dona una variable aleatòria amb una distribució normal. En particular tenim que X té densitat

$$f(x; \mu, \sigma) = \begin{cases} \frac{1}{x\sigma\sqrt{2\pi}} e^{\frac{-(ln(x)-\mu)^2}{2\sigma^2}} & x \geq 0 \\ 0 & x < 0 \end{cases}$$

doncs $F(x) = P(X \leq x) = P(Y \leq ln(x)) = \Phi(\frac{ln(x)-\mu}{\sigma})$ on $\Phi(x)$ denota la funció de distribució de $\mathcal{N}(0,1)$. Finalment notem que, si denotem amb $Z \sim \mathcal{N}(0,1)$ i per $\phi(z)$ la seva funció de densitat, aleshores

$$\begin{aligned} E(X) = E(e^{\mu+\sigma Z}) &= \int_{-\infty}^{+\infty} e^{\mu+\sigma z} \phi(z) dz = \frac{1}{\sqrt{2\pi}} \int_{\infty}^{+\infty} e^{\mu+\sigma z} e^{\frac{-z^2}{2}} dz \\ &= \frac{1}{\sqrt{2\pi}} e^{\mu} \int_{\infty}^{+\infty} e^{\sigma z} e^{\frac{-z^2}{2}} dz. \end{aligned}$$

Completant el quadrat, $(\sigma z - \frac{z^2}{2}) = \frac{1}{2}(z^2 - 2\sigma z + \sigma^2) - \frac{1}{2}\sigma^2 = \frac{1}{2}(z-\sigma)^2 - \frac{1}{2}\sigma^2$. Per tant la integral resultant és

$$\frac{1}{\sqrt{2\pi}} e^{\mu + \frac{\sigma^2}{2}} \int_{\infty}^{+\infty} e^{-\frac{(z-\sigma)^2}{2}} dz = e^{\mu + \frac{\sigma^2}{2}}.$$

En particular, el resultat és directe si considerem la funció generatriu de moments doncs, $M_Y(s) = E(e^{sY})$ i llavors $E(e^Y) = M_Y(1)$. Usant això, i que $M_Y(s) = e^{s\mu + \frac{1}{2}\sigma^2 s^2}$obtenim directament que

$$E(X^2) = e^{2\mu + 2\sigma^2} \quad Var(X) = e^{2\mu+2\sigma^2}(e^{\sigma^2} - 1).$$

Igualment podem calcular els moments del moviment brownià geomètric. Com $X_t = at + \sigma W_t$ tenim

$$M_{X_t}(s) = E(e^{sX_t}) = e^{ats + \frac{1}{2}\sigma^2 s^2 t}, \quad -\infty < s < \infty$$

En particular tenim

$$E(S(t)) = S_0 E(e^{X_t}) = S_0 M_{X_t}(1) = S_0 e^{(a+\frac{\sigma^2}{2})t} \tag{3.3}$$

$$E(S(t)^2) = E(S_0^2 e^2 X_t) = S_0^2 E(e^2 X_t) = S_0^2 M_{X_t}(2) = S_0^2 e^{2at+2\sigma^2 ts} \tag{3.4}$$

$$Var(S(t)) = E(S(t)^2) - E(S(t))^2 \tag{3.5}$$

4 El model de Black-Scholes

Considerarem dos tipus d'actius financers:
Un actiu financer sense risc (com un compte corrent bancari) i que descrivim amb una funció determinista $A(t)$ tal que

$$dA(t) = rA(t)dt, \tag{4.1}$$

amb A(0)=1 i $r > 0$ és un tipus d'interès fixat. L'actiu ve determinat per una equació diferencial ordinària ($A'(t) = rA(t)$) i té una única solució $A(t) = e^{rt}$.
Un actiu financer amb risc, el preu del qual ve donat per

$$S(t) = S_0 exp\{(\mu - \frac{\sigma^2}{2})t + \sigma W(t)\}, \tag{4.2}$$

on μ s'anomena desplaçament i $\sigma > 0$ s'anomena la volatilitat de l'actiu amb risc. En particular notem que el preu de l'actiu amb risc segueix un moviment brownià geomètric, amb $a = (\mu - \frac{1}{2}\sigma^2)$.
Usarem la notació $S \sim GBM(\mu, \sigma^2)$ per dir que S és un procés determinat per l'equació (4.2)[4]
En particular notem que si $S \sim GBM(\mu, \sigma^2)$ aleshores $(S(t)/S(0)) \sim LN([\mu - \frac{1}{2}\sigma^2]t, \sigma^2 t)$ i per tant

$$E[S(t)] = e^{\mu t}S(0) \quad Var[S(t)] = e^{2\mu t}S(0)^2(e^{\sigma^2 t} - 1).$$

Més generalment si $u < t$ tenim que

$$S(t) = S(u)e^{(\mu - \frac{\sigma^2}{2})(t-u) + \sigma(W_t - W_u)}. \tag{4.3}$$

Notem que la filtració generada pel procés $S = \{S(t); t \in \mathbb{T}\}$ que ve donada per $\mathbb{F}^S = \{\mathcal{F}_t^S; t \in \mathbb{T}\}$, amb $\mathcal{F}_t^S = \sigma\{S(u) : u \leq t\}$, és la mateixa que la definida anteriorment $\mathbb{F}^W$, doncs de (4.2) és despren que W és la única font d'aleatorietat en S: si sabem quant val $S(t)$ sabem quant val $W(t)$ i viceversa.

4.1 Trobar la probabilitat neutral al risc

Definició 4.1. *Definim el preu descomptat de l'actiu amb risc com*

$$\tilde{S}_t = exp\{-rt\}S(t) = S(0)exp\{(\mu - r)t - \frac{\sigma^2}{2}t + \sigma W_t\}.$$

Proposició 4.2. *Per* $u < t, \mathbb{E}(\tilde{S}_t | \mathcal{F}_u^W) = \tilde{S}_u exp\{(\mu - r)t\}$

[4]L'elecció de (4.2) com a equació per modelitzar els preus de l'actiu amb risc és ni molt menys evident ni trivial però la discussió queda fora de l'abast d'aquest projecte. Està directament relacionada amb equacions diferencials estocàstiques i la solució d'aquestes. El lector interessat pot trobar un desevolupament ampli del tema referir-se [1], [2], [3].

Demostració. Partint de (4.3) tenim

$$\begin{aligned}
\mathbb{E}(\tilde{S}_t|\mathcal{F}_u^W) &= \mathbb{E}(S(u)exp\{(\mu - \frac{\sigma^2}{2})(t-u) - rt + \sigma(W_t - W_u)\}|\mathcal{F}_u^W) \\
&= \tilde{S}_u exp\{(\mu - \frac{\sigma^2}{2})(t-u)\}\mathbb{E}(exp^{\sigma(W_t - W_u)}|\mathcal{F}_u^W) \\
&= \tilde{S}_u exp\{(\mu - \frac{\sigma^2}{2})(t-u)\}E(exp^{\sigma(W_t - W_u)}) \\
&= \tilde{S}_u exp\{(\mu - \frac{\sigma^2}{2})(t-u)\}exp\{\frac{1}{2}\sigma^2(t-u)\} \\
&= \tilde{S}_u exp\{(\mu - r)t\}.
\end{aligned}$$

□

Com més endavant veurem, és important que sota una certa probabilitat, els preus descomptats siguin una martingala. El càlcul anterior ens diu que l'esperança és constant si i només si $\mu = r$ (on r és la taxa d'interés de l'actiu sense risc). En general però això no és així. Ens interessa trobar una nova mesura de probabilitat, que anomenarem Q, amb els mateixos conjunts de probabilitat 0 que P (i per tant Q és equivalent a P), sota la qual els preus descomptats siguin martingala (i per tant tinguin esperança constant). Sabem que, donat un moviment brownià W, el procés $exp\{\frac{1}{2}\sigma^2 t + \sigma W(t)\}$ és també una martingala. Per tant modifiquen el preu descomptat de la següent manera:

$$\tilde{S}(t) = S_0 e^{-\frac{1}{2}\sigma^2 t + \sigma((\frac{\mu - r}{\sigma})t + W(t))}$$

Designant $W_Q(t) = \frac{\mu - r}{\sigma}t + W(t)$
Obtenim

$$\tilde{S}(t) = S_0 e^{-\frac{1}{2}\sigma^2 t + \sigma W_Q(t)}$$

Si podem construir una mesura de probabilitat Q, sota la qual W_Q sigui un moviment brownià, aleshores $\tilde{S}(t)$ serà una martingala respecte Q i respecte la filtració natural $\mathbb{F}^W$ (W_Q i W difereixen només en una constant, cosa que no afecta a la filtració). Notem que $W_Q(0) = 0$ i que els increments venen descrits per $W_Q(t) - W_Q(s) = \frac{\mu - r}{\sigma}(t-s) + W(t) - W(s)$ i per tant segueixen una distribució normal amb variància $(t-s)$, ja que la constant additiva no afecta al càlcul de la variància. Pel mateix motiu, els increments són independents. En canvi però, els increments tenen, en general, esperança no nul·la .
Per no carregar la notació denotem $W_Q(t) = bt + W(t)$ on $b = \frac{\mu - r}{\sigma}$. Clarament

$$\mathbb{E}_P(W_Q(t)) = \mathbb{E}_P(bt + W(t)) = bt.$$

Usant la densitat de W(t) ho podem reescriure com

$$\mathbb{E}_P(W_Q(t)) = \int_{-\infty}^{\infty}(bt + x)f_{W(t)}(x)dx = \int_{-\infty}^{\infty}(bt + x)\frac{1}{\sqrt{2\pi t}}e^{-\frac{x^2}{2t}}dx.$$

Notem que

$$\begin{aligned}
0 = \int_{-\infty}^{\infty}(bt + x)\frac{1}{\sqrt{2\pi t}}e^{-\frac{x^2}{2t}} &= \int_{-\infty}^{\infty}(bt + x)e^{-\frac{1}{2}b^2 t - bx}\frac{1}{\sqrt{2\pi t}}e^{-\frac{x^2}{2t}}dx \\
&= \int_{-\infty}^{\infty}(bt + x)e^{-\frac{1}{2}b^2 t - bx}f_{W(t)}(x)dx \\
&= \int_{\Omega}(bt + W(t))e^{-\frac{1}{2}b^2 t - bW(t)}dP.
\end{aligned}$$

Fent el canvi de variable $y = x + bt$ tenim la primera igualtat, i l'última és deguda a una formulació de l'esperança en termes de la teoria de mesura. Per $A \subset \Omega$ i per $a \in \mathcal{F}$ podem definir una nova mesura Q, usant el teorema de Radon-Nikodym,

$$Q(A) = \int_A e^{-\frac{1}{2}b^2t - bW(t)} dP.$$

i aleshores

$$0 = \int_\Omega (bt + W(t)) e^{-\frac{1}{2}b^2t - bW(t)} dP = \int_\Omega (bt + W(t)) dQ = \int_\Omega W_Q(t) dQ = E_Q(W_Q(t))$$

En particular la variable $M(t) = e^{-\frac{1}{2}b^2t - bW(t)}$ és estrictament positiva, i el procés $M(t)$ és una martingala i per tant té esperança constant. Finalment per $t = 0$, tenim $M(0) = 1$, i per tant $\mathbb{E}_P(M(t)) = 1$. Equivalentment,

$$\int_\Omega M dP = 1.$$

Finalment, com $\mathbb{E}_P(W_Q(t)M(t)) = \mathbb{E}_Q(W_Q(t)) = 0$, la variable M(t) fa el paper de densitat de Q respecte P.
Podem provar llavors que:

Teorema 4.3. *El procés $W_Q(t) = bt + W(t)$ és un moviment brownià sota la probabilitat Q definida per*

$$Q(A) = \int_A e^{-\frac{1}{2}b^2T - bW(T)} dP$$

Per la demostració del teorema anterior veure [2].

Definició 4.4. *Diem que una probabilitat $\mathbb{P}$ és una mesura neutral al risc, si sota aquesta probabilitat el preu descomptat de l'actiu amb risc és una martingala.*

4.1.1 Estratègies admissibles i replicables

Definició 4.5. *Una estratègia financera en un mercat d'un sol actiu és un procés adaptat $(x, y) = \{(x(t), y(t)) : t \in [0, T]\}$ on $x(t)$ indica el número d'accions de l'actiu amb risc i $y(t)$ indica el número d'unitats monetàries en un instant de temps t.*

Definició 4.6. *El valor d'una estratègia (x, y) és un procés $V_{(x,y)} = (V_{(x,y)}(t))_{t\in[0,T]}$ definit per*

$$V_{(x,y)}(t) = x(t)S(t) + y(t)A(t)$$

Definició 4.7. *Diem que una estratègia $(x(t), y(t))$ és un arbritratge si $V_{(x,y)} = 0, V_{(x,y)}(t) \geq -L$ i per tot t $P(V_{(x,y)}(t) \geq 0) = 1$ i $P(V_{(x,y)}(t) > 0) > 0$*

Observem que, en el fons, un arbritatge no és més que una oportunitat de guanyar diners sense risc. Com és lògic, un model realista ha d'assumir que aquestes oportunitats no existeixen: a la pràctica es considera que els mercats fan desaparèixer molt ràpidament els arbitratges gràcies a la llei de la oferta i la demanda. Tenim llavors el següent teorema:

Teorema 4.8. *Primer teorema fonamental de les finances. Si un mercat admet una probabilitat neutral al risc, aleshores no admet arbitratges.*

La demostració es pot trobar a [4].
Suposem ara que estem comerciant amb una opció europea. De la introducció es desprèn que podem identificar el perfil de beneficis d'un derivat financer amb una certa variable aleatòria:

Definició 4.9. *Un perfil de beneficis és una variable aleatòria $H \geq 0$ que és F_T-mesurable.*

En particular per cada ω, és a dir, per cada evolució possible de mercat, la variable representa el possible benefici del derivat financer en la data T. Si denotem per $H(t)$ el preu a temps t del derivat financer, Tenim $H(T) = H$.

Definició 4.10. *Diem que un perfil de beneficis H és replicable si existeix una estratègia (x, y) tal que $V_{(x,y)}(T) = H$. Diem que un mercat és complet si tota variable aleatòria $\mathcal{F}_T^S-$mesurable H és replicable.*

Teorema 4.11. *Suposem que un mercat té una probabilitat neutral al risc. Aleshores el mercat és complet si i només si la mesura neutral al risc és única.*

La demostració es pot trobar també a [4].

Teorema 4.12. *El model de Black-Scholes és complet en el sentit següent: Per a cada variable aleatòria H, $\mathcal{F}_T$ − mesurable tal que $H \geq 0$ i $\mathbb{E}(H^\alpha) < \infty$ per algun $\alpha > 2$ existeix una estratègia admissible i replicable (x,y) tal que*

$$V_{(x,y)}(t) = \mathbb{E}_Q(e^{-r(T-t)}H|\mathcal{F}_t),$$

on Q denota la probabilitat neutral al risc.

Podem provar ara que el mercat financer donat pel model de Black-Scholes és complet:

Teorema 4.13. *La probabilitat neutral al risc en el model de Black-Scholes és única.*

Demostració. Posem $B \in \mathcal{F}_T^S = \mathcal{F}, H = \mathbb{1}_B$ i $\tilde{H} = e^{-rT}H$. Notem que H és una variable aleatòria acotada i per tant pot ser perfil de beneficis d'un derivat financer. Suposem que Q_1 és una altra probabilitat sota la qual els preus descomptats són martingala. Aleshores,

$$\begin{aligned} E_{Q_1}(\tilde{H}) &= E_{Q_1}(\tilde{V}(T)) \\ &= E_{Q_1}(V(0)) \\ &= V(0) \\ &= E_Q(\tilde{H}), \end{aligned}$$

on a la primera igualtat hem usat el teorema anterior: H és replicable. Com el B és arbitrari, Q i Q_1 coincideixen sobre $\mathcal{F}$ i per tant són iguals. □

Definició 4.14. *El valor d'una opció financera H a temps t és $V_t = \mathbb{E}_Q(e^{-r(T-t)}H|\mathcal{F}_t)$.*

Si la variable aleatòria H es pot escriure com $H = f(S_t)$ aleshores podem expressar V_t com una funció de t i S_t. Tenim doncs que, per exemple, el preu d'una opció de compra a temps t és $C_t = \mathbb{E}_Q(e^{-r(T-t)}f(S_t)|\mathcal{F}_t)$. Resolem així, a priori, el problema de la valoració d'opcions financeres.

4.2 Valoració d'opcions europees

L'objectiu d'aquest capítol és trobar formules tancades per a les opcions europees. Seguint la descripció feta a la introducció podem definir una opció europea com una variable aleatòria positiva $\mathcal{F}_T$-mesurable. Generalment s'escriu com $f(S_T)$ on $f(x) = (x - K)^+$ en el cas d'una opció de compra i $f(x) = (K - x)^+$ en el cas d'una opció de venda, on K denota el preu d'exercici de l'opció.
Com hem vist abans, per tal que els procés que ens determina el preus descomptats de l'actiu amb risc sigui una martingala sota la mesura neutral al risc, cal imposar que $\mu = r$. Si denotem $X^*(t) := (r - \frac{\sigma^2}{2})t + \sigma W_t$ aleshores el preu del l'actiu ve donat per l'expressió $S(t) = S(0)exp\{r - \frac{\sigma^2}{2}t + \sigma W_t\}$=$S(0)exp\{X^*(t)\}$ Tenim que $X^*(T) \sim \mathcal{N}((r - \frac{\sigma^2}{2})T, \sigma^2 T)$ i per tant $X^*(T)$ té per funció de densitat

$$\frac{1}{\sigma\sqrt{2\pi T}} e^{\frac{-(x-\mu^* T)^2}{2\sigma^2\sqrt{T}}}, \quad on \quad \mu^* = r - \frac{\sigma^2}{2}.$$

Per trobar el preu d'un put partim del Teorema 4.12 i la Definició 4.14. Usarem el següent lema de probabilitats condicionades:

Lema 4.15. *Sigui $\mathcal{G}$ una $\sigma-$àlgebra tal que $\mathcal{G} \subset \mathcal{F}$ i siguin X, Y dues variables aleatòries tals que Y és independent de $\mathcal{G}$ i X és $\mathcal{G}$-mesurable. Si h és una funció acotada aleshores*

$$E(h(X,Y)|\mathcal{G}) = E(h(z,Y))|_{z=X}.$$

És a dir podem calcular $E(h(X,Y)|\mathcal{G})$ com si X fos constant.

En el nostre cas tenim que volem calcular

$$\mathbb{E}_Q(e^{-r(T-t)} f(S_T)|\mathcal{F}_t) = \mathbb{E}_Q(e^{-r(T-t)} f(S(t)e^{(r-\frac{\sigma^2}{2})(T-t)+\sigma(W_T - W_t)})|\mathcal{F}_t).$$

Observem que sota la mesura de probabilitat neutral al risc, S_t és una variable $\mathcal{F}_t-$mesurable, i que $(W_T - W_t)$ és independent de $\mathcal{F}_t$. Podem aplicar doncs el lema anterior amb

$$\mathcal{G} = \mathcal{F}_t,$$

$$Y = W_T - W_t \ ,$$

$$X = S(t) \ ,$$

$$h(x,y) = e^{-r(T-t)} f(xe^{(r-\frac{\sigma^2}{2})(T-t)+\sigma y})) \ .$$

Si definim $\psi(x) = \mathbb{E}(h(x,Y))$ tenim que $\mathbb{E}(h(X,Y)|\mathcal{G}) = \psi(X)$. Calculem ara l'expressió de $\psi(x)$.
Com $h(t,x)$ depèn de t, també depèn de t la funció $\psi(x)$. Com $(W_T - W_t) \sim \mathcal{N}(0, T-t) \sim \sqrt{T-t}Z$ on Z és una variable normal estàndard amb funció de densitat $f_Z(z)$, tenim que l'expressió anterior de l'esperança condicionada ens queda com,

$$\begin{aligned}\psi(t,x) &= \mathbb{E}_Q(e^{-r(T-t)} f(xe^{(r-\frac{\sigma^2}{2})(T-t)+\sigma\sqrt{T-t}Z})) \\ &= e^{-r(T-t)} \int_{-\infty}^{+\infty} f(xe^{(r-\frac{\sigma^2}{2})(T-t)+\sigma\sqrt{T-t}z})) f_Z(z) dz.\end{aligned}$$

Així doncs ,

$$\psi(t,x) = e^{-r(T-t)} \int_{-\infty}^{+\infty} f(xe^{(r-\frac{\sigma^2}{2})(T-t)+\sigma\sqrt{T-t}z})) \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} dz. \tag{4.4}$$

En cas d'una opció europea de venda és necessari calcular (4.4) amb el perfil de beneficis $f(x) = (K-x)^+$. Notem que per poder aplicar el Teorema 4.12 cal comprovar la condició d'integrabilitat $\mathbb{E}(H^\alpha) < \infty$ i per poder aplicar el Lema 4.15 cal que la funció g sigui acotada. Totes dues condicions es compleixen en el cas del put, doncs el perfil de beneficis de l'opció de venda està acotat,

$$0 \leq max(0,(k - S(T))) \leq K$$

i per tant és integrable a qualsevol potència. Aleshores

$$\psi(t,x) = e^{-r(T-t)} \int_{-\infty}^{+\infty} (K - xe^{(r-\frac{\sigma^2}{2})(T-t)+\sigma\sqrt{T-t}z})^+ \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} dz.$$

La funció sota la integral és no nul·la quan

$$K - xe^{(r-\frac{\sigma^2}{2})(T-t)+\sigma\sqrt{T-t}z} \geq 0$$

Resolent,

$$\begin{aligned}(r-\frac{\sigma^2}{2})(T-t) + \sigma\sqrt{T-t}z \leq ln(\frac{K}{x})& \\ \sigma\sqrt{T-t}z &\leq ln(\frac{K}{x}) - (r-\frac{\sigma^2}{2})(T-t) \\ z &\leq \frac{ln(\frac{K}{x}) - (r-\frac{\sigma^2}{2})(T-t)}{\sigma\sqrt{T-t}} := d(t,x).\end{aligned}$$

Aleshores

$$\begin{aligned}\psi(t,x) &= e^{-r(T-t)} \int_{-\infty}^{d(t,x)} \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} (K - xe^{(r-\frac{\sigma^2}{2})(T-t)+\sigma\sqrt{T-t}z}) dz \\ &= Ke^{-r(T-t)}\Phi(d(t,x)) - e^{-r(T-t)} \int_{-\infty}^{d(t,x)} \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} (xe^{(r-\frac{\sigma^2}{2})(T-t)+\sigma\sqrt{T-t}z}) dz \\ &= Ke^{-r(T-t)}\Phi(d(t,x)) - e^{-r(T-t)} e^{(r-\frac{\sigma^2}{2})(T-t)} x \int_{-\infty}^{d(t,x)} \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}+\sigma\sqrt{T-t}z} dz \\ &= Ke^{-r(T-t)}\Phi(d(t,x)) - e^{-r(T-t)} e^{(r-\frac{\sigma^2}{2})(T-t)} x \int_{-\infty}^{d(t,x)} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(z-\sigma\sqrt{T-t})^2+\frac{1}{2}\sigma^2(T-t)} dz \\ &= Ke^{-r(T-t)}\Phi(d(t,x)) - x \int_{-\infty}^{d(t,x)-\sigma\sqrt{T-t}} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}y^2} dz \\ &= Ke^{-r(T-t)}\Phi(d(t,x)) - x\Phi(d(t,x) - \sigma\sqrt{T-t}),\end{aligned}$$

on hem fet el canvi de variables $y = z + \sigma\sqrt{T-t}$ i per tant $dy = dz$ i si $z = d(t,x)$ aleshores $y = d(t,x) - \sigma\sqrt{T-t}$.
Tornant al Lema 4.15 tenim que

$$P(t) = e^{-r(T-t)}\mathbb{E}_Q(f(S(T))|\mathcal{F}_t) = \psi(t, S(t)).$$

Així doncs s'ha provat el següent:

Teorema 4.16. *El preu a temps $t \leq T$ d'una opció de venda europea amb preu d'exercici K i data de venciment T és*

$$P(t) = Ke^{-r(T-t)}\Phi(d(t,S(t))) - S(t)\Phi(d(t,S(t)) - \sigma\sqrt{T-t})$$

amb

$$d(t,x) = \frac{ln(\frac{K}{x}) - (r - \frac{\sigma^2}{2})(T-t)}{\sigma\sqrt{T-t}}.$$

El preu d'una opció de compra europea ve donat llavors per l'anomenada paritat Call-Put:

Teorema 4.17. *Siguin $C(t)$, $P(t)$ els preus a temps t d'una opció de compra i d'una opció de venda europees respectivament amb data de venciment T i preu d'exercici K. Aleshores,*

$$C(t) - P(t) = S(t) - Ke^{-r(T-t)}$$

Demostració. Observem que si $S(T) - K \geq 0$ aleshores

$$S(T) - (S(T) - K)^+ + (K - S(T))^+ = S(T) - (S(T) - K) = K.$$

I si $S(T) - K \leq 0$ aleshores

$$S(T) - (S(T) - K)^+ + (K - S(T))^+ = S(T) - 0 + (K - S(T)) = K.$$

Per tant,

$$S(T) - C(T) + P(T) = S(T) - (S(T) - K)^+ + (K - S(T))^+ = K.$$

Multiplicant a tots dos costats per e^{-rT} tenim que

$$\tilde{S}(T) - \tilde{C}(T) + \tilde{P}(T) = Ke^{-rT}.$$

Sabem que $\tilde{S}(t), \tilde{C}(t), \tilde{P}(t)$ son martingales sota la mesura neutral al risc Q. Així doncs

$$\mathbb{E}_Q(\tilde{S}(T) - \tilde{C}(T) + \tilde{P}(T)|\mathcal{F}_t) = \tilde{S}(t) - \tilde{C}(t) + \tilde{P}(t) = Ke^{-rT}$$

Finalment, multiplicant a ambdós costats per e^{rt} tenim el resultat. □

Teorema 4.18. *El preu a temps $t \leq T$ d'una opció de compra europea amb preu d'exercici K i data de venciment T és*

$$C(t) = S(t)\Phi(-d(t,S(t)) + \sigma\sqrt{T-t}) - Ke^{-r(T-t)}\Phi(-d(t,S(t)))$$

amb

$$d(t,x) = \frac{ln(\frac{K}{x}) - (r - \frac{\sigma^2}{2})(T-t)}{\sigma\sqrt{T-t}}$$

Fem una petita reformulació en termes de notació. Posem

$$d_+(t,x) = \frac{ln(\frac{x}{K}) + (r + \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}}$$
$$d_-(t,x) = \frac{ln(\frac{x}{K}) + (r - \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}}$$

Aleshores $d(t,x) = -d_-(t,x)$ i

$$\begin{aligned} d(t,x) - \sigma\sqrt{T-t} &= \frac{ln(\frac{K}{x}) - (r - \frac{\sigma^2}{2})(T-t)}{\sigma\sqrt{T-t}} - \frac{\sigma^2(T-t)}{\sigma\sqrt{T-t}} \\ &= \frac{ln(\frac{K}{x}) - (r + \frac{\sigma^2}{2})(T-t)}{\sigma\sqrt{T-t}} \\ &= -\frac{ln(\frac{x}{K}) + (r + \frac{\sigma^2}{2})(T-t)}{\sigma\sqrt{T-t}} \\ &= -d_+(t,x). \end{aligned}$$

Les corresponents fórmules de Black-Scholes queden com

Teorema 4.19. *Fórmules de Black-Scholes*

$$C((S(t),\sigma,r,T,t,K) = S(t)\Phi(d_+(t,S(t))) - Ke^{-r(T-t)}\Phi(d_-(t,S(t)))$$
$$P(S(t),\sigma,r,T,t,K) = Ke^{-r(T-t)}\Phi(-d_-(t,S(t))) - S(t)\Phi(-d_+(t,S(t)))$$

amb $d_+(t,x), d_-(t,x)$ *com abans.*

Per acabar el capítol resolem el problema de la valoració d'opcions europees. El preu d'una opció de venda o compra europea a temps $t = 0$ és

$$\begin{aligned} &C(0) = S(0)\Phi(d_+(S(0))) - Ke^{-rT}\Phi(d_-S(0)) \\ &P(0) = Ke^{-rT}\Phi(-d_-(S(0))) - S(0)\Phi(-d_+(S(0))) \\ &amb \\ &d_+(x) = \frac{ln(\frac{x}{K}) + (r + \frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}} \\ &d_-(x) = \frac{ln(\frac{x}{K}) + (r - \frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}}. \end{aligned}$$

Notem que el paràmetre més important en el model de Black-Scholes és la volatilitat σ, els altres són constants conegudes. En el model s'assumeix que σ també és constant però a la pràctica per valorar opcions cal aproximar-la.
Fórmules semblants es poden trobar per les opcions barrera i look-back descrites a la introducció. Trobar-les però depèn de l'estudi de $\max_{t\in[0,T]} S(t)$ que al seu temps depèn de $\max_{t\in[0,T]} W(t)$. El desenvolupament de les fórmules d'aquestes opcions financeres es pot trobar a [2] i en l'apèndix es poden trobar algunes que s'han usat en els següents apartats d'aquest treball. En altres casos, com per exemple en el cas de les opcions asiàtiques amb mitjana aritmètica, no hi ha fórmula tancada ni per l'opció de compra ni per l'opció de venda.

5 El mètode de Monte Carlo

Comencem recordant alguns conceptes de la Teoria de la Probabilitat així com alguns dels teoremes més importants que resultaran clau en el desenvolupament d'aquest capítol.
Recordem:
Si X és una variable aleatòria amb llei absolutament contínua i g és una funció mesurable, aleshores $Y = g(X)$ és també una variable aleatòria i en particular, si denotem per f la funció de densitat de X tenim que,

$$E(Y) = \int_{-\infty}^{\infty} g(x)f(x)dx.$$

Teorema 5.1. *Llei forta de grans nombres: Sigui $\{X_n; n \geq 1\}$ una successió de variables aleàtories indepedents i idènticament distruibuïdes i denotem per $S_n = X_1 + ... + X_n$.*
1. Si $E(|X_1|) < \infty$, aleshores,

$$\lim_{n\to\infty} \frac{S_n}{n} = E(X_1) \ \ q.s$$

2. Recíprocament, si $E(|X_1|) = \infty$, aleshores,

$$\limsup_{n\to\infty} \frac{S_n}{n} = \infty \ \ q.s$$

Teorema 5.2. *Teorema del límit central: Sigui $\{X_n; n \geq 1\}$ una successió de variables aleatòries independents i idènticament distribuïdes de quadrat integrable, $E(X_1^2) < \infty$. Sigui $E(X_i) = \mu$ i $Var(X_i) = \sigma^2$. Suposem $\sigma^2 > 0$ i denotem $S_n = X_1 + ... + X_n$. Aleshores*

$$\frac{S_n - n\mu}{\sigma\sqrt{n}} \Rightarrow \ Y$$

on Y és una variable amb distribució $\mathcal{N}(0, 1)$ ($\Rightarrow$ denota convergència en llei)

La demostració dels dos teoremes anteriors es pot trobar a [11].
El mètode de Monte Carlo és una aplicació directa de la llei forta dels grans nombres.
Suposem que volem aproximar numèricament una certa quantitat $E[Y] = E[g(X)] = \theta$ on X és una variable aleatòria i g és una funció mesurable. Sigui ara $\{X_n; n \geq 1\}$ una família de variables aleatòries independents i idènticament distribuïdes amb mateixa llei que X, per les quals tenim un algoritme de complexitat baixa per generar seqüències de mostres independents de la seva funció de probabilitat. Aleshores, generant una mostra independent $X_1, ..., X_n$ la llei forta dels grans nombres ens garantitza que, si $E(|g(X)|) < \infty$, podem aproximar θ amb la mitjana artimètica:

$$\theta = E[g(X)] \simeq \frac{1}{n}\sum_{i=1}^{n} g(X_i).$$

Anomenarem estimador de Monte Carlo a

$$\hat{v}_n = \frac{1}{n}\sum_{i=1}^{n} g(X_i) := \frac{1}{n}\sum_{i=1}^{n} Y_i. \tag{5.1}$$

Notem que en realitat el mètode de Monte Carlo resulta un mètode per aproximar integrals: En el context de la probabilitat, la integral que determina l'esperança.

Com molts mètodes numèrics, el de Monte Carlo no és un métode exacte i per tant resulta important saber quina és la convergència d'aquest, així com una certa estimació de l'error (observem que per tal que el resultat sigui exacte cal que $n \to \infty$). Per això usem el teorema del límit central:
Si tenim una successió $Y_1, ..., Y_n$ de variables aleatòries amb funció de distribució la mateixa que la de la variable Y i amb $E(Y) = \mu$ i $Var(Y) = \sigma^2$, l'error en el mètode de Monte Carlo, $\varepsilon_n := \hat{v}_n - E(Y)$ podem dir que segueix una distribució $\mathcal{N}(0, \frac{\sigma^2}{n})$. En efecte, una petita reformulació del teorema del límit central ens dona

$$\frac{S_n - n\mu}{\sigma\sqrt{n}} = \frac{n(\frac{S_n}{n} - \mu)}{\sigma\sqrt{n}} = \frac{\sqrt{n}(\frac{S_n}{n} - \mu)}{\sigma} = \frac{\sqrt{n}(\hat{v}_n - E(Y))}{\sigma} = \frac{\sqrt{n}\epsilon_n}{\sigma} \to \mathcal{N}(0,1).$$

Notem doncs que la convergència del mètode és $\mathcal{O}(\frac{1}{\sqrt{n}})$. En particular per reduir l'error en un factor 10 cal multiplicar per 100 la mida de la mostra. La convergència és doncs més aviat lenta. Equivalentment, com la convergència és en llei obtenim

$$P(\frac{\sqrt{n}(\hat{v}_n - E(Y))}{\sigma} \leq x) \to \Phi(x).$$

on Φ és la funció de distribució de $\mathcal{N}(0,1)$ o bé que per tot $c_1 < c_2$

$$\lim_{n\to+\infty} P(\frac{\sigma}{\sqrt{n}}c_1 \leq \epsilon_n \leq \frac{\sigma}{\sqrt{n}}c_2) = \int_{c_1}^{c_2} e^{-\frac{x^2}{2}} \frac{1}{\sqrt{2\pi}} dx.$$

Si denotem per $Z \sim \mathcal{N}(0,1)$, com que sabem que $P(|Z| \leq 1.96) \approx 0.95$ tenim que, per n prou gran,

$$|\epsilon_n| \leq 1.96\frac{\sigma}{\sqrt{n}}.$$

Així doncs, també podem crear intervals de confiança d'una manera ja coneguda. Si volem que el límit de la probabilitat sigui $1-\delta$ amb $\delta > 0$, escollim $z_{\delta/2}$ tal que $1-\Phi(z_{\delta/2}) = \delta/2$. Aleshores

$$\hat{v}_n \pm z_{\delta/2}\frac{\sigma}{\sqrt{n}},$$

és un interval de confiança al $(1 - \delta)\%$ de E[Y] quan $n \to \infty$.
A la pràctica, la variància resulta desconeguda. Podem utilitzar, en canvi, la desviació típica modificada de la mostra $Y_1, ..., Y_n$ i que és fàcilment computable mitjançant

$$s_n := \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(Y_i - \hat{v_n})^2}$$

Observem que s_n^2 es pot reescriure com

$$s_n^2 = \frac{n}{n-1}\left(\frac{1}{n}\sum_{i=1}^{n} Y_i^2 - \hat{v_n}^2\right).$$

Per tant s_n^2 i $\hat{v}_n$ es poden calcular només usant $\sum_{i=1}^n Y_i$ i $\sum_{i=1}^n Y_i^2$. A més a més s_n^2 és un estimador sense biaix de σ^2 i consistent, en el sentit que $\lim_{n\to+\infty} s_n^2 = \sigma^2$. Com la convergència en probabilitat implica convergència en llei tenim que $s_n \Rightarrow \sigma$. Com $\sigma > 0$ tenim $\sigma/s_n \Rightarrow 1$. Usant el teorema de Slutsky obtenim doncs

$$\frac{\sqrt{n}(\hat{v}_n - E(Y))}{s_n} \Rightarrow \mathcal{N}(0,1)$$

I per tant, un nou interval de confiança per E[Y] és

$$\hat{v}_n \pm z_{\delta/2} \frac{s_n}{\sqrt{n}} \tag{5.2}$$

Per exemple, amb una probabilitat propera a 0.95, $E[Y]$ pertany a l'interval

$$\left[\hat{v}_n - \frac{1.96 s_n}{\sqrt{n}}, \hat{v}_n + \frac{1.96 s_n}{\sqrt{n}} \right].$$

Calcular la desviació típica d'una mostra ja generada és computacionalment parlant poc costós. Per tant, amb un nombre petit de càlculs podem donar un resultat prou bo de l'error d'aproximar E[Y] i un interval de confiança prou acurat. La possibilitat de donar un error estimat amb un cost numèric baix és una de les fortaleses del mètode.
Hem vist que el preu d'un actiu financer ve donat per $\mathbb{E}_Q(e^{-rT} f(S(T)))$, on $\mathbb{E}_Q$ denota l'esperança sota la mesura neutral al risc, i f denota la funció perfil de beneficis de l'actiu. Un esquema general de com usar el mètode de Monte Carlo per la valoració d'opcions és el següent:

1. Generar una mostra suficientment gran de possibles trajectòries que pot seguir S(t).
2. Per cada trajectòria calcular el perfil de beneficis.
3. Actualitzar el perfil de beneficis descomptant l'actiu sense risc.
4. Fer la mitjana de tots els preus descomptats.

n	Monte Carlo price	Error absolut
20000	6.93121	0.0404541
200000	6.94443	0.0272373
2000000	6.96715	0.00451909
4000000	6.97177	0.0000984354

Taula 2: Convergència del mètode de Monte Carlo. $S(0) = 100, T = 0.5, \mu = 0.08, \sigma = 0.35, K = 110, r = 0.05$. El preu de l'opció de compra és, usant la fórmula de Black-Scholes, 6.97167

6 Valoració d'opcions amb el metode de Monte Carlo

6.1 Generació del moviment brownià

Sigui $W = \{W_t; t \geq 0\}$ un moviment brownià estàndard, tenim que $W_t \sim \mathcal{N}(0, t)$. Degut a que els increments de W són independents i com $W_t - W_s \sim N(0, t-s) \sim \sqrt{t-s}\mathcal{N}(0,1)$ generar una mostra de W a partir dels increments és directe. En efecte, sigui $Z_1, ...Z_n$ una mostra de varables aleatòries independents normals (estàndard). Posant $t_0 = 0$, i com per definició $W(0) = 0$, que

$$W(t_{i+1}) = W(t_i) + \sqrt{t_{i+1} - t_i} Z_{i+1} \quad i = 0, ..., n-1$$

En el cas que $X \sim BM(\mu, \sigma^2)$, amb constants μ i σ, donat X(0), tenim

$$X(t_{i+1}) = X(t_i) + \mu(t_{i+1} - t_i) + \sigma\sqrt{t_{i+1} - t_i} Z_{i+1} \quad i = 0, ..., n-1 \tag{6.1}$$

Per tant, el problema de generar un moviment brownià es redueix a generar una mostra de variables aleatòries independents normals estàndard .
Finalment, seguint un raonament semblant, de l'equació (4.3) tenim un mètode per generar una mostra dels preus de l'actiu amb risc. Posant $0 = t_0 < t_1 < .... < t_n$ tenim

$$S(t_{i+1}) = S(t_i) exp(\mu - \frac{\sigma^2}{2})(t_{i+1} - t_i) + \sigma\sqrt{t_{i+1} - t_i} Z_{i+1} \ \ i = 0, 1, ..., n-1$$

on $Z_1, Z_2, ..., Z_n$ són variables aleatòries independents normals.

6.2 Generació de mostres aleatòries

6.2.1 Mètode de la transformada inversa

Suposem que volem generar una mostra aleatòria d'una funció de distribució. En altres paraules, volem generar una variable aleatòria X amb la propietat $P(X \leq x) = F(x)$ per a tot x. Un procediment intuïtiu és el següent: generem una variable aleatòria $U \sim Unif[0, 1]$. Aleshores $X = F^{-1}(U)$ on F^{-1} denota la funció inversa de F . Notem però que la inversa de la funció de distribució només està definida si F és estrictament creixent. Com la funció de distribució és creixent i contínua per la dreta, podem definir $F^{-1}(u) = inf\{x : F(x) \geq u\}$. Així solucionem els possibles problemes que poden sorgir si $F(x)$ té salts. A més a més, si F és constant en un interval [a,b] i X té distribució F aleshores $P(a < X \leq b) = F(b) - F(a) = 0$. Per tant les seccions planes corresponen a intervals amb probabilitat 0. Finalment notem que si F té per funció de densitat una funció contínua, aleshores F és estrictament creixent en tots aquells punts on la densitat no s'anul·la. Així doncs

$$P(X \leq x) = P(F^{-1}(U) \leq x) = P(U \leq F(x)) = F(x),$$

on la segona igualtat és degut a que per la definició anterior de F^{-1} els esdeveniments $\{F^{-1}(u) \leq x\}, \{u \leq F(x)\}$ coincideixen per a tot u i x.

Exemple 6.1. Recordem que una distribució exponencial amb mitjana λ té com a funció de distribució $F(x) = 1 - e^{-\frac{x}{\lambda}}$. Aleshores $X = -\lambda log(1 - U)$ amb $U = F(X)$. En particular com la distribució de U i $1 - U$ és la mateixa tenim que $X = -\lambda log(U)$.

Tant la distribució normal com la seva inversa es poden expressar en termes de la funció error:

$$Erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt.$$

Tenim llavors

$$Erf(x) = 2\Phi(x\sqrt{2}) - 1, \quad \Phi(x) = \frac{1}{2}(Erf(x/\sqrt{2}) + 1)$$

i

$$Erf(x)^{-1} = \frac{2}{\sqrt{2}}\Phi^{-1}(\frac{u+1}{2}), \quad \Phi(x)^{-1} = \sqrt{2}Erf^{-1}(2u+1).$$

Introduïm ara un algoritme clàssic per generar mostres aleatòries d'una distribució normal:

6.2.2 Mètode de Box-Muller

Volem transformar dues variables aleatòries independents amb distribució uniforme en l'interval [0,1], U_1, U_2 , en dues variables aleatòries normals independents estàndard: Z_1, Z_2. Usarem coordenades polars. Posem $R^2 = Z_1^2 + Z_2^2$ i notem que $R^2 \sim \chi_2^2$ i per tant R^2 segueix una distribució exponencial amb paràmetre $\lambda = \frac{1}{2}$. Ara bé, la variable aleatòria $-2logU_1$ té la mateixa llei, doncs per $x > 0$ tenim,

$$P\{-2log(U_1) < x\} = P\{log(U_1) > -\frac{x}{2}\} = P\{U_1 > e^{-\frac{x}{2}}\} = 1 - e^{-\frac{x}{2}}$$

A més a més $\Theta = 2\pi U_2$, és una variable aleatòria uniforme en $(0, 2\pi)$. Per tant posant

$$R^2 = -2log(U_1)$$

$$\Theta = 2\pi U_2$$

tenim una expressió en coordenades polars: Donat R, el punt (Z_1, Z_2) està distribuït uniformement en el cercle de radi $\sqrt{R}$ centrat en l'origen. Per generar un punt en el cercle, generem un angle entre 0 i 2π aleatòriament i després projectem l'angle a un punt cercle. En particular Θ ens dona l'angle, i el punt en el cercle té coordenades $(Rcos(\Theta), Rcos(\Theta))$ Així doncs, les variables

$$Z_1 = \sqrt{-2log(U_1)}cos(\pi U_2)$$
$$Z_2 = \sqrt{-2log(U_1)}sin(\pi U_2)$$

són dues variables aleatòries indepedents amb distribució normal estàndard.

6.3 El pont Brownià

La recursió (6.1) genera un vector $(W(t_1), ..., W(t_n))$ d'esquerra a dreta. No obstant això, potser voldríem generar primer el valor final $W(t_n)$, i després donat també $W(0) = 0$ acabar generant els punts intermedis per tal de fer les trajectòries contínues. Aquesta construcció es pot fer i s'anomena construcció del pont brownià. L'estratègia es basa en prendre distribucions condicionades:
Suposem que tenim valors $W(s_1) = x_1, W(s_2) = x_2, ...W(s_k) = x_k$ d'una trajectòria browniana generada per valors $s_1 < s_2 < \cdots < s_k$ i que volem calcular $W(s)$ condicionat als anteriors valors generats. Suposem $s_i < s < s_i$. Aleshores tenim

$$(W(s)|W(s_j), j = 1, ...k) = (W(s)|W(s_i) = x_i, W(s_{i+1}) = x_{i+1}),$$

és a dir el valor la distribució de $W(s)$ condicionada a tots els valors anteriors és la mateixa que resulta de prendre la distribució condicionada de $W(s)$ respecte als valors obtinguts en els temps immediatament anterior, s_i i immediatament posterior s_{i+1}. Aquest fet es deu a que donat $W(s_i)$, aleshores $W(s)$ és independent de tot $W(t)$ per a $t < s_i$ i donat $W(s_{i+1})$, $W(s_i)$ és independent de tot $W(t)$ per $t > s_{i+1}$.
Es pot provar llavors que (veure secció 3.1 de [3]) que

$$\begin{aligned}&(W(s)|W(s_1) = x_1, W(s_2) = x_2, ..., W(s_k) = x_k) = \\ &\mathcal{N}(\frac{(s_{i+1} - s)x_i + (s - s_i)x_{i+1}}{(s_{i+1} - s_i)}, \frac{(s_{i+1} - s)(s - s_i)x_{i+1}}{(s_{i+1} - s_i)}).\end{aligned}$$

6.4 Valoració d'opcions europees

Començarem amb les opcions més simples, les europees (european plain vanilla). Recordem que en el cas de les opcions europees, el perfil de beneficis ve donat per la fórmula $(S(T) - K)^+$ on T és el moment de venciment de l'opció i K és el preu d'exerici. Notem que només és rellevant el preu de l'actiu a data T, independentment del preu que hagi pogut tenir per a tot $t < T$. Així doncs, en aquest cas, no cal discretitzar les trajectòries del preu del l'actiu, nomès considerar possibles preus finals. Per tant, donada una mostra $Z_1, Z_2, ...Z_n$ de variables aleatòries normals independents, podem calcular el preu de l'actiu, que recordem que ve donat seguint l'esquema Monte Carlo amb el següent algoritme:

Algorithm 1 Mètode de Monte Carlo per a un call europeu

for `i=1,...,n` **do**
  `generate` $Z_i \sim \mathcal{N}(0, 1)$
  $S_i(T) \leftarrow S(0)exp(\mu - \sigma^2/2)T + \sigma\sqrt{T}Z_i$
  $C_i \leftarrow e^{-rT}(S(T) - K)^+$
end for
$\hat{S}_n = (C_1 + ... + C_n)/n$

6.5 Valoració d'opcions que depenen de la trajectòria

Discretitzem l'interval de temps [0,T] amb una partició equidistant, $t_i = i\frac{T}{m}$ i aleshores $t_i - t_{i-1} = \frac{T}{m}$. Seguint la filosofia Monte Carlo podem valorar opcions asiàtiques amb el següent algoritme:

Algorithm 2 Mètode de Monte Carlo per a un call asiàtic amb mitjana aritmètica

```
for i=1,...,n do
    for j=1,...,m do
        generate Z_ij ~ N(0,1)
        S_i(t_j) ← S(t_{i-1}) exp{(μ − σ^2/2)(t_i − t_{i-1}) + σ sqrt(t_i − t_{i-1}) Z_ij}
        S̄ ← (S_i(t_1) + .... + S_i(t_m)) / m
        C_i ← e^{-rT} (S̄ − K)^+
    end for
    Ŝ_n = (C_1 + ..... + C_n)/n
end for
```

7 Tècniques per la reducció de la variància

La fortalesa de l'estimador de Monte Carlo és precisament que és un estimador sense biaix de $\theta = E[g(X)]$. Observem però que $Var(\hat{v}_n) = \frac{1}{n}Var(g(X))$. Clarament incrementant la mida de mostra la variància tendeix a 0, però com hem vist, l'error en el mètode convergeix més aviat lentament. L'objectiu d'aquest capítol és desenvolupar mètodes per millorar l'eficiència i l'eficàcia del mètode de Monte Carlo reduint la variància de les simulacions.

7.1 Varietats de control

Suposem que volem estimar $\mu = E(Y)$ usant el mètode de Monte Carlo, és a dir usant l'estimador $\bar{Y}_n = \frac{Y_1+....+Y_n}{n}$. Suposem que en cada iteració podem calcular X_i juntament amb Y_i, tal que els parells $(X_i, Y_i) \quad i = 0, ..., n$ són i.i.d i que $E[X_i] = E(X)$ és coneguda (denotem amb el parell (X,Y) variables aleatòries genèriques amb la mateixa distribució que cada (X_i, Y_i)). Aleshores, per una certa constant b, podem definir

$$Y_i(b) = Y_i - b(X_i - E[X]).$$

a cada iteració i-èssima. Definim llavors

$$\bar{Y}(b) = \bar{Y} - b(\bar{X} - E[X]) = \frac{1}{n}\sum_{i=1}^{n}(Y_i - b(X_i - E[X])).$$

Notem que l'error observat $\bar{X} - E[X]$ controla l'error a l'estimar E[Y].
Com a estimador de E[Y], l'estimador anterior és un estimador sense biaix ja que

$$E[\bar{Y}(b)] = E[\bar{Y} - b(\bar{X} - E[X])] = E[\bar{Y}] = E[Y].$$

i és un estimador consistent, doncs, amb probabilitat 1,

$$\lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^{n}Y_i(b) = \lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^{n}(Y_i - b(X_i - E[X])) = E[Y - b(X - E[X])] = E[Y].$$

A més a més, usant que X,Y són variables aleatòries independents amb la mateixa distribució que (X_i, Y_i) cada $Y_i(b)$ té variància,

$$Var[Y_i(b)] = Var[Y_i - b(X_i - E[X])] = \sigma_Y^2 + b^2\sigma_X^2 - 2b\sigma_X\sigma_Y\rho_{XY} := \sigma^2(b) \qquad (7.1)$$

on $\sigma_X^2 = Var[X], \sigma_Y^2 = Var(Y)$ i ρ_{XY} és el coeficient de correlació entre X,Y que recordem que és $\rho_{XY} = \frac{E[(X-E[X])(Y-E[Y])]}{\sigma_X\sigma_Y}$, σ_X és la desviació estàndard de X i σ_Y la desviació estàndard de la variable Y. Notem que mentre l'estimador $\hat{v}_n$ basat en la mitjana mostral(corresponent al cas b=0) té variància $\frac{\sigma_Y^2}{n}$, l'estimador $\bar{Y}(b)$ té variància $\frac{\sigma^2(b)}{n}$. Per tant $\bar{Y}(b)$ té menys variància que $\hat{v}_n$ si $b^2\sigma_X < 2b\sigma_Y\rho_{XY}$. Trobem el valor b^* que minimitza el valor de (7.1):
Si posem $g(b) = \sigma^2(b)$ aleshores $g'(b) = 2\sigma_X^2 - 2b\sigma_X\sigma_Y\rho_{XY}$ i per tant $b^* = \frac{\sigma_Y}{\sigma_X}\rho_{XY} = \frac{COV(X,Y)}{Var[X]}$. Substituint, tenim doncs $g(b^*) = \sigma_Y^2(1 - \rho_{XY}^2)$ i la ràtio de reducció de la variància és

$$\frac{Var[\bar{Y} - b^*(\bar{X} - E[X])]}{Var[\bar{Y_n}]} = (1 - \rho_{XY}^2). \qquad (7.2)$$

Dues observacions sobre el mètode:
1-Treballant amb el coeficient òptim b^*, l'efectivitat del mètode depèn clarament de la correlació entre la variable que ens interessa Y i la variable control X. A més correlació, més reducció de variància.
2-Observem que tot el mètode es basa, en gran part, en trobar el valor òptim b^*. A la pràctica, si no sabem E[Y] difícilment podrem obtenir σ_Y o ρ_{XY}. Partint de (7.2), ho podem reformular com,

$$\hat{b}_n = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sum_{i=1}^n (X_i - \bar{X})^2}$$

Clarament $\hat{b}_n$ no depèn de E(X) i notem que $\hat{b}_n \to b^*$ amb probabilitat 1 (dividint abaix i a dalt per n i usant la llei forta dels grans nombres). Podem usar així doncs l'estimador $\bar{Y}(\hat{b}_n)$ que ve donat per la mitjana aritmètica de $Y_i(\hat{b}_n) = Y_i - \hat{b_n}(X_i - E[X]), i = 1, ..., n$.

7.2 Variacions antitètques

Recordem que volem aproximar $\theta = E[g(X)] = E[Y]$. Observem el següent:
Siguin Y_1, Y_2 dues copies de Y. Aleshores clarament un estimador sense biaix de θ és $\tilde{\gamma} := \frac{Y_1+Y_2}{2}$. Tenim

$$\begin{aligned}
Var(\frac{Y_1+Y_2}{2}) &= \frac{1}{4}E(Y_1+Y_2)^2 - (E(Y_1)+E(Y_2))^2 \\
&= \frac{1}{4}E[Y_1^2] - E(Y_1)^2 + E[Y_2^2] - E(Y_2)^2 + 2(E[Y_1Y_2] - E[Y_1]E[Y_2] \\
&= \frac{1}{4}Var(Y_1) + Var(Y_2) + 2Cov(Y_1, Y_2) \\
&= \frac{1}{2}(Var(Y) + Cov(Y_1, Y_2)).
\end{aligned}$$

En particular, si les variables no són independents, aleshores $Cov(Y_1, Y_2)$ és no nul·la i en particular si la covariància és negativa, la variància de $\frac{Y_1+Y_2}{2}$ és menor que la suma de variàncies (en el cas que haguem usat variables independents). Així doncs, intuïtivament, podem pensar en usar variables aleatòries amb covariància negativa per tal de reduir la variància. Més generalment, considerem una mostra de mida $2n$ de parells de variables aleatòries $Y_1, \tilde{Y_1}$ tals que:

- Els parells $(Y_1, \tilde{Y_1}), (Y_2, \tilde{Y_2}), ..., (Y_n, \tilde{Y_n})$ són independents i idènticament distribuïts.
- Per cada i, Y_i i $\tilde{Y_i}$ segueixen la mateixa distribució i per tant no són independents.

Aleshores l'estimador antitètic és la mitjana de les 2n observacions:

$$\hat{Y}_{AV} = \frac{1}{2n}(\sum_{i=1}^n Y_i + \sum_{i=1}^n \tilde{Y_i}) = \frac{1}{n}\sum_{i=1}^n (\frac{Y_i + \tilde{Y_i}}{2}). \tag{7.3}$$

En particular observem que $E[\frac{Y_i+\tilde{Y_i}}{2}] = E[Y] = \theta$ i a més a més

$$\begin{aligned}
Var(\hat{Y}_{AV}) = \frac{1}{n^2}\frac{n(Var(Y) + Cov(Y_1, Y_2)}{2} &= \frac{1}{2n}(Var(Y) + Cov(Y_1, Y_2)) \\
&= Var(\hat{v}_{2N}) + \frac{Cov(Y_1, Y_2)}{2n}.
\end{aligned}$$

Per tant, com abans, si les variables $Y_i, \tilde{Y_i}$ tenen correlació negativa, la variància de l'estimador $\hat{Y}_{AV}$ és menor que la variància de l'estimador de Monte Carlo, on s'usen 2n variables aleatòries independents. La pregunta que sorgeix és com generar parells de variables aleatòries amb correlació negativa i la mateixa distribució.

Definició 7.1. *Diem que dues variables aleatòries $Y, \tilde{Y}$ a valors reals són variables antitètiques si tenen la mateixa distribució i correlació negativa.*

Observem que si $U \sim Unif[0,1]$ aleshores 1- U també segueix una distribució uniforme. Per tant, generant una mostra $U_1, ..., U_n$ podem generar una segona mostra $1-U_1, ..., 1-U_n$ sense modificar la llei del procés. Diem que les variables U_i i $1-U_i$ formen un parell de variables antitètiques en el sentit que quan més gran és el valor de U_i més petit és el valor de $1-U_i$.
Podem estendre això a variables amb funció de distribució F i el mètode de la transformada inversa: $F^{-1}(U)$ i $F^{-1}(1-U)$ tenen totes dues distribució F i són variables antitètiques doncs F^{-1} és monòtona. A més a més, com la distribució uniforme és simètrica, $F^{-1}(U)$ i $F^{-1}(1-U)$ tenen el mateix valor però signe oposat (el mateix raonament serveix per la distribució normal, que també és simètrica). En particular tenim el següent resultat.

Proposició 7.2. *Sigui $Y = g(U_1, ..., U_n)$ on g és una funció monòtona en cada variable. Aleshores $Y_1 = g(U_1, ..., U_n), Y_2 = g(1-U_1, ..., 1-U_n)$ formen un parell de variables aleatòries antitètiques.*

En el cas que ens pertoca, com ens interessa generar variables aleatòries normals estàndard indepedents i idènticament distribuïdes $Z_1, ..., Z_n$, podem generar parells de variables antitètiques considerant la seqüència de variables aleatòries $-Z_1, ..., -Z_n$; notem que $Cov(Z, -Z) = -1$. Reformulant la proposició anterior:

Proposició 7.3. *$Y = g(Z_1, ..., Z_n)$ on g és una funció monòtona en cada variable. Aleshores $Y_1 = g(Z_1, ..., Z_n), Y_2 = g(-Z_1, ..., -Z_n)$ formen un parell de variables aleatòries antitètiques, amb $Z_i \sim \mathcal{N}(0,1)$.*

En el cas que això s'usi per generar un moviment brownià notem que si Z_i simula els increments d'una trajecòria d'aquest, aleshores $-Z_i$ simula els increments d'una trajectòria amb signe oposat. Finalment, si $X \sim \mathcal{N}(\mu, \sigma^2)$, aleshores podem usar com a variables antitètiques el parell de variables amb correlació negativa $X, 2\mu - X$.
De la definició (7.23) s'observa que $\hat{Y}_{AV}$ és la mitjana de n variables independents,

$$(\frac{Y_1 + \tilde{Y}_1}{2}), (\frac{Y_2 + \tilde{Y}_2}{2}), ..., (\frac{Y_n + \tilde{Y}_n}{2}) \tag{7.4}$$

Aplicant el teorema del límit central obtenim

$$\frac{\hat{Y}_{AV} - E[Y]}{\frac{\sigma_{AV}}{\sqrt{n}}}$$

on

$$\sigma_{AV}^2 = Var[\frac{Y_i + \tilde{Y}_i}{2}]$$

Com abans, la convergència en límit es manté si substituïm la variància per la desviació estàndard modificada s_{AV} de la mostra (7.4). Així doncs tenim un $1-\delta$ interval de confiança de la forma

$$\hat{Y}_{AV} \pm z_{\frac{\delta}{2}} \frac{\sigma_{AV}}{\sqrt{n}}$$

on $1 - \Phi(z_{\frac{\delta}{2}}) = \frac{\delta}{2}$.

7.3 Mostra estratificada

Com sempre, volem estimar E[Y]. Siguin subconjunts $A_1, ..., A_K$ subconjunts reals disjunts, anomenats estrats, tals que $P(Y \in \bigcup_i A_i) = 1$. Aleshores

$$E[Y] = \sum_{i=1}^{K} P(Y \in A_i) E[Y|Y \in A_i] = \sum_{i=1}^{K} p_i E[Y|Y \in A_i]$$

amb $p_i = P(Y \in A_i)$. Fins ara hem generat variables independents aleatòries $Y_1, ..., Y_n$ amb la mateixa distribució que Y. La proporció d'aquesta mostra Y_i que es troba en A_i no serà en general igual a p_i, però sí que tendirà a aquest valor a mesura que la mida de la mostra n es faci gran. El que volem a partir d'ara es construir una mostra escollint des de bon principi quina fracció de les observacions cal generar de cada estrat A_i.
Podem generalitzar això de dues maneres: Primer, fent que els estrats A_i siguin definits en termes d'una segona variable X, que pren valors reals, tal que $P(X \in v \bigcup_i A_i) = 1$. Aleshores

$$E[Y] = \sum_{i=1}^{K} P(X \in A_i) E[Y|X \in A_i] = \sum_{i=1}^{K} p_i E[Y|X \in A_i]$$

amb $p_i = P(X \in A_i)$ denota la probabilitat de cada estrat (cal que $\sum_{i=1}^{K} p_i = 1$ i cada p_i ha de ser estrictament positiu). En el nostre cas Y serà una funció de X, amb X una trajectòria d'un actiu amb risc i Y el preu descomptat d'aquest actiu. L'altre generalització consisteix en fer els valors $n_i,\ i = 1, ..., K$ (que corresponen al número de mostres que prenem de cada estrat) valors arbitraris tal que la seva suma és n i no pas valors proporcionals a $p_1, ..., p_K$.
Per cada $i = 1, ..., K$ sigui $Y_{ij},\ j = 1, ..., n_i$ una mostra independent de la distribució condicionada de Y donat $X \in A_i$. Denotem per $q_i = \frac{n_i}{n}$ la proporció d'observacions de cada estrat A_i. Aleshores tenim un estimador de E[Y] que ve donat per

$$\hat{Y} = \sum_{i=1}^{K} p_i \cdot \frac{1}{n_i} \sum_{j=1}^{n_i} Y_{ij} = \frac{1}{n} \sum_{i=1}^{K} \frac{p_i}{q_i} \sum_{j=1}^{n_i} Y_{ij} \tag{7.5}$$

Els punts clau del mètode són dos:
1-Escollir la variable que volem estratificar X, els estrats $A_1, ..., A_k$ i els valors $n_1, ..., n_k$.
2-Generar una mostra de la distribució de (X,Y) condicionat a $X \in A_i$.

Exemple 7.4. Siguin F una funció de distribució i $p_1, ..., p_K$ probabilitats tals que la seva suma és 1. Definim $a_0 = -\infty$ i

$$a_1 = F^{-1}(p_1), a_2 = F^{-1}(p_1 + p_2), ..., a_k = F^{-1}(p_1 + p_2 + \cdots + p_K) = F^{-1}(1),$$

i definim també els estrats

$$A_1 = (a_0, a_1], A_2 = (a_1, a_2], ..., A_K = (a_{k-1}, a_K]$$

(o bé amb $A_K = (a_{k-1}, a_K)$ si $a_K = \infty$). Notem que per construcció cada estrat A_i té probabilitat p_i sota la funció de distribució F: si Y té funció de distribució F, aleshores

$$P(Y \in A_i) = F(a_i) - F(a_{i-1}) = p_i.$$

Per tant, definir els estrats és directe donat els quantils a_i.

Finalment, donat els estrats $A_1, ..., A_K$, per generar una mostra de la distribució de Y condicionada a $Y \in A_i$ usem el mètode de la transformada inversa: Si $U \sim Unif[0,1]$ notem que

$$V = a_{i-1} + U(a_i - a_{i-1})$$

està uniformament distribuïda entre a_{i-1} i a_i. A més a més,

$$\begin{aligned} P(F^{-1}(V) \le x) &= P(a_{i-1} + U(a_i - a_{i-1}) \le F(x)) \\ &= P(U \le \left(\frac{F(x) - a_{i-1}}{a_i - a_{i-1}}\right)) \\ &= \frac{F(x) - a_{i-1}}{a_i - a_{i-1}}, \end{aligned}$$

i per tant $F^{-1}(V)$ té la distribució que buscàvem, és a dir, la distribució de Y condicionada a $Y \in A_i$.
Per simplificar les coses suposarem que escollim mostres dels diferents estrats d'una manera equiprobable, és a dir, tal que $p_i = q_i = \frac{n_i}{n}$. Denotem per

$$\mu_i = E[Y_{ij}] = E[Y|X \in A_i].$$

$$\sigma_i^2 = Var[Y_{ij}] = Var[Y|X \in A_i].$$

Aleshores

$$Var[\hat{Y}] = \sum_{i=1}^{K} p_i^2 Var[\frac{1}{n_i} \sum_{j=1}^{n_i} Y_{ij}] = \sum_{i=1}^{K} p_i^2 \frac{\sigma_i^2}{n_i} = \sum_{i=1}^{K} \left(\frac{n_i}{n}\right)^2 \frac{\sigma_i^2}{n_i} = \frac{1}{n} \sum_{i=1}^{K} p_i \sigma_i^2 \tag{7.6}$$

on n_i denota el valor de mostres pres en cada estrat A_i i per tant $n_1 + ... + n_K = n$. Si ho comparem amb la variància sense estratificació notem que

$$E[Y^2] = \sum_{i=1}^{K} p_i E[Y^2|X \in A_i] = \sum_{i=1}^{K} p_i(\sigma_i^2 + \mu_i^2).$$

Si usem que $\mu = E[Y] = \sum_{i=1}^{K} p_i \mu_i$ obtenim que

$$Var[Y] = E[Y^2] - \mu^2 = \sum_{i=1}^{K} p_i \sigma_i^2 + \sum_{i=1}^{K} p_i \mu_i^2 - \left(\sum_{i=1}^{K} p_i \mu_i\right)^2. \tag{7.7}$$

Per la desigualtat de Jensen obtenim que

$$\sum_{i=1}^{K} p_i \mu_i^2 \ge \left(\sum_{i=1}^{K} p_i \mu_i\right)^2$$

amb una desigualtat estricta excepte quan els μ_i són tots iguals. Comparant doncs (7.6) i (7.7) observem que usant una estratificació amb una mostra equiprobable dels estrats sempre aconseguim una reducció de la variància.

7.3.1 Estratificació terminal

En el projecte que ens pertoca, i en general, a l'hora de valorar opcions financeres, la part més important d'una possible trajectòria que pot seguir el preu d'un actiu és el valor d'aquest en la data T (data de venciment). Observem però que el preu d'un actiu ve donat per un moviment geomètric brownià, que al seu temps és una transformació monòtona d'un moviment brownià. Volem generar una mostra estratificada d'aquest últim procés per millorar l'eficàcia del mètode de Monte Carlo.
Suposem que volem generar una trajectòria browniana $W(t_1), ..., W(t_m)$. La idea és estratificar el valor final $W(t_m)$ i generar els valors intermedis de la trajectòria $W(t_1), ..., W(t_{m-1})$ usant la construcció del pont brownià. Considerem un estrat equiprobable de dimensió K, amb $n_1, ..., n_k$ proporcionals i siguin $U_1, ..., U_K$ variables aleatòries independents uniformes en [0,1]. Posem

$$V_i = \frac{i-1}{K} + \frac{U_i}{K}, \quad i = 1, ..., K$$

Aleshores, pel que hem vist abans $\Phi^{-1}(V_1), ..., \Phi^{-1}(V_K)$ és una mostra estratificada d'una variable amb distribució normal estàndard, i per tant $\sqrt{t_m}\Phi^{-1}(V_1), ..., \sqrt{t_m}\Phi^{-1}(V_K)$ és una mostra estratificada de $\mathcal{N}(0, t_m)$ que recordem que és la mateixa distribució que té $W(t_m)$. Per acabar de generar la trajectòria browniana, recordem que la distribució condicionada de $W(t_j)$ donats $W(t_{j-1})$ i $W(t_m)$ és,

$$\mathcal{N}\left(\frac{t_m - t_j}{t_m - t_{j-1}}W(t_{j-1}) + \frac{t_j - t_{j-1}}{t_m - t_{j-1}}W(t_m), \frac{(t_m - t_j)(t_j - t_{j-1})}{(t_m - t_{j-1})}\right),$$

amb $t_0 = 0$ i $W(0) = 0$. El següent algoritme genera K trajectòries brownianes estratificades sobre $W(t_m)$.

Algorithm 3 [3], secció 4.3.2

for `i=1,...,K` **do**
  `generate` $U \sim Unif[0,1]$
  $V \leftarrow (I - 1 + U)K$
  $W(t_m) \leftarrow \sqrt{t_m}\Phi^{-1}(V)$
  for `j=1,...,m-1` **do**
    `generate` $Z \sim \mathcal{N}(0,1)$
    $W(t_m) \leftarrow \frac{t_m - t_j}{t_m - t_{j-1}}W(t_{j-1}) + \frac{t_j - t_{j-1}}{t_m - t_{j-1}}W(t_m) + \sqrt{\frac{(t_m - t_j)(t_j - t_{j-1})}{(t_m - t_{j-1})}}Z$
  end for
end for

7.4 Monte Carlo condicionat

Considerem una opció barrera. La idea és la següent: Si la barrera és del tipus knock-out, un cop es creua la barrera el valor de l'opció val 0. Si en canvi l'opció és del tipus knock-in, un cop s'assoleix el preu el preu barrera, el valor de l'opció ve donat per la fórmula de Black-Scholes: A la pràctica, això es tradueix en que podem aturar la simulació i obtenir un preu analític de l'opció. La formulació matemàtica és la següent:
Siguin X,Z,Y variables aleatòries, i com fins ara, volem estimar $E[Y] = \theta$. De la teoria d'esperances condicionades respecte una variable discreta, sabem que si posem $V = E[Y|Z]$ aleshores,

$$E[V] = E[E[Y|Z]] = E[Y]$$

Notem que

$$Var[Y] = E[Var[Y|Z]] + Var[E[Y|Z]].$$

En particular $Var[Y|Z]$ és també una variable aleatòria, per tant $E[Var[Y|Z]] \geq 0$ i aleshores,

$$Var[Y] \geq Var[E[Y|Z]].$$

Dit d'una altra manera, per estimar θ, si simulem Z i podem calcular $E[Y|Z]$ d'una manera analítica podem reduir la variància sense necessitat de calcular Y com una transformació de Z.

Exemple 7.5. Considerem per exemple una opció barrera de venda del tipus Down-and-In discretitzada en m intervals de temps, és a dir, $t_i = iT/m$, amb preu d'exercici K, i preu barrera L, tal que $L \leq K$. Tenim valors $\{S(0), S(t_1), ..., S(T)\}$. Sigui $\tau, \tau < m$ el moment en que $S(t_\tau) \leq L$ .Aleshores

$$\begin{aligned} E[(K - S_T)^+ \mathbb{1}_{\{\min_{0\leq t\leq T} S_t \leq L\}}] &= E[E[(K - S_T)^+ \mathbb{1}_{\{\min_{0\leq t\leq T} S_t \leq L\}} | \tau = k, S(t_\tau) \leq L]] \\ &= E[E[(K - S_T)^+ | S(t_\tau)]] \\ &= P(S(t_\tau), \sigma, r, T, t_\tau, K) \end{aligned}$$

on $P(S(t_\tau), \sigma, r, T, t_\tau, K)$ denota el preu d'un call europeu amb valor inicial $S(t_\tau)$, data d'exercici T i temps inicial t_τ donat per la fórmula del Teorema 4.19.
Així doncs, si generem N trajectòries possibles, els diferents estimadors que usarem seran

$$\begin{cases} \mathrm{e}^{-r(T-t_\tau)} C(S(t_\tau), \sigma, r, T, t_\tau, K) & \text{si} \quad \tau = k < m \\ 0 & \text{en altre cas} \end{cases}$$

8 Part pràctica

8.1 Valoració d'opcions europees

Aquest capítol mostra els resultats de la valoració d'un call europeu amb el mètode de Monte Carlo. Per fer-ho s'ha usat el mètode estàndard, el mètode antitètic, i el mètode amb variable de control. Com a variable de control s'ha usat el preu del mateix actiu que estem valorant: Si S(t) és el preu de l'actiu, com estem treballant sota la probabilitat neutral al risc i la taxa d'interés r és constant, $e^{-rT}S(t)$ és una martingala i per tant $E_Q(e^{-rT}S(T)) = S(0)$. En particular si S_i és el preu de l'actiu en la i-èssima simulació i Y_i és el preu descomptat en la mateixa iteració podem usar l'estimador

$$\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{b_n}[S_i - e^{-rT}S(0)])$$

Observem que en el cas que estiguem valorant una opció europea, és a dir $Y = e^{-rT}(S(T) - K)^+$ la correlació entre Y i S(T) depèn K. En efecte, en K=0 tenim una correlació perfecta però per grans valors de K la correlació decau. Com que la correlació de les variables té un impacte en el mètode, l'eficàcia d'usar el preu de l'actiu com a variable de control depèn en gran part dels paràmetres usats. Les següents taules reflecteixen això:

K	30	40	45	50	55	60	65
$\hat{\rho}$	0.999	0.969	0.911	0.823	0.692	0.569	0.462
$\hat{b_n}$	0.962	0.801	0.620	0.423	0.249	0.137	0.080

Taula 3: Correlació estimada entre S(T) i $(S(T) - K)^+$ per diferents valors de K, amb S(0) =45, $\sigma = 20\% r = 5\%$ i T=1

K	30	40	45	50	55	60	65
$\hat{\rho}$	1	0.994	0.968	0.892	0.770	0.599	0.435
$\hat{b_n}$	0.987	0.940	0.815	0.577	0.340	0.162	0.065

Taula 4: Correlació estimada entre S(T) i $(S(T) - K)^+$ per diferents valors de K, amb S(0) =50, $\sigma = 30\% r = 5\%$ i T=0.25

La següent taula mostra el resultat de la valoració. Observem primer que a mesura que la volatilitat creix també ho fa el preu de l'opció així com l'error estàndard. El mètode antitètic resulta eficaç en tots els casos per tal de reduir l'error estàndard. El mètode de varietat de control resulta eficaç quan el preu d'exercici de l'opció és baix, per exemple K=30, pel que hem comentat abans. Per la mateixa raó no resulta tant eficaç quan el preu d'exercici de l'opció és alt. Finalment el mètode de mostra estratificada funciona prou bé amb tots els valors.

El fet que una volatilitat més alta comporti un preu més alt de l'opció de venda té sentit si entenem la volatilitat com una mesura de la variació del preu de l'actiu subjacent: quan més alta és, més pot canviar el preu de l'actiu subjacent ja sigui pujar o baixar: que el preu pugi afecta positivament al comprador d'una opció de compra, i que el preu baixi afecta positivament al comprador d'una opció de venda.

Preu d'un Call europeu

r	K	BS	C^1	C^2	C^3	C^4
$\sigma = 0.1$						
	30	10.300	10.343(0.0399)	10.301(0.00304)	10.299(0.000228)	10.298(0.00263)
0.01	40	1.794	1.796(0.0266)	1.78269(0.0136322)	1.807(0.0120)	1.792(0.00252)
	50	0.0264	0.0300(0.00338)	0.0277(0.00214)	0.0264(0.00292)	0.0258(0.00192)
	30	10.887	10.809(0.0398)	10.886(0.00273)	10.887(0.000263)	10.885(0.00231)
0.03	40	2.233	2.270(0.0230)	2.258(0.0131)	2.248(0.0114)	2.235(0.00248)
	50	0.0448	0.0464(0.00408)	0.0430(0.00272)	0.0460(0.00391)	0.0449(0.00271)
	30	11.463	11.4330(0.0408)	11.471(0.0029)	11.463(0.000186)	11.461(0.00270)
0.05	40	2.722	2.724(0.0311)	2.717(0.0122)	2.734(0.0107)	2.719(0.00182)
	50	0.0738	0.0706(0.00491)	0.0754(0.00360)	0.0828(0.00523)	0.07515(0.00253)
$\sigma = 0.2$						
	30	10.505	10.48(0.0779)	10.52(0.0181)	10.512(0.00829)	10.517(0.0110)
0.01	40	3.373	3.359(0.0544)	3.321(0.0295)	3.379(0.0237)	3.377(0.00595)
	50	0.651	0.595(0.0236)	0.666(0.0167)	0.676(0.0197)	0.660(0.00642)
	30	11.049	11.0567(0.0795)	11.0907(0.0174)	11.0485(0.00730)	11.0495(0.00693)
0.03	40	3.765	3.736(0.0560)	3.7482(0.0297)	3.758(0.0232)	3.775(0.00780)
	50	0.781	0.7464(0.0260)	0.752(0.0177)	0.752(0.0193)	0.7812(0.00693)
	30	11.590	11.461(0.0776)	11.5621(0.0148)	11.6034(0.00690)	11.5936(0.00710)
0.05	40	4.180	4.160(0.0596)	4.209(0.0301)	4.200(0.0226)	4.173(0.00524)
	50	0.930	0.876(0.0290)	0.931(0.0193)	0.967(0.0217)	0.925(0.00536)
$\sigma = 0.4$						
	30	12.104	11.950(0.150)	12.315(0.0699)	12.072(0.0296)	12.102(0.0229)
0.01	40	6.5102	6.3423(0.119)	6.5502(0.0709)	6.539(0.0467)	6.499(0.0188)
	50	3.309	3.186(0.0877)	3.239(0.0582)	3.2120(0.0497)	3.308(0.0193)
	30	12.525	12.245(0.147)	12.493(0.0665)	12.493(0.0284)	12.530(0.0238)
0.03	40	6.855	6.543(0.119)	6.902(0.0713)	6.877(0.0469)	6.884(0.0283)
	50	3.544	3.608(0.0954)	3.540(0.060)	3.504(0.0509)	3.585(0.0246)
	30	12.947	13.191(0.153)	13.0079(0.0666)	12.892(0.026)	12.977(0.0265)
0.05	40	7.209	7.0259(0.121)	7.135(0.0717)	7.251(0.0457)	7.181(0.0165)
	50	3.789	3.91310.0983)	3.7995(0.0625)	3.737(0.0503)	3.773(0.0188)

Taula 5: El preu inicial és S(0)=40, la data de venciment és T=1 (1 any) i el número d'iteracions és n=10.000. La resta de dades es poden veure a la taula[5].
C^1 és el preu del mètode de Monte Carlo estàndard.
C^2 és el preu del mètode de Monte Carlo amb el mètode antitètic de reducció de variància.
C^3 denota el preu amb varietat de control.
Entre parèntesis l'error estàndard. (desviació estàndard / n).
C^4 denota el preu amb el mètode de mostra estratificada. S'ha usat un estrat de 100 conjunts (K=100), n=5000 i 50 mostres en cada estrat($q_i = 100/5000$).

[5]Els resultats han estat arrodonits a tres xifres significatives per poder-los encabir d'una manera llegible en una mateixa taula

8.2 Valoració de diferents opcions barrera

Denotem d'ara en endavant per L el preu barrera de l'opció. Observem la següent relació entre les opcions barrera de venda i les opcions europees:

$$P_{UO}(T) + P_{UI}(T) = (K - S(T))^{+}(1 + \mathbb{1}_A),$$

on $A = \{\omega : \max_{t\in[0,T]} S(t,\omega) = L\}$. Es pot provar però, que la variable aleatòria $M_T = \max_{t\in[0,T]} S(T)$ ve donada per una funció de densitat(veure [2]) i per tant P(A)=0. La igualtat anterior queda doncs com

$$P_{UO}(T) + P_{UI}(T) = (K - S(T))^{+}.$$

De la mateixa manera, com el mínim té distribució contínua tenim que $P(\{\min_{t\in[0,T]} S(t,\omega) = L\}) = 0$ i aleshores

$$P_{DO}(T) + P_{DI}(T) = (K - S(T))^{+}.$$

El mateix raonament serveix per les opcions de compra barrera.
Les opcions barrera poden ser monitoritzades a temps continu o discret. En el primer cas existeixen fórmules tancades algunes de les quals s'han usat en aquest apartat i es poden trobar a l'apèndix. És intuïtiu pensar que una major discretització del temps fa tendir més el preu d'una opció barrera discreta al preu d'una opció barrera contínua. La primera taula mostra això:

L	m	C	Cl
85	50	16.5442(0.327)	
	100	16.3965(0.332)	
	250	15.4431(0.312)	
	500	15.2694(0.312)	14.813
90	50	13.4672(0.312)	
	100	12.7674(0.298)	
	250	12.0241(0.291)	
	500	11.4864(0.283)	11.202
95	50	9.5765(0.269)	
	100	8.78142(0.262)	
	250	7.50173(0.250)	
	500	7.21167(0.245)	6.29118

Taula 6: valoració d'una opció barrera de venda del tipus Down-and-Out.
Els paràmetres restants són S(0)=100, T=1, $\sigma = 4\%$, r= 1%,K=100. El paràmetre m es tal que $t_i = iT/m$.
C denota el preu de l'opció amb una discretització discreta
Cl denota el preu amb una monitorització a temps continu.

Per saber com de bones son les aproximacions en el cas d'una discretització del temps podem usar el següent resultat avançat de Brodie, Glasserman, i Kou [7].

Teorema 8.1. *Sigui $V_m(H)$ el preu d'una opció de compra o venda barrera, del tipus knock-in o knock-out, amb preu barrera K, i m tal que $\Delta t = T/m$. Sigui V(H) el preu de la mateixa opció barrera a temps continu. Aleshores,*

$$V_m(H) = V(He^{\pm\beta\sigma\sqrt{T/m}}) + o(\frac{1}{\sqrt{m}}),$$

on + correspon al cas $K > S(0)$, - correspon al cas $K < S(0)$ i $\beta = -\zeta(\frac{1}{2})/\sqrt{2\pi} \approx 0.5826$, on ζ és la funció zeta de Riemann.

La següent taula mostra el resultat de la valoració d'una opció barrera de compra del tipus down-and-out. Per valorar-la s'ha usat una discretització amb diferents mides. Com abans, per millorar el mètode de Monte Carlo s'ha usat el mètode antitètic i el mètode de variable de control. En aquest últim cas, com a variable de control s'ha usat una opció de compra europea. La taula també mostra el preu de l'opció barrera a temps continu, i el preu corregit de l'opció barrera discreta donat pel Teorema 8.1. Observem:

		Call down-and-out			
K	m	C^2	C^3	C^4	C^5
$C^1 = 7.92164$					
95	10	8.35028	8.2413(0.10141)	8.13508(0.0429339)	8.38749(0.0172858)
	20	8.25367	8.22226(0.104081)	8.20261(0.0458122)	8.31745(0.0193727)
	25	8.22564	8.26261(0.102862)	8.28969(0.0470011)	8.25996(0.0207578)
	50	8.14947	8.12692(0.103879)	8.05809(0.0482459)	8.17153(0.023836)
$C^1 = 5.48346$					
100	10	5.6943	5.61504(0.0859695)	5.29943(0.0444055)	5.72399(0.00995588)
	20	5.64904	5.52149(0.0851788)	5.47058(0.0459979)	5.66983(0.0127232)
	25	5.63562	5.54784(0.0858777)	5.52453(0.0471099)	5.67481(0.0121956)
	50	5.59858	5.52917(0.0862137)	5.58966(0.0479376)	5.61122(0.0147242)
$C^1 = 3.55355$					
110	10	3.64925	3.35675(0.0659129)	3.34991(0.0399008)	3.67452(0.00474484)
	20	3.62969	3.37692(0.0683854)	3.55102(0.0425572)	3.64291(0.00754787)
	25	3.62376	3.39938(0.0680493)	3.50826(0.0432746)	3.65158(0.00683694)
	50	3.60716	3.50099(0.070718)	3.57879(0.0429917)	3.62581(0.0080371)

Taula 7: El preu inicial és S(0)=100, la data de venciment és T=0.2, el número d'iteracions és n=10.000, $\sigma = 0.3$ i $r = 0.05$.
C^2 és el preu de l'opció barrera corregit donat pel teorema (8.1).
C^3 és el preu del mètode de Monte Carlo estàndard
C^4 és el preu del mètode de Monte Carlo amb el mètode antitètic de reducció de variància.
C^5 denota el preu amb varietat de control.
Entre parèntesis l'error estàndard. (desviació estàndard / n)

A mesura que fem una discretizació més fina, és a dir, amb més punts, veiem que com el preu corregit de l'opció barrera donat pel preu 8.1 tendeix al preu de l'opció barrera a temps continu tot i que en queda lluny pel fet que hem usat una discretització realista.

Com abans, el mètode de control de variables aconsegueix una millor reducció de la variància que el mètode antitètic. A més a més, amb el mètode de control de variables el preu de l'opció tendeix millor, en la majoria de casos, al preu corregit de l'opció barrera que amb el mètode antitètic.

Finalment veiem quins són els resultats de valorar una opció barrera de venda del tipus down-and-in usant el mètode de Monte Carlo condicionat:

			Put down-and-in		
σ	C^1	C^2	C^3	C^4	C^5
0.2	3.857	3.8022(0.0544)	3.739(0.0280)	3.784(0.0183)	3.787(0.0419)
0.3	6.0675	5.981(0.0734)	5.998(0.0304)	5.990(0.0128)	6.0002(0.0449)
0.4	7.990	7.882(0.0906)	7.945(0.0318)	7.970(0.00797)	7.9709(0.0465)
0.5	9.818	9.851(0.105)	9.775(0.0325)	9.794(0.00666)	9.719(0.0489)

Taula 8: El preu inicial és S(0)=45, la data de venciment és T=1, el número d'iteracions és n=10.000, $m = 250$ i $r = 0.07$ i el preu barrera és 40.
C^1 és el preu de l'opció barrera corregit donat pel teorema (8.1).
C^2 és el preu del mètode de Monte Carlo estàndard
C^3 és el preu del mètode de Monte Carlo amb el mètode antitètic de reducció de variància.
C^4 és el preu del mètode de Monte Carlo amb una opció de venda europea com a varietat de control.
C^5 és el preu de Monte Carlo condicionat.
Entre parèntesis l'error estàndard. (desviació estàndard / n)

Observem que el mètode de Monte Carlo condicionat redueix l'error en comparació al mètode estàndard en magnitud semblant al mètode antitètic, però resulta menys efectiu que el mètode de varietat de control. Pel mateix motiu comentat amb les opcions financeres, una alta volatilitat repercuteix en un augment del preu de l'opció: quan més alta és, més factible resulta que el valor de l'actiu subjacent caigui per sota el valor barrera.

8.3 Opcions asiàtiques

Les opcions asiàtiques amb perfil de beneficis $A_{arithm}(T) - K)^+$ amb

$$A_{arithm}(t_n) = \frac{1}{n}\sum_{k=1}^{n} S(t_k),$$

on $t_i = i\frac{T}{n}$ no tenen fórmules tancades pels preus. La raó és que la suma de variables aleatòries amb distribució lognormal no és una nova variable aleatòria amb distribució lognormal. Per tant, per tal de valorar aquest tipus d'opcions cal usar mètodes numèrics. Denotem per

$$A_{geo}(t_n) = \sqrt[n]{\prod_{k=1}^{n} S(t_k)}$$

la mitjana geomètrica discreta. Observem que si la distància entre els punts discretitzats és petita podem usar la següent aproximació:

$$\begin{aligned} A_{geo}(t_n) &= \left(\prod_{k=1}^{n} exp\{ln(S(t_k))\}\right)^{\frac{1}{n}} = exp\left(\frac{1}{n}\sum_{k=1}^{n} lnS(t_k)\right) \\ &\approx exp\left(\frac{1}{t_n}\int_0^{t_n} lnS(t)dt\right) \end{aligned}$$

Aquesta última expressió s'anomena mitjana geomètrica contínua. Per les opcions asiàtiques amb mitjana geomètrica (tant discreta com contínua) sí que existeixen fórmules analítiques que usen la fórmula de Black-Scholes. Seguidament, en derivem una:

Teorema 8.2. *El preu d'un call i d'un put d'opcions asiàtiques amb mitjana geomètrica discreta ve donat per $C(\hat{S(0)}, \hat{\sigma}, r, T, 0, K)$ i $P(\hat{S(0)}, \hat{\sigma}, r, T, 0, K)$.*

Demostració. Denotem per $t_k = \frac{k}{n}T$ i considerem el perfil de beneficis

$$H(T) = max\{0, K - exp\left(\frac{1}{n}\sum_{k=1}^{n} lnS(t_k)\right)\},$$

amb

$$S(t_k) = S(0)exp\{(r - \frac{1}{2}\sigma^2)t_k + \sigma W_Q(t_k)\}.$$

Analitzem l'exponencial:

$$\begin{aligned} \frac{1}{n}\sum_{k=1}^{n} lnS(t_k) &= \frac{1}{n}\sum_{k=1}^{n}(ln(S(0)) + (r - \frac{1}{2}\sigma^2)t_k + \sigma W_Q(t_k)) \\ &= ln(S(0)) + \frac{1}{n}(r - \frac{1}{2}\sigma^2)\sum_{k=1}^{n}\frac{k}{n}T + \sigma\frac{1}{n}\sum_{k=1}^{n} W_Q(t_k) \end{aligned}$$

Introduïm la suma telescòpica per tal de poder treballar amb els increments del procés de Wiener:

$$W_Q(t_k) = \sum_{i=1}^{k}(W_Q(t_i) - W_Q(t_{i-1})),$$

i aleshores

$$\sum_{k=1}^{n} W_Q(t_k) = \sum_{k=1}^{n}\sum_{i=1}^{k}[W_Q(t_i) - W_Q(t_{i-1})]$$
$$= \sum_{k=1}^{n}(n+1-k)[W_Q(t_k) - W_Q(t_{k-1})].$$

Sabem que la suma d'increments independents $[W_Q(t_k) - W_Q(t_{k-1})]$ és una variable aleatòria normal amb esperança zero i variància

$$\begin{aligned}\sum_{k=1}^{n}(n+1-k)^2 Var[W_Q(t_k) - W_Q(t_{k-1})] &= \sum_{k=1}^{n}(n+1-k)^2\frac{T}{n}\\ &= \sum_{k=1}^{n}(n+1-k)^2\frac{T}{n}\\ &= \frac{T}{n}\sum_{k=1}^{n} j^2\\ &= \frac{T}{n}\frac{n(n+1)(2n+1)}{6},\end{aligned}$$

on hem usat la fórmula de la suma de quadrats. Ens queda doncs, posant $Y \sim \mathcal{N}(0,1)$,

$$\sum_{k=1}^{n}(n+1-k)[W_Q(t_k) - W_Q(t_{k-1})] = \frac{1}{n}\sqrt{T\frac{(n+1)(2n+1)}{6}}Y.$$

Igualment tenim

$$\sum_{k=1}^{n} k = \frac{1}{2}n(n+1).$$

Així doncs el perfil de beneficis ens queda com,

$$H(T) = max\{0, K - S(0)exp\left(\frac{1}{2}(r - \frac{1}{2}\sigma^2)\frac{n(n+1)}{n^2}T + \sigma\sqrt{T}\sqrt{\frac{(n+1)(2n+1)}{6n^2}}Y\right)\}$$

Posem

$$\hat{\sigma} = \sigma\sqrt{\frac{n+1)(2n+1)}{6n^2}},$$

i busquem un x tal que $\hat{S(0)} = S(0)exp\{x\}$ i

$$\frac{1}{2}(r - \frac{1}{2}\sigma^2)\frac{n(n+1)}{n^2}T = (r - \frac{1}{2}\hat{\sigma}^2)T + x.$$

Insertant $\hat{\sigma}$, ens queda,

$$\hat{S(0)} = e^{-rT}S(0)exp\{\frac{(n+1)T}{2n}\left(r + \frac{\sigma^2}{2}\left(\frac{(2n+1)}{3n} - 1\right)\right)\}.$$

Finalment ens queda doncs,

$$H(0) = e^{-rT}\mathbb{E}_{\mathbb{Q}}\left(max\{0, K - \hat{S(0)}exp\{(r - \frac{1}{2}\hat{\sigma}^2)T + \hat{\sigma}\sqrt{T}Y\}\}\right)$$

$\square$

Una fórmula semblant es pot trobar per les opcions asiàtiques amb mitjana geomètrica contínua a [2].

Observació 8.3. La distància entre la mitjana aritmètica i la mitjana geomètrica és més aviat petita. Per exemple, si simulem una trajectòria , amb $\mu = 1\%, \sigma = 30\%, T = 1, m = 100, S(0) = 100, K = 100$ obtenim que la mitjana aritmètica de la trajectòria és 113.662 mentre que la mitjana geomètrica és 113.304.

De l'observació anterior i del fet que les opcions asiàtiques tenen una fórmula tancada es desprén que podem usar una opció asiàtica amb mitjana geomètrica discreta com a variable de control per calcular el preu d'opcions asiàtiques amb mitjana aritmètica. La idea original es pot trobar a [9].
En la següent taula mostrem els resultats de la valoració d'una opció asiàtica de compra amb mitjana aritmètica. La implementació es basa en l'esquema mostrat en l'algoritme 2 mostrat en la secció 6.5. La primera columna mostra el preu d'una opció europea. La segona el preu de l'opció asiàtica amb el mètode de Monte Carlo estàndard, la tercera mostra el preu de l'opció amb el mètode antitètic. Finalment, la quarta columna mostra el preu de l'opció usant com a variable de control una opció europea, i la cinquena mostra el preu de l'opció usant com a variable de control una opció asiàtica amb mitjana geomètrica discreta.
Tant el mètode antitètic com el de variable de control produeixen una reducció significativa de l'error estàndard (i en consequència de la variància), que és el paràmetre més important. En aquest cas, usar una opció europea com a variable de control, no és tant eficaç com anteriorment. Això és important doncs, com s'ha mostrat teòricament, pel mètode de variable de control no només és important usar una variable de la qual coneguem analíticament la seva esperança, sinó també usar una variable amb una forta correlació amb la que volem valorar. Resulta en canvi molt més eficaç usar una opció asiàtica amb mitjana geomètrica discreta com a variable de control amb la qual s'aconsegueix una reducció del paràmetre anteriorment esmentat proper al 50%. Observem també que en el mètode de Monte Carlo estàndard la desviació estàndard tendeix a augmentar a mesura que ho fa la volatilitat. Amb els mètodes implementats es tendeix a corregir això, especialment amb l'últim.

Preu d'una opció asiàtica amb mitjana aritmètica					
r	K	C^1	C^2	C^3	C^4
$\sigma = 0.2$					
	35	5.097(0.020)	5.0915(0.00111)	5.101(0.0101)	5.0945(0.0002)
0.03	40	0.885(0.013)	0.868(0.00658)	0.874(0.00707)	0.896(0.00015)
	45	0.00816(0.00116)	0.0111(0.0009)	0.0102(0.00109)	0.0123(0.00008)
	35	5.144(0.0202)	5.148(0.00109)	5.153(0.0101)	5.155(0.00023)
0.05	40	0.933893(0.0131)	0.905(0.00648)	0.913(0.00699)	0.936(0.000152)
	45	0.0122(0.00144)	0.0101(0.000853)	0.0104(0.000953)	0.0134(7.07266e-05)
	35	5.204(0.0204)	5.20384(0.00114)	5.2073(0.0102)	5.216(0.000239)
0.07	40	0.944671(0.0131)	0.936564(0.00643)	0.959887(0.00731)	0.978001(0.000155)
	45	0.0108752(0.00126)	0.0119681(0.000941)	0.0119876(0.00108)	0.0152612(0.000101)
$\sigma = 0.3$					
	35	5.14739(0.0300)	5.12707(0.00374)	5.13181(0.0154)	5.13897(0.000503)
0.03	40	1.27044(0.019)	1.27237(0.0102)	1.27368(0.0104)	1.31177(0.000336)
	45	0.105576(0.00545)	0.109231(0.00398)	0.103613(0.00366)	0.112927(0.000253)
	35	5.19517(0.0298)	5.18142(0.00363)	5.20153(0.0153)	5.19682(0.000489)
0.05	40	1.31891(0.0195)	1.29601(0.0100)	1.3269(0.0106)	1.35027(0.000340)
	45	0.108317(0.00558)	0.102783(0.00375)	0.103512(0.00389)	0.118902(0.000249)
	35	5.20666(0.0298)	5.23659(0.00341)	5.25035(0.0152)	5.25347(0.0005)
0.07	40	1.36372(0.01967)	1.36363(0.0103)	1.34168(0.0109)	1.39037(0.000369)
	45	0.107993(0.00540)	0.111548(0.00382)	0.111537(0.00402)	0.125794(0.000273)
$\sigma = 0.4$					
	35	5.20128(0.0382)	5.23835(0.00767)	5.22488(0.0204)	5.26006(0.000824)
0.03	40	1.64332(0.0256)	1.67108(0.0138)	1.70517(0.0144)	1.72824(0.000607)
	45	0.275558(0.0108)	0.30597(0.00777)	0.29735(0.00722)	0.319985(0.000519)
	35	5.22777(0.0384)	5.29136(0.00751)	5.28961(0.0201)	5.3144(0.000806)
0.05	40	1.6953(0.0259)	1.72554(0.0141)	1.72591(0.0144)	1.76627(0.000648)
	45	0.285543(0.0105)	0.302168(0.00748)	0.318084(0.00746)	0.331168(0.000510)
	35	5.36755(0.0384)	5.33158(0.00731)	5.33994(0.02)	5.36736(0.000825)
0.07	40	1.7775(0.0269)	1.74944(0.0141)	1.76459(0.0145)	1.80359(0.000636)
	45	0.325067(0.0121)	0.318696(0.00770)	0.323944(0.00731)	0.34272(0.000517)

Taula 9: El preu inicial és S(0)=40, la data de venciment és T=0.2, el número d'iteracions és n=10.000, $\sigma = 0.3$ i $r = 0.05$.
C^1 és el preu del mètode de Monte Carlo estàndard
C^2 és el preu del mètode de Monte Carlo amb el mètode antitètic.
C^3 és el preu del mètode de Monte Carlo amb una opció europea com a variable de control
C^4 És el preu amb una opció aritmètica amb mitjana geomètrica discreta com a variable de control.

9 Conclusions

El model de Black-Scholes ens proporciona un marc teòric, sota el qual podem valorar opcions financeres calculant esperances. A la pràctica però, a vegades resulta impossible trobar fórmules tancades per algunes opcions financeres: és llavors quan sorgeix la necessitat de mètodes numèrics que calculin aquests valors.
La llei forta dels grans nombres ens proporciona, juntament amb el desenvolupament d'eines informàtiques, un marc molt bo per poder aproximar esperances: el mètode de Monte Carlo. La seva simplicitat és un gran avantatge a l'hora d'implementar-lo, però a la pràctica, degut a que la convergència del mètode al valor real resulta més aviat lenta, caldria un número massa de grans de simulacions per poder arribar a resultats acurats. A causa d'això, i de la limitació dels recursos físics, resulta clau el desenvolupament de maneres de millorar tant l'eficàcia com l'eficiència del mètode. En particular, en aquest treball s'han implementat 4 mètodes: el mètode antitètic, el mètode de variable de control, el mètode de mostra estratificada i el mètode de Monte Carlo condicionat. Tots els mètodes esmentats permeten millorar l'eficàcia del mètode de Monte Carlo estàndard, però cal fer algunes observacions importants al respecte de cada un d'ells:
El mètode antitètic és, probablament, el més fàcil d'implementar i no necessita cap informació addicional sobre el model que estem simulant.
El mètode de control amb variables, en general, dona resultats semblants al mètode antitètic. Cal remarcar però que cal trobar sempre una variable de control de la qual cal conèixer l'esperança. Això comporta conèixer a priori alguna cosa sobre el model que estem simulant en general. Però conèixer l'esperança d'una manera analítica no resulta suficient per l'eficàcia del mètode: és clau trobar una variable aleatòria amb una correlació elevada amb la variable que volem valorar i això no sempre és fàcil. Per exemple, usar una opció de compra europea com a variable de control resulta molt eficaç en la valoració de certes opcions barrera i no tant en la valoració d'opcions asiàtiques amb mitjana aritmètica discreta. En aquest últim cas sí que resulta molt eficaç usar com a variable de control una opció de compra asiàtica amb mitjana geomètrica.
El mètode de mostra estratificada ha resultat eficaç, però és més complex que els anteriors en el sentit següent: per poder estratificar una variable no només cal conèixer la seva esperança, sinó també la seva distribució, i la variable ha d'estar lligada al model que estem simulant. Si disposem d'aquesta informació, un número de mostres proporcional en cada estrat produeix sempre una reducció de la variància.
Finalment, el mètode de Monte Carlo condicionat produeix resultats semblants en termes de reducció de l'error estàndard semblants al mètode antitètic, tot i que és més eficient al reduir el número d'iteracions necessàries.

10 Referències

Referències

[1] Lamberton, Damien; Lapeyre, Bernard. *Introduction to Stochastic Calculus Applied to Finance* . CRC Press, 2008. Print.

[2] Capiński, Marek; Kopp, Ekkehard. *The Black-Scholes Model.* Cambridge University Press, 2012. Print.

[3] Glasserman, Paul. *Monte Carlo Methods in Financial Engineering* . Springer, 2003. Print.

[4] Shreve, Steven E. *Stochastic calculus for finance 2, Continuous-time models* . Springer, 2004. Print

[5] Capiński, Marek; Zastawniak, Tomasz. *Numerical Methods in Finance with C++* . Cambridge University Press, 2012. Print.

[6] Capiński, Marek; Kopp, Ekkehard; Traple, Janusz. *Stochastic Calculus for Finance.* Cambridge University Press, 2012. Print.

[7] Broadie, Mark; Glasserman, Paul; Kou, Steven. *A Continuity Correction for Discrete Barrier Options.* Mathematical finance 7.4 (1997): 325-349. Web.

[8] Hull, John C. *Options, futures, and other derivatives.* Pearson Prentice Hall, 2009.

[9] Kemna, Angelien G.Z; Vorst, Antonius C.F. *A Pricing Method for Options Based on Average Asset Values.* Journal of banking finance 14.1 (1990): 113-129. Web.

[10] Boyle, Phelim P. *Options: A Monte Carlo Approach.* Journal of financial economics 4.3 (1977): 323-338. Web.

[11] Sanz, Marta. *Probabilitats.* Edicions Universitat de Barcelona, 1999. Print.

11 Annexos

11.1 Apèndix

Proposició 11.1. *Sigui* $X = (X_1, ..., X_n)$ *un vector de variables aleatòries independents i idènticament distribuïdes i siguin* f *i* g *funcions creixents. Aleshores*

$$E[f(X)g(X)] \geq E[f(X)]E[g(X)]$$

Demostració. Fem inducció en n. Si n=1, aleshores $(f(x) - f(y))(g(x) - g(y)) \geq 0$ per tot $x, y \in \mathbb{R}$. Per tant $E[(f(X) - f(Y))(g(X) - g(Y))] \geq 0$ és a dir,

$$E[(f(X)g(X)] + E[f(Y)g(Y))] \geq E[(f(X)g(Y)] + E[f(Y)g(X))]$$

Com les variables X e Y són independents i identicament distribuïdes tenim que

$$\begin{aligned} 2E[(f(X)g(X)] = E[(f(X)g(X)] + E[f(Y)g(Y))] \geq \\ \geq E[(f(X)g(Y)] + E[f(Y)g(X))] = 2E[(f(X)g(Y)] \end{aligned}$$

Suposem que l'enunciat és cert per $X \in \mathbb{R}^{n-1}$. Aleshores

$$\begin{aligned} E[f(X)g(X)|X_n = x_n] &= E[f(X_1, ...X_{n-1}, x_n)g(X_1, ...X_{n-1}, x_n)] \\ &\geq E[f(X_1, ...X_{n-1}, x_n)]E[g(X_1, ...X_{n-1}, x_n)] \\ &= E[f(X)|X_x = x_n)]E[g(X)|X_x = x_n)] \end{aligned}$$

Prenent esperances a ambdós costats tenim el resultat. □

Corol·lari 11.2. *Sigui* g *una funció monòtona i* $X_1, ..., X_n$ *variables aleatòries amb funció de distribució* F_X*. Aleshores les variables*

$$Y = g(F_X^{-1}(U_1), ..., F_X^{-1}(U_n))$$

i

$$Y' = g(F_X^{-1}(1 - U_1), ..., F_X^{-1}(1 - U_n))$$

són un parell de variables aleatòries antitètiques.

Demostració. Sense pèrdua de generalitat podem suposar que g és una funció creixent. Aleshores, tant Y com $f := -Y'$són funcións creixents. Per tant $= E[YY'] = E[g(U)f(U)] \geq E[g(U)]E[f(U)] = E[Y]E[Y']$ el que implica que

$$Cov(Y, Y') = E[YY'] - E[Y]E[Y'] \leq 0$$

□

Les següents fórmules han estat extretes de [8].

Teorema 11.3. *El preu d'un call Down-And-Out d'un actiu financer amb preu inicial S(0), preu d'exercici K, preu barrera L, volatilitat σ, i taxa d'interès r, i amb $K \geq L$ ve donat per l'expressió*

$$Call_{do} = C(r, T, K, S(0), \sigma) - Call_{di}$$

amb

$$S(0)\left(\frac{L}{S(0)}\right)^{2\lambda}\Phi(y) - Ke^{-rT}\left(\frac{L}{S(0)}\right)^{2\lambda-2}\Phi(y-\sigma\sqrt{T}),$$

on

$$\lambda = \frac{r+\frac{\sigma^2}{2}}{\sigma^2}$$
$$y = \frac{ln(\frac{L^2}{S(0)K})}{\sigma\sqrt{T}} + \lambda\sigma\sqrt{T}.$$

Teorema 11.4. *El preu d'un put Up-and-In d'un actiu financer amb preu inicial S(0), data de venciment T,preu d'exercici K, preu barrera L, volatilitat σ, i taxa d'interès r, i amb $K \geq L$ ve donat per l'expressió,*

$$P_{ui} = -S(0)\left(\frac{L}{S(0)}\right)^{2\lambda}\Phi(-y) + Ke^{-rT}\left(\frac{L}{S(0)}\right)^{2\lambda-2}\Phi(-y+\sigma\sqrt{T}),$$

amb y, λ com en el teorema anterior. Finalment el preu d'un put Up-and-Out ve donat per

$$P_{uo} = P(r, T, K, S(0), \sigma) - P_{ui},$$

on $P(r, T, K, S(0), \sigma)$ denota el preu d'un put europeu.

11.2 Codi

El següent codi ha estat creat i usat en les simulacions descrites en aquest treball. Funcions genèriques:

```
using namespace std;

#include <cmath>
#include <iostream>
#include <fstream>
#include <vector>
#include <algorithm>
//


const double pi=4.0*atan(1.0);
const int N=10000;
const int T=1;

double vector_mean(vector<double>& v){
    int n=v.size();
    double sum=0.0;
    for(int j=0; j<n;j++){
        sum+=v[j];
    }
    return sum/n;
}
double covariance(vector<double> Xi,vector<double> Yi){
    double sum = 0;
    int n=Xi.size();
    double mean_arr1 = vector_mean(Xi);
    double mean_arr2 = vector_mean(Yi);
    for(int i = 0; i < n; i++)
        sum = sum + (Xi[i] - mean_arr1) *
                    (Yi[i] - mean_arr2);
    return sum / (n-1);
}
double sd(vector<double>& v, double mean){
    double n=v.size();
    double sum=0.0;
    for(int i=0; i < n; i++){
        sum= sum +((v[i] - mean)* (v[i] - mean));
    }
    double a= sum/ (n-1);
    cout<<sqrt(a)<<endl;
    return sqrt(a);

}
double variance(vector<double>& v, double mean){
    double n=v.size();
    double sum=0.0;
    for(int i=0; i < n; i++){
        sum= sum +((v[i] - mean)* (v[i] - mean));
    }
    double a= sum/ (n);
    return sqrt(a);
```

```
}
double bcoefficient(vector<double> Xi,vector<double> Yi){
    double b=0.0;
    double num=0.0, div=0.0;
    double xmean= vector_mean(Xi);
    double ymean= vector_mean(Yi);
    for(int i=0;i<Xi.size(); i++){
        num=num+((Xi[i]-xmean)*(Yi[i]-ymean));
        div=div + ((Xi[i]-xmean)*(Xi[i]-xmean));
    }
    //cout<<"num"<<num<<endl;
    //cout<<"div"<<div<<endl;
    return num/div;
}

/*
 * Sample correlation
 */
double correlation(vector<double> Xi,vector<double> Yi){
    double num=0.0,div=0.0, div1=0.0,div2=0.0;
    double xmean= vector_mean(Xi);
    double ymean= vector_mean(Yi);
    for(int i=0;i<Xi.size(); i++){
        num=num+((Xi[i]-xmean)*(Yi[i]-ymean));
        div1=div1 + ((Xi[i]-xmean)*(Xi[i]-xmean));
        div2=div2 + ((Yi[i]-ymean)*(Yi[i]-ymean));
    }
    div= sqrt(div1*div2);
    return num/div;


}

double gaussian_distribution(double x){
    // we use the relation between the erf and the CDF function of
    normal distribution
    return 0.5 * ( 1.0 + erf(x / sqrt(2.0)));

}
/*
 * Beasley-Springer-Boro algorithm for approximating the inverse method.
 */
double inverse_gaussian_distribution(double u){
    static double a[4] = {   2.50662823884,
    -18.61500062529,41.39119773534,-25.44106049637};

    static double b[4] = {  -8.47351093090,
                          23.08336743743,
                          -21.06224101826,
                          3.13082909833};

    static double c[9] = {0.3374754822726147,
                         0.9761690190917186,
                         0.1607979714918209,
                         0.0276438810333863,
```

```
                    0.0038405729373609,
                    0.0003951896511919,
                    0.0000321767881768,
                    0.0000002888167364,
                    0.0000003960315187};
    double y= u - 0.5;
    double r=0.0;
    double div=0.0, num=0.0;
    double x=0.0;
    if (abs(y)< 0.42) {
        r=y*y;
        num= y* (((a[3]*r + a[2])* r + a[1])* r + a[0]);
        div=((((b[3]*r + b[2])*r + b[1])* r + b[0])*r +1);
        x= num/div;
    }
    else{
        r=u;
        if(y > 0){
            r=1-u;

        }
        r=log(-log(r));
        x=c[0] + r*(c[1] + r*(c[2] + r*(c[3] + r*
        (c[4]+r*(c[5] + r*(c[6] + r*(c[7] + r*c[8])))))));
        if(y < 0){
        x=-x;

        }
    }

    return x;

}
double BSC(double S0, double volatility, double rate, double time, double strike){
    double d1=0.0,d2=0.0;
    d1=(log(S0/strike) + ((rate + volatility*volatility/2)*time))/
    (volatility*sqrt(time));
    d2=(log(S0/strike) + ((rate - volatility*volatility/2)*time))/
    (volatility*sqrt(time));
    return S0*gaussian_distribution(d1)-(strike*(exp(-rate*time))
    *gaussian_distribution(d2));

}


double Box_Muller(){
    //We use the function rand() to generate an integer between 0
    //and RAND_MAX and then reescale it to the interval [0,1]
    double U1= (rand()+1.0) / (RAND_MAX +1.0);
    double U2= (rand()+1.0) / (RAND_MAX +1.0);
    return sqrt(-2.0*log(U1))*cos(2.0*pi*U2);
}
/*
 * Time=Time to maturity in years
 *
 */
```

Codi per les opcions europees:

```
double  EuropeanCall_MC_Convergence(long  numiter,double S0,
double volatility, double rate, double time, double strike){
    vector<double> prices(numiter,0);
    double c_0=0.0;//price of the call
    double rprice= BSC(S0,volatility,rate,time,strike);
    for(long i=0; i<numiter ; ++i){

        double zi= Box_Muller();
        double  b=volatility*sqrt(time)*zi;
        double a=(rate - 0.5*volatility*volatility)*time;
        double Si=S0*exp(a + b);

        prices[i]=exp(-rate*time)*max(Si-strike,0.0);
        if(i==20000){
            vector<double>  v(prices.begin(),prices.begin()+9999);
            double a= vector_mean(v);
            cout<<"20.000 iterations \t "<< a <<endl;
            cout<<"error "<<a - rprice<<endl;
        }
        if(i==200000){
            vector<double>  v2(prices.begin(),prices.begin()+99999);
            double b= vector_mean(v2);
            cout<<"200.000 iterations \t "<< vector_mean(v2) <<endl;
            cout<<"error "<<b - rprice<<endl;
        }
        if(i==2000000){
            vector<double>  v3(prices.begin(),prices.begin()+999999);
            double c= vector_mean(v3);
            cout<<"2.000.000 iterations \t "<< vector_mean(v3) <<endl;
            cout<<"error "<<c - rprice<<endl;
        }
    }

    c_0=vector_mean(prices);
    cout<<"Final error "<<c_0 - rprice << endl;
    return c_0;
}

/*
 *
 * Crude Monte Carlo method for european call
 * numiter= number of iterations
 * SO= initial price of the asset
 * volatility= annual volatility of the asset. Assumed to be constant.
 * Time= time to maturity of the option in years.
 * strike= strike price of the option.
 */

vector<double>  CEuropeanCall_MC(long  numiter,double S0, double
volatility, double rate, double time, double strike,ofstream& fout){
    vector<double> prices(numiter,0);
    double c0=0.0;//price of the call
    double rprice= BSC(S0,volatility,rate,time,strike);
    for(long i=0; i<numiter ; ++i){
```

```
        double  zi= Box_Muller();
        double  b=volatility*sqrt(time)*zi;
        double a=(rate - 0.5*volatility*volatility)*time;
        double Si=S0*exp(a + b);
        //prices[i]=Si;
        prices[i]=exp(-rate*time)*max(Si-strike,0.0);
    }

    c0=vector_mean(prices);
    fout<<c0<<"\t";
    double stdeerror= (sd(prices,c0)/sqrt(numiter));
    fout<<"("<<stdeerror<<")"<<"\t";
    //fout<<"["<<c0-1.96*stdeerror<<","<<c0 + 1.96*stdeerror<<"]";
    //cout<<"Final error "<<c0 - rprice << endl;
    return prices;
}

/*
 *
 * Monte Carlo method for european call. We also used
 antithetic variate in the same method. We use two paths:
 One with Z, and one with -Z.
 */
void  EuropeanCall_MC(long  numiter,double S0, double volatility,
double rate, double time, double strike,ofstream& fout){

    vector<double> prices1(numiter,0);
    // used for crude montecarlo
    vector<double> prices2(numiter,0);
    // used to generate de -S path for antithetic variate method
    vector<double> prices(numiter/2.0,0);
    double c0=0.0,c=0.0;// price of the call
    double rprice= BSC(S0,volatility,rate,time,strike);
    for(long i=0; i<numiter ; ++i){

        double  zi= Box_Muller();
        double  b=volatility*sqrt(time)*zi;
        double a=(rate - 0.5*volatility*volatility)*time;
        double S1=S0*exp(a + b);
        //this condition is to use the anthetic variate method
        with n=10000, and not with n=20000
        if(i % 2==0){
            double S2=S0*exp(a + (-1* b));
            prices2[i]=exp(-rate*time)*max(S2-strike,0.0);
        }
        prices1[i]=exp(-rate*time)*max(S1-strike,0.0);

    }

    c=vector_mean(prices1);//  //call price with the crude montecarlo estimator
    fout<<c;
    double stdeerror0=(sd(prices1,c)/sqrt(numiter));
    fout<<"("<<stdeerror0<<")"<<"\t";
    //generatee the antithetic estimator
    for(long j=0;j <numiter;j++){
```

```
        if(j%2==0){
            prices[j/2]= (prices1[j]+prices2[j])/2.0;
           // cout<<j<<endl;
        }

    }
    c0=vector_mean(prices);// call price with antithetic variate method
    fout<<c0;
    double stdeerror= (sd(prices,c0)/sqrt(numiter));
    fout<<"("<<stdeerror<<")"<<"\t";
    fout<<"["<<c0-1.96*stdeerror<<","<<c0 + 1.96*stdeerror<<"]";
}
/*
 * European Call with control variate:We use the uderlying asset
 price as control variate.
 *
 */
vector<double> EuropeanCall_control(long  numiter,double S0,
double volatility, double rate, double time, double strike,ofstream& fout){
    vector<double> prices(numiter,0);
    vector<double> s(numiter,0);
    double c0=0.0;//price of the call
    for(long i=0; i<numiter ; ++i){

        double zi= Box_Muller();
        double  b=volatility*sqrt(time)*zi;
        double a=(rate - 0.5*volatility*volatility)*time;
        double Si=S0*exp(a + b);
        s[i]=Si;
        prices[i]=exp(-rate*time)*max(Si-strike,0.0);
    }

    double bcoef=  bcoefficient(s,prices);
    //cout<<bcoef<<endl;
    double correla= correlation(s,prices);
    //cout<<"correlation "<<correla<<endl;
    vector<double>  Y(numiter,0);
    for(long j=0;j <numiter;++j){
        Y[j]= prices[j] - (bcoef*(s[j]-((exp(rate*time))*S0)));

    }
    c0=vector_mean(Y);
    fout<<c0;
    double stdeerror= (sd(Y,c0)/sqrt(numiter));
    fout<<"("<<stdeerror<<")"<<"\t";
    //fout<<"["<<c0-1.96*stdeerror<<","<<c0 + 1.96*stdeerror<<"]";
    //cout<<"Final error "<<c0 - rprice << endl;
    return prices;
}
vector<double> EuropeanCall_Stratifying(long  numiter,double S0,
double volatility, double rate, double time, double strike,
ofstream& fout){


    double c0=0.0;//price of the call
    double M=100;//num of strata
```

```
    double L=numiter/M;
    double std=0.0;
    double rprice= BSC(S0,volatility,rate,time,strike);
    vector<double> prices(M,0);
    vector<double> prices1(L,0);
    for(long i=1; i<=M ; ++i){

        for(int j=1;j<=L;j++){
            double random= (rand()+1) / (RAND_MAX +1.0);
            double v=(i-1+random)/M;
            double inv=inverse_gaussian_distribution(v);
            double b=volatility*sqrt(time)*inv;
            double a=(rate - 0.5*volatility*volatility)*time;
            double Si=S0*exp(a + b);
            prices1[j-1]=exp(-rate*time)*max(Si-strike,0.0);

        }
        double ave = vector_mean(prices1);
        prices[i-1]=ave;

        double st=sd(prices1,ave);
        std=std+(L/numiter)*pow(st,2);

    }

    c0=vector_mean(prices);
    double stdeerror= (sd(prices,c0)/sqrt(numiter));
    fout<<c0;
    fout<<"("<<sqrt(std)/sqrt(numiter)<<")"<<"\t";
    return prices;
}
double EuropeanCallCorrelation(long numiter,double S0, double
volatility, double rate, double time, double strike,ofstream& fout){

    vector<double> prices(numiter,0);
    vector<double> paths(numiter,0);
    double c0=0.0;//price of the call
    for(long i=0; i<numiter ; ++i){

        double zi= Box_Muller();
        double b=volatility*sqrt(time)*zi;
        double a=(rate - 0.5*volatility*volatility)*time;
        double Si=S0*exp(a + b);
        paths[i]=Si;
        prices[i]=exp(-rate*time)*max(Si-strike,0.0);
    }

    double bcoef= bcoefficient(paths,prices);
    fout<<"bcoef \t"<<bcoef<<"\t";
    double correla= correlation(paths,prices);
    fout<<"correlation \t"<<correla<<endl;
    return correla;
}
```

```
int main(){
    //this for random number:
    srand( (unsigned int) time( NULL ) ) ;
    ofstream fout;

    vector<double> volatilities{0.1, 0.2,0.3,0.4};
    //sigma values of 10%, 20%, 30%
    vector<double> rates{0.01,0.03, 0.05}; //interest rates of 1%, 3%,5%,.
    vector<double> strikes1{30,40,50};
    vector<double> strikes2{90,100,110};
    double S0=40;
    double S1=100;
    /*
     * European Call
    */

    fout.open("EuropeanCall.out");
    fout<<"European Call Table "<<endl;
    fout<<"Sigma \t BS Formula \t Crude Monte Carlo \t MonteCarlo
    Antithetic \t COnfidence interval  "<<endl;
    for(int i=0; i< volatilities.size();i++){
    fout<<volatilities[i]<<endl;
    for(int j=0; j< rates.size();j++){
    fout<<"r:"<<rates[j]<<endl;
    for(int z=0; z< strikes1.size();z++){
        fout<<strikes1[z]<<"\t" <<BSC(S0,volatilities[i],rates[j],1,
        strikes1[z])<<"\t";
        EuropeanCall_MC(N,S0,volatilities[i],rates[j],T,strikes1[z],fout);
        fout<<endl;

    }
    }
    }
    fout.close();



    fout.open("EuropeanCallControl.out");
    fout<<"With European Call with underlying asset price as
    control Variate "<<endl;
    fout<<"Sigma \t BS  \t Crude MonteCarlo \t Stratified"<<endl;
    for(int i=0; i< volatilities.size();i++){
    fout<<volatilities[i]<<endl;
    for(int j=0; j< rates.size();j++){
    fout<<"r:"<<rates[j]<<endl;
    for(int z=0; z< strikes2.size();z++){
    fout<<strikes1[z]<<"\t";
    fout<<BSC(S0,volatilities[i],rates[j],1,strikes1[z])<<"\t";
    //CEuropeanCall_MC(N,S0,volatilities[i],rates[j],1,strikes2[z],fout);
    //EuropeanCall_control(N,S0,volatilities[i],rates[j],1,strikes1[z],fout);
    EuropeanCall_control(N,S0,volatilities[i],rates[j],1,strikes1[z],fout);
    EuropeanCall_Stratifying(N/2,S0,volatilities[i],rates[j],1,strikes1[z],fout);
    fout<<endl;

    }
    }
```

```
	}
	fout.close();
	/*
	vector<double> strikes3{30,40,45,50,55,60,65};
	fout.open("correlation.out");
	fout<<"correlation between S(T) and (S(T)-K)^+ "<<endl;
	fout<<"T = 1 "  <<endl;
	for(int a=0;a<strikes3.size();a++){
		fout<<"K :"<<strikes3[a]<<"\t";
		EuropeanCallCorrelation(N,40.00,0.2,0.03,1,strikes3[a],fout);
		fout<<endl;

	}
       */
	cout << "Press enter to finish..." << endl;
		cin.get();

}
```

Funcions per les opcions asiàtiques:

```
/*
 * Closed price for the  discrete  geometric average asian call.
 * m is the number of points: t_{1}........, t_{m}
 */
double GAsian_Call(double S0,int m,  double sigma, double r,
double T, double K){
	double a=0.0, b=0.0, d1=0.0,d2=0.0;
	a = exp(-r*T)*S0*exp( (m+1.0)*T/(2.0*m)*(r+sigma*sigma*((2.0*m+1.0)
	/(3.0*m) -1.0)/2.0));
	b = sigma *sqrt((m+1.0)*(2.0*m+1.0)/(6.0*m*m));
	return BSC(a,b,r,T,K);

}


/*
 *
 * Simulation of numiter paths of european plain vanilla call.
 */
vector<double> CEuropeanCall_MC(long  numiter,double S0, double volatility,
double rate, double time, double strike,ofstream& fout){
	vector<double> prices(numiter,0);
	double c0=0.0;//price of the call
	double rprice= BSC(S0,volatility,rate,time,strike);
	for(long i=0; i<numiter ; ++i){

		double zi= Box_Muller();
		double  b=volatility*sqrt(time)*zi;
		double a=(rate - 0.5*volatility*volatility)*time;
		double Si=S0*exp(a + b);
		//prices[i]=Si;
		prices[i]=exp(-rate*time)*max(Si-strike,0.0);
	}

	c0=vector_mean(prices);
```

```
    //fout<<c0<<"\t";
    //double stdeerror= (sd(prices,c0)/sqrt(numiter));
    //fout<<"("<<stdeerror<<")"<<"\t";
    //fout<<"["<<c0-1.96*stdeerror<<","<<c0 + 1.96*stdeerror<<"]";
    //cout<<"Final error "<<c0 - rprice << endl;
    return prices;
}
/*
 *
 * Crude MonteCarlo method for Asian Call
 */
double AsianCall_CMC(int numiter,double S0, double volatility,
double rate, double time, int timestep, double strike,ofstream& fout){
    vector<double> paths(timestep,0);
    vector<double> prices(numiter,0);
    double dt=time/timestep;
    double zi,b,a,S;
    double c0=0.0;
    for(long i=0; i< numiter; i++){
        for(long j=0; j <timestep; ++j){
            if(j==0){
                paths[0]=S0;
            }
            else{
            zi= Box_Muller();
            b=volatility*sqrt(dt)*zi;
            //cout<<" b1 "<<b<<endl;
            a=(rate - 0.5*volatility*volatility)*dt;
            paths[j]=paths[j-1]*exp(a + b);
            }
        }
        double mean=0.0, geomean=0.0;
        mean= vector_mean(paths);
        //to print the distance between geometric and arithmetic mean
        //geomean= geometric_mean(paths)
        //cout<<"mean "<<mean<<endl;
        //cout<<"geo mean "<<geomean<<endl;
        double a1= exp(-rate*time)* max(mean-strike,0.0);
        prices[i]=a1;

    }
    //crude monte carlo
    double c=vector_mean(prices);
    fout<<c;
    double stdeerror0=(sd(prices,c)/sqrt(numiter));
    fout<<"("<<stdeerror0<<")"<<"\t";
    //fout<<"["<<c0-1.96*stdeerror0<<","<<c0 + 1.96*stdeerror0<<"]";
    return c0;

}
/*
 * antithetic variate
 */
double AsianCall_Antithetic(int numiter,double S0, double
volatility, double rate,
double time, int timestep, double strike,ofstream& fout){
```

```
    vector<double> paths1(timestep,0);
    vector<double> paths2(timestep,0);
    vector<double> prices1(numiter,0);
    vector<double> prices2(numiter,0);
    vector<double> prices(numiter,0);
    double sum=0.0;
    double dt=time/timestep;
    double zi,b,a,S;
    double c0=0.0;
    for(long i=0; i< numiter; i++){
        for(long j=0; j < timestep; ++j){
            if(j==0){
                paths1[0]=S0;
                paths2[0]=S0;
            }
            else{
                zi= Box_Muller();
                b=volatility*sqrt(dt)*zi;
                a=(rate - 0.5*volatility*volatility)*dt;
                paths1[j]=paths1[j-1]*exp(a + b);
                paths2[j]=paths2[j-1]* exp(a+ (-1)*b);
            }
        }
        double a1= exp(-rate*time)* max(vector_mean(paths1)-strike,0.0);
        double a2= exp(-rate*time)* max(vector_mean(paths2)-strike,0.0);
        prices1[i]=a1;
        prices2[i]=a2;

    }
    //antithetic control
    for(long j=0;j <numiter;++j){
        prices[j]= (prices1[j]+prices2[j])/2.0;

    }
    c0= vector_mean(prices);
    fout<<c0;
    double stdeerror= (sd(prices,c0)/sqrt(numiter));
    fout<<"("<<stdeerror<<")"<<"\t";
    //fout<<"["<<c0-1.96*stdeerror<<","<<c0 + 1.96*stdeerror<<"]";
    return c0;

}
/*
 *
 * Asian Call with with an European call as control .
 */
double AsianCall_Control(int numiter,double S0, double volatility, double rate,
double time, int timestep, double strike,ofstream& fout){
    vector<double> paths1(timestep,0);
    vector<double> eurprices(numiter,0);
    vector<double> prices(numiter,0);
    double BSPrice=0.0;
    double sum=0.0;
    double dt=time/timestep;
    double zi,b,a,S;
    double c0=0.0;
```

```
    double bcoef=0.0;
    vector<double>  Y(numiter,0);
    //Control Variate: European Call, We geta sample, and the expected value
    of the sample which is given by the bsformula.
    We also generate the optimal coefficient b that reduces variance.

    for(long i=0; i<= numiter; i++){
        for(long j=0; j < timestep; ++j){
            if(j==0){
                paths1[0]=S0;
            }
            else{
            zi= Box_Muller();
            b=volatility*sqrt(dt)*zi;
            a=(rate - 0.5*volatility*volatility)*dt;
            paths1[j]=paths1[j-1]*exp(a + b);
            }
        }
        double a1= exp(-rate*time)* max(vector_mean(paths1)-strike,0.0);
        eurprices[i]=exp(-rate*time)*max(paths1[timestep-1]-strike,0.0);
        prices[i]=a1;


    }
    //cout<<"correlation "<<correlation(prices,eurprices)<<endl;
    BSPrice =  BSC (S0,volatility,rate,time,strike);
    // bcoef=covariance(eurprices,prices)/covariance(eurprices,eurprices);
    bcoef=bcoefficient(eurprices,prices);
    //cout<<"bcoef "<<bcoef <<endl;
    //cout<<bcoef<<endl;
    for(long j=0;j <numiter;++j){
        Y[j]= prices[j] + (-bcoef*(eurprices[j]-BSPrice));

    }
    c0= vector_mean(Y);
    fout<<c0;
    double stdeerror= (sd(Y,c0)/sqrt(numiter));
    fout<<"("<<stdeerror<<")"<<"("<<bcoef<<")"<<"\t";
    //fout<<"["<<c0-1.96*stdeerror<<","<<c0 + 1.96*stdeerror<<"]";
    return c0;

}
/*
 *
 * Asian Call with with an Geometric asian  call as control .
 */

double AsianCall_GeoControl(int numiter,double S0,
double volatility, double rate,double time, int timestep,
double strike,ofstream& fout){
    vector<double> paths1(timestep,0);
    vector<double> prices(numiter,0);
     vector<double> geoprices(numiter,0);
    double BSPrice=0.0;
    double sum=0.0;
    double dt=time/timestep;
```

```
    double zi,b,a,S;
    double c0=0.0;
    double bcoef=0.0;


    BSPrice =  GAsian_Call(S0,timestep,volatility,rate,time,strike);
    for(int  i=0; i<=prices.size() ; i++){
        double geopr=S0;
        for(long j=0; j < timestep; ++j){
            if(j==0){
                paths1[0]=S0;
            }
            else{
                zi= Box_Muller();
                b=volatility*sqrt(dt)*zi;
                a=(rate - 0.5*volatility*volatility)*dt;
                paths1[j]=paths1[j-1]*exp(a + b);
            }
        }
        double mean=0.0, geomean=0.0;
        mean= vector_mean(paths1);
        geomean= geometric_mean(paths1);
        //cout<<"mean "<<mean<<endl;
        //cout<<"geo mean "<<geomean<<endl;
        double a1= exp(-rate*time)* max(mean-strike,0.0);
        double a2= exp(-rate*time)* max(geomean-strike,0.0);
        prices[i]=a1;
        geoprices[i]=a2;

    }
    //bcoef=covariance(geo,prices)/covariance(eurprices,eurprices);
    bcoef=bcoefficient(geoprices,prices);
    //cout<<"bcoef"<<bcoef<<endl;
    //crude monte carlo
    vector<double>  Y(numiter,0);
    for(long j=0;j <=numiter;++j){
        Y[j]= prices[j] + (-bcoef*(geoprices[j]-BSPrice));

    }
    c0= vector_mean(Y);
    fout<<c0;
    double stdeerror= (sd(Y,c0)/sqrt(numiter));
    fout<<"("<<stdeerror<<")"<<"\t";
    fout<<"["<<c0-1.96*stdeerror<<","<<c0 + 1.96*stdeerror<<"]";
    return c0;
}
int main(){
    //this for random number:
    srand( (unsigned int) time( NULL ) ) ;
    ofstream fout;
    //0.1 volatility gives very low values.
    vector<double> volatilities{ 0.2,0.3,0.4};
    //volatilities values of 10%, 20%, 30%
    vector<double> rates{0.03, 0.05, 0.07};
    //interest rates of  3%,5%,7%.
    vector<double> strikes1{35,40,45};
```

```
vector<double> strikes2{90,100,110};
double S0=40;
double S1=100;

//suposing 250 trading days during a year there are 50 days in 0.2 years.
fout.open("AsianCall.out");
fout<<"Asian  Call Table "<<endl;
fout<<"Sigma \t Crude european Call \t Crude Asian Call
\t MonteCarlo Antithetic "<<endl;
for(int i=0; i< volatilities.size();i++){
fout<<volatilities[i]<<endl;
for(int j=0; j< rates.size();j++){
fout<<"r:"<<rates[j]<<endl;
for(int z=0; z< strikes1.size();z++){
fout<<strikes1[z]<<"\t";
fout<<BSC(S0,volatilities[i],rates[j],0.2,strikes1[z])<<"\t";
//CEuropeanCall_MC(N,S1,volatilities[i],rates[j],1,strikes2[z],fout);
AsianCall_CMC(N,S0,volatilities[i],rates[j],0.2,50,strikes1[z],fout);
AsianCall_Antithetic(N,S0,volatilities[i],rates[j],0.2,50,strikes1[z],fout);
fout<<endl;

}
}
}
fout.close();

fout.open("AsianCallControl.out");
fout<<"Asian  Call Table "<<endl;
fout<<"Sigma \t Crude european Call \t European Control
\t Geometric Control "<<endl;
for(int i=0; i< volatilities.size();i++){
fout<<volatilities[i]<<endl;
for(int j=0; j< rates.size();j++){
fout<<"r:"<<rates[j]<<endl;
for(int z=0; z< strikes1.size();z++){
fout<<strikes1[z]<<"\t";
fout<<BSC(S0,volatilities[i],rates[j],0.2,strikes1[z])<<"\t";
AsianCall_Control(N,S0,volatilities[i],rates[j],0.2,50,strikes1[z],fout);
AsianCall_GeoControl(N,S0,volatilities[i],rates[j],0.2,50,strikes1[z],fout);
fout<<endl;

}
}
}
fout.close();

cout << "Press enter to finish..." << endl;
    cin.get();

}
```

Codi per les opcions barrera:

```
/*
 * Up and In put with barrier >=K
 */
```

```
double UpandInPutPrice(double S0, double sigma, double r,
double T, double K,double L){
    double lambda=0.0,y=0.0, bs=0.0,price1=0.0,priceput=0.0;
    lambda= (r +0.5*sigma*sigma)/(sigma*sigma);
    y=log(L*L/(S0*K))/(sigma*sqrt(T)) + lambda*sigma*sqrt(T);
    price1=-S0*pow((L/S0),2*lambda)*gaussian_distribution(-y) +
    K*exp(-r*T)*pow((L/S0),(2*lambda -2))
    *gaussian_distribution(-y + sigma*sqrt(T));

    return  price1 ;
}
double UpandOutPutPrice(double S0, double volatility, double r,
double T, double strike,double barrier){
    double lambda=0.0,y=0.0, bs=0.0,price1=0.0,priceput=0.0;
    bs=BSC(S0,volatility,r,T,strike);// price of an european call.
    price1=UpandInPutPrice(S0,volatility,r,T,strike,barrier);
    priceput=bs- S0 + strike*exp(-r*T);
    return priceput - price1 ;

}


/*Call down and out with barrier <= K
 *
 */
double DownandOutPrice(double S0, double sigma, double r,
double T, double K,double L){
    double lambda=0.0,y=0.0, bs=0.0,price1=0.0;
    lambda= (r +0.5*sigma*sigma)/(sigma*sigma);
    y=log(L*L/(S0*K))/(sigma*sqrt(T)) + lambda*sigma*sqrt(T);
    bs=BSC(S0,sigma,r,T,K);
    price1= S0*pow(L/S0,2*lambda)*gaussian_distribution(y)
    -K*exp(-r*T)*pow(L/S0,(2*lambda -2) )*gaussian_distribution(y -sigma*sqrt(T));
    return bs - price1;

}
double DownandInPrice(double S0, double sigma, double r,
double T, double K,double L){
    double lambda= (r +0.5*sigma*sigma)/(sigma*sigma);
    double y= log(L*L/(S0*K))/(sigma*sqrt(T)) + lambda*sigma*sqrt(T);
    return S0*pow(L/S0,2*lambda)*gaussian_distribution(y)
    -K*exp(-r*T)*pow(L/S0,(2*lambda -2) )*gaussian_distribution(y -sigma*sqrt(T));

}
/*
 *
 * Down and In Put with barrier L <K
 */
double DownandInPutPrice(double S0, double sigma, double r,
double T, double K,double L){
     double lambda= (r +0.5*sigma*sigma)/(sigma*sigma);
     double x1=(log(S0/L)/(sigma*sqrt(T))+ lambda*sigma*sqrt(T));
     double y= (log(L*L/(S0*K))/(sigma*sqrt(T))) + lambda*sigma*sqrt(T);
     double y1=(log(L/S0)/(sigma*sqrt(T) ))+ lambda*sigma*sqrt(T);
     return -1*S0*gaussian_distribution(-x1)+K*exp(-r*T)
     *gaussian_distribution(-x1+sigma*sqrt(T)) + S0*pow(L/S0,2*lambda)*
```

```
    (gaussian_distribution(y)-gaussian_distribution(y1))
   - K*exp(-r*T)*pow(L/S0,2*lambda -2)*
    (gaussian_distribution(y-sigma*sqrt(T))
   -gaussian_distribution(y1-sigma*sqrt(T)));



}

double DownInPut_MC(int numiter,double S0, double volatility,
double rate, double time, double timestep,double strike ,
double barrier,ofstream& fout){

    vector<double> prices(N,0);
    vector<double> paths(timestep+1,0);
    //cout<<paths.size()<<endl;
    double a=0.0, b=0.0;
    double dt=time/timestep;
    double C0=0.0;
    for(long i=0; i< numiter; i++){
        double minim=S0;

        for(long j=0; j <= timestep; ++j){
            if(j==0){
                paths[j]=S0;
            }
            else{
                double zi= Box_Muller();
                double  b=volatility*sqrt(dt)*zi;
                double a=(rate - 0.5*volatility*volatility)*dt;
                double S=paths[j-1]*exp(a + b);
                paths[j]=S;
                if( S < minim){
                    minim=S;
                }
            }

        }
        if(minim <= barrier){
            prices[i]= exp(-rate*time)*max(strike- paths[timestep-1],0.0);


        }
        else{
            prices[i]= 0.0;

        }
        //cout<<paths[0]<<endl;
        // cout<<"i "<<i<<" "<<s<<endl;;
    }
    C0=vector_mean(prices);

    fout<<C0;
    double stdeerror= (sd(prices,C0)/sqrt(prices.size()));
    fout<<"("<<stdeerror<<")"<<"\t";
    //fout<<"["<<C0-1.96*stdeerror<<","<<C0 + 1.96*stdeerror<<"]";
```

```
    //cout<<b<<endl;
    return C0;
}
double UpOutPut_MC(int numiter,double S0, double volatility ,
double rate, double time, double timestep,double strike ,
double barrier,ofstream& fout){

    vector<double> prices(N,0);
    vector<double> paths(timestep+1,0);
    //cout<<paths.size()<<endl;
    double a=0.0, b=0.0;
    double dt=time/timestep;
    double C0=0.0;
    for(long i=0; i< numiter; i++){
        double maxim=0.00;

        for(long j=0; j <= timestep; ++j){
            if(j==0){
                paths[j]=S0;
            }
            else{
                double zi= Box_Muller();
                double  b=volatility*sqrt(dt)*zi;
                double a=(rate - 0.5*volatility*volatility)*dt;
                double S=paths[j-1]*exp(a + b);
                paths[j]=S;
                if( S > maxim){
                    maxim=S;
                }
            }

        }
        if(maxim > barrier){
            prices[i]= 0.0;

        }
        else{

            prices[i]= exp(-rate*time)*max(strike - paths[timestep-1],0.0);

        }
        //cout<<paths[0]<<endl;
        // cout<<"i "<<i<<" "<<s<<endl;;
    }
    C0=vector_mean(prices);

    fout<<C0;
    double stdeerror= (sd(prices,C0)/sqrt(prices.size()));
    fout<<"("<<stdeerror<<")"<<"\t";
    //fout<<"["<<C0-1.96*stdeerror<<","<<C0 + 1.96*stdeerror<<"]";
    //cout<<b<<endl;
    return C0;
}
double UpOutPut_Antithetic(int numiter,double S0, double volatility ,
double rate, double time, double timestep,double strike ,
double barrier,ofstream& fout){
```

```
vector<double> prices(numiter/2,0);
vector<double> paths(timestep+1,0);
vector<double> antitheticprices(numiter/2,0);
vector<double> antitheticpaths(timestep+1,0);
double a=0.0, b=0.0;
double dt=time/timestep;
double C0=0.0;
for(long i=0; i< prices.size(); i++){
    bool t=false;
    double maximum1=S0;
    double maximum2=S0;
    for(long j=0; j <= timestep; ++j){
        if(j==0){
            paths[j]=S0;
            antitheticpaths[j]=S0;
        }
        else{
            double zi= Box_Muller();
            double  b=volatility*sqrt(dt)*zi;
            double a=(rate - 0.5*volatility*volatility)*dt;
            double S1=paths[j-1]*exp(a + b);
            double S2=antitheticpaths[j-1]*exp(a + (-1)*b);
            paths[j]=S1;
            antitheticpaths[j]=S2;
            //cout<<S<<endl;
            if( S1 > maximum1){
                maximum1=S1;
            }
            if( S2 > maximum2){
                maximum2=S2;
            }
        }
    }
    if(maximum1 > barrier){
        prices[i]=0.0;

    }
    else{
        prices[i]= exp(-rate*time)*max(strike - paths[timestep-1],0.0);
    }
    if(maximum2 > barrier){
        antitheticprices[i]=0.0;

    }
    else{
        antitheticprices[i]= exp(-rate*time)*
        max(strike - antitheticpaths[timestep-1],0.0);
    }

}
vector<double> Y(numiter/2,0);
for(int z=0; z< Y.size();z++){
    Y[z]=(antitheticprices[z]+ prices[z])/2.0;

}
```

```
    C0=vector_mean(Y);
    fout<<C0;
    double stdeerror= (sd(Y,C0)/sqrt(numiter));
    fout<<"("<<stdeerror<<")"<<"\t";
    //fout<<"["<<C0-1.96*stdeerror<<","<<C0 + 1.96*stdeerror<<"]";
    //cout<<b<<endl;
    return C0;


}
double UpOutPut_Control(int numiter,double S0, double volatility,
double rate, double time, double timestep,double strike ,
double barrier,ofstream& fout){

    vector<double> prices(numiter,0);
    vector<double> paths(timestep+1,0);
    vector<double> pathseur(numiter,0);
    double a=0.0, b=0.0;
    double dt=time/timestep;
    double C0=0.0;
    for(long i=0; i< prices.size(); i++){
        bool t=false;
        double maximum=S0;
        for(long j=0; j <= timestep; ++j){
            if(j==0){
                paths[j]=S0;
            }
            else{
                double zi= Box_Muller();
                double  b=volatility*sqrt(dt)*zi;
                double a=(rate - 0.5*volatility*volatility)*dt;
                double S=paths[j-1]*exp(a + b);
                paths[j]=S;
                //cout<<S<<endl;
                if( S > maximum){
                    maximum=S;
                }
            }

        }
        pathseur[i]=exp(-rate*time)*max(strike- paths[timestep -1],0.0);
        if(maximum > barrier){
            prices[i]=0.0;

        }
        else{
            prices[i]= exp(-rate*time)*max(strike- paths[timestep -1],0.0);

        }
    }

    double bcoef=  bcoefficient(prices,pathseur);
    double bscall= BSC(S0,volatility,rate,time,strike);
    double bsput=bscall- S0 + strike*exp(-rate*time);
    vector<double>  Y(numiter,0);
    for(long j=0;j <numiter;++j){
```

```
        Y[j]= prices[j] - (bcoef*(pathseur[j]- bsput));

    }
    C0=vector_mean(Y);
    fout<<C0;
    double stdeerror= (sd(Y,C0)/sqrt(numiter));
    fout<<"("<<stdeerror<<")"<<"\t";
    //fout<<"["<<C0-1.96*stdeerror<<","<<C0 + 1.96*stdeerror<<"]";
    return C0;


}
/*
 * Down and out crude monte carlo
 */
double DownOutCall_MC(int numiter,double S0, double volatility,
double rate, double time, double timestep,double strike ,
double barrier,ofstream& fout){

    vector<double> prices(numiter,0);
    vector<double> paths(timestep+1,0);
    double a=0.0, b=0.0;
    double dt=time/timestep;
    double C0=0.0;
    for(long i=0; i< prices.size(); i++){
        bool t=false;
        double minimum=S0;
        for(long j=0; j <= timestep; ++j){
            if(j==0){
                paths[j]=S0;
            }
            else{
                double zi= Box_Muller();
                double  b=volatility*sqrt(dt)*zi;
                double a=(rate - 0.5*volatility*volatility)*dt;
                double S=paths[j-1]*exp(a + b);
                paths[j]=S;
                //cout<<S<<endl;
                if( S < minimum){
                    minimum=S;
                }
            }
        }
        if(minimum >= barrier){
            prices[i]= exp(-rate*time)*max(paths[timestep-1]-strike,0.0);
        }
        else{
            prices[i]=0.0;
        }

    }
    C0=vector_mean(prices);
    fout<<C0;
    double stdeerror= (sd(prices,C0)/sqrt(numiter));
    fout<<"("<<stdeerror<<")"<<"\t";
    //cout<<b<<endl;
```

```
    return C0;


}
double DownOutCall_Antithetic(int numiter,double S0,
double volatility, double rate, double time, double timestep,double strike ,
double barrier,ofstream& fout){

    vector<double> prices(numiter/2,0);
    vector<double> paths(timestep+1,0);
    vector<double> antitheticprices(numiter/2,0);
    vector<double> antitheticpaths(timestep+1,0);
    double a=0.0, b=0.0;
    double dt=time/timestep;
    double C0=0.0;
    for(long i=0; i< prices.size(); i++){
        bool t=false;
        double minimum1=S0;
        double minimum2=S0;
        for(long j=0; j <= timestep; ++j){
            if(j==0){
                paths[j]=S0;
                antitheticpaths[j]=S0;
            }
            else{
                double zi= Box_Muller();
                double  b=volatility*sqrt(dt)*zi;
                double a=(rate - 0.5*volatility*volatility)*dt;
                double S1=paths[j-1]*exp(a + b);
                double S2=antitheticpaths[j-1]*exp(a + (-1)*b);
                paths[j]=S1;
                antitheticpaths[j]=S2;
                //cout<<S<<endl;
                if( S1 < minimum1){
                    minimum1=S1;
                }
                if( S2 < minimum2){
                    minimum2=S2;
                }
            }
        }
        if(minimum1 >= barrier){
            prices[i]= exp(-rate*time)*max(paths[timestep-1]-strike,0.0);
        }
        else{
            prices[i]=0.0;
        }
        if(minimum2 >= barrier){
            antitheticprices[i]= exp(-rate*time)*
            max(antitheticpaths[timestep-1]-strike,0.0);
        }
        else{
            antitheticprices[i]=0.0;
        }

    }
```

```
    vector<double> Y(numiter/2,0);
    for(int  z=0;  z< Y.size();z++){
        Y[z]=(antitheticprices[z]+  prices[z])/2.0;

    }
    C0=vector_mean(Y);
    fout<<C0;
    double  stdeerror=  (sd(Y,C0)/sqrt(numiter));
    fout<<"("<<stdeerror<<")"<<"\t";
    //cout<<b<<endl;
    return  C0;


}
/*
 *
 * We use as a control variate a european vanila call.
 */

double  DownOutCall_Control(int  numiter,double  S0,  double  volatility ,
double  rate,  double  time,  double  timestep,double  strike  ,
double  barrier,ofstream&  fout){

    vector<double>  prices(numiter,0);
    vector<double>  paths(timestep+1,0);
    vector<double>  pathseur(numiter,0);
    double  a=0.0,  b=0.0;
    double  dt=time/timestep;
    double  C0=0.0;
    for(long  i=0;  i<  prices.size();  i++){
        bool  t=false;
        double  minimum=S0;
        for(long  j=0;  j  <=  timestep;  ++j){
            if(j==0){
                paths[j]=S0;
            }
            else{
                double  zi=  Box_Muller();
                double   b=volatility*sqrt(dt)*zi;
                double  a=(rate  -  0.5*volatility*volatility)*dt;
                double  S=paths[j-1]*exp(a  +  b);
                paths[j]=S;
                //cout<<S<<endl;
                if(  S  <  minimum){
                    minimum=S;
                }
            }

        }
        pathseur[i]=exp(-rate*time)*max(paths[timestep-1]-strike,0.0);
        if(minimum  >=  barrier){
            prices[i]=  exp(-rate*time)*max(paths[timestep-1]-strike,0.0);
        }
        else{
            prices[i]=0.0;
        }
```

```
    }

    double bcoef=  bcoefficient(prices,pathseur);
    //cout<<"Down and out b "<<bcoef<<endl;
    double bs= BSC(S0,volatility ,rate,time,strike);
    vector<double>  Y(numiter,0);
    for(long j=0;j <numiter;++j){
        Y[j]= prices[j] - (bcoef*(pathseur[j]- bs));

    }
    C0=vector_mean(Y);
    fout<<C0;
    double stdeerror= (sd(Y,C0)/sqrt(numiter));
    fout<<"("<<stdeerror<<")"<<"\t";
    //cout<<b<<endl;
    return C0;


}
double DownInPut_Conditional(int numiter,double S0,
double volatility, double rate, double time, double timestep,double strike ,
double barrier,ofstream& fout){

    vector<double> prices(numiter,0);
    vector<double> paths(timestep+1,0);
    double a=0.0, b=0.0;
    double dt=time/timestep;

    double C0=0.0;
    for(long i=0; i< prices.size(); i++){
        double crossingtime=0.0;
        double crossingvalue=0.0;
        bool t=false;
        for(long j=0; j <=timestep; ++j){
            if(j==0){
                paths[j]=S0;
            }
            else{
                double zi= Box_Muller();
                double  b=volatility*sqrt(dt)*zi;
                double a=(rate - 0.5*volatility*volatility)*dt;
                double S=paths[j-1]*exp(a + b);
                paths[j]=S;
                //cout<<S<<endl;
                if( S <= barrier){
                    t=true;
                    crossingtime=j*dt;
                    crossingvalue=paths[j];
                    break;
                }
            }

        }
        if(t){
            prices[i]=exp(-rate*crossingtime)*
            BSP(crossingvalue,volatility,rate, T, crossingtime, strike);
```

```
                t=false;

            }
            else{
                prices[i]=0.0;
                //cout<<"prices[0] "<<prices[i]<<endl;
            }
        }

        C0=vector_mean(prices);
        fout<<C0;
        double stdeerror= (sd(prices,C0)/sqrt(numiter));
        fout<<"("<<stdeerror<<")"<<"\t";
        //cout<<b<<endl;
        return C0;


    }
    double DownInPut_Control(int numiter,double S0,
    double volatility, double rate, double time, double timestep,double strike ,
    double barrier,ofstream& fout){

        vector<double> prices(numiter,0);
        vector<double> paths(timestep+1,0);
        vector<double> pathseur(numiter,0);
        double a=0.0, b=0.0;
        double dt=time/timestep;
        double C0=0.0;
        for(long i=0; i< prices.size(); i++){
            bool t=false;
            double minimum=S0;
            for(long j=0; j <=timestep; ++j){
                if(j==0){
                    paths[j]=S0;
                }
                else{
                    double zi= Box_Muller();
                    double  b=volatility*sqrt(dt)*zi;
                    double a=(rate - 0.5*volatility*volatility)*dt;
                    double S=paths[j-1]*exp(a + b);
                    paths[j]=S;
                    //cout<<S<<endl;
                    if( S < minimum){
                        minimum=S;
                    }
                }

            }
            pathseur[i]=exp(-rate*time)*max(strike - paths[timestep-1],0.0);
            if(minimum <= barrier){
                prices[i]= exp(-rate*time)*max(strike -paths[timestep-1],0.0);
            }
            else{
                prices[i]=0.0;
            }
        }
```

```
    double bcoef=  bcoefficient(prices,pathseur);
    cout<<"Down and out b "<<bcoef<<endl;
    double bs= BSP(S0,volatility,rate,T,0,strike);
    vector<double>  Y(numiter,0);
    for(long j=0;j <numiter;++j){
        Y[j]= prices[j] - (bcoef*(pathseur[j]- bs));

    }
    C0=vector_mean(Y);
    fout<<C0;
    double stdeerror= (sd(Y,C0)/sqrt(numiter));
    fout<<"("<<stdeerror<<")"<<"\t";
    //cout<<b<<endl;
    return C0;


}
double DownInPut_Antithetic(int numiter,double S0, double volatility,
double rate, double time, double timestep,double strike ,
double barrier,ofstream& fout){

    vector<double> prices(numiter/2,0);
    vector<double> paths(timestep+1,0);
    vector<double> antitheticprices(numiter/2,0);
    vector<double> antitheticpaths(timestep+1,0);
    double a=0.0, b=0.0;
    double dt=time/timestep;
    double C0=0.0;
    for(long i=0; i< prices.size(); i++){
        bool t=false;
        double minimum1=S0;
        double minimum2=S0;
        for(long j=0; j <= timestep; ++j){
            if(j==0){
                paths[j]=S0;
                antitheticpaths[j]=S0;
            }
            else{
                double zi= Box_Muller();
                double  b=volatility*sqrt(dt)*zi;
                double a=(rate - 0.5*volatility*volatility)*dt;
                double S1=paths[j-1]*exp(a + b);
                double S2=antitheticpaths[j-1]*exp(a + (-1)*b);
                paths[j]=S1;
                antitheticpaths[j]=S2;
                //cout<<S<<endl;
                if( S1 < minimum1){
                    minimum1=S1;
                }
                if( S2 < minimum2){
                    minimum2=S2;
                }
            }
        }
        if(minimum1 <= barrier){
```

```
            prices[i]= exp(-rate*time)*
            max(strike - paths[timestep -1],0.0);
        }
        else{
            prices[i]=0.0;
        }
        if(minimum2 <= barrier){
            antitheticprices[i]= exp(-rate*time)*max(strike -antitheticpaths[timestep
        }
        else{
            antitheticprices[i]=0.0;
        }

    }
    vector<double> Y(numiter/2,0);
    for(int z=0; z< Y.size();z++){
        Y[z]=(antitheticprices[z]+ prices[z])/2.0;

    }
    C0=vector_mean(Y);
    fout<<C0;
    double stdeerror= (sd(Y,C0)/sqrt(numiter));
    fout<<"("<<stdeerror<<")"<<"\t";
    //cout<<b<<endl;
    return C0;


}
```