Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université de Carthage

École Supérieure de la Statistique et de l'Analyse de l'Information

Rapport de projet de simulation Monte Carlo

Élaboré par : Nada Akrmi

Table des matières

1 Introduction et Objectif **1**

2 Partie 1 : Méthode par le schéma d'Euler **1**
2.1 Question 1 :Simulation avec la méthode de Monte Carlo Classique 2
2.2 Question 2 : Simulation par la méthode antithétique 2
2.3 Question 3 : Simulation par la méthode de contrôle 3

3 Partie 2 : Méthode par un schéma Sans Biais **4**
3.1 Question1 :Monte Carlo classique 4
3.2 Question 2 : Stratification 4

4 Résultat **5**

5 Conclusion **5**

6 Annexe **6**

1 Introduction et Objectif

Le terme Monte Carlo désigne une famille de méthodes algorithmiques visant à calculer une valeur numérique approchée en utilisant des procédés aléatoires, c'est-à-dire des techniques probabilistes.

Les méthodes de Monte Carlo sont indispensables dans des domaines aussi variés que la finance, les télécommunications, la biologie ou encore les sciences sociales. Elles permettent de résoudre des problèmes centrés sur un calcul à l'aide du hasard

La méthode de simulation de Monte-Carlo permet d'introduire une approche statistique du risque dans une décision financière.

On cherche à estimer $V = E[g(X_t]$ avec X un processus défini par une EDS (équation différentielle stochastique) :

$X_t = x_0 + \int_0^t \mu(X_s)ds + \int_0^t \sigma dW_s$

avec

$\mu(x) = 0.1(\sqrt{\exp x - 1} - 1/8)$

$W = (W_t)$ un mouvement brownien standard

$g(x) = (\exp X_t - K)_+$

Le but de ce projet est d'évaluer V avec plusieurs méthodes de simulation et de les comparer pour choisir la meilleure avec le maximum de précision.

on prend les constantes suivantes :

$x_0 = 0$

$\sigma = 0.5$

$K = 1$

$T = 1$

2 Partie 1 : Méthode par le schéma d'Euler

La méthode d'Euler, nommée ainsi en l'honneur du mathématicien Leonhard Euler, est une procédure numérique pour résoudre par approximation des équations différentielles du premier ordre avec une condition initiale. C'est la plus simple des méthodes de résolution numérique des équations différentielles.

On considère les variables suivantes :

n= 10
$\Delta = T/n = 0.1$
$t_k = k\Delta = k/10$

On définit $X_{t_0}^{\Delta} = x_0 = 0$

et $X_{t_{k+1}}^{\Delta} = X_{t_k}^{\Delta} + \mu(X_{t_k}^{\Delta})\Delta + \sigma(W_{t_{k+1}} - W_{t_k})$

2.1 Question 1 :Simulation avec la méthode de Monte Carlo Classique

La première méthode qu'on va utiliser c'est la méthode de Monte Carlo classique.

- On génère un mouvement brownien W comme une loi normale d'espérenece nulle et de variance $= t_k$ avec un accroissement.

- On génère les $X_{t_k}^{\Delta}$ suivant la formule donnée.

- On calcule l'estimateur de Monte Carlo standard MMC : $S_n = \frac{1}{n}\sum\limits_{k=1}^{n} g(X_{t_k}^{\Delta})$

- Cet estimateur converge P presque sûrement vers $V = E[g(X_{T)}] = E[g(X_1)]$ selon la loi forte des grands nombres. C'est un estimateur sans biais de V.

- On calcule à la fin la variance de S_n : $var(S_n) = \frac{var(g(X_1))}{n}$

2.2 Question 2 : Simulation par la méthode antithétique

On génère $(W_{t_0}, W_{t_1}, ..., W_{t_n})$ un mouvement brownien sur la grille $t_0 < t_1 < ... < t_n$ => $(-W_{t_0}, -W_{t_1}, ..., -W_{t_n})$ est aussi un mouvement brownien sur cette grille

Les 2 mouvements suivent la même loi.

On génère par la suite

$$X_{1t_{k+1}}^{\Delta} = X_{1t_k}^{\Delta} + \mu(X_{1t_k}^{\Delta})\Delta + \sigma(W_{t_{k+1}} - W_{t_k})$$

et

$$X_{2t_{k+1}}^{\Delta} = X_{2t_k}^{\Delta} + \mu(X_{2t_k}^{\Delta})\Delta - \sigma(W_{t_{k+1}} - W_{t_k})$$

On pose $\tilde{S_n} = \frac{1}{2n}\sum\limits_{k=1}^{n}(g(X_{1t_k}^{\Delta}) + g(X_{2t_k}^{\Delta}))$ l'estimateur par la méthode antithétique.

On calcule l'espérance de cet estimateur.

$$E(\tilde{S_n}) = \frac{1}{2n}\sum_{k=1}^{n}(E(g(X_{1t_k}^{\Delta})) + E(g(X_{2t_k}^{\Delta})))$$

$$= \frac{1}{2n}\sum_{k=1}^{n}(E(g(X_{1t_1}^{\Delta})) + E(g(X_{2t_1}^{\Delta})))$$

$$= \frac{1}{2n}\sum_{k=1}^{n}(E(g(X_{1t_k}^{\Delta})) + E(g(X_{2t_k}^{\Delta})))$$

converge P presque $\frac{1}{2n}\sum\limits_{k=1}^{n}(E[g(X_1)]) + E[g(X_1)]))$

$= \frac{1}{2n} 2n(E[g(X_1)])$

$= V$

On calcule à la fin la variance de cet estimateur :

$var(\tilde{S_n}) = \frac{1}{4n^2} \sum_{k=1}^{n} (var(g(X^{\Delta}_{1t_k})) + var(g(X^{\Delta}_{2t_k})))$

$= \frac{1}{4n} var(g(X_{11}) + g(X_{21}))$

$= \frac{1}{4n} var(g(X_{11})) + var(g(X_{21})) + 2cov(g(X_1), g(X_2))$

$cov(g(X_{11}), g(X_{11})) \leq 0$ car les 2 fonctions monotones ont un sens de monotonie différents.

$==> var(\tilde{S_n}) \leq \frac{1}{2n} var(g(X_1) \leq var(S_n)$

2.3 Question 3 : Simulation par la méthode de contrôle

Étape 1 :On génère la variable de contrôle

On calcule $m = E[g(\tilde{X_T})]$ avec $\tilde{X_T} = \sigma W_t$

On génère W comme on a déjà fait pour les 2 dernières questions.

Étape 2 : On génère l'estimateur par la méthode de Monte Carlo classique X_3

Étape 3 :On calcule $\alpha = \frac{cov(\tilde{X_T}, X_{3t})}{var(\tilde{X_T})}$

Étape 4 : On calcule finalement $Y_t = X_{3t} - \alpha(\tilde{X_T} - m)$

- L'estimateur par la méthode de contrôle $\bar{S_n} = \frac{1}{n} \sum_{k=1}^{n} Y_{t_k}$
On calcule l'espérence de cet estimateur

$E(\bar{S_n}) = \frac{1}{n} \sum_{k=1}^{n} E(Y_{t_k})$

$= \frac{1}{n}(\sum_{k=1}^{n} E(g(X_{3t_k})) - \alpha \sum_{k=1}^{n} (E(g(\tilde{X_{t_k}})) - m))$

$= \frac{1}{n} \sum_{k=1}^{n} E(g(X_{3t_k}))$
converge presque sûrement vers $V = E[g(X_{T)}] = E[g(X_1)]$

On calcule à la fin la variance de cet estimateur

$var(\bar{S_n}) = \frac{var(g(Y_1))}{n}$

$= \frac{var(g(X_{31})-\alpha(g(\tilde{X_1})-m))}{n}$

$= \frac{var(g(X_{31})-\alpha g(\tilde{X_1})}{n} \leq var(S_n).$

3 Partie 2 : Méthode par un schéma Sans Biais

3.1 Question1 :Monte Carlo classique

Soit $(\tau_k)_{k\geq 1}$ une suite i.i.d de variables aléatoires de loi exponentielle de paramètre β (qu'on a pris 6 assez grande pour ne par sortir de la boucle rapidement), indépendantes du mouvement brownien W.

On a simulé cette suite avec la méthode d'inversion :
On a genèré une loi uniforme U ([0,1]) telle que les τ_k sont égles à $-log(1-U)/Beta$

On définit

$T_0 = 0$ et $T_k = (\sum_{i=1}^{k} \tau_i) \wedge T$

$N_t = maxk : T_k < T$ avec $T_{N_T} < T$ et $T_{n+1} = T$

Pour générer les T_k on a sommé les τ_k jusqu'à atteidre T=1, notre compteur s'arrête à p alors $N_t = p-1$ et $T_k = T = 1$.

$\Delta T_k = T_k - T_{k-1}$

$\Delta W_k = W_{T_k} - W_{T_{k-1}}$

$X_{T_0} = X_0 = x_0 = 0$

et $X_{t_{k+1}} = X_{t_k} + \mu(X_{t_k})\Delta T_{k+1} + \sigma \Delta W_{k+1}$

$$\psi := e^{\beta T}\Big(g(\widehat{X}_{T_{N_T+1}}) - g(\widehat{X}_{T_{N_T}})\mathbf{1}_{N_T>0}\Big)\prod_{k=1}^{N_T}\frac{\big(\mu(\widehat{X}_{T_k}) - \mu(\widehat{X}_{T_{k-1}})\big)\Delta W_{k+1}}{\sigma\beta\Delta T_{k+1}}$$

On a d'après l'énoncé $E(\psi) = V$

3.2 Question 2 : Stratification

On applique la méthode de statification surdes variables discrètes (N_t) qui suivent la loi de poisson.
L'estimateur proposé par la méthode de stratifcation est donné par :

$E(\psi) = E(\psi 1(\cup N_t = n)) = \sum E(\psi 1(N_t = n) = E(\psi / N_t = n).P(N_t)$

Pour estimer V=$E(\psi)$ on génère U_k sont iid suivent la loi uniforme entre 0 et t, tel que

$$L(T_1, ..., T_n/N_t = n) = L(U_1, U_2, .., U_n)$$

avec
$U_1 = min(U_1, U_2, .., U_n)$
et
$U_n = max(U_1, U_2, .., U_n)$

4 Résultat

Après avoir implémenté les méthodes citées ci- dessus, on a trouvé les résultats suivants.

RStudio: Notebook Output

	E	Var	temps
Monte Carlo classique	0.05826868	0.0083231003	30.5585220
Antithétique	0.04197985	0.0018170453	0.9546969
Métode de controle	0.05826868	0.0003496462	1.9379790
schema Sans Biais :Monte-Carlo simple	0.05826868	0.0001234492	32.9513273

Figure 1 – Résultat

Ce tableau présente l'espérance , la variance et le temps d'exécution obtenus pour chaque estimateur.

5 Conclusion

L'évaluation de V par les différentes méthodes a donné des bons résultats.

Le meilleur estimateur est celui obtenu par la méthode par un schéma sans biais par l'estimateur de Monte Carlo classique (variance minimale) mais avec le plus grand temps d'exécution.

L'estimation par la méthode de schéma d'Euler par la méthode antithétique a donné un temps d'exécution minimale.

La méthode de schéma d'Euler avec l'estimateur de Monte Carlo classique a donné la plus grande variance avec un temps d'exécution un peu grand.

6 Annexe

```
Méthode par le schéma d'Euler
Déclaration des variables
Question 1
x0=0
sigma=0.5
k=1
T=1
n=10
delta= T/n
t= k/10
deltaX=c(rep(0,n+1))
g=function(x)
gx=max(exp(x)-k,0)
return(gx)
mu=function(x)
ux=0.1*(sqrt(exp(x))-1)-1/8
return(ux)
simulation de W mouvement brownien
W = rnorm(1000,sd=sqrt(0.1))
dW=cumsum(W )
temps1=Sys.time()
X<-c(rep(0,11))
X[0]=0
for (j in 1 :11)
X[j+1]=X[j]+delta*mu(X[j])+sigma*(dW[j+1]-dW[j])
simuler V
V1=c((rep(0,11)))
for (i in 0 :11)
V1[i]=max(g(X[i]),0)
V=mean(V1)
V
Variance
var1=sd(V1)*sd(V1)
var1
Temps d'éxecution
t1= Sys.time()-temps1
Question2
temps2=Sys.time()
Xt1=c(rep(0,11))
Xt2=c(rep(0,11))
X1<-c(rep(0,11))
X2<-c(rep(0,11))
X1[0]=0
X2[0]=0
for(j in 1 :11)
X1[j+1]=X1[j]+delta*mu(X1[j])+sigma*(dW[j+1]-dW[j])
X2[j+1]=X2[j]+delta*mu(X2[j])-sigma*(dW[j+1]-dW[j])
X2
Vt1=c(rep(0,11))
Vt2=c(rep(0,11))
```

```
for (i in 0 :11)
Vt1[i]=max(g(X1[i]),0)
for (i in 0 :11)
Vt2[i]=max(g(X2[i]),0)
G=0.5*(Vt1+Vt2 )
G
V2=mean(G)
variance
var2=sd(G)*sd(G )
var2
t2=Sys.time()-temps2
var1
Question3
temps3=Sys.time()
Xt3
Xt3=c(rep(0,11))
X3<-c(rep(0,11))
X3[0]=0
for(j in 1 :11)
X3[j+1]=X3[j]+delta*mu(X3[j])+sigma*(dW[j+1]-dW[j])
X3
Vt3=c(rep(0,11))
for (i in 0 :11)
Vt3[i]=max(g(X3[i]),0)
Vt3
Variable de contrôle
Xttild=c(rep(0,11))
for(i in 0 :11)
Xttild[i]= sigma*dW[i]
Xttild
Xtild=c(rep(0,11))
for (i in 0 :11)
Xtild[i]=max(exp( Xttild[i])-1,0)
Xtild
m=mean(Xtild)
m
méthode de contrôle
alpha=cov(Xtild,Vt3)/var(Xtild)
alpha
Y=Vt3-alpha*(Xtild-m)
Y
Ym=mean(Y)
Ym
var3=var(Y)
var3
t3=Sys.time()- temps3
var2
2eme partie
temps4=Sys.time()
Mc=100
Beta=6
```

```
U=runif(Mc)
E=-log(1-U)/Beta
génerer T
Tc=c(rep(0,Mc+1))
j=1
c=0
Tk1=c(rep(0,11))
while((c < 1) )
c=c+E[j]
Tk1[j]=c
j=j+1
Tk1
c
if (c>=1)
Tk1[j-1]=1
T=Tk1[1 :j]
Nt=j-1
T
Nt
générer les delta T
Y=c(rep(0,Nt))
for (i in 1 : Nt)
Y[i]=T[i+1]-T[i]
Y
Générer les X chapeau
Xchap_T=c(rep(0,11))
Xchap=c(rep(0,11))
for (j in 1 :(Nt-1))
Xchap[j+1]=Xchap[j]+Y[j+1]*mu(Xchap[j])+sigma*(dW[j+1]-dW[j])
Xchap
générer psi
p=1
for (i in 2 :(Nt-1)) p=p*(mu(Xchap[i])-mu(Xchap[i-1]))*(dW[i+1]-dW[i])/(0.5*Beta*Y[i+1])
p
z=1
for (i in 2 :(Nt-1))
z=p*(exp(Beta*T)*( g(Xchap[i])-g(Xchap[i-1])))
V_chap=mean(z)
var_chap=var(z)
t4=Sys.time()- temps4
E=c(V,V2,Ym,v_chap)
Var=c(var1, var2 ,var3 ,var_chap)
temps=c(t1,t2,t3,t4)
tabl<-cbind(E ,Var,temps)
rownames(tabl)<-c(”Monte Carlo classique” , ”Antithétique”, ”Métode de controle” ,”schema Sans Biais :Monte-Carlo simple”)
tabl
```