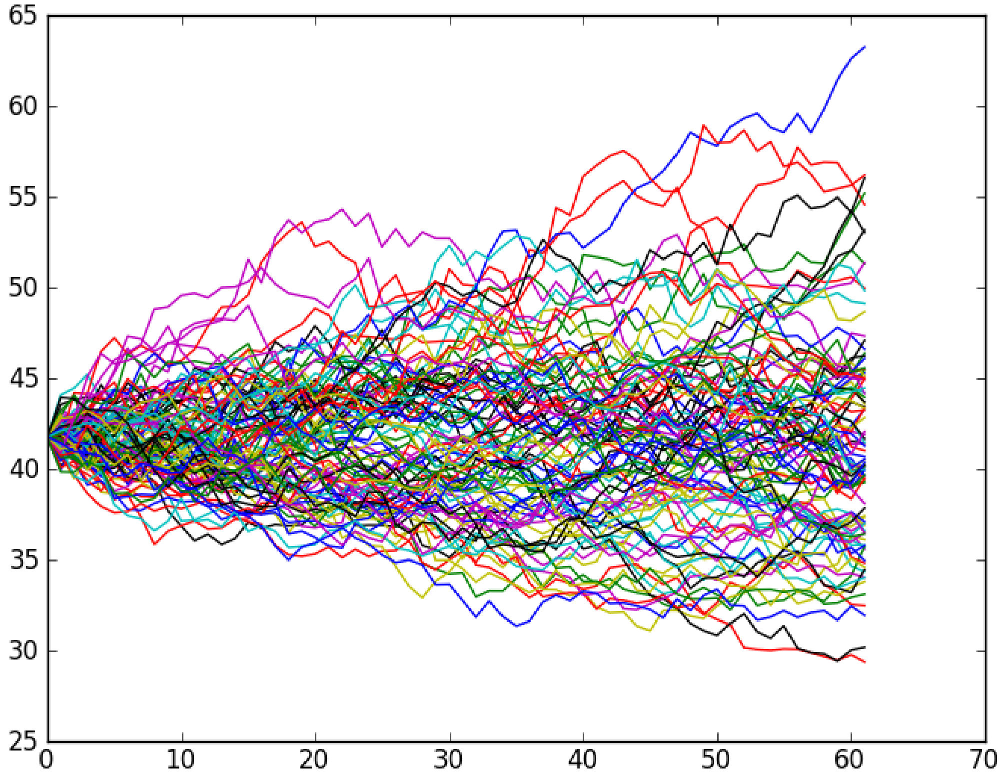
Méthodes de Monte Carlo pour la Finance
EXAMEN-MF : 2020 - 2021



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

Table des matières

1 Introduction

1 - 1 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.

1 - 2 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.

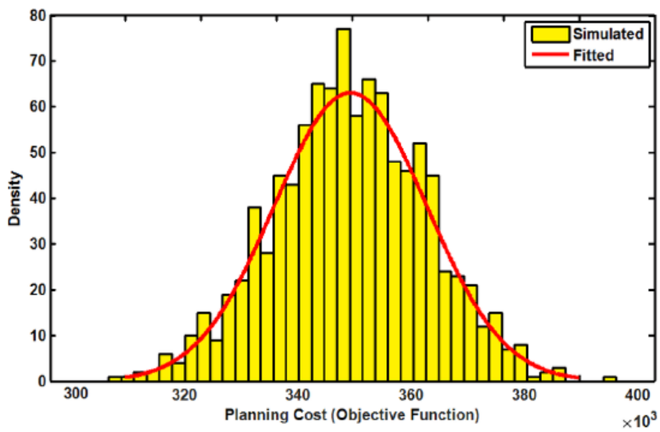
1 - 3 Principe

Les différentes méthodes de simulation permettent de reproduire artificiellement des réalisations de certaines variables aléatoires (qui modélisent souvent le fonctionnement des différents systèmes physiques, et qui respectent certaines lois de probabilités).
Ces réalisations permettent de faire des estimations des paramètres statistiques les plus importants des systèmes et ainsi tenter une étude plus efficace pour optimiser le fonctionnement et la fiabilité de ces systèmes.

1 - 4 Rappels sur les méthodes de Monte Carlo

Les méthodes de Monte Carlo sont nées grâce à Louis Buffon mathématicien du XVII ème siècle. Il a ainsi entrevu une possibilité pour calculer le nombre π numériquement. En effet, en lançant un grand nombre d'aiguilles et en comptant celles qui touchaient deux stries il détenait une valeur approximative de la probabilité. Cette expérience repose sur un théorème : **la loi forte des grands nombres.** La méthode de simulation de Monte-Carlo permet aussi d'introduire une approche statistique du risque dans une décision financière.

- ✓ **Avantage** : Facilité de mise en œuvre.
- ▪ **Inconvénients** : Il s'agit seulement d'une approximation des variables aléatoires.



2 Réponses :

2 - 1 QUESTION 1.

Soit W_t un Mouvement Brownien sur l'intérvalle $t \in [0, T = 3]$, Calculons par les simulations Monte-Carlo(MC) $(N_{mc} = 10000)$ $Var(W_T)$ et $\mathbb{E}[(W_T)^4]$:

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

$$Var(W_T) = \frac{1}{N_{mc}} \sum_{k=1}^{N_{mc}} (W_T^{(k)})^2 = T = 3.$$

$$\mathbb{E}[(W_T)^4] = \frac{1}{N_{mc}} \sum_{k=1}^{N_{mc}} (W_T^{(k)})^4.$$

Question1.m

```matlab

fprintf("E[WT^4]=%f \n",Esperance_WT_4())
fprintf("V[WT]=%f \n",Variance_WT())

function[esperance] = Esperance_WT_4()
    T = 3;
    N = 100;
    delta_t = T/N;
    Nmc = 10000;
    W(1) = 0;
    for n = 1:Nmc
        for i = 1:N - 1
            W(i+1) = W(i) + sqrt(delta_t) * randn;
        end
        last_value(n) = W(N) ^ 4;
    end
    esperance = mean(last_value);
end

function[variance] = Variance_WT()
    T = 3;
    N = 100;
    delta_t = T/N;
    Nmc = 10000;
    W(1) = 0;
    for n = 1:Nmc
        for i = 1:N - 1
            W(i+1) = W(i) + sqrt(delta_t) * randn;
        end
        last_value(n) = W(N);
    end
    esperance = mean(last_value);
    variance = mean((last_value -  esperance).^2);
end
```

```
>> Question1
E[WT^4]=25.603318
V[WT]=3.066302
>>
```

FIGURE 1 – Code Sous MATLAB du fichier Question1.m

On trouve bien le resultat par simulation Monte Carlo.

2 - 2 QUESTION 2.

Soit W_t un Mouvement Brownien sur l'intérvalle $t \in [0, T = 3]$, Vérifions par les simulations Monte-Carlo le résultat suivant :

$$\int_0^T W_t^2 dW_t = \frac{1}{3} W_T^3 - \int_0^T W_t dt$$

Question2.m

```
Integrales_Stochastiques()

function[]= Integrales_Stochastiques()
    T = 3;
    N = 100000;
    delta_t = T/N;
    W(1) = 0;
    Integrale_gauche(1) = 0;
    Integrale_Wt(1) = 0;
    t = (1:N) * delta_t;
    for i = 1:N - 1
        W(i + 1) = W(i) + sqrt(delta_t) * randn;
        Integrale_gauche(i+1) = Integrale_gauche(i) + (W(i) ^ 2) * sqrt(delta_t) * randn;
        Integrale_Wt(i + 1) = Integrale_Wt(i) + W(i) * delta_t;
        Integrale_droite(i + 1) = (1/3) * (W(i+1) ^ 3) - Integrale_Wt(i+1);
    end
    hold on;
    plot(t,Integrale_gauche,'g');
    plot(t,Integrale_droite,'b');
    xlabel 't_n'
    ylabel 'Integrales'
    title 'Simulation Monte Carlo Integrales'
end
```

FIGURE 2 – Code Sous MATLAB du fichier Question2.m

Graphes des Trajectoires :

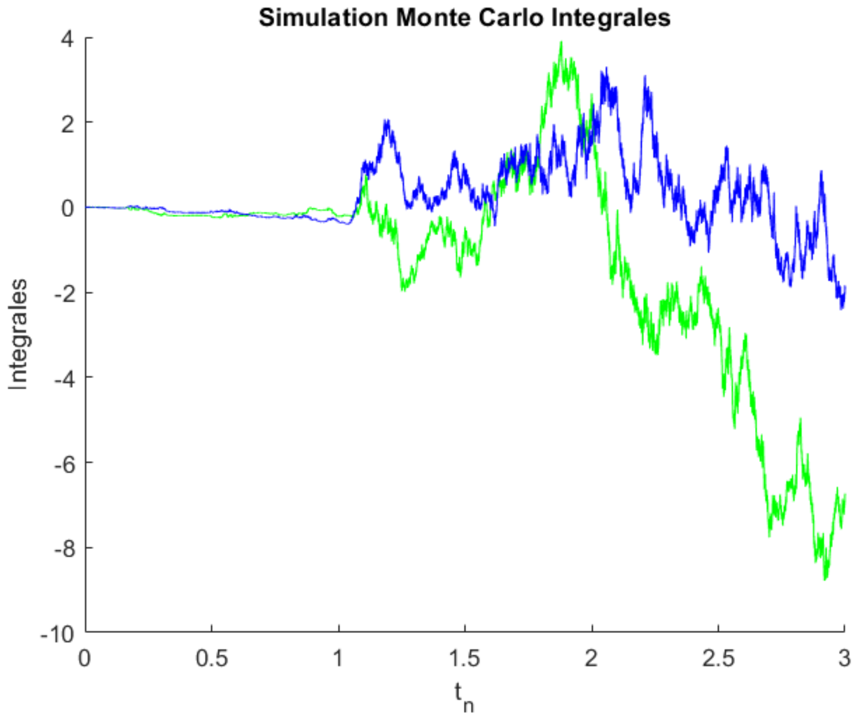


FIGURE 3 – Trajectoire des integrales : Integrale à gauche en Vert et integrale à droite en Bleu

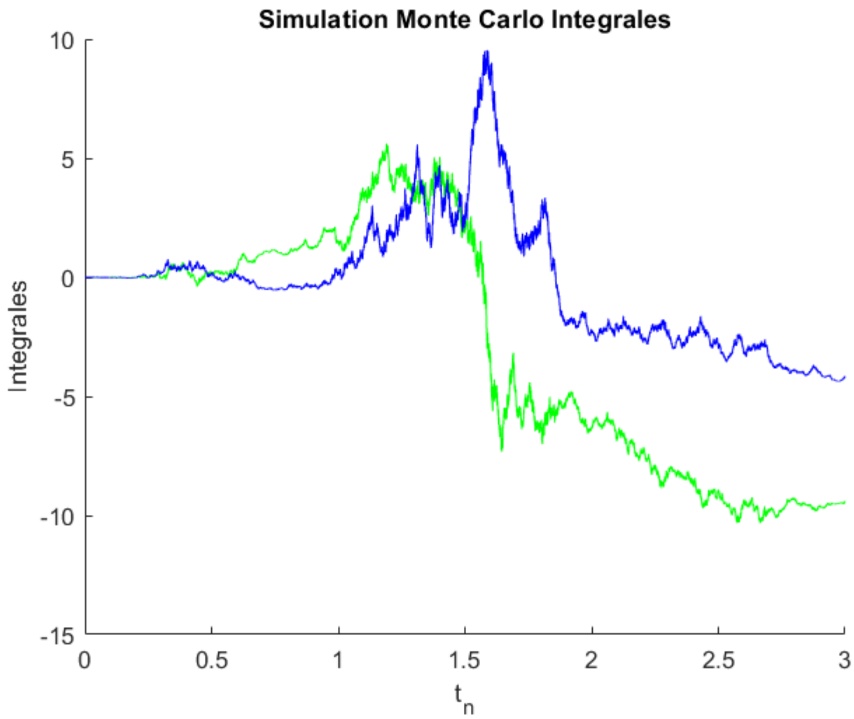


FIGURE 4 – Trajectoire des integrales : Integrale à gauche en Vert et integrale à droite en Bleu

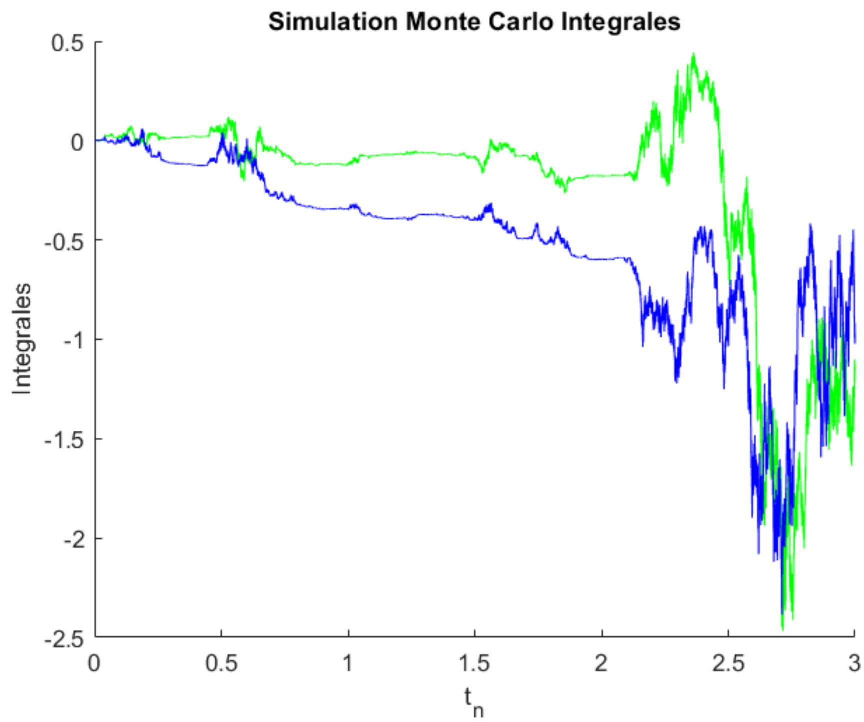


FIGURE 5 – Trajectoire des integrales : Integrale à gauche en Vert et integrale à droite en Bleu

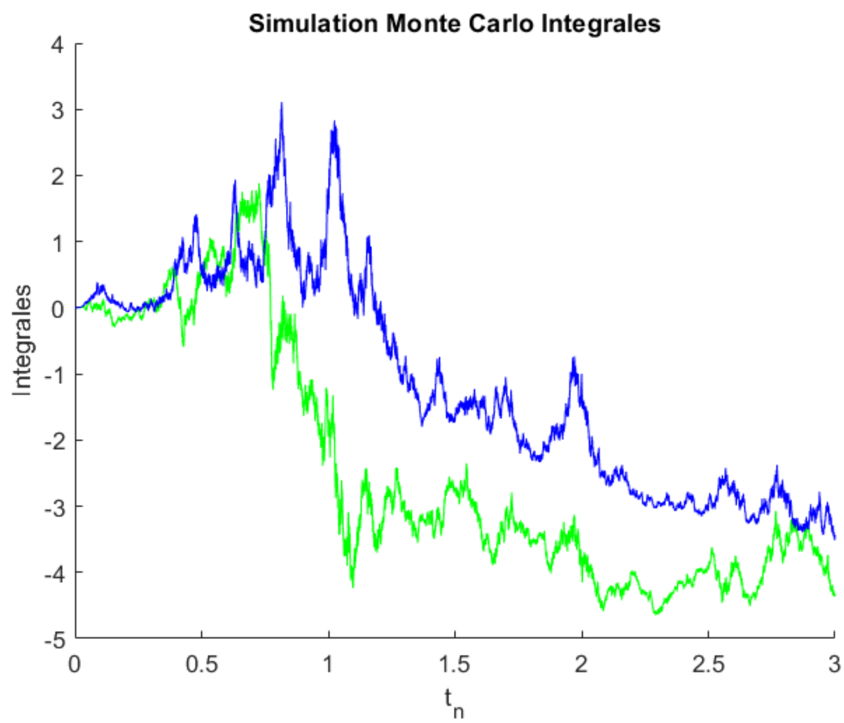


FIGURE 6 – Trajectoire des integrales : Integrale à gauche en Vert et integrale à droite en Bleu

2 - 3 QUESTION 3.

Soit l'actif vérifie l'équation différentielle stochastique suivante :

$$dS_t = S_t(r(t)dt + \sigma(t)dW_t), \quad t \in [0, T], \quad S(t = 0) = S_0$$

Le taux d'interêt et la volatilité ne sont plus des constantes. C'est le cas du modèle de Black et Scholes généralisé.

$$\sigma(t) = \sigma_0 \frac{t}{T}, \quad r(t) = r_0 \frac{t^2}{T^2}, \qquad \sigma_0 = 0.5, \quad r_0 = 0.3, \quad T = 0.5.$$

Simulation ($N_{mc} = 1000$) car c'est lourd pour 10000 trajectoires de l'actif avec $S_0 = 10$.

```
Question3.m

Simulation_Actif()

function[] = Simulation_Actif()
    T = 0.5;
    N = 1000;
    delta_t = T/N;
    sigma(1) = 0.5;
    r(1) = 0.3;
    S(1) = 10;
    Nmc = 1000;
    t = (1:N)*delta_t;
    for n = 1:Nmc
        for i = 1:N - 1
            sigma(i + 1) = sigma(1) * delta_t*(i + 1) / T;
            r(i + 1) = (r(1) * (delta_t * (i + 1)) ^ 2)/(T ^ 2);
            S(i+1) = S(i) * (1 + r(i) * delta_t + sigma(i) * sqrt(delta_t) * randn);
        end
        hold on;
        plot(t,S);
        xlabel 't_n'
        ylabel 'Actif'
       title 'Simulation de l actif'
    end
end
```

FIGURE 7 – Code Sous MATLAB du fichier Question3.m

Graphe de l'actif :

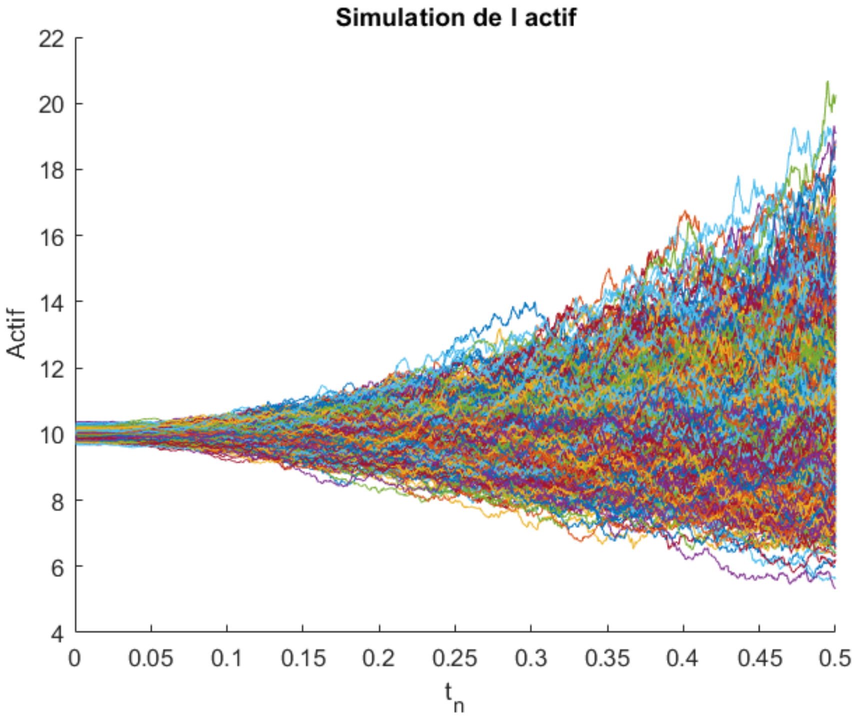


FIGURE 8 – Simulation de l'actif

2 - 4 QUESTION 4.

Dans le modèle Cox-Ingerson-Ross le taux d'interêt vérifie l'équation différentielle stochastique suivante :

$$dr_t = (\alpha - \beta r_t)dt + \omega\sqrt{|r_t|}dW_t, \quad t \in [0, T], \quad r(t = 0) = r_0$$

```
Simulation_rt()


function[last_value] = Simulation_rt()
    T = 0.5;
    N = 100;
    delta_t = T/N;
    r(1) = 0.1;
    alpha = 0.2;
    beta = 0.1;
    w = 0.3;
    sigma = 0.5;
    Nmc = 1000;
    t = (1:N) * delta_t;
    for n = 1:Nmc
        for i = 1:N-1
            r(i+1) = r(i) + (alpha - beta*r(i)) * delta_t + w * sqrt(abs(r(i))) * sqrt(delta_t) * randn;
        end
        hold on;
        last_value(n) = r(N);
        plot(t,r);
        xlabel 't_n'
        ylabel 'taux d interet'
       title 'Simulation de taut d interet'
    end
end
```

FIGURE 9 – Code Sous MATLAB du fichier Question41.m

Graphe du modèle :

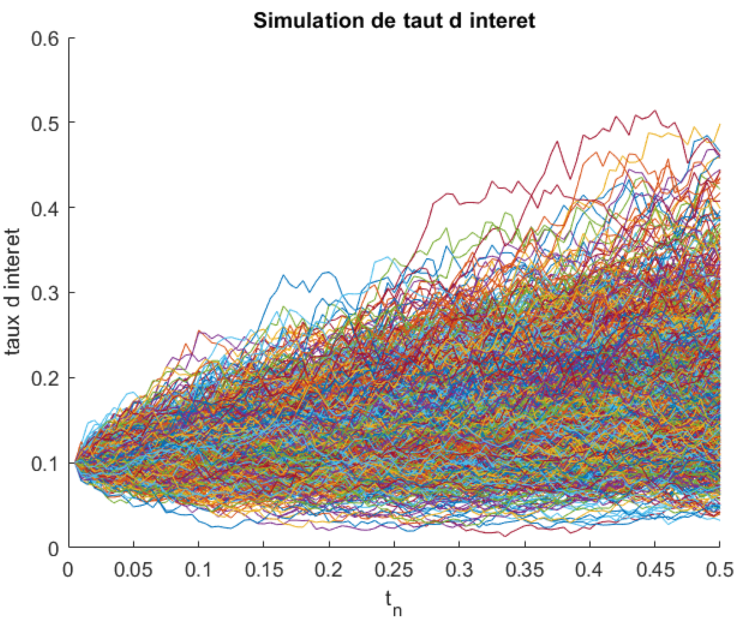


FIGURE 10 – Simulation de l'actif

```
Question3_1.m  ×  Question3_2.m  ×  +

densite_empirique_rt()

function[last_value] = Simulation_rt()
    T = 0.5;
    N = 100;
    delta_t = T/N;
    r(1) = 0.1;
    alpha = 0.2;
    beta = 0.1;
    w = 0.3;
    sigma = 0.5;
    Nmc = 1000;
    t = (1:N) * delta_t;
    for n = 1:Nmc
        for i = 1:N-1
            r(i+1) = r(i) + (alpha - beta*r(i)) * delta_t + w * sqrt(abs(r(i))) * sqrt(delta_t) * randn;
        end
        hold on;
        last_value(n) = r(N);
    end
end

function[] = densite_empirique_rt()
    X = Simulation_rt();
    Nmc = length(X);
    a = 0; %debut de l'intervalle
    b = 0.5; % fin de l'intervalle
    N = 18; %Nombre pour discretiser l'intervalle
    delta_t = (b-a)/N; %pas de la discretisation
    for i = 1:N
        x(i) = a + delta_t * (i - 1);
        counter = 0;
        for n = 1:Nmc
            if X(n) <=  x(i) + delta_t && X(n) >= x(i)
                counter = counter + 1;
            end
        end
        densite(i) = counter/(Nmc*delta_t);
    end
    plot(x,densite);
       xlabel 'x_n'
       ylabel 'fonction densite rt'
       title 'Simulation de fonction densite rt'
end
```

Figure 11 – Code Sous MATLAB du fichier Question42.m

Graphe de la fonction de densité empirique :

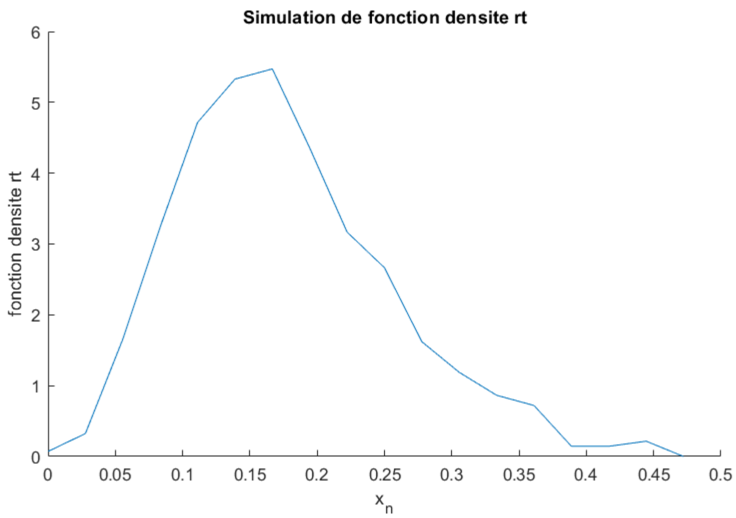


Figure 12 – Simulation de l'actif

Dans le cadre du modèle Cox-Ingerson-Ross l'actif vérefie l'équation différentielle stochastique conduite par le même MB Wt que le le processus stochastique rt.

$$dS_t = S_t(r_t dt + \sigma dW_t), \quad t \in [0, T], \quad S(t = 0) = S_0$$

Question43.m

```
function[] = Simulation_modele()
    T = 0.5;
    N = 100;
    delta_t = T/N;
    r(1) = 0.1;
    alpha = 0.2;
    beta = 0.1;
    w = 0.3;
    sigma = 0.5;
    Nmc = 1000;
    S(1) = 10;
    t = (1:N) * delta_t;
    counter = 0;
    for n = 1:Nmc
        for i = 1:N-1
            r(i+1) = r(i) + (alpha - beta*r(i)) * delta_t + w * sqrt(abs(r(i))) * sqrt(delta_t) * randn;
            S(i+1) = S(i) * (1 + r(i) * delta_t + sigma * sqrt(delta_t) * randn);
        end
        if S(N) < S(1)
            counter = counter + 1;
        end
        hold on;
        plot(t,S);
        xlabel 't_n'
        ylabel 'S'
       title 'Simulation de S'
    end
    proba = counter/Nmc;
    fprintf("P[St<S0]=%f \n",proba)
end
```

```
P[St<S0]=0.501000
fx >>
```

Figure 13 – Code Sous MATLAB du fichier Question43.m

Graphe du modèle :

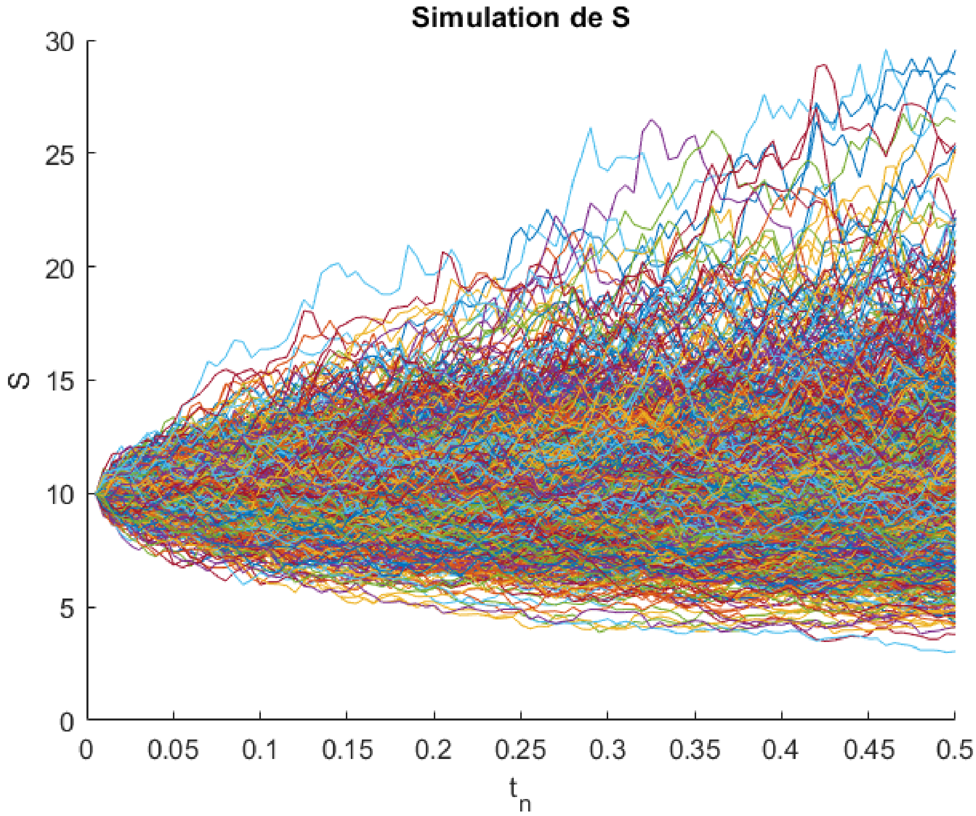


FIGURE 14 – Simulation de l'actif

2 - 5 QUESTION 5.

On observe parfois sur le marché une situation qui s'appelle Oscillateur Brownien. Les deux portefeulles correlés Pt et Mt vériffient le système des équations différentielles stochastiques dans l'enoncé.

```
function ResolutionMC()


    T=15;
    n=100;
    M(1)=10;
    P(1)=10;
    sigma=1.5;
    a=0.3;
    b=0.5;
    delta_t=T/n;
    t=(0:n)*delta_t;
    figure;

    variation_M(1)=0;
    variation_P(1)=0;

    for i=1:n
        M(i+1)=M(i)+P(i)*delta_t;
        P(i+1)=P(i)-a*P(i)*delta_t-b*M(i)*delta_t+sigma*sqrt(delta_t)*randn;
        variation_M(i+1)=variation_M(i)+(M(i+1)-M(i))^2;
        variation_P(i+1)=variation_P(i)+(P(i+1)-P(i))^2;
    end
    plot(t,variation_M);
    xlabel('t');
    ylabel('Variation quadratique de M');
    title('Variation quadratique de M ');


end
```

FIGURE 15 – Code Sous MATLAB du fichier Question43.m

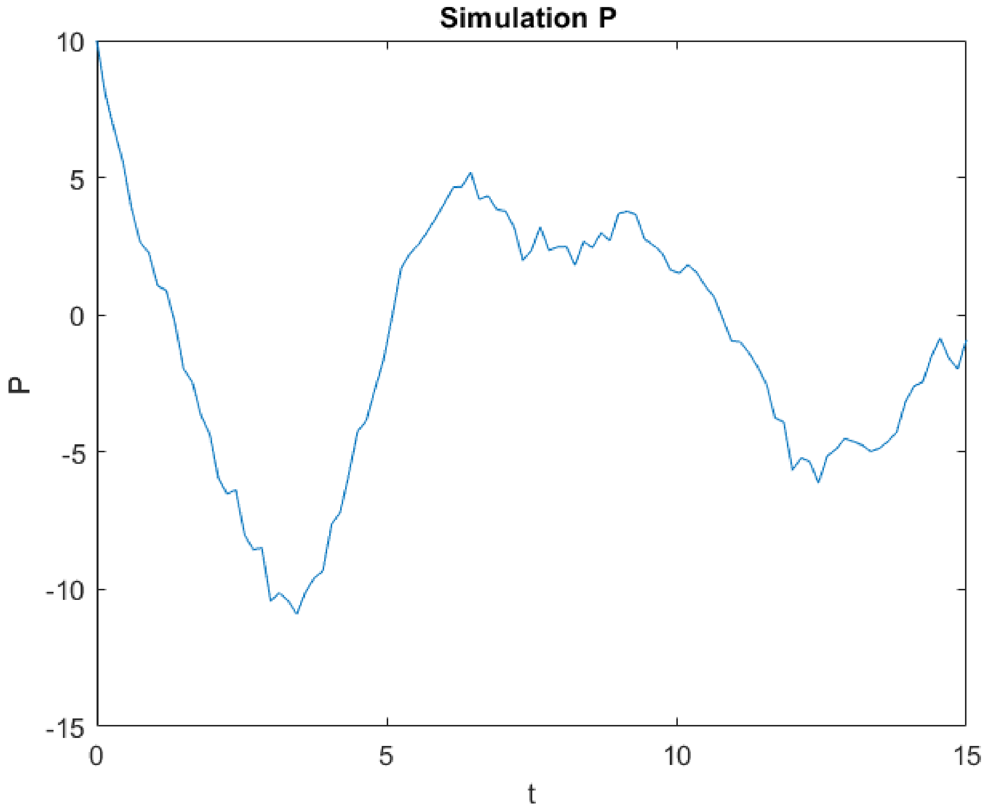


FIGURE 16

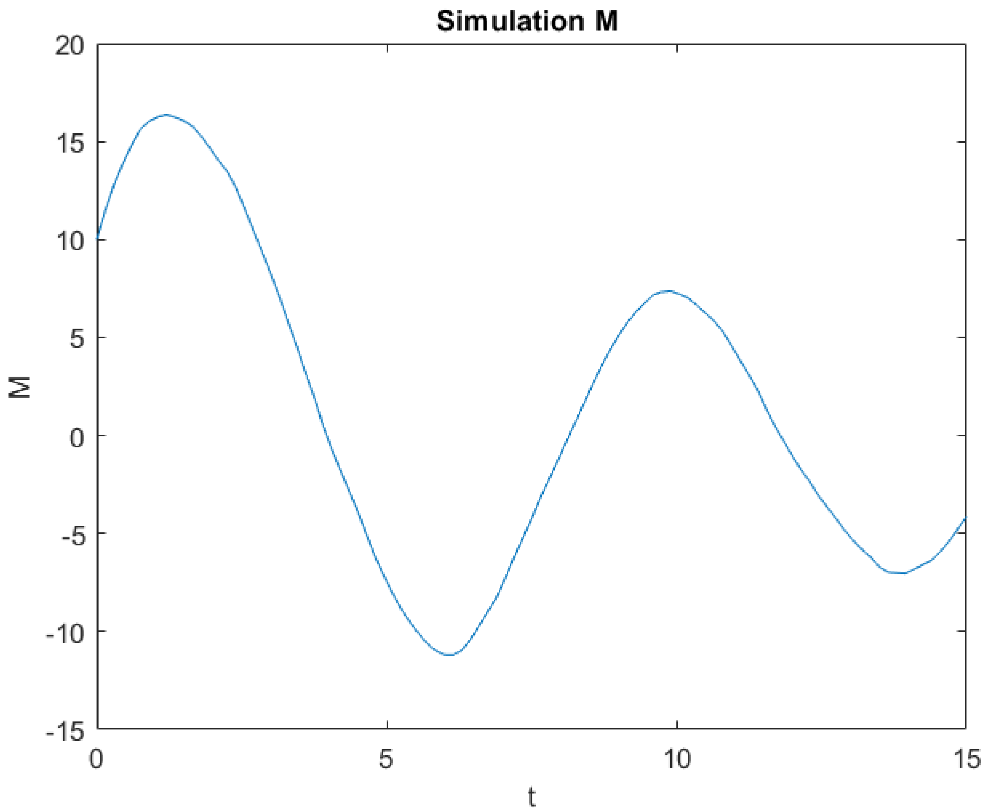


FIGURE 17

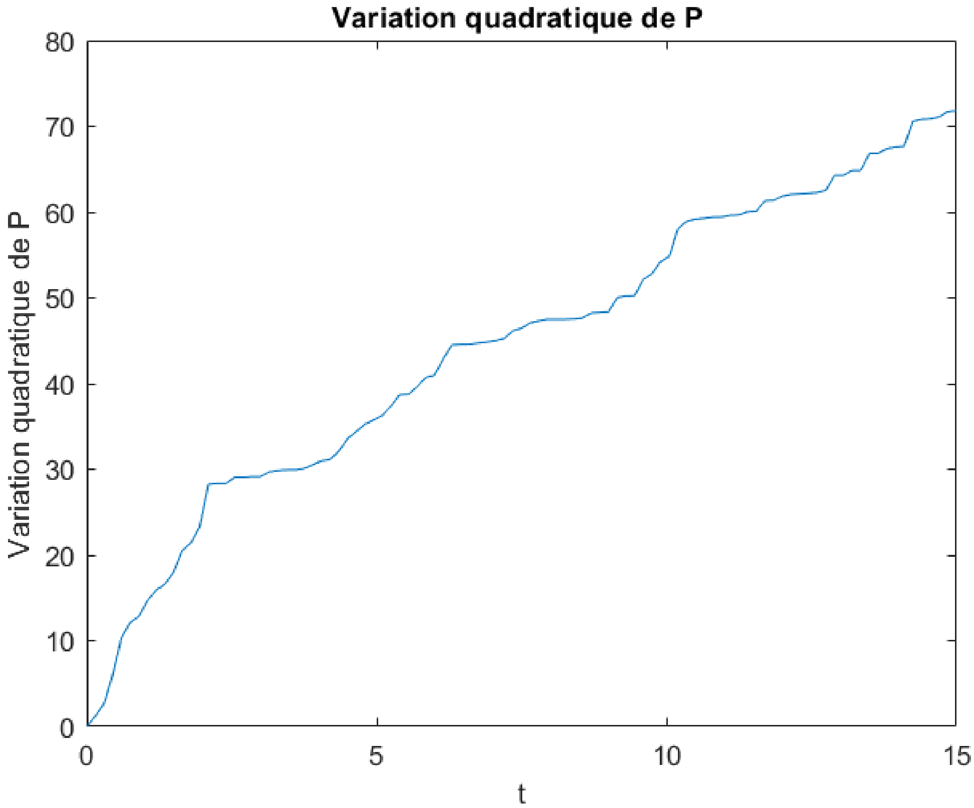


FIGURE 18

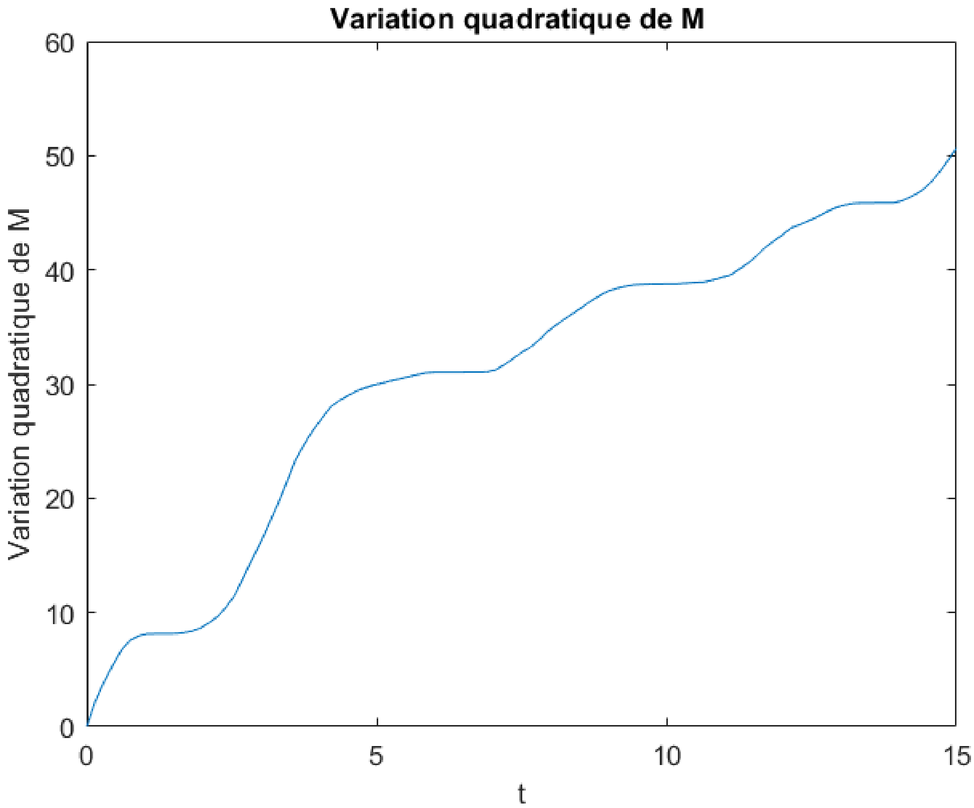


FIGURE 19

3 Conclusion

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

Le cours Méthodes de Monte Carlo pour la Finance 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.

Grâce au module "Méthodes de Monte Carlo pour la Finance " pour l'année 2020-2021, j'ai accumulé de nombreuses connaissances :

→ Modélisation probabiliste du marché au temps discret et continue.
→ Simulation des trajectoires des actifs dans le cadre du modèle Binomial.
→ Vérification de la propriété "martingale" pour le modèle "Marche aléatoire".
→ Simulation du Mouvement Brownien.
→ Vérification de la propriété "martingale" pour le mouvement Brownien.
→ Variation quadratique et le Lemme d'Ito.
→ Simulation de l'intégrale stochastique.
→ Vérification de la propriété d'isométrie de l'intégrale stochastique par Monte Carlo.
→ Calcule des intégrales stochastiques à l'aide du Lemme d'Ito.
→ Application du Lemme d'Ito à la résolution de l'EDS de Black et Scholes.
→ Deux méthodes de résolutions des équations différentielles stochastique par Monte Carlo.
→ Application du Lemme d'Ito à la résolution théorique de l'EDS .
→ Solution par Monte Carlo de l'EDS de Vasicek à l'aide du méthode d'Euler.
→ Calcule de la Fonction d'utlisité et de l'espérances d'un portefeuille.
→ L'équation différentielle stochastique pour l'actif avec des sauts et sa solution analytique.