Wissal GHOUDI

ING 2 GFM1

Simulation de MC

Exo 1 :

Code :

```
disp("Exo 1: Partiel 2021/2022");
question1();
function [] = question1()
 T = 2;
 lambda = 2;
 W(1) = 0;
 N = 100;
 delta_t = T / N;
 Nmc = 1000;
 proba = 0;
 cpt = 0;
 t=(0:N)*delta_t;
 for k = 1:Nmc % simulation de Nmc trajectoire
     for i = 1:N
     W(i + 1) = W(i) + sqrt(delta_t) * randn;
     end
 plot(t,W); % affiche une trajectoire du mouvement brownien
% xlabel 't'
% ylabel 'Processus W_t'
% title 'Trajectoires du Mouvement Brownien'  hold on;
 last_value(k) = W(N + 1);
 if (abs(last_value(k)) < 0.5)
    cpt = cpt + 1;
 end
 a(k) = exp(lambda * last_value(k) - (lambda^2) * T  / 2);
 end
 proba = cpt / Nmc; % calcule proba demander
 disp("proba[ |Wt| < 0.5 ] = " + proba);
 disp("esperance = " + mean(a)); % affichage proba demander
end
```

Résultat :

proba[|Wt| < 0.5] = 0.265

esperance = 0.63777

Exo 2 :

Code :

```
disp("Exo 2: Partiel 2021/2022");
question2();
function [] = question2()
 T = 2;
 W(1) = 0;
 N = 100;
 delta_t = T / N;
 Nmc = 1000;
 I1 = 0;
 I2 = 0;
 t = (0:N) * delta_t;
 %t=linspace(0,T,N+1);

 for i = 1:N
     W(i + 1) = W(i) + sqrt(delta_t) * randn;
     I1 = I1 + t(i) * (W(i + 1) - W(i));
     I2 = I2 + W(i) * delta_t;
 end
 I2 = T * W(N + 1) - I2;
 disp("membre de gauche " + I1); %affiche l'integrale de
gauche
 disp("Membre de droite " + I2); %affiche l'egalite de
droite
 disp("Difference " + abs(I1 - I2)); % verifie si elles
sont egale (soustraction proche de zeros)

end
```

Résultat :

membre de gauche = 0.18501

Membre de droite = 0.18209

Difference = 0.0029188

On a donc montré la formule d'intégration par partie car c'est porche de 0

Exo 3 :

Code :

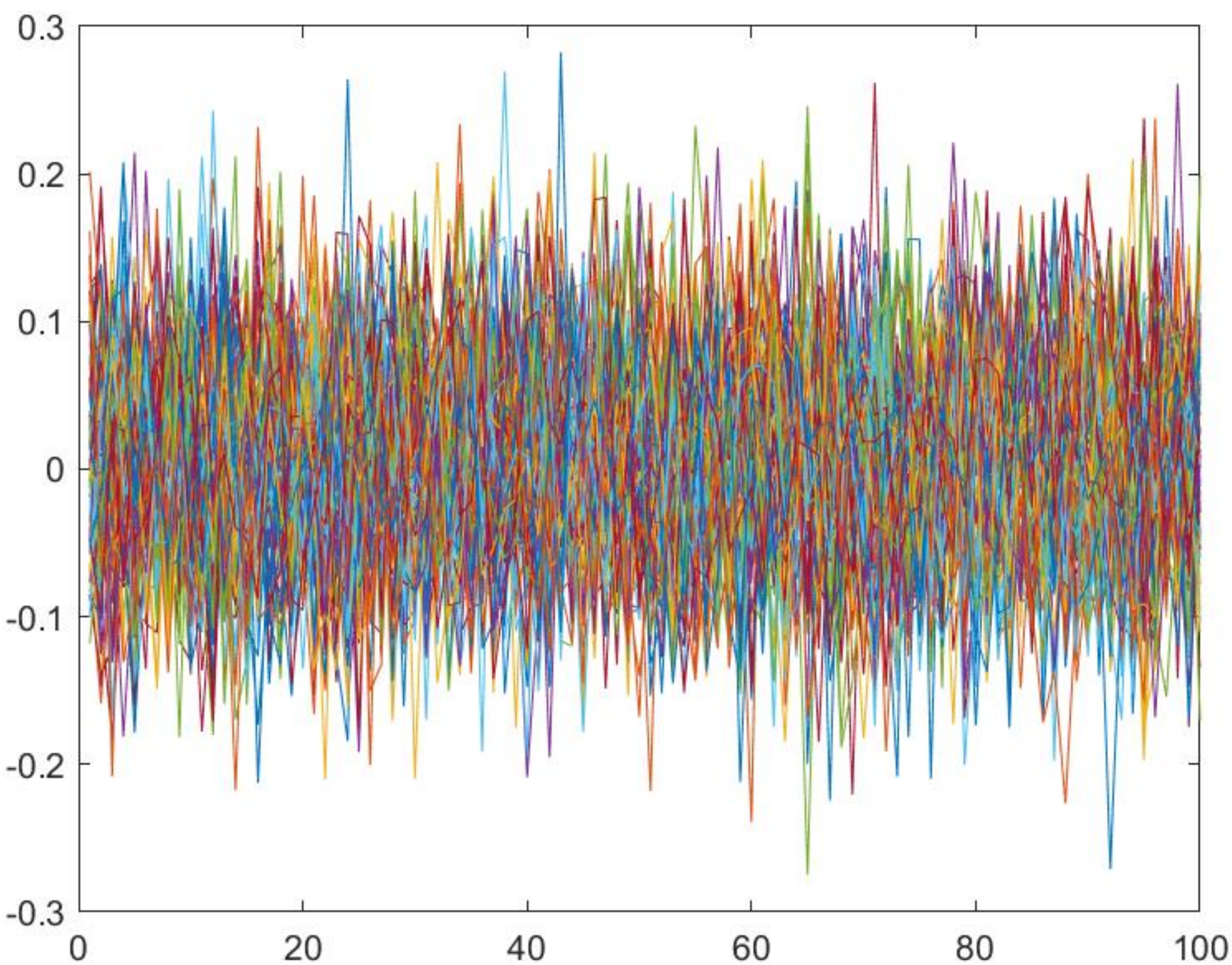
```
disp("Exo 3: Partiel 2021/2022");
%question3_Cas1();
question3_Cas2();
function [] = question3_Cas1()
 T = 2;
 s0 = 10;
 r = 0.4;
 sigma0 = 0.5;
 Nmc = 100;
 N = 100;
 delta_t = T / N;
 W(1) = 0;
 S(1) = s0;
 R(1) = 0;
 t = (0:N) * delta_t;
 for k = 1:Nmc
     for i = 1:N
         W(i + 1) = W(i) + sqrt(delta_t) * randn;
         S(i + 1) = S(i) + S(i)*(r * delta_t + sigma0*(W(i
+ 1) - W(i)));
         R(i)=(S(i+1)-S(i))/S(i);
     end
% plot(t,S);
% hold on;
plot (R); % on plot 100 trajectoire de rendement
hold on;
 DailyReturn(k) = R(N);
 end
densite_Emp_graphe(-0.3,0.01,DailyReturn); % on affiche la
fonction de densité
end
```

```
function[P,x]=fonction_Emp_densite(X,a,delta)
N_x=length(X);
for i =1:N_x+1
 x(i)=a+delta*(i-1);
 cont=0;
     for n=1:length(X)
         if X(n)<=x(i)+delta && X(n)>x(i)
            cont=cont+1;
         end
 end
 P(i)=cont/(length(X));
end
end

function[]=densite_Emp_graphe(a,delta,X)
[P,x]=fonction_Emp_densite(X,a,delta);
figure;
plot(x,P,'ro','MarkerSize',2,'MarkerFaceColor', 'r' );
xlabel 'x'
ylabel 'f_X(x)'
title 'Fonctions de densité '
end

function [k] = sigma(t,r,T,S,sigma0)
 k=sigma0*exp(5*r*t/T)*(sin(S/10)^2);
end

function [] = question3_Cas2()
 T = 2;
 s0 = 10;
 r = 0.4;
 sigma0 = 0.5;
 Nmc = 100;
 N = 100;
 delta_t = T / N;
 W(1) = 0;
 S(1) = s0;
 t = (0:N) * delta_t;
 R(1)=0;

 for k = 1:Nmc
     for i = 1:N
         W(i + 1) = W(i) + sqrt(delta_t) * randn;
```

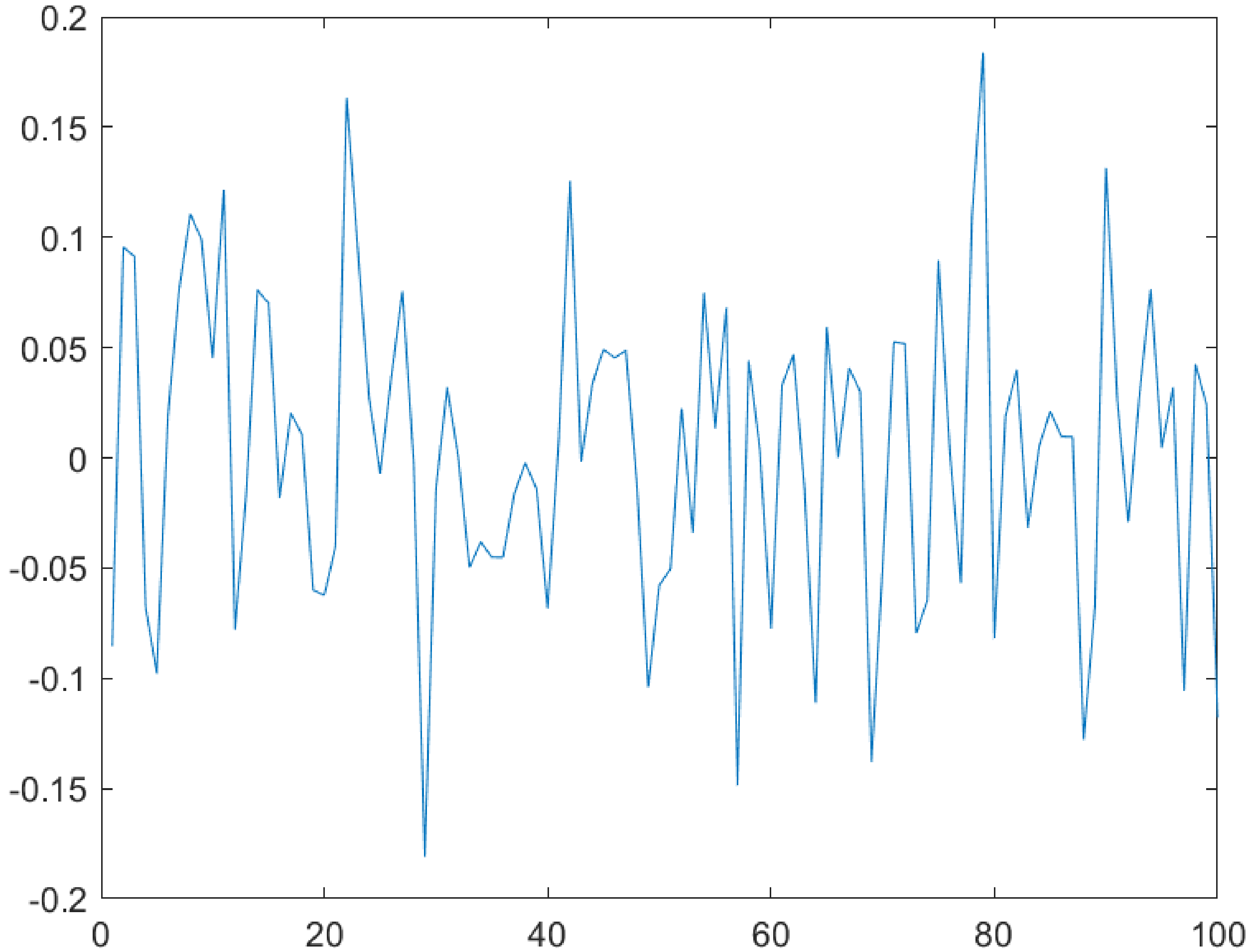
```
        S(i + 1) = S(i)*(1 + r * delta_t +
sigma(t(i),r,T,S(i),sigma0) * (W(i + 1) - W(i)));
        R(i+1) = (S(i+1) - S(i)) / S(i);
    end
        plot(R);
        hold on;
        %plot(t,S);
        %RR=rmmissing(R);
        % RRR=R(N+1);
        DailyReturn(k) = R(N);
 end
 densite_Emp_graphe(-0.3,0.015,DailyReturn);
end
```

Résultat :

1)

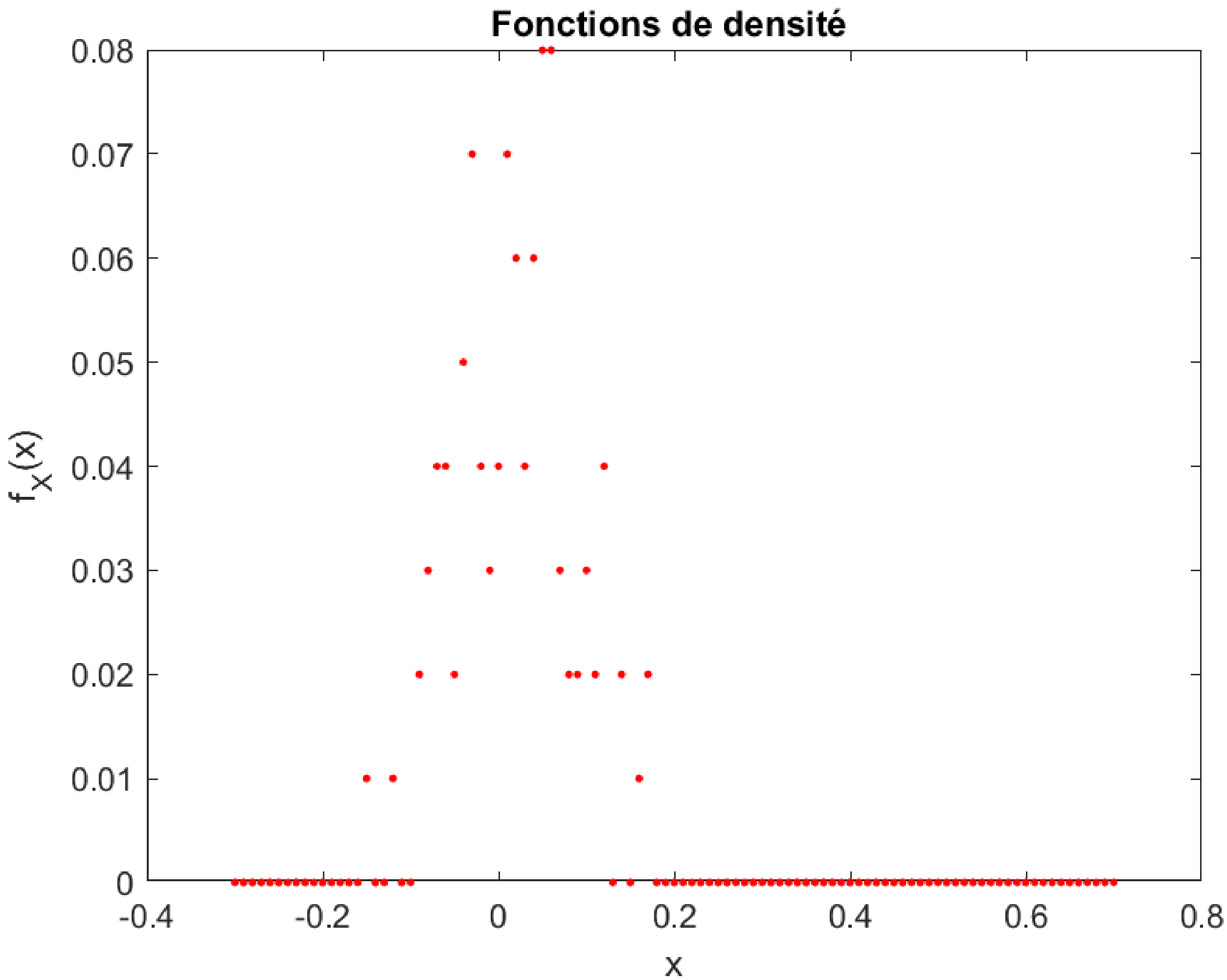


Ci-dessus on voit la simulation de Nmc = 100 trajectoires de rendement qui dépendent de ti



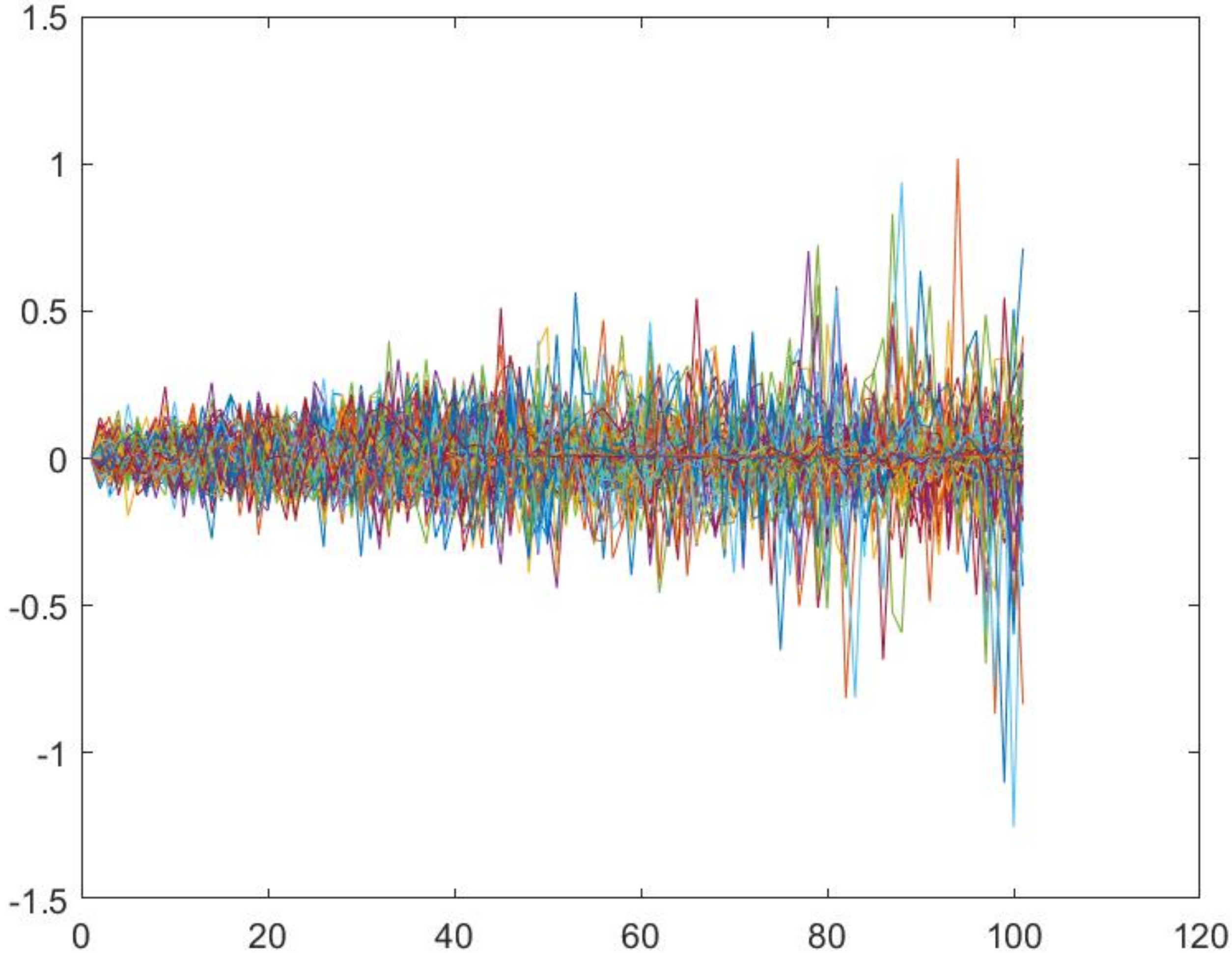
Ce graphique est pour une trajectoire de rendement en fonction de ti

2)



Ci-dessus le graphe de la fonction de densité empirique du rendement final RT

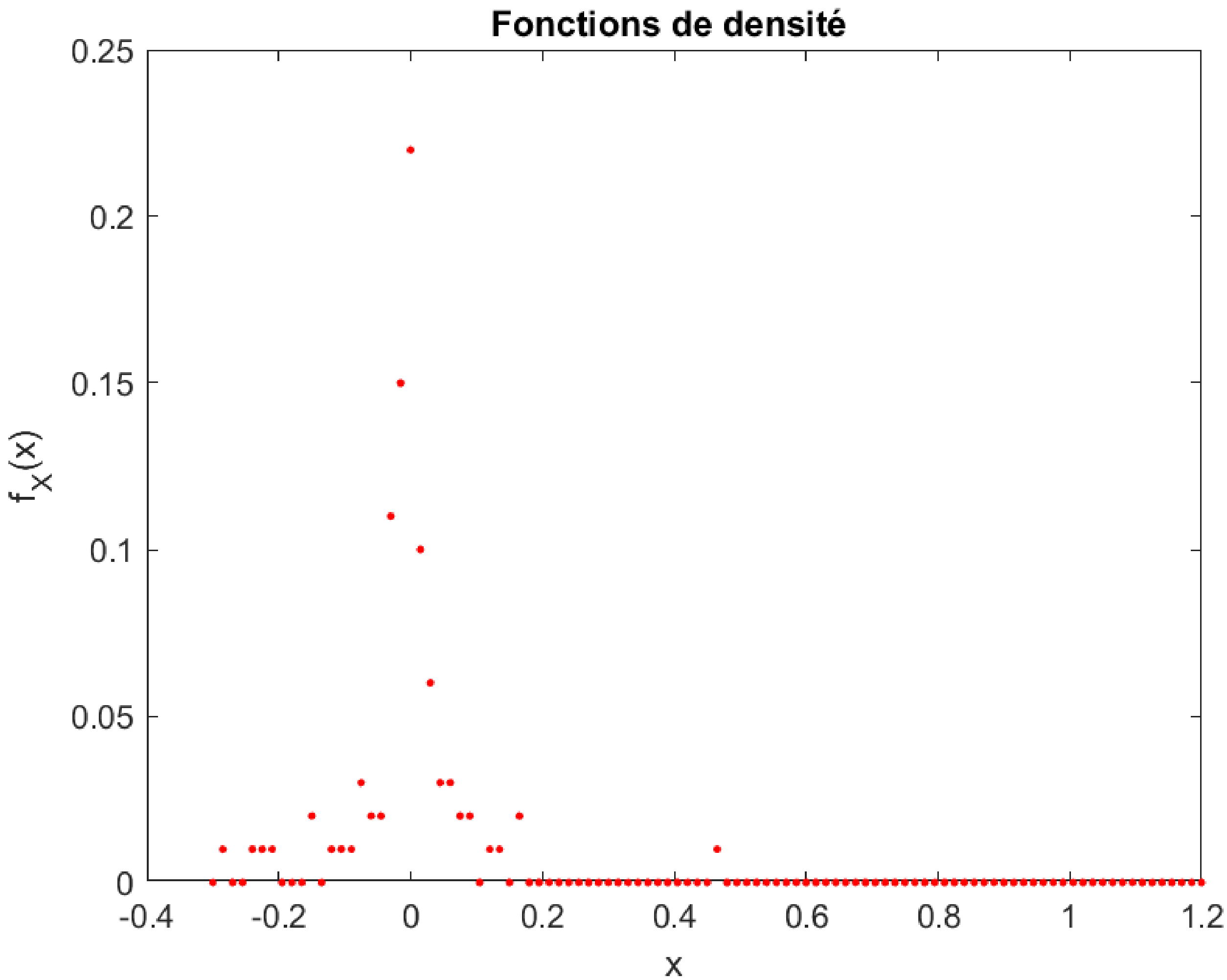
3)



Ci-dessus on voit la simulation de Nmc = 100 trajectoires de rendement qui dépendent de ti.

On peut voir que plus ti est grand plus le rendement devient volatile (dispersion plus élevé)

4)



Ci-dessus la fonction de densité empirique du rendement final RT. On peut voire que comparer a la première avec sigma constant on remarque que Rt est plus volatile et que l'amplitude de la fonction de densité est plus faible ce que peut signifier un rendement plus faible que dans le première modèle

Exo 4 :

Code :

```
disp("Exo 4: Partiel 2021/2022");
question4_Cas1();
%question4_Cas2();
function [] = question4_Cas1()
 T = 0.5;
 r = 0.12;
 sa0 = 100;
 sb0 = 75;
```

```
 na = 110;
 nb = 100;
 sigmaa = 0.2;
 sigmab = 0.3;
 Sa(1) = sa0;
 Sb(1) = sb0;
 W(1) = 0;
 N = 100;
 delta_t = T / N;
 delta_ta = T / na;
 delta_tb = T / nb;
 p0 = na * sa0 + nb * sb0;
 ta = (0:na) * delta_ta;
 tb =(0:nb) * delta_tb;
 P(1) = p0;
 %P = 0;
 cpt = 0;
 proba = 0;
 Nmc=1000;
 for k = 1:Nmc
     for i = 1:na
        W(i + 1) = W(i) + sqrt(delta_ta) * randn;
        Sa(i + 1) = Sa(i)*(1 + r * delta_ta + sigmaa *
(W(i + 1) - W(i)));
     end
     for i = 1:nb
        W(i + 1) = W(i) + sqrt(delta_tb) * randn;
        Sb(i + 1) = Sb(i)*(1 + r * delta_tb + sigmab *
(W(i + 1) - W(i)));
     end
     P(k) = na * Sa(N + 1) + nb * Sb(N + 1);
     if ((P(k) / P(1)) < 0.9)
        cpt = cpt + 1;
     end
% plot(ta, Sa);
% hold on;
% plot(tb, Sb);
% hold on;
% plot( P);
% hold on;
 end
 proba = cpt / Nmc; % calcule la proba en comptent le
nombre d'iteration tels que P(k) /P0 < 0.5 et on divise par
Nmc pour avoir la proba
```

```
 disp("proba [PT/P0<0.9] = " + proba); % affiche la proba
 disp(" Esperance " + mean(P));
end


function [] = question4_Cas2()
 T = 0.5;
 r = 0.12;
 sa0 = 100;
 sb0 = 75;
 na = 110;
 nb = 100;
 sigmaa = 0.2;
 sigmab = 0.3;
 Sa(1) = sa0;
 Sb(1) = sb0;
 W(1) = 0;
 N = 100;
 delta_t = T / N;
 delta_ta = T / na;
 delta_tb = T / nb;
 p0 = na * sa0 + nb * sb0;
 ta = (0:na) * delta_ta;
 tb =(0:nb) * delta_tb;
 P(1) = p0;
 %P = 0;

 proba = 0;
 Nmc=1000;
 p=-1;
 for j =1:200
      cpt = 0;
     for k = 1:Nmc

         for i = 1:na
            W(i + 1) = W(i) + sqrt(delta_ta) * randn;
            Sa(i + 1) = Sa(i)*(1 + r * delta_ta + sigmaa *
(W(i + 1) - W(i)));
         end
         for i = 1:nb
            W(i + 1) = W(i) + sqrt(delta_tb) * randn;
            W(i+1)=p*W(i)+ sqrt(1-p*p)*W(i)*W(i);
            Sb(i + 1) = Sb(i)*(1 + r * delta_tb + sigmab *
(W(i + 1) - W(i)));
```

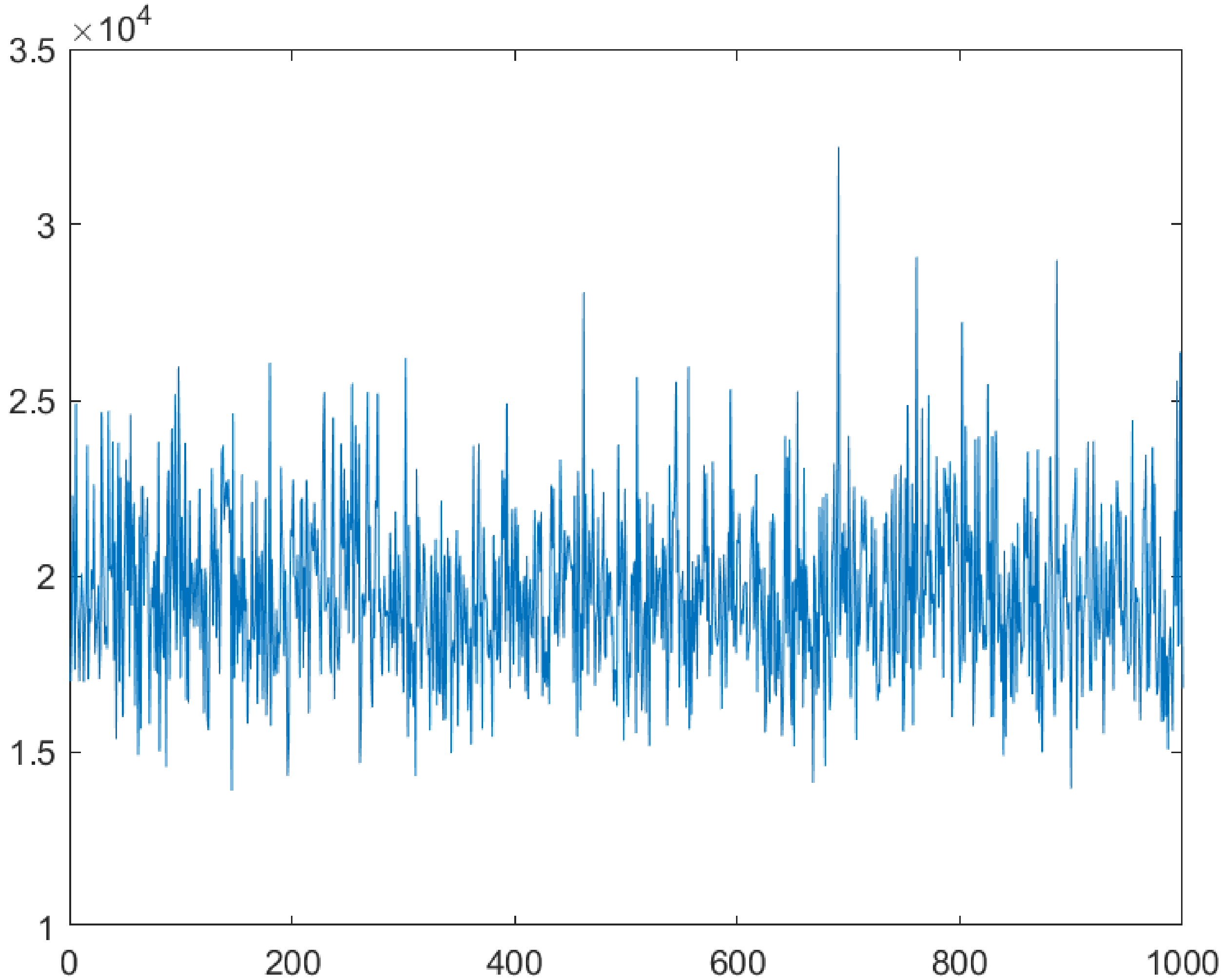
```
        end
        P(k) = na * Sa(N + 1) + nb * Sb(N + 1);
        if ((P(k) / P(1)) < 0.9)
           cpt = cpt + 1;
        end
    % plot(ta, Sa);
    % hold on;
    % plot(tb, Sb);
    % hold on;

    %hold on;
     end

     tp(j)=cpt / Nmc;% stock la proba pour chaque p
     pp(j)=p;% stock le p correspondant
     %proba = cpt / Nmc; % calcule la proba en comptent le
nombre d'iteration tels que P(k) /P0 < 0.5 et on divise par
Nmc pour avoir la proba
     p=p+0.01;
 end
 disp(pp(1));
 disp(tp(1));
plot(pp,tp); % affiche le graphe [PT/P0<0.9] en fonction de
p
 % disp("proba [PT/P0<0.9] = " + proba); % affiche la proba
 disp(" esperance = " + mean(P));
end
```

Résultat :

Cas 1 :

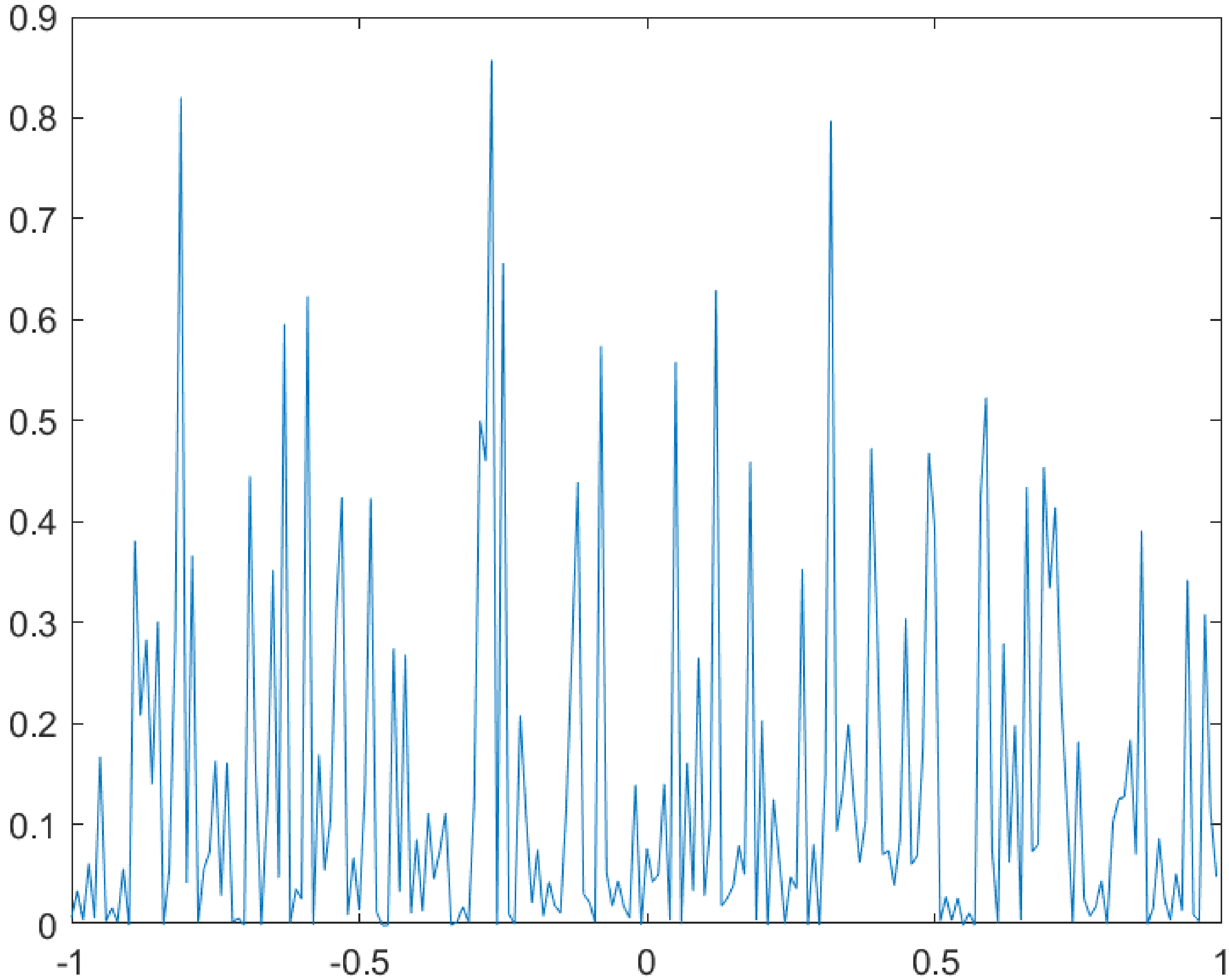
proba [PT/P0<0.9] = 0.209

Esperance = 19506.8469



ci-dessus un graphe pour une trajectoire de P

<u>Cas 2 :</u>



On peut voir ci-dessus proba([PT/P0<0.9]) en fonction de p qui varie entre -1 et 1 j'ai mis un pas de 0.01 que j'incrément à chaque fois. On peut voir que le coefficient de corrélation influe énormément sur la probabilité que la valeur du portefeuille baisse de 10%.

Exo 5 :

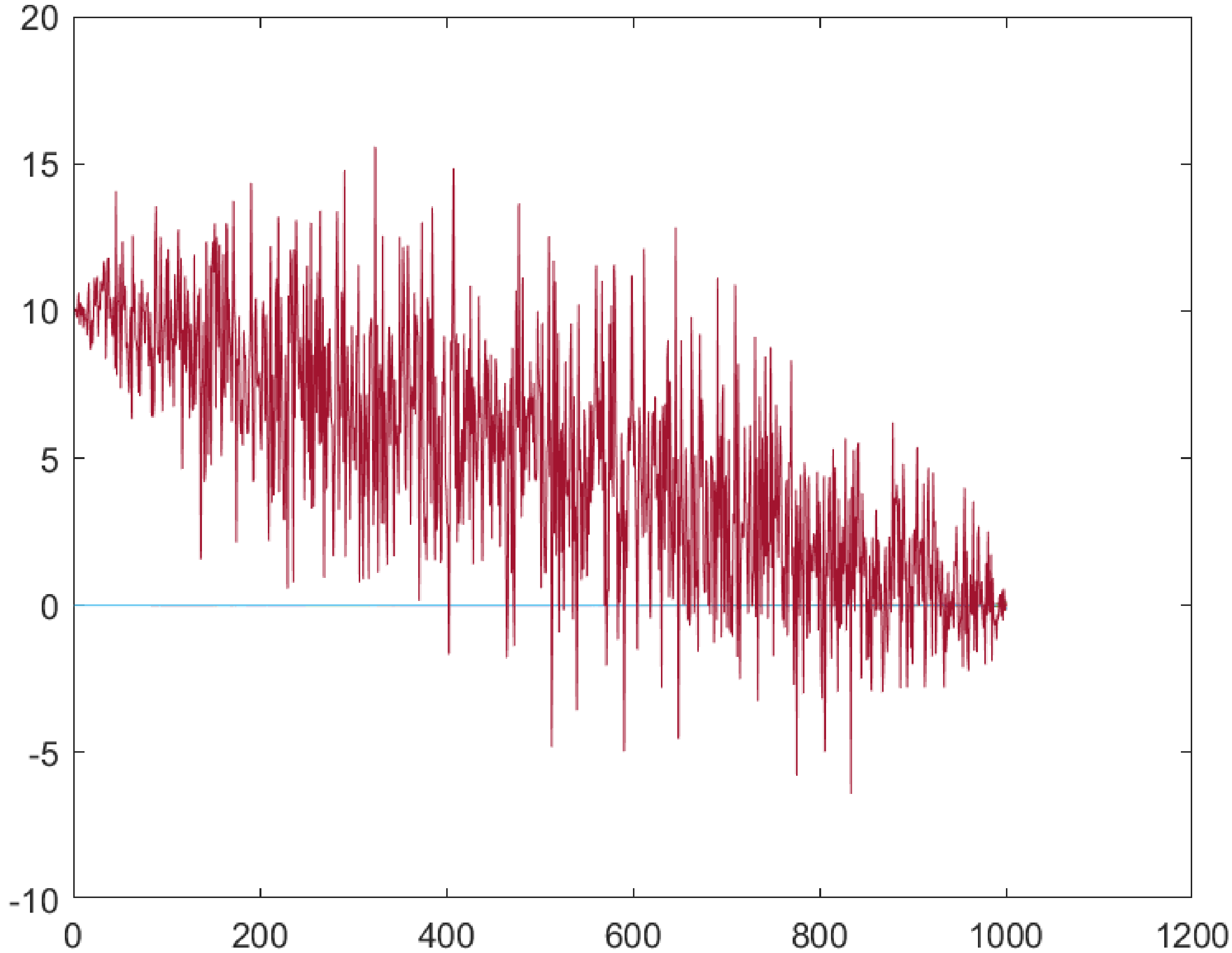
<u>Code :</u>

```
disp("Exo 5: Partiel 2021/2022");
%question5();
question5_2();
```

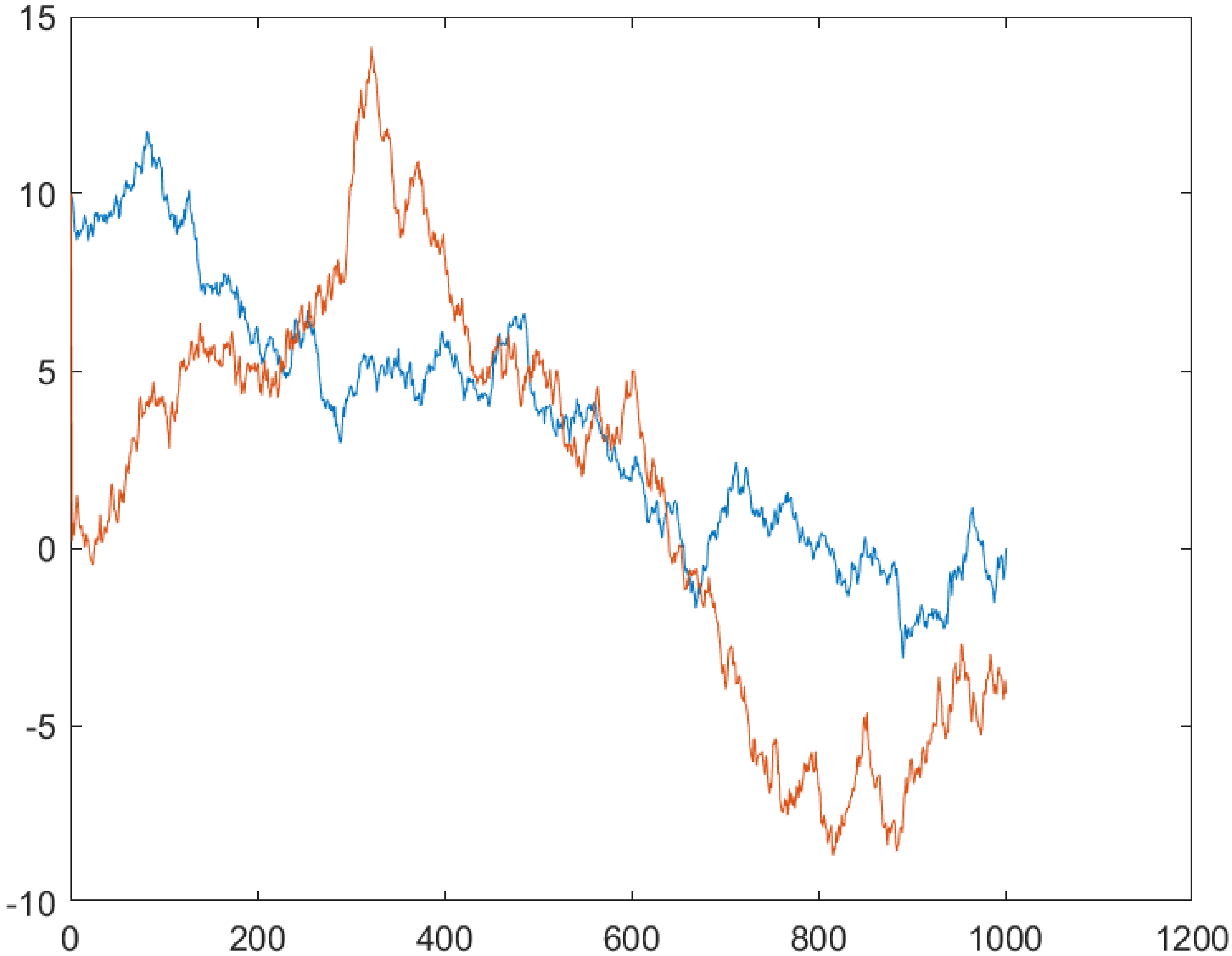
```
function[] = question5()
 T= 2;
 N=1000;
 r=0.1;
 sigma=5;
 delta_t=T/N;
 x0=10;
 X(1)=x0;
 X2(1)=x0;
 W(1)=0;
 t=(0:N)*delta_t;
 Nmc=1000;
 I2=0;
 for k=1:Nmc
     for i=1:N
         W(i+1)=W(i)+sqrt(delta_t)*randn;
         X(i+1)=X(i)-X(i)/(T - t(i))*delta_t+sigma*(W(i+1)-
W(i)); % avec l'equation stoquastique
         I2=I2+(W(i+1)-W(i))/(T-t(i));  % avec la solution
de l'equation

     end
 %plot(X);
 % hold on; % affiche Nmc X
 X2(k+1)=x0*(T-t(N))/T+sigma*(T-t(N))*I2;    %  resultat
avec la solution de l'equation

 end
% figure;
 plot(X); % affiche le gaphe avec l'equation stoquastique
 hold on;
 %plot(X2); % affiche le gaphe avec la solutionde
l'equation
 %plot(X2,'ro','MarkerSize',4,'MarkerFaceColor', 'r');

end

function[] = question5_2()
 T= 2;
 N=1000;
 r=0.1;
 sigma=5;
 delta_t=T/N;
 x0=10;
```

```
 X(1)=x0;
 X2(1)=x0;
 W(1)=0;
 t=(0:N)*delta_t;
 Nmc=1000;
 I(1)=0;
 for k=1:Nmc
 for i=1:N
 W(i+1)=W(i)+sqrt(delta_t)*randn;
 I(i+1)=I(i)+(W(i+1)-W(i))/(N-i);

 end
 plot(I);
 hold on;
 X2(k+1)=x0*(N-k)/N+sigma*(N-k)*I(k);

 end
 plot(X2);
end
```

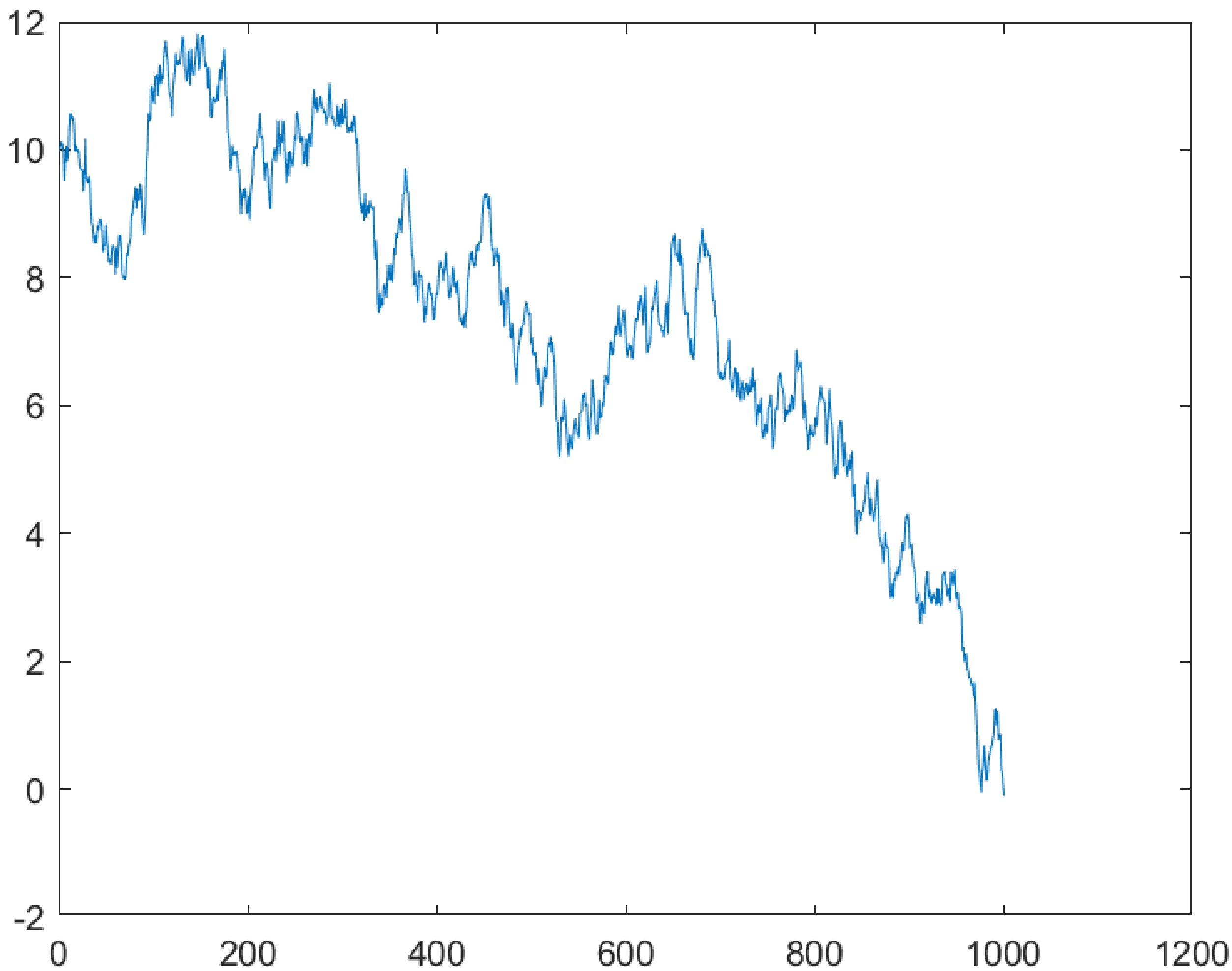
<u>Résultat :</u>



ci-dessus la simulation d'une trajectoire Xt avec la solution de l'équation



Ci-dessus on peut voir un graphe avec les deux méthodes de simulation (avec l'équation différentielle stochastique et avec la solution de l'équation) d'une trajectoire on peut voire que les trajectoires sont assez proches malgré quelque différence



Ci-dessus une trajectoire de Xt simuler avec la méthode aux différences finies

Pour vérifier que la solution de l'équation est bonne on peut vérifier de multiple paramètre comme en vérifiant que les espérances pour les mêmes mouvements brownien sont proches en utilisant la méthode de MC. On peut aussi compare les graphiques obtenus pour chaque trajectoire. Pour conclure