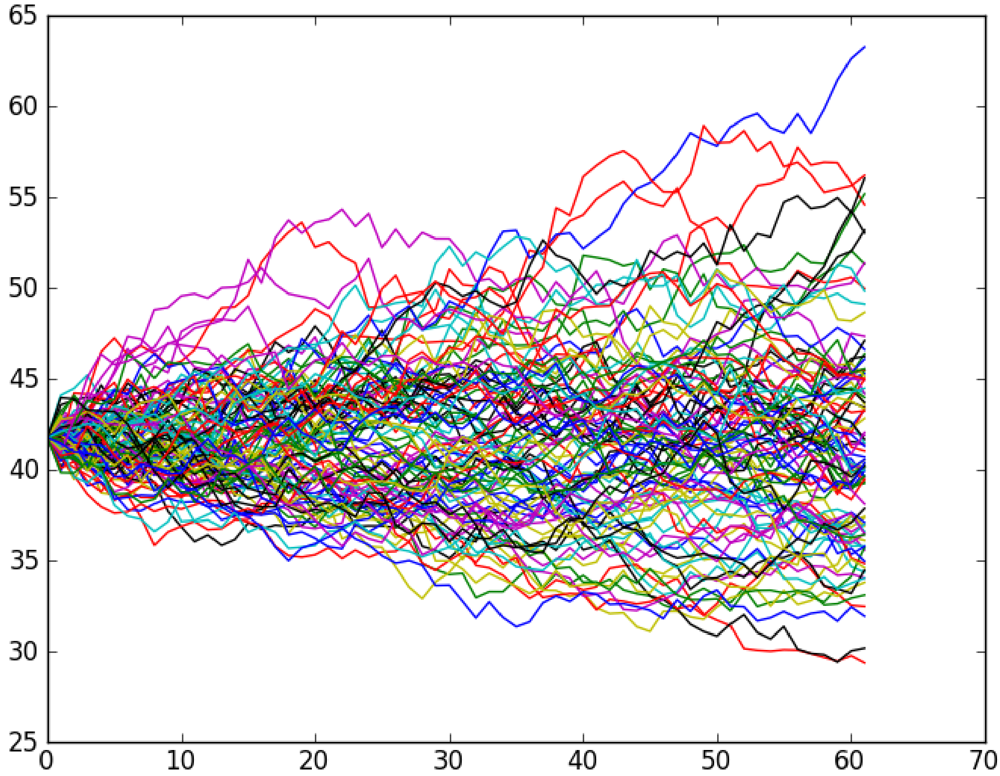
Méthodes de Monte-Carlo pour la Finance

Devoir 1 : Simulation des Objets Aléatoires Fondamentaux



Adnane EL KASMI (ING2 MF01)
à l'attention de : Irina Kortchemski

Table des matières

Devoir 1: Simulation des Objets Aléatoires Fondamentaux

I Introduction

A Présentation

Dans un modèle stochastique, les prix des actifs sont représentés comme la solution d'une équation différentielle stochastique. Le problème essentiel posé les mathématiques financières est d'évaluer les produits dérivés sur ces actifs. Dans la plupart des cas, le payoff de ces produits est donné par une fonction de l'actif sousjacent à une (ou des) date(s) future(s). Le prix, dans un modèle complet, est alors l'espérance sous l'unique probabilité risque neutre du payoff actualisé. L'intérêt numérique est donc de calculer cette espérance de la manière la plus efficace et la plus rapide possible.

B Objectifs

Connaître les principales techniques de simulations utilisées en finance pour valoriser les produits dérivés et estimer les risques liés à leur couverture ou à la gestion d'un portefeuille.

C Exemple : Simulateur de portefeuille par Monte-Carlo

Avec un simulateur de portefeuille par Monte-Carlo, vous pouvez facilement créer, modifier et supprimer vos portefeuilles et simuler leurs risques et rendements attendus. Nous prenons des rendements historiques quotidiens pour les 10 dernières années et les projetons pour un an à l'avenir :

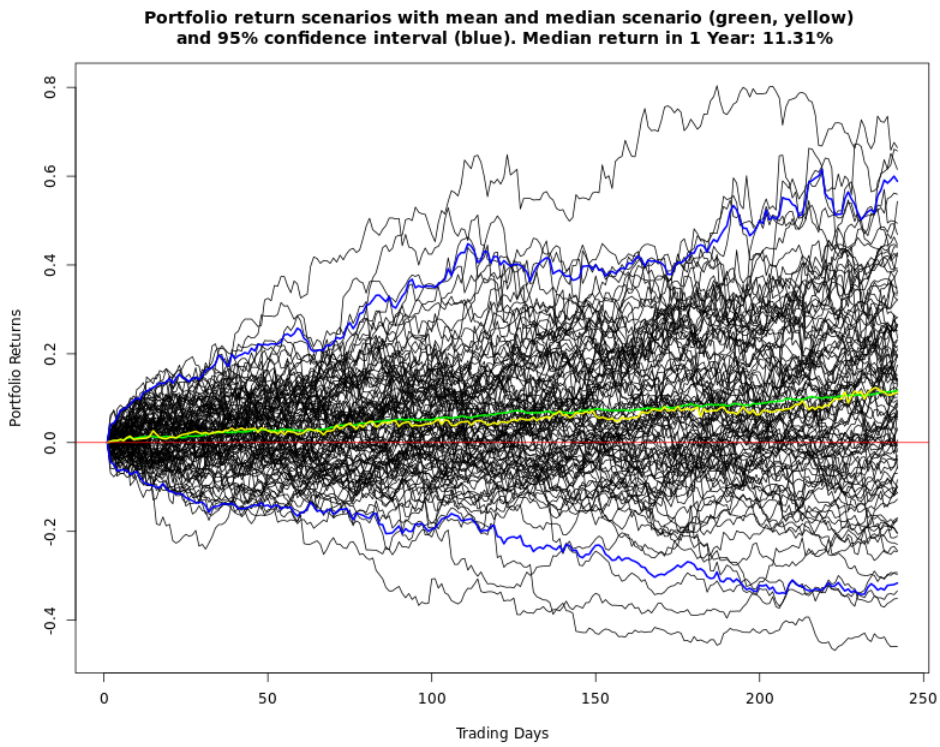


FIGURE 1 – Portfolio Simulator - estimate the expected risk and return of your investments.

II Arbre Binomial

A Modèle binomial

En finance, le modèle binomial (ou modèle CRR du nom de ses auteurs) fournit une méthode numérique pour l'évaluation des options. Il a été proposé pour la première fois par Cox, Ross et Rubinstein (1979). Le modèle est un modèle discret pour la dynamique du sous-jacent. L'évaluation de l'option est calculée par application de la probabilité risque-neutre pour laquelle les prix actualisés sont des martingales.

B Utilisation du modèle

La méthode binomiale, pour valoriser les options, est très largement utilisée car elle est capable de prendre en compte un nombre important de conditions pour lesquelles l'application d'autres modèles n'est pas aisée. Cela vient en grande partie du fait que la méthode binomiale prend en compte les variations de l'actif sous-jacent (contrairement aux autres méthodes qui ne prennent en compte qu'un point fixe). Par exemple la méthode binomiale est utilisée pour les options américaines (celles-ci peuvent être exercées à tout moment) et les options Bermudiennes (celles-ci peuvent être exercées à différents moments). La méthode binomiale est de plus mathématiquement relativement simple et peut être facilement programmée en logiciel (ou éventuellement sur une feuille de calcul). Bien que plus lente que la méthode de Black-Scholes, la méthode binomiale est considérée comme plus précise, particulièrement pour les options à long terme et les options sur titre versant des dividendes. C'est pourquoi il existe plusieurs versions du modèle binomial qui sont utilisées par les personnes travaillant sur le marché des options. Pour les options comportant plusieurs sources d'incertitudes (par exemple les options réelles) ou pour les options complexes (par exemple les options asiatiques) l'application de la méthode binomiale en « arbre » présente des difficultés et n'est pas pratique. Dans ces cas-là il vaut mieux utiliser la Méthode de Monte-Carlo.

C Définition de l'Arbre Binomial

L'Arbre Binomial est un processus stochastique discret qui modélise l'évolution d'un actif au cours du temps. L'intervalle [0,T] est discrétisé : $t_n = \Delta t.n$. A chaque instant t_n on fait correspondre le prix S_n discret : $t_n \to S_n$ et $t_0 \to S_0$.

L'idée principale de la méthode Binômiale : $S_{n+1} = \begin{cases} u.\mathrm{S}_n \text{ avec probabilité p} \\ d.\mathrm{S}_n \text{ avec probabilité (1-p)} \end{cases}$

avec u>1 et d<1.

D L'arbre Binomial pour l'actif

Dans cette partie, on va présenter l'évolution possible d'un actif sur une durée de 20 périodes.

→ Simulation de l'Arbre Binomial pour l'Actif sous MATLAB :

On va prendre $u = 1.1$, $d = 0.9$, $S_0 = 10$ et $N = 20$. Le code sous MATLAB :

```
u=1.1;
d=0.9;
S0=10;
N=20;

ArbreBinomialActif(N,S0,u,d);

function [arbre_binomial_actif] = ArbreBinomialActif(N,S0,u,d)
for n=1:N+1
    for i=1:n
        S(n,i)=S0*(u^(i-1))*(d^(n+1-i));
    end
end
plot(S,'*')
xlabel 't_n'
ylabel 'S_n'
title 'Simulation de Arbre Binomial pour Actif'
end
```

FIGURE 2 – Code Sous MATLAB du fichier SimulationArbreBinomialActif.m

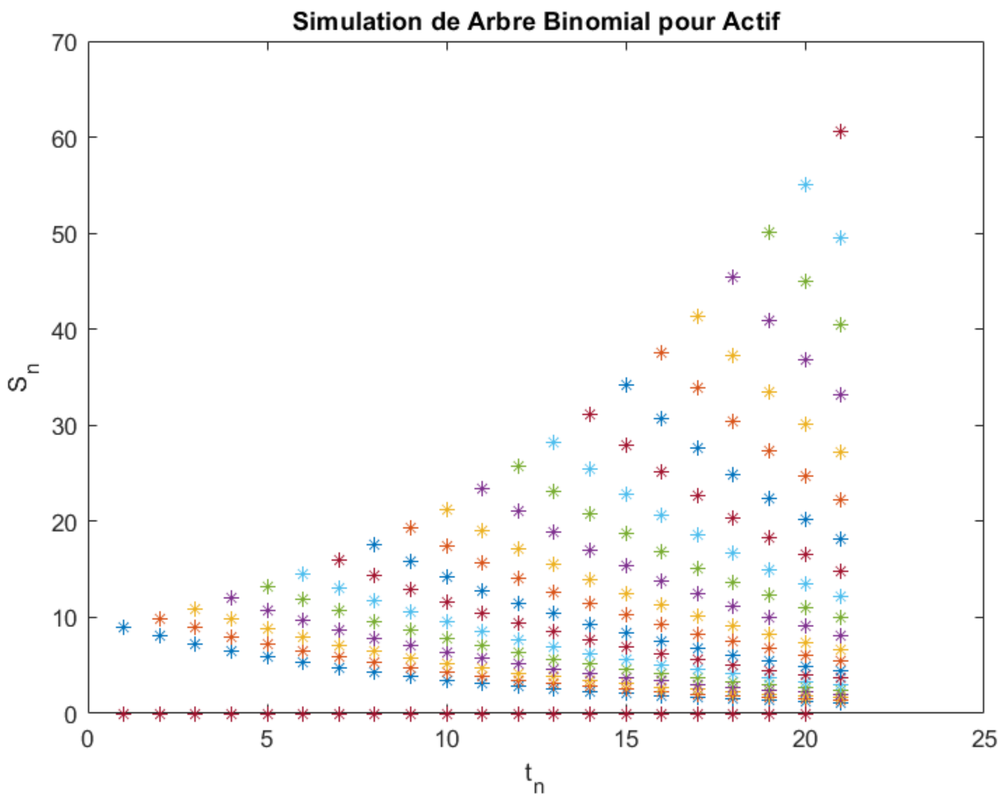


FIGURE 3 – Graphe sous MATLAB de Simulation de l'Arbre Binomial pour l'Actif.

E Evolution d'un actif dans l'Arbre Binomial

L'actif est une chaine de Markov.

→ Simulation d'évolution d'un actif dans l'Arbre Binomial sous MATLAB :

On va prendre $u = 1.1$, $d = 0.9$, $S_0 = 10$, $N = 20$ et $p = 0.4$. Le code sous MATLAB :

```
p=0.4;
u=1.1;
d=0.9;
S0=10;
N=20;

EvolutionActifArbre(N,S0,u,d,p);

function [evolution_actif_arbre] = EvolutionActifArbre(N,S0,u,d,p)
actif(1)=S0;
for n=1:N+1
    if rand<p
        actif(n+1)=u*actif(n);
    else
        actif(n+1)=d*actif(n);
    end
    for i=1:n
        S(n,i)=S0*(u^(i-1))*(d^(n+1-i));
    end
end
hold on;
plot(S,'*');
plot(actif)
xlabel 'Temps discret'
ylabel 'Valeurs de actif S'
title 'Simulation Evolution Actif dans Arbre Binomial'
end
```

Figure 4 – Code Sous MATLAB du fichier EvolutionActifArbreBinomial.m

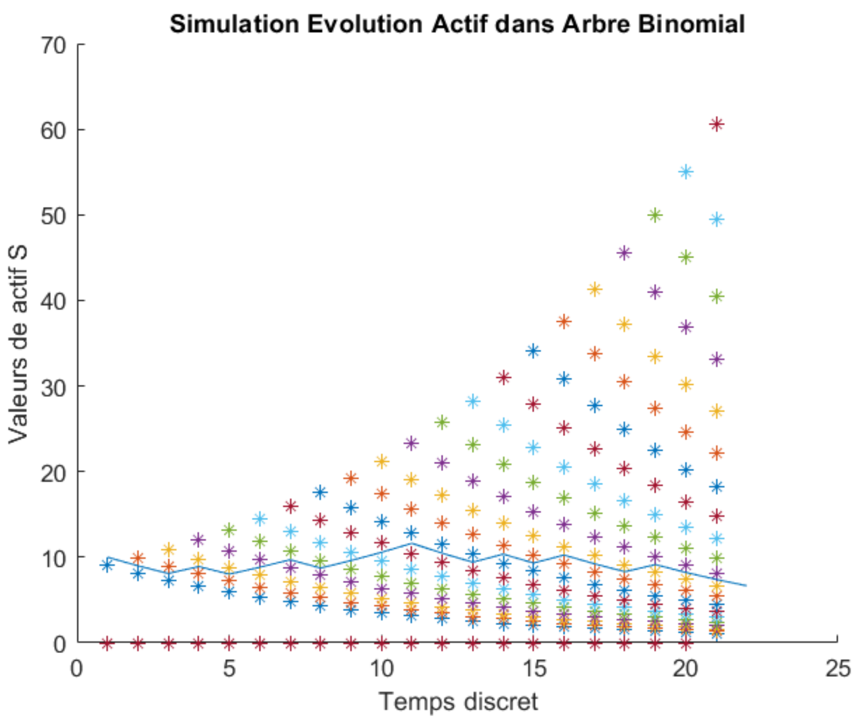


Figure 5 – Graphe sous MATLAB de Simulation d'évolution d'un actif dans l'Arbre Binomial.

III Démonstration Monte-Carlo Martingale pour une marche aléatoire

A Définition d'une Martingale

Soit $(\mathcal{F}_n)_{n\geq 0}$ une filtration et $(X_n)_{n\geq 0}$ un processus.

*$(X_n)_{n\geq 0}$ est une **martingale** (ou $(\mathcal{F}_n)$-martingale si on veut préciser la filtration) si*

1. *pour tout $n \geq 0$, X_n est $\mathcal{F}_n$-mesurable (le processus est adapté),*
2. *pour tout $n \geq 0$, $X_n \in \mathrm{L}^1$ (le processus est intégrable),*
3. *pour tout $n \geq 0$, $\mathrm{E}(X_{n+1}|\mathcal{F}_n) = X_n$.*

B Définition d'une Marche Aléatoire sur Z

Soit $(X_n)_{n\geq 1}$ une suite de variables aléatoires i.i.d. de loi $\mathbb{P}(X_n = -1) = \mathbb{P}(X_n = 1) = \frac{1}{2}$. Pour $n \geq 0$, $M_n = \sum_{k=1}^{n} X_k$. La suite $(M_n)_{n\geq 0}$ modélise une marche aléatoire centrée sur Z sous la condition initiale $M_0 = 0$.

C La Marche Aléatoire $(M_n)_{n\geq 0}$ est une Martingale

C.1 Démonstration Mathématique

On note $\mathcal{F}_n = \sigma(X_k | k \leq n)$ la filtration naturelle de $(X_n)_{n\geq 1}$.

1) Pour $n \geq 0$, $M_n = f(X_1, .., X_n)$ avec $f(x_1, ..., x_n) = \sum_{k=1}^{n} x_k$ $\mathcal{F}_n$-mesurable, alors $M_n \in \mathcal{F}_n$ c'est à dire que M_n est adapté à $\mathcal{F}_n$.

2) Pour $n \geq 0$, $\mathbb{E}[|M_n|] \leq \sum_{k=1}^{n} \mathbb{E}[|X_k|] \leq n < +\infty$ puisque $\mathbb{E}[|X_k|] = 1.\frac{1}{2} + 1.\frac{1}{2} = 1$, alors M_n est integrable.

3) Pour $n \geq 0$, $\mathbb{E}[M_{n+1}|\mathcal{F}_n] = \mathbb{E}[M_n + X_{n+1}|\mathcal{F}_n]$ puisque $M_n \in \mathcal{F}_n$ et X_{n+1} est indépendante de $\mathcal{F}_n$ alors : $\mathbb{E}[M_{n+1}|\mathcal{F}_n] = M_n + \mathbb{E}[X_{n+1}]$ or $\mathbb{E}[X_{n+1}] = -1.\frac{1}{2} + 1.\frac{1}{2} = 0$,
donc : Pour $n \geq 0$, $\mathbb{E}[M_{n+1}|\mathcal{F}_n] = M_n$.

On a donc $(M_n)_{n\geq 0}$ est une Martingale.

C.2 Démonstration par Simulation Monte-Carlo sous MATLAB

D'après la Loi des Grands Nombres : $M_k = \mathbb{E}[M_n|\mathcal{F}_k] = \frac{1}{N_{mc}} \sum_{p=1}^{N_{mc}} M_n^{(p)}$.

$\rightarrow$ Simulation Monte-Carlo de la Martingale Marche Aléatoire sous MATLAB :

```
k=100;
n=300;
Nmc=1000;

[esperance_Mn,M_Fk] = martingale(k,n,Nmc);

disp(esperance_Mn);
disp(M_Fk);

function [esperance_Mn,M_Fk] = martingale(k,n,Nmc)
[M_Fk,M] = processus_M(k);
figure
for j=1:Nmc
for i=k:(n-1)
M(i+1)=M(i)+pas();
end
last_value(j)=M(n);
hold on;
plot(M,'LineWidth',2)
xlabel 't_n'
ylabel 'Processus M_n'
title 'Simulation Monte Carlo de la Marche Aléatoire)'
end
esperance_Mn=sum(last_value)/Nmc;
end

function [M_Fk,M] = processus_M(k)
M(1)=0;
for i=1:k-1
M(i+1)=M(i)+pas();
end
hold on
plot(M,'black','LineWidth',2);
xlabel 't_n'
ylabel 'Processus M_n'
title 'La Marche Aléatoire sur Z'
M_Fk=M(k);
end

function [step] = pas()
if rand < 1/2
        step=1;
    else
        step=-1;
end
end
```

FIGURE 6 – Code Sous MATLAB du fichier SimulationMarcheAleatoire.m

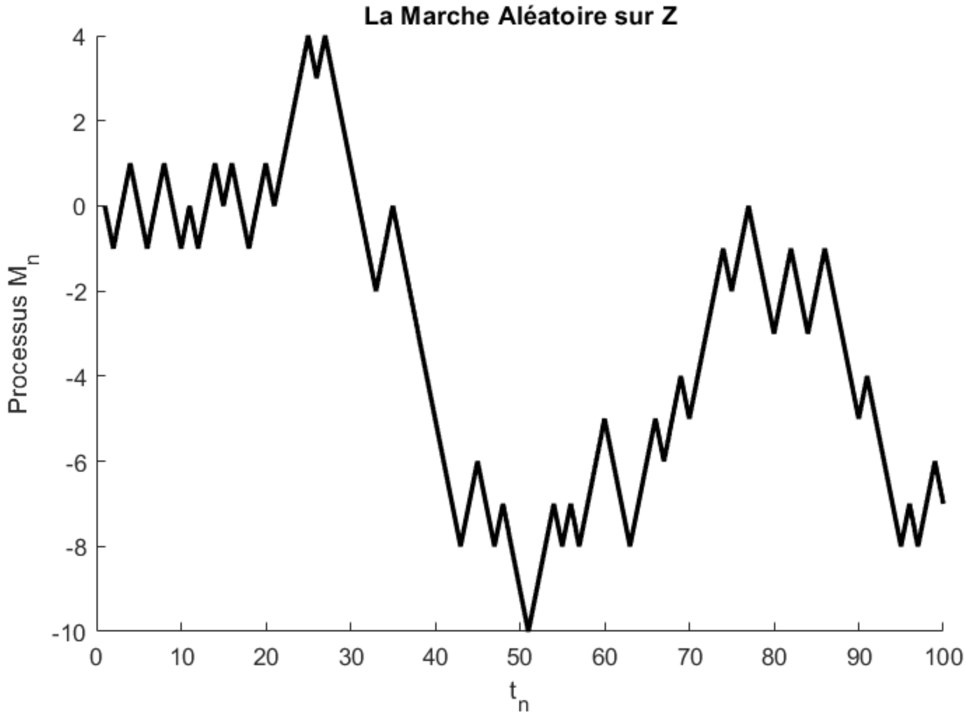


FIGURE 7 – Marche Aléatoire sur Z

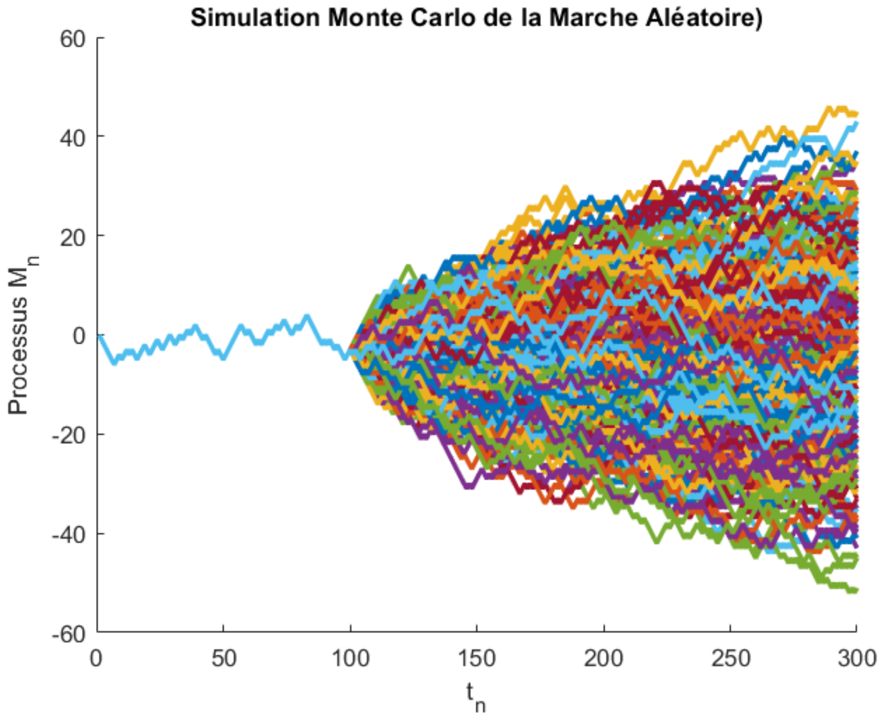


FIGURE 8 – Simulation Monte-Carlo Martingale Marche Aléatoire sur Z.

Par simulation de Monte-Carlo, on vérifie que la moyenne des N_{mc} valeurs à l'instant $n = 300$ est bien égale à la valeur en $n = 100$. Cette valeur tend vers 0 qui est la valeur de l'ésperance de cette Martingale $(M_n)_{n\geq 0}$ puisque $(\mathbb{E}[M_n] = \mathbb{E}[M_0] = 0)$.

D Le processus $(A_n)_{n\geq 0} = (M_n^2 - n)_{n\geq 0}$ est une Martingale

D.1 Démonstration Mathématique

On note $\mathcal{F}_n = \sigma(X_k|k \leq n)$ la filtration naturelle de $(X_n)_{n\geq 1}$ et $A_n = M_n^2 - n$.

1) Pour $n \geq 0$, $A_n = f(X_1, .., X_n)$ avec $f(x_1, ..., x_n) = (\sum_{k=1}^n x_k)^2 - n$ $\mathcal{F}_n$-mesurable, alors $A_n \in \mathcal{F}_n$ c'est à dire que A_n est adapté à $\mathcal{F}_n$.

2) Pour $n \geq 0, \mathbb{E}[|A_n|] = \mathbb{E}[|M_n^2 - n|] \leq \mathbb{E}[M_n^2] + n$ or $\mathbb{E}[M_n^2] = \mathbb{E}[(\sum_{k=1}^n X_k)^2] = \mathbb{E}[\sum_{k=1}^n X_k^2 + 2\sum_{1\leq i<j\leq n} X_i X_j]$ or pour $i < j$ on a par indépendance $\mathbb{E}[X_i X_j] = \mathbb{E}[X_i]\mathbb{E}[X_i] = 0$, on a aussi $\mathbb{E}[X_k^2] = 1$ alors par linéarité de l'intégrale : $\mathbb{E}[|A_n|] = \mathbb{E}[|M_n^2 - n|] \leq \mathbb{E}[M_n^2] + n \leq 2n < +\infty$.
Alors Pour $n \geq 0$, A_n est integrable.

3) Pour $n \geq 0$, $\mathbb{E}[A_{n+1}|\mathcal{F}_n] = \mathbb{E}[M_{n+1}^2 - (n+1)|\mathcal{F}_n] = \mathbb{E}[(M_n + X_{n+1})^2 - (n+1)|\mathcal{F}_n] = \mathbb{E}[(M_n^2 - n) + X_{n+1}^2 + 2M_n X_{n+1} - 1|\mathcal{F}_n]$ puisque $M_n^2 - n, M_n, 1 \in \mathcal{F}_n$ et X_{n+1} est indépendante de $\mathcal{F}_n$ alors : $\mathbb{E}[A_{n+1}|\mathcal{F}_n] = A_n + \mathbb{E}[X_{n+1}^2] + 2M_n\mathbb{E}[X_{n+1}] - 1 = A_n + 1 + 0 - 1 = A_n$ car $\mathbb{E}[X_{n+1}] = -1.\frac{1}{2} + 1.\frac{1}{2} = 0$ et $\mathbb{E}[X_{n+1}^2] = 1.\frac{1}{2} + 1.\frac{1}{2} = 1$,
donc : Pour $n \geq 0$, $\mathbb{E}[A_{n+1}|\mathcal{F}_n] = A_n$.

On a donc $(A_n)_{n\geq 0}$ c'est à dire $(M_n^2 - n)_{n\geq 0}$ est une Martingale.

C.2 Démonstration par Simulation Monte-Carlo sous MATLAB

D'après la Loi des Grands Nombres : $A_k = \mathbb{E}[A_n|\mathcal{F}_k] = \frac{1}{N_{mc}} \sum_{p=1}^{N_{mc}} A_n^{(p)}$.
D'autre part : Pour $n \geq 0$: $A_0 = M_0^2 - 0 = 0$ et $A_{n+1} = A_n + X_{n+1}^2 + 2M_n X_{n+1} - 1$

$\rightarrow$ Simulation Monte-Carlo de la Martingale Marche Aléatoire sous MATLAB :

Par simulation de Monte-Carlo, on vérifie que la moyenne des N_{mc} valeurs à l'instant $n = 300$ est bien égale à la valeur en $n = 100$. Cette valeur tend vers 0 qui est la valeur de l'ésperance de cette Martingale $(A_n)_{n\geq 0} = (M_n^2 - n)_{n\geq 0}$ puisque $(\mathbb{E}[A_n] = \mathbb{E}[A_0] = \mathbb{E}[M_0^2 - 0] = 0)$.

```matlab
k=100;n=300;Nmc=2000;
martingale(Nmc,n,k)

function martingale(Nmc,n,k)
A=zeros(1,n+1);
M=zeros(1,n+1);
somme=0;
figure;
for i=1:k-1
M(i+1)=M(i)+pas();
A(i+1)=M(i+1)^2-(i+1);
end
for j=1:Nmc
for i=k:n
M(i+1)=M(i)+pas();
A(i+1)=M(i+1)^2-(i+1);
end
somme=somme+A(n+1);
hold on;
plot(A,'LineWidth',2)
xlabel 't_n'
ylabel 'Processus A_n'
title 'Simulation Monte Carlo de A_n=(M_n^2-n)'
end
end

function [step] = pas()
if rand < 1/2
        step=1;
    else
        step=-1;
end
end
```

FIGURE 9 – Code Sous MATLAB du fichier SimulationProcessusAn.m

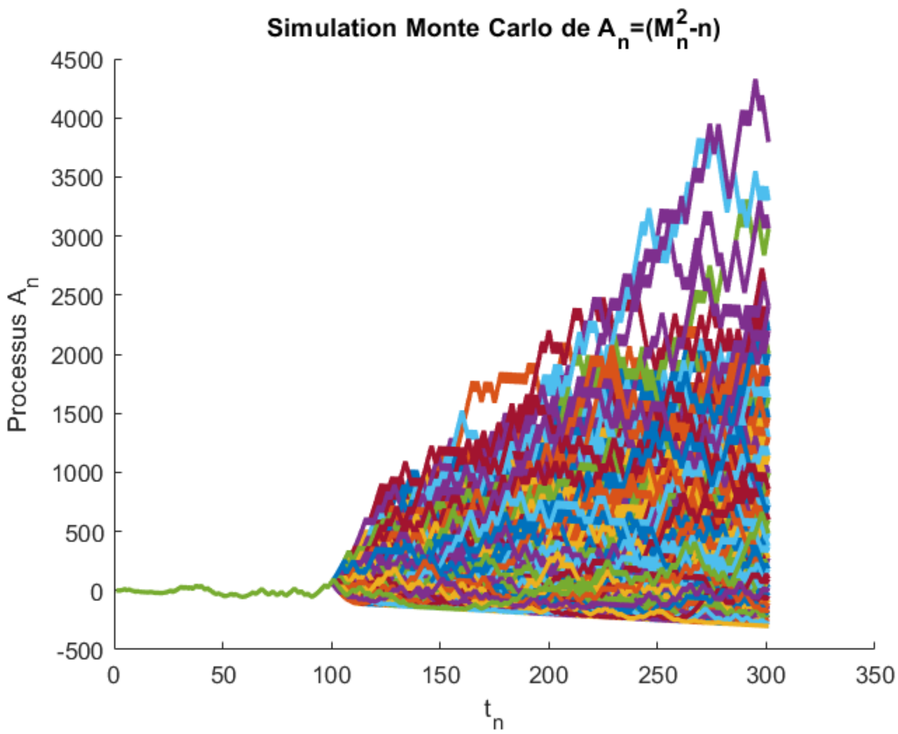


FIGURE 10 – Simulation Monte-Carlo Martingale $(A_n)_{n\geq 0} = (M_n^2 - n)_{n\geq 0}$.

IV Mouvement Brownien

A Définition d'un Mouvement Brownien

Le mouvement brownien, ou processus de Wiener, est une description mathématique du mouvement aléatoire d'une « grosse » particule immergée dans un fluide et qui n'est soumise à aucune autre interaction que des chocs avec les « petites » molécules du fluide environnant.

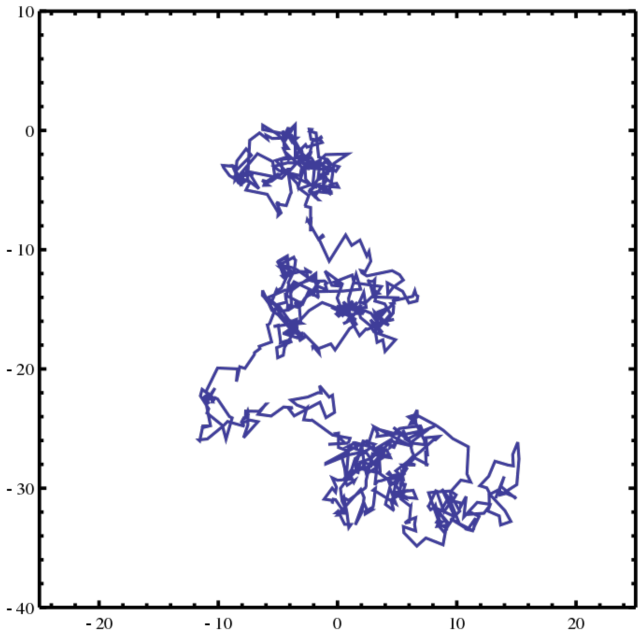


Figure 11 – Mouvement brownien d'une particule.

Le Mouvement Brownien est un processus W_t vérifiant :

•W_t est adapté à sa filtration naturelle $\mathcal{F}_n \sigma(W_k | k \leq t)$.

•Si t=0 alors $\mathrm{W}_t = 0$ p.s.

•W_t est à accroissements indépendants : $(W_t - W_s)$ est indépendant de $\mathcal{F}_s$ pour tous $t, s \in [0, T]$ tels que $s \leq t$.

•W_t est à accroissements stationnaires et gaussiens : $W_t - W_s = W_{t-s}$ suit la loi $N(0, t-s)$ pour tous $t, s \in [0, T]$ tels que $s \leq t$. En particulier $W_{t+1} = W_t + W_1$ avec W_1 est indépendant de $\mathcal{F}_t$ puisque $W_1 = W_{t+1} - W_n$.

•W_t est continu sa densité : $f_{W_t}(x) = \dfrac{1}{\sqrt{2\pi t}} \exp\left(-\dfrac{x^2}{2t}\right)$ pour tout réel x.

•$\mathrm{E}[\mathrm{W}_t] = 0$ et $Var(W_t) = t$.

B Simulation d'une trajectoire du Mouvement Brownien

→ Simulation d'une trajectoire du Mouvement Brownien sous MATLAB :

```
T=1; N=100;

Brownien_Motion(T,N)

function[]=Brownien_Motion(T,N)
t=zeros(1,N+1);
W=zeros(1,N+1);
delta_t=T/N;
t(1)=0;
W(1)=0;
for n=1:N
t(n+1)=t(n)+delta_t;
W(n+1)=W(n)+sqrt(delta_t)*randn;
end
plot(t,W);
xlabel 't_n'
ylabel 'Processus W_n'
title("Trajectoire Mouvement Brownien")
end
```

FIGURE 12 – Code Sous MATLAB du fichier SimulationTrajectoireMB.m

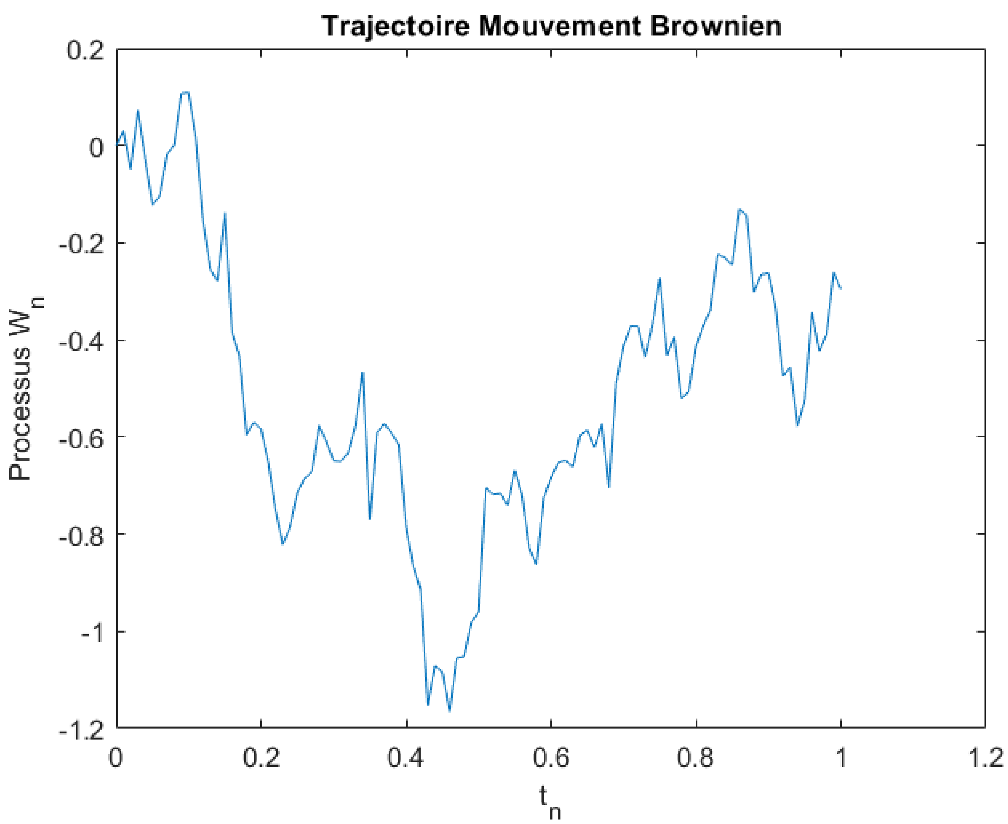


FIGURE 13 – Simulation d'une trajectoire du $(W_n)_{n\geq 0}$.

C Le Mouvement Brownien $(W_t)_{t\geq 0}$ est une Martingale

C.1 Démonstration Mathématique

On note $\mathcal{F}_t = \sigma(W_k|k \leq t)$ la filtration naturelle de $(W_t)_{t\geq 0}$.

1) Pour $t \geq 0$, on a par définition $W_t \in \mathcal{F}_t$ c'est à dire que W_t est adapté à $\mathcal{F}_t$.

2) Pour $n \geq 0$, $\mathbb{E}[|W_t|] =\leq \mathbb{E}[\frac{1}{2}(W_t^2 + 1)] = \frac{1}{2}(t+1) < +\infty$ puisque $\mathbb{E}[W_t^2] = Var(W_t) + \mathbb{E}[W_t]^2 = t+0 = t$ et $|W_t| \leq \frac{1}{2}(W_t^2+1)$ d'après l'inégalité $(|W_t|-1)^2 \geq 0$, alors W_t est integrable.

3) Pour $t \geq 0$, $\mathbb{E}[W_{t+1}|\mathcal{F}_t] = \mathbb{E}[W_t + W_1|\mathcal{F}_t]$ puisque $W_t \in \mathcal{F}_t$ et W_1 est indépendant de $\mathcal{F}_t$ alors : $\mathbb{E}[W_{t+1}|\mathcal{F}_t] = W_t + \mathbb{E}[W_1]$ or $\mathbb{E}[X_1] = 0$,
donc : Pour $n \geq 0$, $\mathbb{E}[W_{t+1}|\mathcal{F}_t] = W_t$.

On a donc $(W_t)_{t\geq 0}$ est une Martingale.

C.2 Démonstration par Simulation Monte-Carlo sous MATLAB

D'après la Loi des Grands Nombres, on vérifie les formules théoriques par simulation :

$\mathbb{E}[W_t] = \frac{1}{N_{mc}} \sum_{k=1}^{N_{mc}} W_t^{(k)} = 0.$

$Var(W_t) = \frac{1}{N_{mc}} \sum_{k=1}^{N_{mc}} (W_t^{(k)})^2 = t.$

$\rightarrow$ Simulation Monte-Carlo de la Martingale Mouvement Brownien :

Par simulation de Monte-Carlo, on vérifie que la moyenne des N_{mc} valeurs à l'instant $n = 300$ est bien égale à la valeur en $n = 100$. Cette valeur tend vers 0 qui est la valeur de l'ésperance de cette Martingale Mouvement Brownien $(W_n)_{n\geq 0}$ puisque $(\mathbb{E}[W_n] = \mathbb{E}[W_0] = 0)$.

```
k=100; n=300; Nmc=2000;
martingale(k,n,Nmc);

function [esperance_Wn,W_Fk] = martingale(k,n,Nmc)
T=2;
delta_t=T/(n+k);
[W_Fk,W] = processus_W(k,delta_t);
figure
for j=1:Nmc
for i=k:(n-1)
W(i+1)=W(i)+sqrt(delta_t)*randn;
end
last_value(j)=W(n);
hold on;
plot(W,"LineWidth",2)
xlabel 't_n'
ylabel 'Mouvement Brownien W_n'
title 'Simulation Monte-Carlo Martingale MB'
end
esperance_Wn=sum(last_value)/Nmc;
end

function [W_Fk,W] = processus_W(k,delta_t)
W(1)=0;
for i=1:k-1
W(i+1)=W(i)+sqrt(delta_t)*randn;
end
W_Fk=W(k);
end
```

FIGURE 14 – Code Sous MATLAB du fichier SimulationMartingaleMB.m

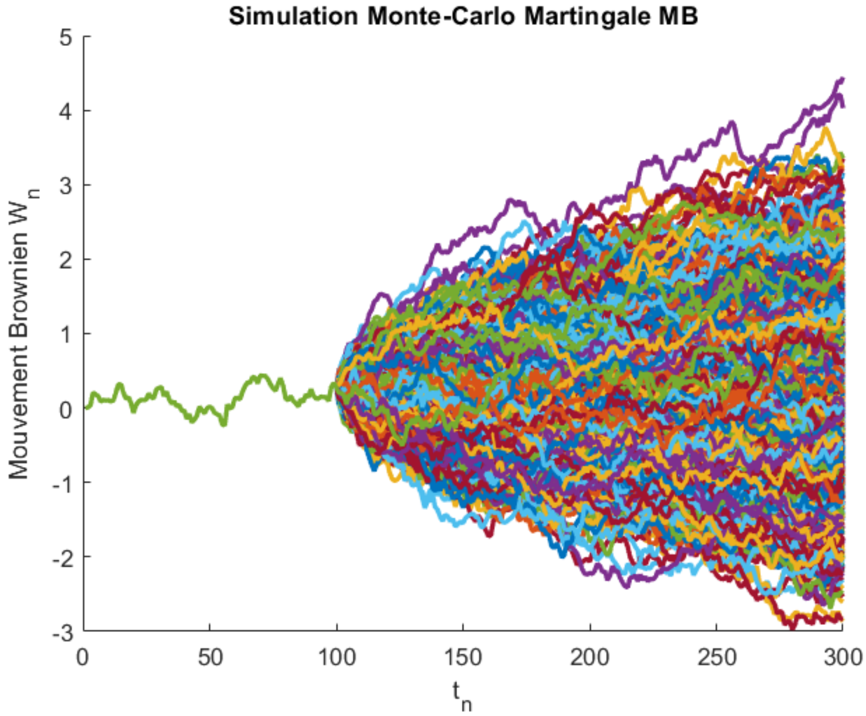


FIGURE 15 – Simulation Monte-Carlo Martingale $(W_t)_{t\geq 0}$.

V Démonstration Mont-Carlo des Martingales $(B_t)_{t\geq 0} = (W_t^2 - t)_{t\geq 0}$ et $(C_t)_{t\geq 0} = (\exp(\sigma W_t - \frac{\sigma^2 t}{2}))_{t\geq 0}$

A Simulation Monte-Carlo de $(B_t)_{t\geq 0} = (W_t^2 - t)_{t\geq 0}$

$\rightarrow$ Simulation Monte-Carlo de la Martingale $(B_t)_{t\geq 0} = (W_t^2 - t)_{t\geq 0}$ sous MATLAB :

```
k=100;n=300;Nmc=2000;
martingale(Nmc,n,k)

function martingale(Nmc,n,k)
B=zeros(1,n+1);
W=zeros(1,n+1);
somme=0;
delta_t=1/n;
t=(0:n)*delta_t;
figure;
for i=1:k-1
W(i+1)=W(i)+sqrt(delta_t)*randn(1,1);
B(i+1)=W(i+1)^2-t(i+1);
end
for j=1:Nmc
for i=k:n
W(i+1)=W(i)+sqrt(delta_t)*randn(1,1);
B(i+1)=W(i+1)^2-t(i+1);
end
somme=somme+B(n+1);
hold on;
plot(B,'LineWidth',2);
xlabel('t_n');
ylabel('Processus B_n');
title('Simulation Monte Carlo de B_n=(W_n^2-n)');
end
disp(B(k))
disp(somme/Nmc);
end
```

FIGURE 16 – Code Sous MATLAB du fichier SimulationMartingaleBn.m

```
>> SimulationProcessusBn
   -0.2967

   -0.3180
```

FIGURE 17 – Vérification de la Simulation Monte-Carlo via l'espérance de $(B_t)_{t\geq 0} = (W_t^2 - t)_{t\geq 0}$

Par simulation de Monte-Carlo, on vérifie que la moyenne des N_{mc} valeurs à l'instant $n = 300$ est bien égale à la valeur en $n = 100$. Cette valeur tend vers 0 qui est la valeur de l'ésperance de cette Martingale $(B_t)_{t\geq 0} = (W_t^2 - t)_{t\geq 0}$ puisque $(\mathbb{E}[B_t] = \mathbb{E}[B_0] = \mathbb{E}[W_0^2 - 0] = 0)$.

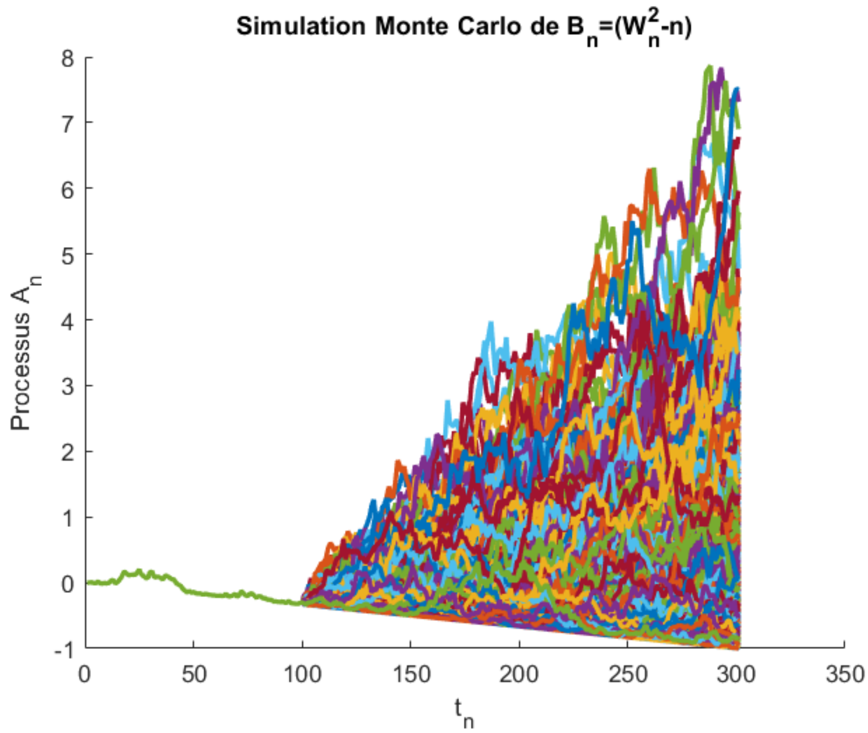


FIGURE 18 – Simulation Monte-Carlo Martingale $(B_t)_{t\geq 0} = (W_t^2 - t)_{t\geq 0}$.

B Simulation Monte-Carlo de $(C_t)_{t\geq 0} = (\exp(\sigma W_t - \frac{\sigma^2 t}{2}))_{t\geq 0}$

→ Simulation Monte-Carlo de la Martingale $(C_t)_{t\geq 0} = (\exp(\sigma W_t - \frac{\sigma^2 t}{2}))_{t\geq 0}$ sous MATLAB :

```
k=100; n=300; Nmc=2000; sigma=0.3;
martingale(Nmc,n,k,sigma)

function martingale(Nmc,n,k,sigma)
C=ones(1,n+1);
W=zeros(1,n+1);
somme=0;
delta_t=1/n;
t=(0:n)*delta_t;
figure;
for i=1:k-1
W(i+1)=W(i)+sqrt(delta_t)*randn(1,1);
C(i+1)=exp((sigma*W(i+1))-(sigma^2*t(i+1))/2);
end
for j=1:Nmc
for i=k:n
W(i+1)=W(i)+sqrt(delta_t)*randn(1,1);
C(i+1)=exp((sigma*W(i+1))-(sigma^2*t(i+1))/2);
end
somme=somme+C(n+1);
hold on;
plot(C,'LineWidth',2);
xlabel('t_n');
ylabel('Processus C_n');
title('Simulation Monte Carlo de C_n=exp((sigma*W_n)-(sigma^2*t_n)/2)');
end
disp(C(k))
disp(somme/Nmc);
end
```

FIGURE 19 – Code Sous MATLAB du fichier SimulationMartingaleCn.m

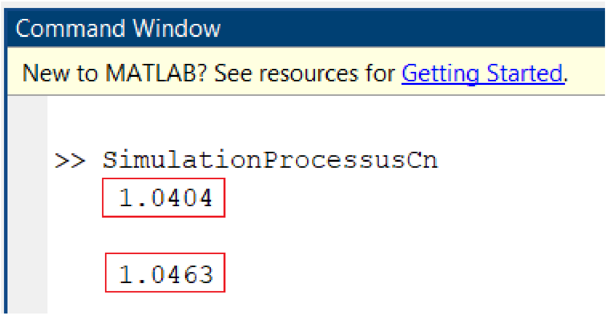


FIGURE 20 – Vérification de la Simulation Monte-Carlo via l'espérance de $(C_t)_{t\geq 0} = (\exp{(\sigma W_t - \frac{\sigma^2 t}{2})})_{t\geq 0}$

Par simulation de Monte-Carlo, avec une valeur de $\sigma = 0.3$, on vérifie que la moyenne des N_{mc} valeurs à l'instant $n = 300$ est bien égale à la valeur en $n = 100$. Cette valeur tend vers 1 qui est la valeur de l'ésperance de cette Martingale $(C_t)_{t\geq 0} = (\exp{(\sigma W_t - \frac{\sigma^2 t}{2})})_{t\geq 0}$ puisque $(\mathbb{E}[C_t] = \mathbb{E}[C_0] = \mathbb{E}[\exp{(\sigma W_0 - \frac{\sigma^2 0}{2})}] = 1)$.

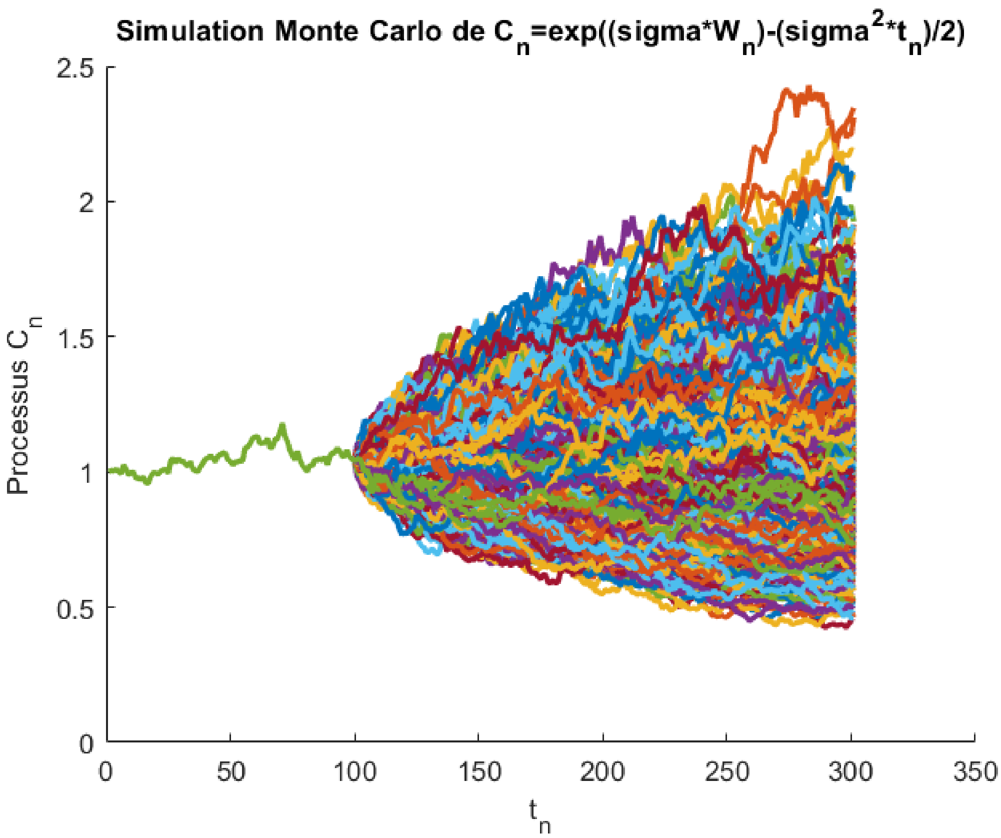


FIGURE 21 – Simulation Monte-Carlo Martingale $(C_t)_{t\geq 0} = (\exp{(\sigma W_t - \frac{\sigma^2 t}{2})})_{t\geq 0}$ avec $\sigma = 0.3$.

$\rightarrow$ Conclusion

Par Simulation de Monte-carlo on a bien les processus : $(B_t)_{t\geq 0} = (W_t^2 - t)_{t\geq 0}$ et $(C_t)_{t\geq 0} = (\exp{({}_t - \frac{\sigma^2 t}{2})})_{t\geq 0}$ sont bien des Martingales.

VI Simulation de la Variation Quadratique (Lemme d'Ito)

A Définition de la Variation Quadratique

la variation quadratique est utilisée dans l'analyse des processus stochastiques, comme le mouvement brownien et autres martingales1. La variation quadratique est un type de variation d'un processus. Définition de la variation quadratique à $t = T$: Soit X_t processus sur [0,T], une subdivision $\pi_n = (t_0, t_2, ...t_N)$ de l'intervalle [0,T]. Variation quadratique s'appelle la limite : $< X >_T = \lim_{N\to+\infty} \sum_{n=1}^{N} (X_{t_n} - X_{t_{n-1}})^2$. On définit la limite dans le sens convergence L^2.

B Simulation de la Variation Quadratique du Mouvement Brownien

On définit la variation quadratique du mouvement Brownien sur l'intervalle [0,T] par : $< W >_T = \lim_{N\to+\infty} \sum_{n=1}^{N} (W_n - W_{n-1})^2 = T$.

On doit avoir que $t \mapsto < W >_T$ est une droite. On la simule et on observe et on observe que la variance de $< W >_T$ tend vers zero quand $N \to +\infty$ et la variation quadratique approche une variable non aléatoire T.

$\to$ Simulation de la Variation Quadratique du Mouvement Brownien :

```
N=100; T=1;

variation_quadratique_Ito(N,T)

function [] = variation_quadratique_Ito(N,T)
 W(1)=0; variation_W(1)=0; delta_t=T/N;
t=(0:N)*delta_t;
for i=1:N
W_dt(i)=sqrt(delta_t)*randn;
W(i+1)=W(i)+ W_dt(i);
variation_W(i+1)=variation_W(i)+(W(i+1)-W(i))^2;
end
plot(t,variation_W,'LineWidth',2);
hold on;
plot(t,t,'yellow','LineWidth',2);
xlabel 't_n'
ylabel 'Variation Quadratique W_n'
title 'Simulation de la Variation Quadratique du Mouvement Brownien'
end
```

FIGURE 22 – Code Sous MATLAB du fichier VariationQuadratiqueIto.m.m

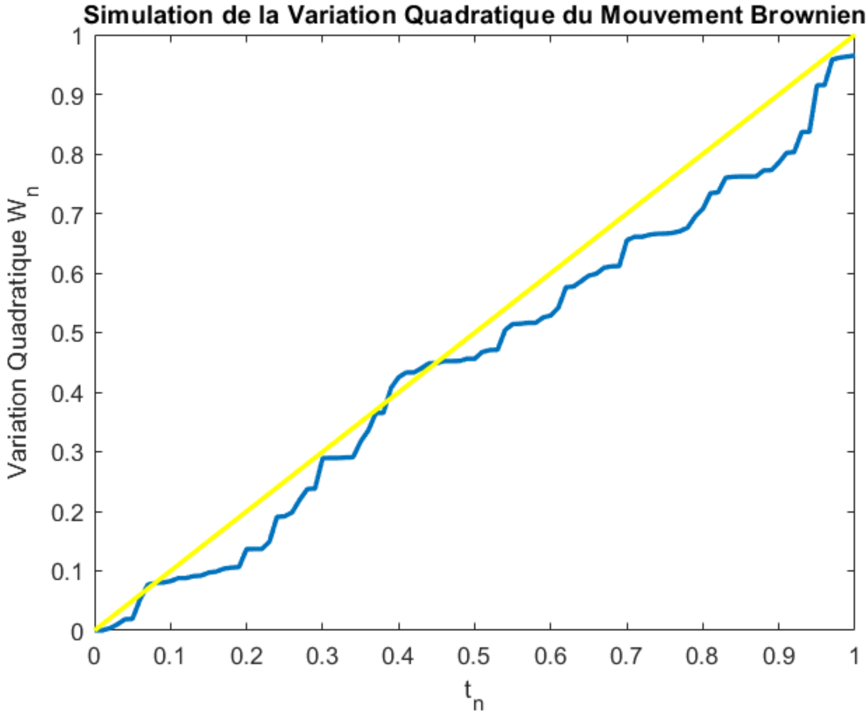


FIGURE 23 – Simulation de la Variation Quadratique du Mouvement Brownien en Jaune $(W_t)_{t\geq 0}$.

VII Simulations pour les Intégrales Stochastiques

A Définition de l'intégrale stochastique

On définit l'Intégrale Stochastique par :

$\int_0^T \Theta_t dW_t = \lim_{N\to+\infty} \sum_{n=0}^{N-1}(W_{t_{i+1}} - W_{t_i})^2$ avec $\mathbb{E}[\int_0^T \Theta_s^2 ds] < +\infty$

Dans les applications en finance, Θ_t représentera la quantité d'actif risqué contenue dans un portefeuille à l'instant t et dW_t la variation infinitésimale de cet actif risqué. Il est donc naturel de vouloir imposer que Θ_t soit $\mathcal{F}$-adapté.

B Simulation de l'intégrale stochastique

→ Simulation de l'intégrale stochastique avec $\Theta_t = W_t$:

```
% teta_t=W_t

N=100; Nmc=1000000; T=1;

Esperance_Integrale_Stochastique(N,Nmc,T)

function [esperance_Integrale] = Esperance_Integrale_Stochastique(N,Nmc,T)
W(1)=0;
delta_t=T/N;
t=(0:N)*delta_t;
for k=1:Nmc
I=0;
for i=1:N
W(i+1)=W(i)+sqrt(delta_t)*randn;
I=I+W(i)*(W(i+1)-W(i));
end
Integrale(k)=I;
end
esperance_Integrale=mean(Integrale);
end
```

Command Window

New to MATLAB? See resources for Getting Started.

```
>> EsperanceIntegraleStachastique

ans =

  8.7750e-04
```

FIGURE 24 – Code Sous MATLAB du fichier EsperanceIntegraleStachastique.m

On remarque que cette espérance tend vers 0 ce qui est absolument cohérent. Donc la simulation est bien vérifiée.

C Vérification Isométrie - Simulation de l'intégrale stochastique cas général

→ Simulation de l'intégrale stochastique cas général :
On choisit la fonction :

$$\Theta_t(t, W_t) == t^2 W_t^2 + \sin W_t + t$$

```
function [esperance_Integrale_stoch_carre,esperance_Integrale_Teta] = Verification_Isometrie()
function[f]=Teta(t,w)
    f=t^2*w^2+sin(w)+t;
end
W(1)=0;N=100; Nmc=100000; T=2; delta_t=T/N;
t=(0:N)*delta_t;
for k=1:Nmc
I_stoch=0;
I_Teta=0;
for i=1:N
W(i+1)=W(i)+sqrt(delta_t)*randn;
I_stoch=I_stoch+Teta(t(i),W(i))*(W(i+1)-W(i));
I_Teta=I_Teta+(Teta(t(i),W(i)))^2*delta_t;
end
Integrale_stoch_carre(k)=I_stoch^2;
Integrale_Teta(k)=I_Teta;
end
esperance_Integrale_stoch_carre=mean(Integrale_stoch_carre);
esperance_Integrale_Teta=mean(Integrale_Teta);
end
```

Command Window

New to MATLAB? See resources for Getting Started.

```
>> Verification_Isometrie

ans =

  68.0205

>> [esperance_Integrale_stoch_carre,esperance_Integrale_Teta] = Verification_Isometrie()

esperance_Integrale_stoch_carre =

  67.6824


esperance_Integrale_Teta =

  68.2034
```

FIGURE 25 – Code Sous MATLAB du fichier VerificationIsometrie.m

Donc la simulation est bien vérifiée.

VIII Conclusion

Je tiens tout d'abord à remercier mon professeur Irina Kortchemski pour ces efforts agréables malgré l'etat sanitaire et l'enseignement à distance.

Ce Devoir m'a permis de bien comprendre les notions et les techniques fondamentales de la méthode de simulation Monte-Carlo grâce à des algorithmes en Matlab et les liens de celle-ci avec les principales méthodes de résolution numérique des Équations Différentielles Stochastiques ainsi que la modélisation et estimation par simulation des différents paramètres intervenant aux problèmes des finances quantitatives.