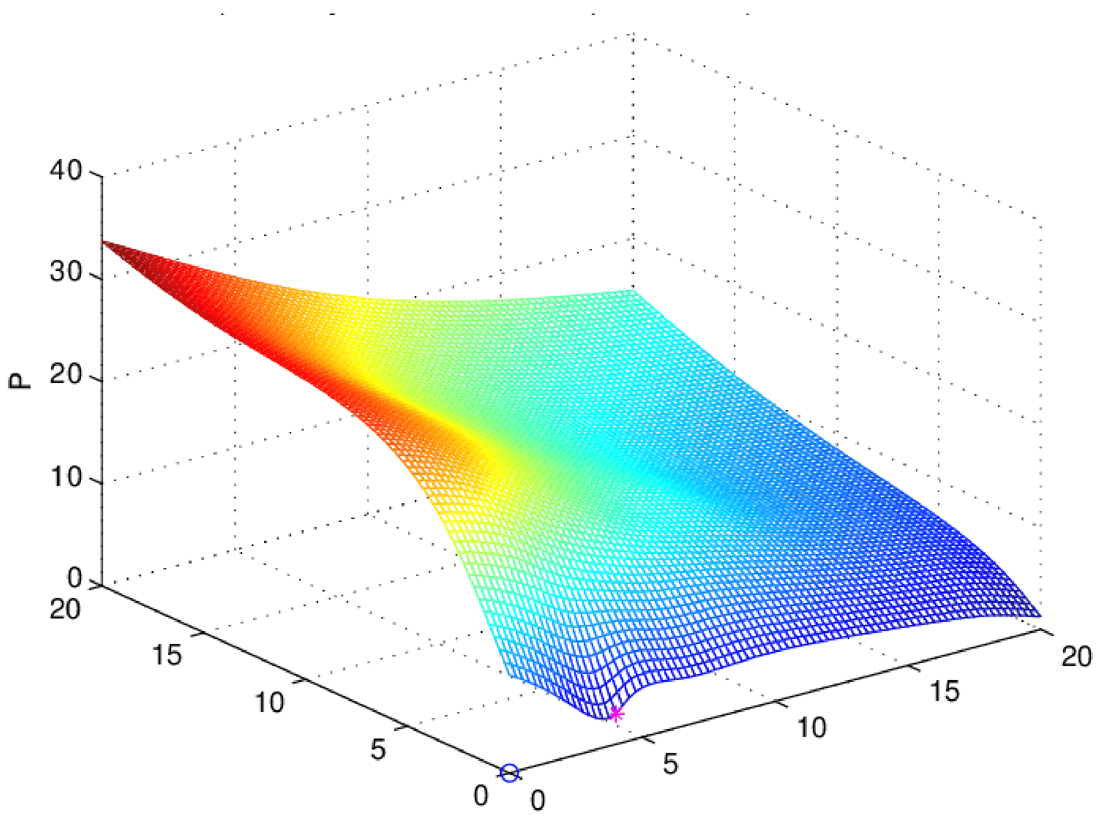
Méthodes Numériques Avancées Pour la Finance : Examen 2020-2021



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

Table des matières

I Introduction

1 Présentation

Les méthodes numériques fondées sur les équations aux dérivées partielles n'étaient pas très populaires Jusqu'à récemment. Les modèles obtenus par des arguments probabilistes et simulés par les méthodes de Monte-Carlo sont en effet plus naturelles et il est plus facile à implémenter des méthodes stochastiques que les algorithmes utilisés pour les EDP. Cependant, quand il est possible discrétiser les EDP, les algorithmes pour la résolution des équations discrétisées sont très efficaces. Les solutions, numériques des EDP donnent plus d'informations. Elles donnent, par exemple, les prix d'une option pour toutes les valeurs initiales de l'actif sous-jacent et pour toutes les valeurs du temps d'exercice, tandis que les méthodes de Monte-Carlo les donnent pour une seule valeur de l'actif et du temps d'exercice. Les EDP sont aussi très efficaces pour le calcul des Greecs.

Les PDF en finance possèdent plusieurs caractéristiques. On travaille sur des domaines temporels et spatiaux finis. On doit impérativement imposer des conditions aux limites et des conditions initiales. Les équations sont souvent paraboliques, mais elles peuvent aussi contenir les termes hyperboliques, comme par exemple l'EDP pour les options asiatiques.

Des modèles de stratégie de hedging du marché non-liquide ont permis d'arriver à une équation de Black et Scholes non linéaire. La resolution numérique des ces équations sont indispensables.

2 Modèle de Black et Scholes

L'actif sous-jacent S est un processus stochastique $S(t)$, $t \in [t, T]$ qui vérifie l'équation différentielle stochastique suivante :

$$dS_t = rS_t dt + \sigma S_t dW_t \text{ avec } S_{t=0} = S_0$$

On note σ la volatilité de l'action, r le taux d'intérêt. La solution de cette équation est donnée par la formule :

$$S(t) = S_0 \exp\left[(r - \tfrac{\sigma^2}{2})t + \sigma W(t)\right]$$

Le prix d'une Option Européenne au moment de temps t est donnée par l'ésperance conditionnelle d'une fonction Pay-Off :

$$V(S_0, t) = e^{-r(T-t)}\mathbb{E}[max(S(T) - K, 0)|S_t = S_0]$$

On peut expliquer ce résultat de façon suivante : on simule un grand nombre de chemins d'évolution de l'actif $S(t)$ sur l'intervalle de temps $[t, T]$, en partant toujours de S_0. Pour chaque chemin on cherche la valeur finale $S(T)$. Puis on calcule la moyenne arithmétique de gains, c'est-à dire de $max(S(T)K, 0)$. Le facteur $e^{-r(T-t)}$ exprime le fait qu'une banque rembourse les intérêt sur l'intervalle du temps $[t, T]$.

3 Simulation d'évolution du prix de l'actif sous-jacent par la méthode aux Différences Finies

L'évolution de l'actif sous-jacent S est un processus stochastique $S(t),\ t \in [0, T]$ qui vérifie l'équation différentielle stochastique suivante:

$$dS_t = rS_t dt + \sigma S_t dW_t, \quad S(t = 0) = S_0$$

On note σ la volatilité de l'actif, r le taux d'intérêt.

La méthode 1 consiste à discrétiser l'équation stochastique par les Différences Finies, puis la simuler.

- Discrétisons l'intérvale $[0, T]$ sur N parties: : $(\Delta t = T/N, \quad t_i = i\,\Delta t)$.
- Discrétisons le différentiel

$$dS(t_i) \sim S(t_i + \Delta t) - S(t_i) \equiv S_{t_{i+1}} - S_i$$

- Discrétisons le différentiel

$$dW(t_i) \sim W(t_i + \Delta t) - W(t_i) \equiv W_{t_{i+1}} - W_{t_i}$$

- Discrétisons l'équation différentielle

$$S_{t_{i+1}} - S_{t_i} = rS_{t_i}\Delta t + S_{t_i}\sigma(W_{t_{i+1}} - W_{t_i})$$

soit

$$S(t_{i+1}) = S_{t_i}((1 + r\Delta t) + \sigma(W_{t_{i+1}} - W_{t_i}))$$

- Simulons pour **un** chemin du mouvement Brownien:

$$\begin{cases} W_0 = 0 \\ W_{t_1} = g_1\sqrt{\Delta t} \\ W_{t_2} = W_{t_1} + g_2\sqrt{\Delta t} \\ W_{t_N} = W_{t_{N-1}} + g_N\sqrt{\Delta t}, \end{cases}$$

où les nombres $\{g_i\}$ suivent la loi Normale $\mathbb{N}(0, 1)$.

- Pour ce chemin du mouvement Brownien simulons le chemin correspondant à l'évolution de l'actif

$$S(t_{i+1}) = S(t_i)((1 + r\Delta t) + \sigma(W_{t_{i+1}} - W_{t_i}))$$

$$\begin{cases} S_0 = S_{t_0} \\ S_{t_1} = S_{t_0}((1 + r\Delta t) + \sigma(W_{t_1} - W_{t_0})) \equiv S_{t_0}((1 + r\Delta t) + \sigma\sqrt{\Delta t}g_1) \\ S_{t_2} = S_{t_1}((1 + r\Delta t) + \sigma(W_{t_2} - W_{t_1})) \equiv S_{t_1}((1 + r\Delta t) + \sigma\sqrt{\Delta t}g_2) \\ S_{t_N} = S(t_{N-1})((1 + r\Delta t) + \sigma(W_{t_N} - W_{t_{N-1}})) \equiv S_{t_{N-1}}((1 + r\Delta t) + \sigma\sqrt{\Delta t}g_N) \end{cases}$$

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.

5 Les grecques des options

Les grecques des options permettent de comprendre le changement du prix d'une prime d'option lorsque les variables changent. Les grecques delta, gamma, véga et thêta permettent de comprendre comment le prix des options se forme. Celles-ci sont primordiales pour le négociateur en options. Ainsi, une prime d'option est déterminée notamment par le cours de la valeur sous-jacente, la durée de l'option jusqu'à son échéance et la volatilité. Ces trois variables ne sont pas constantes. La volatilité sur le marché des actions augmente avec l'incertitude, ce qui génère un gonflement des primes des options.

Le delta Δ : est la plus importante des grecques. Il porte sur la variation du prix d'une option par rapport à la variation du prix de la valeur sous-jacente. Le delta mesure donc l'amplitude de la variation de la prime d'option en cas de hausse ou de baisse de l'action sous-jacente.

Le gamma Γ : mesure la variation du delta qui est engendrée par une variation du cours du sous-jacent. Les options à la monnaie ont le gamma le plus élevé vu qu'une fluctuation de la valeur sous-jacente agit fortement sur le delta de ces options.
(Delta et gamma, vitesse et accélération du prix d'une option).

Le bêta β : désigne la volatilité d'un actif par rapport à son marché. Utilisé dans le Modèle d'évaluation des actifs financiers (MÉDAF), le bêta permet d'évaluer la valeur d'un actif en fonction du taux sans risque et de la rentabilité attendue sur le marché.
(Beta, une mesure relative de la volatilité).

Le véga V : d'une option correspond au taux de variation d'une option consécutive à une variation de la volatilité. Si la volatilité baisse, les cours du sous-jacent fluctueront moins ce qui fera baisser le prix des options. Et à l'inverse, une volatilité qui augmente fera augmenter la fluctuation des cours du sous-jacent et le prix des options monte.
(Véga, le risque lié à la volatilité)

Le thêta Θ : mesure la perte de valeur d'une option au fil du temps. Toutes choses égales par ailleurs, une option perd en effet de la valeur au fur et à mesure qu'elle se rapproche de sa date d'échéance.
(Thêta, le temps c'est de l'argent).

Le rhô ρ : mesure la sensibilité d'une option par rapport à un changement du taux sans risque. Il s'obtient en effectuant la dérivé du prix d'une option par rapport au taux sans risque.
(Rhô, le risque du taux sans risque).

II Problème 1 :

1 Partie 1 :

→ **Question 1 :**

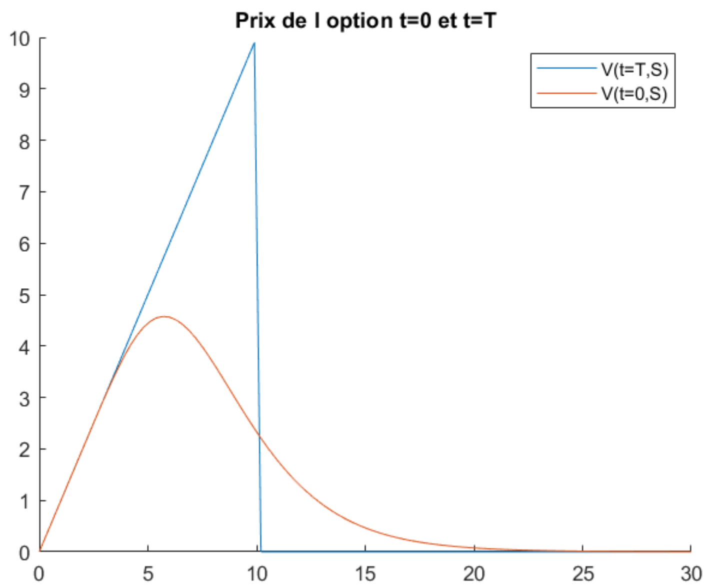


FIGURE 1

→ **Question 2 :**

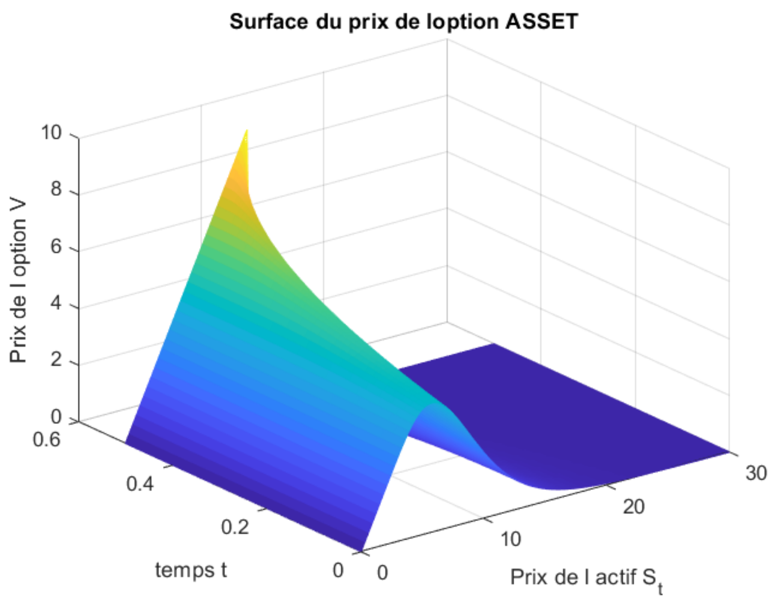


FIGURE 2

→ **Question 3 :**

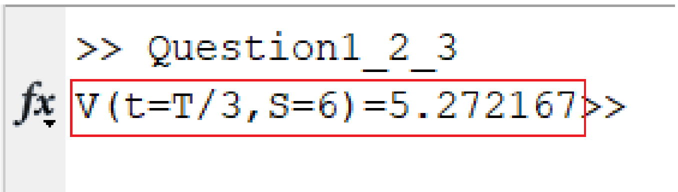


FIGURE 3

→ Code MATLAB Partie 1 :

```
visualiser_fonction_V()

function [] = visualiser_fonction_V()

% Définition des constantes %

L=30;
K=10;
T=0.5;
sigma=0.5;
N=99;
M=4999;

% Définition des vecteurs %

S=linspace(0,L,N+2);
t=linspace(0,T,M+2);
dt=T/(M+2);
ds=L/(N+2);
V=zeros(M+2,N+2);

r=0.4;

% Implementation de la condition finale %

for j=1:N+2
    V(M+2,j)=Pay_off_Asset(S(j),K);
end

% Implementation des conditions aux limites Dirichlet %

for k=1:M+1
    V(k,1)=0;
    V(k,N+2)=0;
end

% Discrétisation de l'équation de Black et Scholes %

for n=M+2:-1:2
for i=2:N+1
    V(n-1,i)=V(n,i)+dt*(r*S(i)*(V(n,i+1)-V(n,i-1))/(2*ds)+(1/2)*(sigma^2)*(S(i)^2)*((V(n,i+1)+V(n,i-1)-2*V(n,i))/(ds^2))-r*V(n,i));
end
end

% Question 1
figure;
hold on;
plot(S,V(M+2,:));
plot(S,V(1,:));
legend('V(t=T,S)','V(t=0,S)');
title('Prix de l option t=0 et t=T')

% Question 2
figure;
mesh(S,t,V)
xlabel('Prix de l actif S_t')
ylabel('temps t')
zlabel('Prix de l option V')
title('Surface du prix de loption ASSET')

% Question 2
fprintf("V(t=T/3,S=6)=%f",V(floor(T/(3*dt))+1,floor(6/(ds))+1));

end

% la condition finale %

function [f] = Pay_off_Asset(S,K)

if (S < K)
```

Figure 4

2 Partie 2 :

→ Question 4 et 5 :

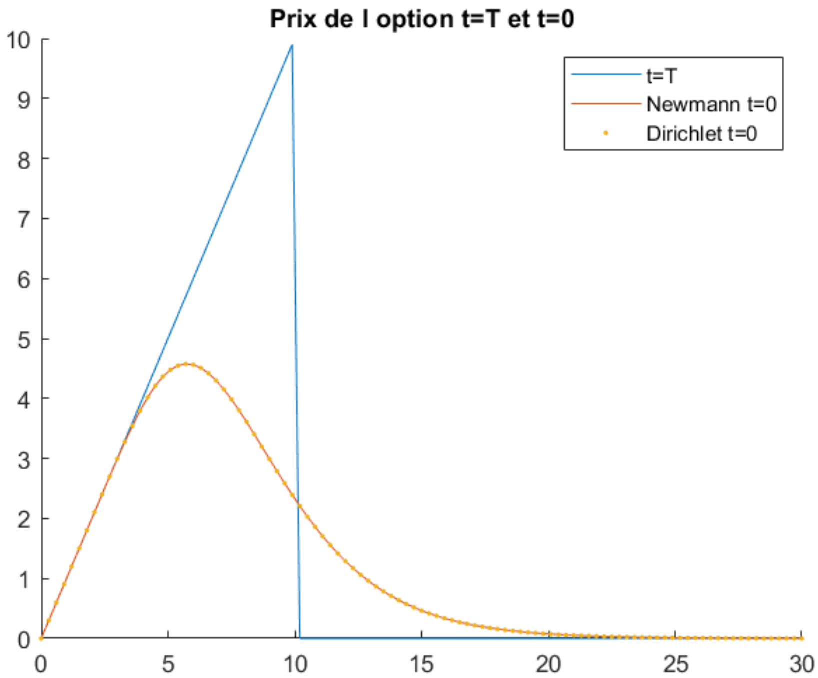


FIGURE 5

→ Code MATLAB Partie 2 :

```
visualiser_fonction_V()

function [] = visualiser_fonction_V()

% Définition des constantes %

L=30;
K=10;
T=0.5;
sigma=0.5;
N=99;
M=4999;

% Définition des vecteurs %

S=linspace(0,L,N+2);
t=linspace(0,T,M+2);
dt=T/(M+2);
ds=L/(N+2);
Vneu=zeros(M+2,N+2);
Vdir=zeros(M+2,N+2);
Vneu=zeros(M+2,N+2);

r=0.4;

% Implementation de la condition finale %

for j=1:N+2
    Vdir(M+2,j)=Pay_off_Asset(S(j),K);
    Vneu(M+2,j)=Pay_off_Asset(S(j),K);
end

% Implementation des conditions aux limites Dirichlet %

for k=1:M+1
    Vdir(k,1)=0;
    Vdir(k,N+2)=0;
end

% Discrétisation de l'équation de Black et Scholes %

for n=M+2:-1:2
for i=2:N+1
    Vdir(n-1,i)=Vdir(n,i)+dt*(r*S(i)*(Vdir(n,i+1)-Vdir(n,i-1))/(2*ds)+(1/2)*(sigma^2)*(S(i)^2)*((Vdir(n,i+1)+Vdir(n,i-1)-2*Vdir(n,i))/(ds^2))-
    Vneu(n-1,i)=Vneu(n,i)+dt*(r*S(i)*(Vneu(n,i+1)-Vneu(n,i-1))/(2*ds)+(1/2)*(sigma^2)*(S(i)^2)*((Vneu(n,i+1)+Vneu(n,i-1)-2*Vneu(n,i))/(ds^2))-

    % Question 4

    % Implementation des conditions aux limites Newmann %

    %Pour S=0
    Vneu(n-1,1)=Vneu(n-1,2)-ds;
    %Pour S=L
    Vneu(n-1,N+2)=Vneu(n-1,N+1);


end
end


% Question 5
figure;
hold on;
plot(S,Vneu(M+2,:));
plot(S,Vneu(1,:));
plot(S,Vdir(1,:),'.','LineWidth',0.01);
legend('t=T','Newmann t=0','Dirichlet t=0');
title('Prix de l option t=T et t=0')



end

% la condition finale %
```

FIGURE 6

3 Partie 3 :

→ Question 6 :

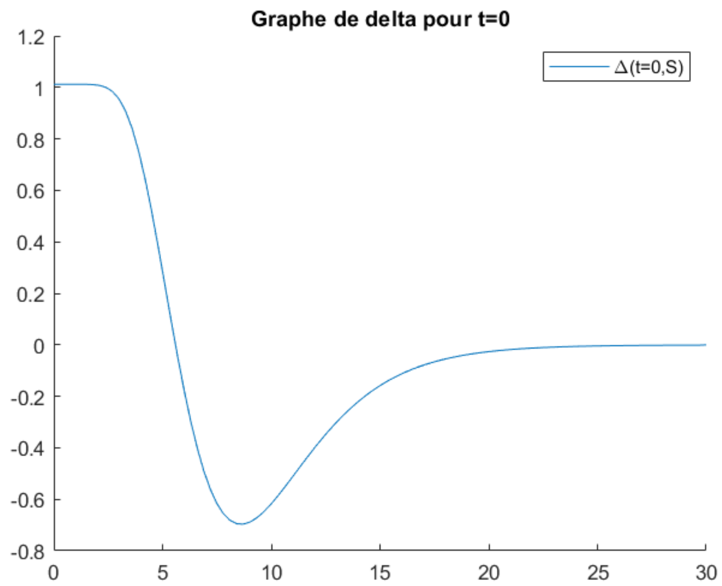


FIGURE 7

→ Question 7 :

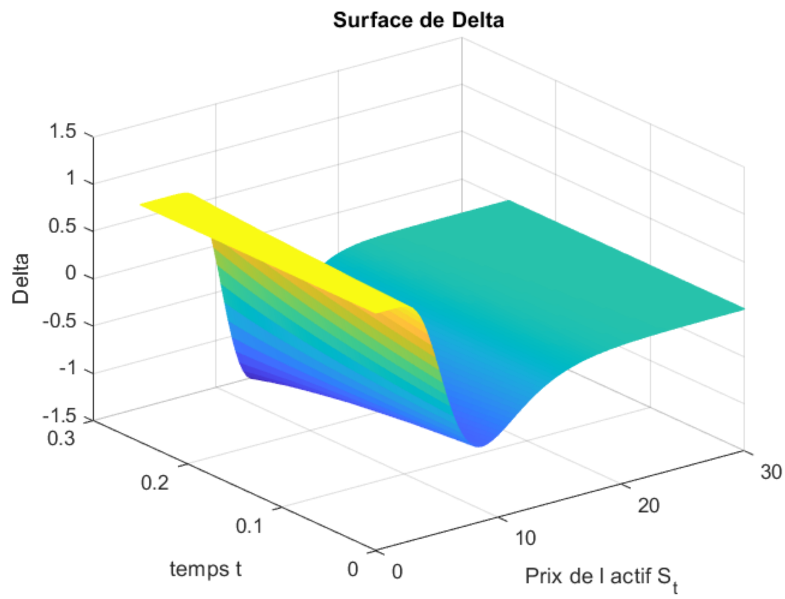


FIGURE 8

→ Code MATLAB Partie 3 :

```
visualiser_delta()

function [] = visualiser_delta()

% Définition des constantes %

L=30;
K=10;
T=0.5;
sigma=0.5;
N=99;
M=4999;

% Définition des vecteurs %

S=linspace(0,L,N+2);
t=linspace(0,T,M+2);
dt=T/(M+2);
ds=L/(N+2);
V=zeros(M+2,N+2);
delta=zeros(M+2,N+2);

r=0.4;

% Implementation de la condition finale %

for j=1:N+2
    V(M+2,j)=Pay_off_Asset(S(j),K);
end

% Implementation des conditions aux limites Dirichlet %

for k=1:M+1
    V(k,1)=0;
    V(k,N+2)=0;
end

% Discrétisation de l'équation de Black et Scholes %

for n=M+2:-1:2
for i=2:N+1
    V(n-1,i)=V(n,i)+dt*(r*S(i)*(V(n,i+1)-V(n,i-1))/(2*ds)+(1/2)*(sigma^2)*(S(i)^2)*((V(n,i+1)+V(n,i-1)-2*V(n,i))/(ds^2))-r*V(n,i));
end
end

for n=1:M+2
    for i=1:N+1
        delta(n,i)=(V(n,i+1)-V(n,i))/ds;
    end
    delta(n,N+2)=0;
end

% Question 6
figure;
hold on;
plot(S,delta(1,:));
legend('\Delta(t=0,S)');
title('Graphe de delta pour t=0')

% Question 2
figure;
mesh(S,t(1:floor(T/(2*dt))+1),delta(1:floor(T/(2*dt))+1,:))
xlabel('Prix de l actif S_t')
ylabel('temps t')
zlabel('Delta')
title('Surface de Delta')

end

% la condition finale %

function [f] = Pay_off_Asset(S,K)

if (S < K)
    f=S;
elseif (S >= K)
    f=0;
end


end
```

FIGURE 9

III Problème 2 :

1 Partie 1 :

→ Question 1 :

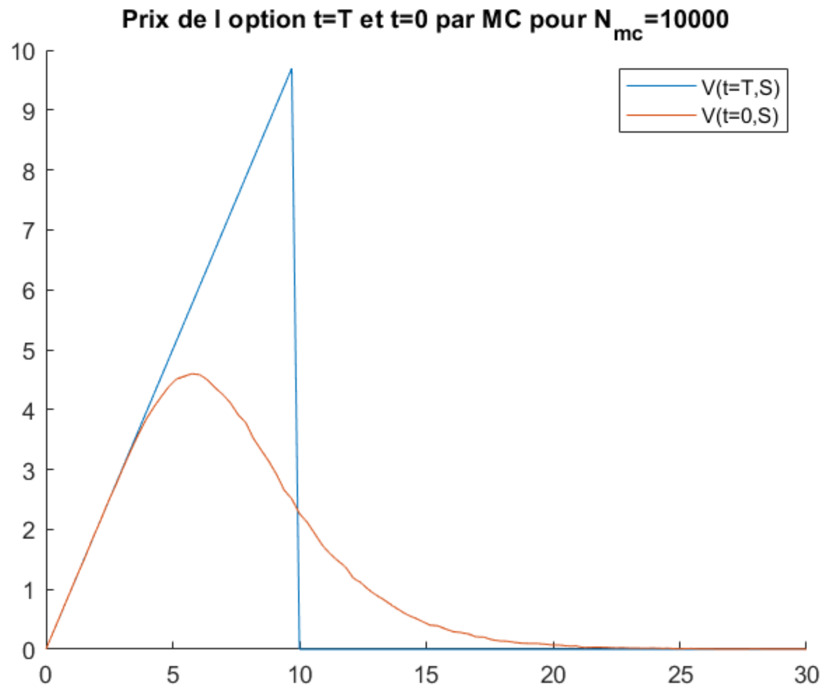


Figure 10

→ Question 2 :

```
>> Question1_2_3
V(t=T/3,S0=6)=4.557992>> Question1_2_3
```

Figure 11

→ Question 3 :

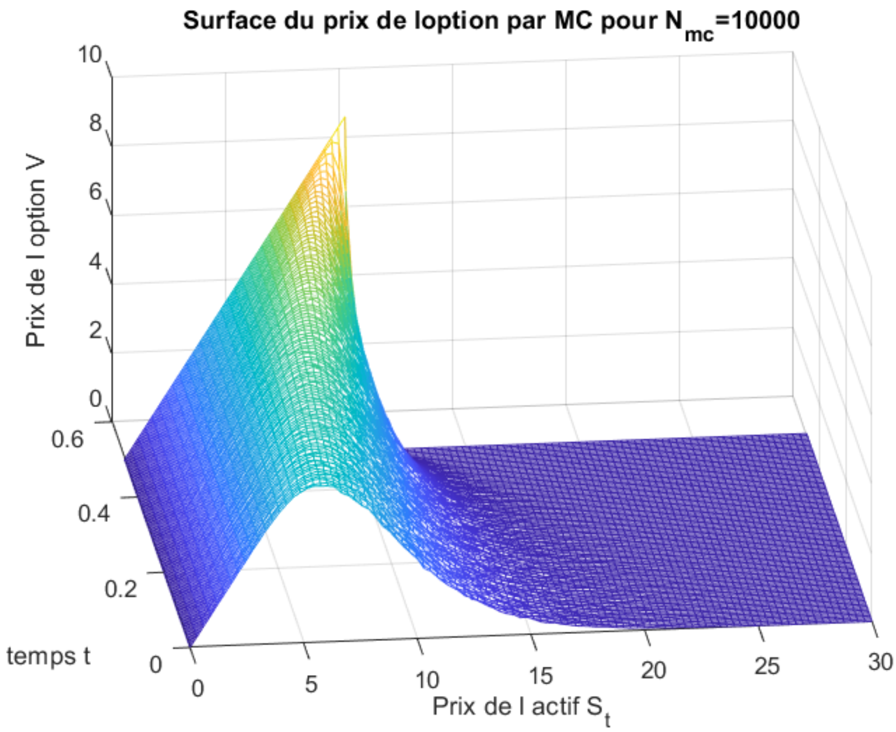


FIGURE 12

→ Code MATLAB Partie 1 :

```matlab

volatilite_fixe()

function [] = volatilite_fixe()

% Définition des constantes et vecteurs %

L=30;
T=0.5;
Ns=99;
Nt=99;
ds=L/Ns;

for j=1:(Ns+1)
    for k=1:(Nt+1)
    St(j)=(L/Ns)*(j-1);
    t(k)=(T/Nt)*(k-1);
    V(k,j)=Prix_Call_St_fixe_t_fixe(t(k),St(j));
    end
end




% Question 1

figure;
hold on;
plot(St,V(Nt+1,:));
plot(St,V(1,:));
legend('V(t=T,S)','V(t=0,S)');
title('Prix de l option t=T et t=0 par MC pour N_{mc}=10000')

% Question 2

fprintf("V(t=T/3,S0=6)=%f",V(floor(T/(3*ds))+1,floor(6/(ds))+1));

% Question 3

figure;
mesh(St,t,V)
xlabel('Prix de l actif S_t')
ylabel('temps t')
zlabel('Prix de l option V')
title('Surface du prix de loption par MC pour N_{mc}=10000')

end

function [prix] = Prix_Call_St_fixe_t_fixe(t,St)

% Définition des constantes et vecteurs %

K=10;
T=0.5;
r=0.4;
sigma=0.5;
Nmc=10000;

for n=1:Nmc
    ST(n)=St*exp((r-(sigma^2)/2)*(T-t)+sigma*sqrt(T-t)*randn);
    gain(n)=Pay_off_Asset(ST(n),K);
end

prix=exp(-r*(T-t))*mean(gain);

end

function [f] = Pay_off_Asset(S,K)

if (S < K)
    f=S;
elseif (S >= K)
    f=0;
end

end
```

Figure 13

2 Partie 2 :

→ Question 4 :

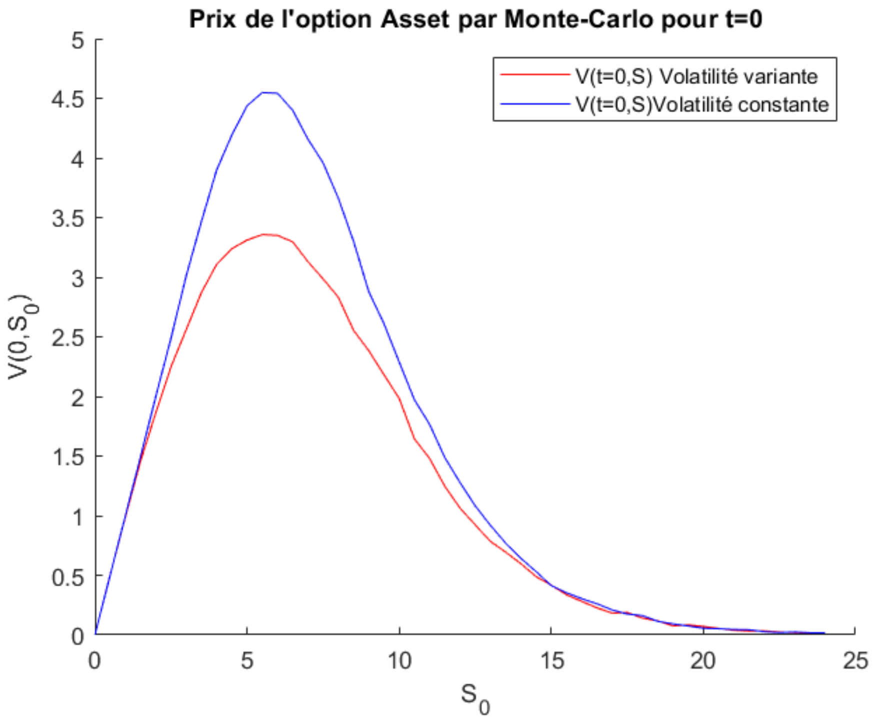


FIGURE 14

→ Code MATLAB Partie 2 :

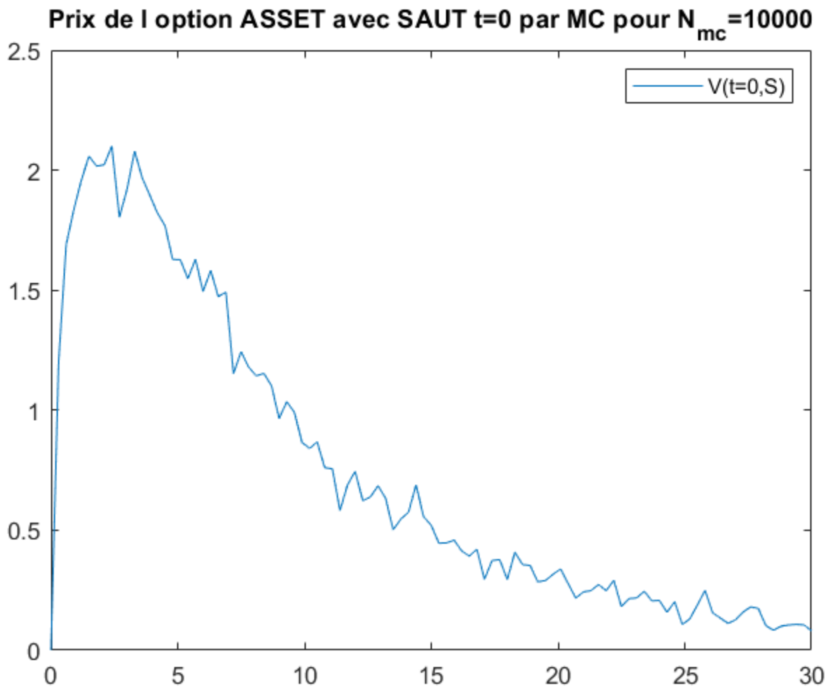
```

% Question 4

Graphe_prix_Asset_valitilite_varie_fixe(10000)

function [] = Graphe_prix_Asset_valitilite_varie_fixe(Nmc)

K=10;

for j=1:49
  S0(j)=0.5*(j-1);
  prix_sigma_varie(j)=prix_Asset_valitilite_varie_S0_fixe(S0(j),Nmc);
  prix_sigma_fixe(j)=prix_Asset_valitilite_fixe_S0_fixe(S0(j),Nmc);
end

figure;
hold on;
plot(S0,prix_sigma_varie,'r');
plot(S0,prix_sigma_fixe,'b');
title("Prix de l'option Asset par Monte-Carlo pour t=0");
legend("V(t=0,S) Volatilité variante","V(t=0,S)Volatilité constante")
xlabel("S_0");
ylabel("V(0,S_0)");
end

function [prix] = prix_Asset_valitilite_varie_S0_fixe(S0,Nmc)

sum=0;
r=0.4;
T=0.5;

for n=1:Nmc
 sum=sum+PayOff_Asset_valitile_varie_S0_fixe(S0);
end

prix=exp(-r*T)*sum/Nmc;

end

function [prix] = prix_Asset_valitilite_fixe_S0_fixe(S0,Nmc)

sum=0;
r=0.4;
T=0.5;

for n=1:Nmc
 sum=sum+PayOff_Asset_valitile_fixe_S0_fixe(S0);
end

prix=exp(-r*T)*sum/Nmc;

end


function [gain] = PayOff_Asset_valitile_varie_S0_fixe(S0)

K=10;
T=0.5;
r=0.4;
N=100;
S(1)=S0;
dt=T/N;

for i=1:N
    g=randn;
    S(i+1)=S(i)*exp((r-(Sigma_variable(S(i),K)^2)/2)*dt+Sigma_variable(S(i),K)*sqrt(dt)*g);
end

gain=Pay_off_Asset(S(N+1),K);

end

function [gain] = PayOff_Asset_valitile_fixe_S0_fixe(S0)

K=10;
T=0.5;
r=0.4;
N=100;
S(1)=S0;
sigma=0.5;
dt=T/N;

for i=1:N
    S(i+1)=S(i)*exp((r-(sigma^2)/2)*dt+sigma*sqrt(dt)*randn);
end

gain=Pay_off_Asset(S(N+1),K);

end


function [f] = Pay_off_Asset(S,K)

if (S < K)
    f=S;
elseif (S >= K)
    f=0;
end

end

function [f] = Sigma_variable(S,K)

sigma=0.5;

if (S < K)
    f=sigma+(K-S)/K;
elseif (S >= K)
    f=sigma;
end

end
```

Figure 15

IV Problème 3 :

1 Partie 1 :

→ **Question 1 :**



→ Code MATLAB Partie 1 :

```
Graphe_Prix_Option_ASSET_SAUT(1000)

function [] = Graphe_Prix_Option_ASSET_SAUT(Nmc)
N=100;
L=30;
for j=1:(N+1)
    S0(j)=L*(j-1)/N;
    prix_option_saut(j)=Prix_S0fixe(S0(j),Nmc);
end

% Question 1

figure;
plot(S0,prix_option_saut);
legend('V(t=0,S)');
title('Prix de l option ASSET avec SAUT t=0 par MC pour N_{mc}=10000')

end


function [prix]=Prix_S0fixe(S0,Nmc)

T=0.5;
K=10;
r=0.4;
sigma=0.5;
lambda=25;
J=0.17;
mu=r-lambda*(exp(J)-1);
sum=0;

for n=1:Nmc
    ST=S0*exp((mu-(sigma^2)/2)*T+sigma*sqrt(T)*randn+Nt(lambda,T)*J);
    if ST<K
        gain=ST;
     else
        gain=0;
     end
    sum=sum+gain;
end
prix=exp(-r*T)*(sum/Nmc);
end


function [Nsauts]=Nt(lambda,t)
n=0;
Tsauts=0;
while Tsauts < t
    ExpSaut=-log(randn)/lambda;
    Tsauts=Tsauts+ExpSaut;
    n=n+1;
end
Nsauts=n-1;
end
```

Figure 16

V Problème 4 :

1 Partie 1 :

→ **Question 1 :**

```
>> Question1_2
V(t=0,S=9)=0.680066>>
```

FIGURE 17

→ **Question 2 :**

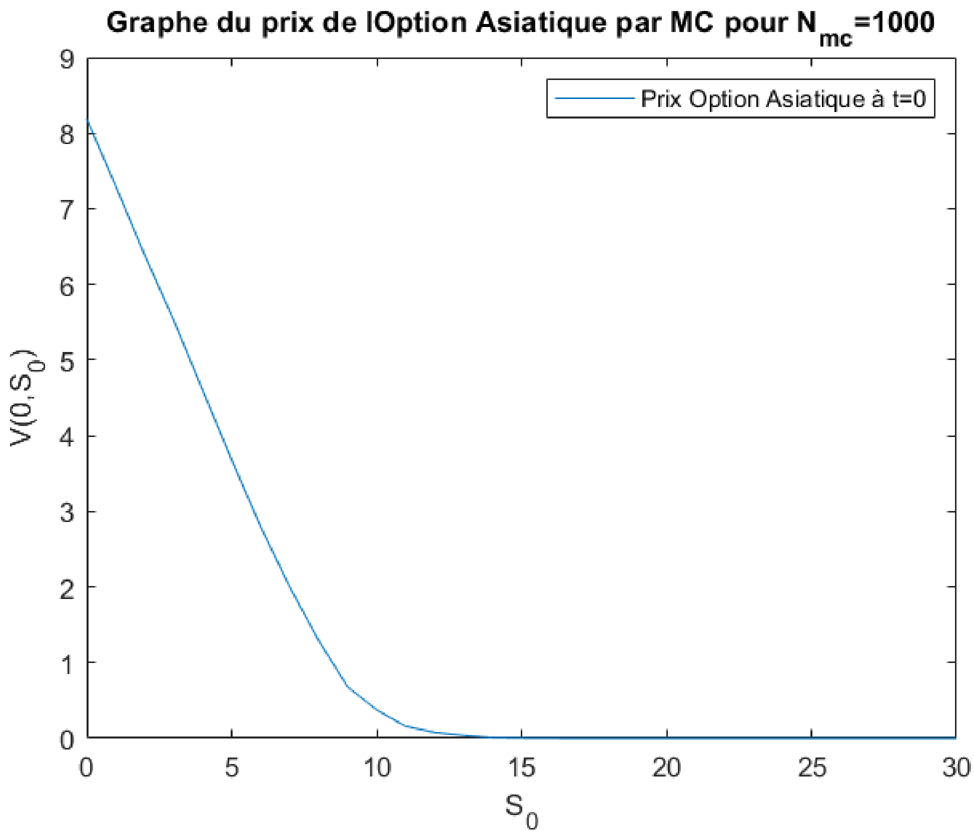


FIGURE 18

→ Code MATLAB Partie 1 :

```matlab
Prix_Asiatique(1000)

function [] = Prix_Asiatique(Nmc)

L=30;
K=10;

ds=1;

for k=1:L+1
    S0(k)=k-1;
    prix_option(k)=Prix_Asiatique_S0_fixe(S0(k),Nmc);
end

% Question 1
fprintf("V(t=0,S=9)=%f",prix_option(1,floor(9/(ds))+1));

% Question 2
plot(S0,prix_option)
xlabel('S_0')
ylabel('V(0,S_0)')
legend('Prix Option Asiatique à t=0')
title('Graphe du prix de lOption Asiatique par MC pour N_{mc}=1000')



end

function [prix] = Prix_Asiatique_S0_fixe(S0,Nmc)

T=0.5;
r=0.4;
sum=0;

for k=1:Nmc
    sum=sum+Payoff_asiatique(S0);
end

prix=exp(-r*T)*(sum/Nmc);

end

function [res] = Payoff_asiatique(S0)

K=10;
T=0.5;
r=0.4;
sigma=0.5;
S(1)=S0;
N=100;
dt=T/N;
produit=1;

for i=1:N
    S(i+1)=S(i)*exp((r-(sigma^2)/2)*dt+sigma*sqrt(dt)*randn);
    produit=produit*S(i);
end

moyenne_geometrique=(produit)^(1/N);

res=max(K-moyenne_geometrique,0);

end
```

FIGURE 19

VI Problème 5 :

1 Partie 1 :

→ Question 1 :

```
V(t=0,S0=6)=4.407756>>
```

FIGURE 20

→ Question 2 :

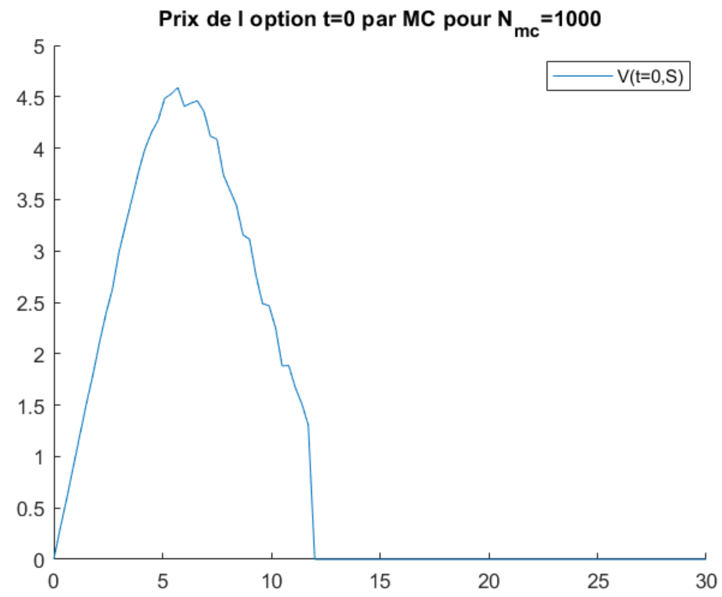


FIGURE 21

→ **Question 3 :**

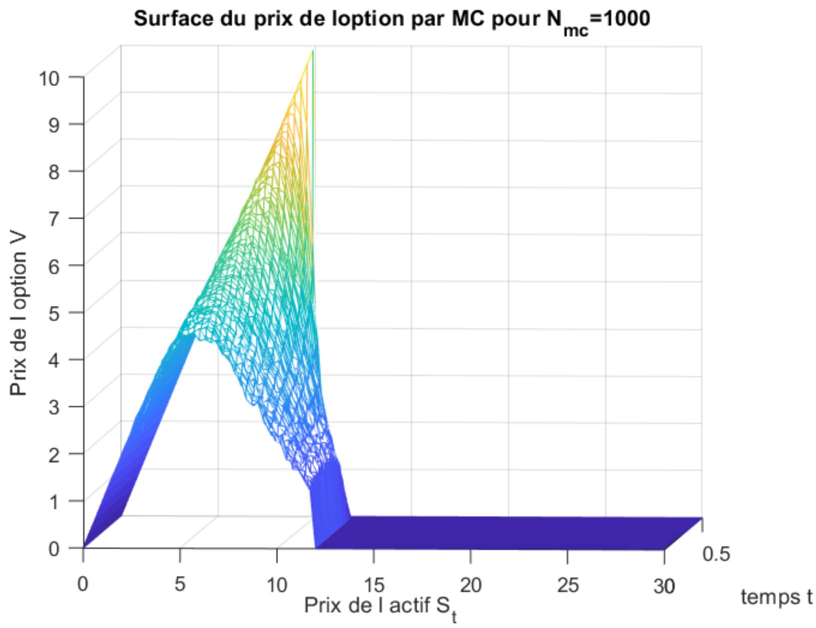


Figure 22

→ Code MATLAB Partie 1 :

```
|
surface_Asset_Barriere()

function [] = surface_Asset_Barriere()

% Définition des constantes et vecteurs %

L=30;
T=0.5;
Ns=100;
Nt=100;
ds=L/Ns;
B=12;

for j=1:(Ns+1)
    for k=1:(Nt+1)
    St(j)=(L/Ns)*(j-1);
    t(k)=(T/Nt)*(k-1);
    if St(j)<B
    V(k,j)=Prix_Asset_Barriere_St_fixe_t_fixe(t(k),St(j));
    else
    V(k,j)=Prix_Asset_Barriere_St_fixe_t_fixe(t(k),0);
    end
    end
end


% Question 1

fprintf("V(t=0,S0=6)=%f",V(1,floor(6/(ds))+1));

% Question 2

figure;
hold on;
plot(St,V(1,:));
legend('V(t=0,S)');
title('Prix de l option t=0 par MC pour N_{mc}=1000')


% Question 3

figure;
mesh(St,t,V)
xlabel('Prix de l actif S_t')
ylabel('temps t')
zlabel('Prix de l option V')
title('Surface du prix de loption par MC pour N_{mc}=1000')

end

function [prix] = Prix_Asset_Barriere_St_fixe_t_fixe(t,St)

% Définition des constantes et vecteurs %

B=12;
K=10;
T=0.5;
r=0.4;
sigma=0.5;
Nmc=1000;

for n=1:Nmc
    if St<B
    ST(n)=St*exp((r-(sigma^2)/2)*(T-t)+sigma*sqrt(T-t)*randn);
    gain(n)=Payoff_Asset_Barriere(ST(n),K);
    else
    gain(n)=Payoff_Asset_Barriere(0,K);
    end
end

prix=exp(-r*(T-t))*mean(gain);

end

function [f] = Payoff_Asset_Barriere(S,K)

B=12;

if (S < K)
    f=S;
```

Figure 23

VII 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, elle a fait un excellant enseignement à distance.

Le cours Méthodes numériques avancée appliquée à 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 numériques avancée appliquée à la finance" pour l'année 2020-2021, j'ai accumulé de nombreuses connaissances :

→ Résolution numérique de l'équation de Black et Scholes.
→ Méthode aux Différences Finies. Discrétisation par les méthodes d'Euler explicite.
→ Programmation des solutions analytiques de Black et Sholes.
→ Volatilité locale.
→ Implémentation des conditions aux limites de Dirichlet et de Neumann.
→ Le lien entre les méthodes déterministe et celle de Monté-Carlo. Surface des prix.
→ Evaluation du prix de l'option Vanilla par la méthode de Monté-Carlo.
→ Options Call, Butterfly par Monte-Carlo.
→ Méthode de la réduction de la variance.
→ Options Exotiques par Monté-Carlo.
→ Options Asiatiques et Lookback. Strike fixe. Strike flotant.
→ Option Lookback par la méthode aux Différences Finies.