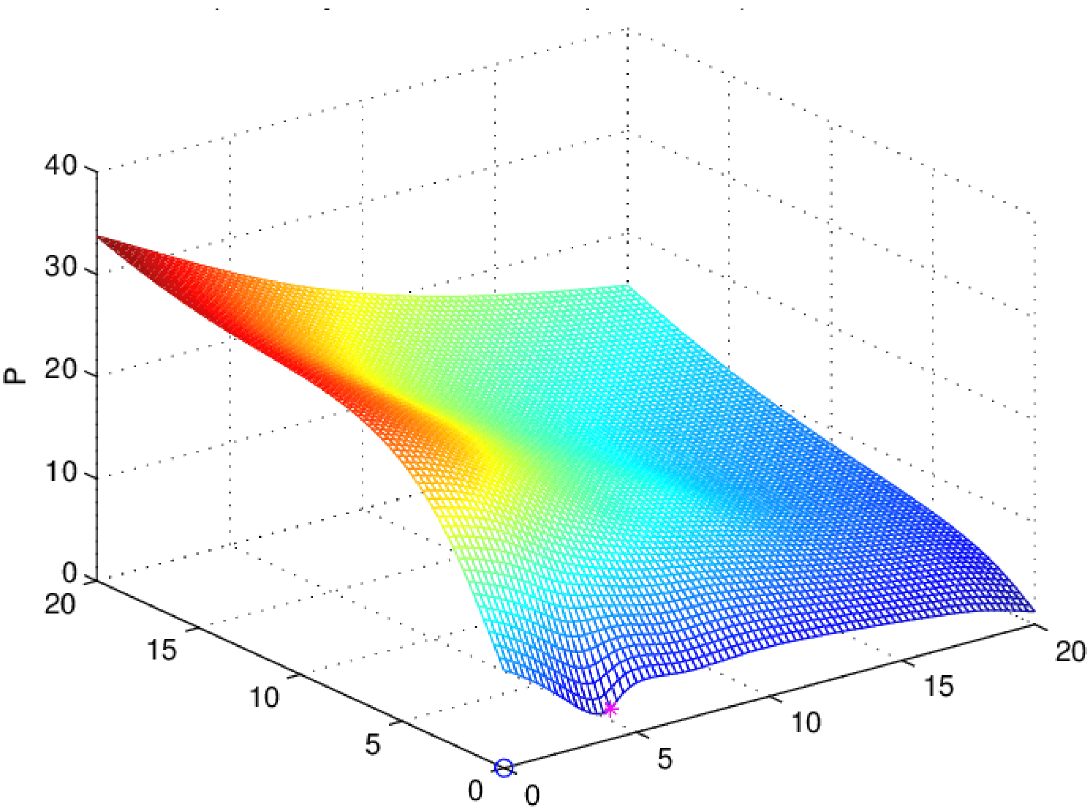
Méthodes Numériques Avancées Pour la Finance : TP3



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

Table des matières

TP: Méthodes numériques avancées appliquées à la finance

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: : $\quad (\Delta t = T/N, \quad t_i = i\,\Delta t)$.
- 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

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

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

II TP3 : Pricing de l'Option Asiatique par la Méthode de Monte-Carlo. Strike fixe. Call.

Le but de ce TP est la simulation numérique du prix de l'option Asiatique par la méthode de Monte-Carlo.

Prix d'une options asiatiques $V(t, S_t, A_t)$ porte sur un actif S_t et sur la valeur moyenne de l'actif :

$$A_t = \frac{1}{t}\int_0^t S_\tau \,\mathrm{d}\tau$$

Prix de l'option à la date T d'écheance :

$$V(T, S(T), A(T)) = max(A(T) - K, 0) \ , \ A(T) = \frac{1}{T}\int_0^T S_\tau \,\mathrm{d}\tau$$

1 Partie I. Simulation d'évolution du prix de l'actif

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:

$$dS_t = rS_t dt + \sigma S_t dW_t, \quad S(t=0) = S_0 \tag{1}$$

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

La solution de cette équation est donnée par la formule:

$$S(t) = S_0 \exp((r - \frac{\sigma^2}{2})t + \sigma W(t))$$

Discrétisons l'intervalle $[0, T]$ sur K parties: $\quad (\Delta t = T/N, \quad t_n = n\,\Delta t)$.

$$S_{t_0} = S_0, \quad S_{t_n} = S_n, \quad S_{t_N} = S(T)$$

On peut écrir pour $S_{t_{n+1}}$:

$$S_{n+1} = S_n \exp((r - \frac{\sigma^2}{2})\,\Delta t + \sigma(W(t_{n+1}) - W(t_n)))$$

Le processus stochastique $W(t_{n+1}) - W(t_n)$ suit la loi $\mathbb{N}(0, t_{n+1} - t_n) \equiv \mathbb{N}(0, \Delta t)$.

On peut donc coder $W(t_{n+1}) - W(t_n)$ par la variable aléatoire :

$$W(t_{n+1}) - W(t_n) = \sqrt{\Delta t}\,\mathbb{N}(0, 1)$$

Pour la suite utiliser les valeurs suivantes

$$\boxed{S_0 = 10} \ \boxed{r = 0.4} \ \boxed{\sigma = 0.3} \ \boxed{T = 1} \ \boxed{K = 10} \ \boxed{\Delta t = T/N} \ \boxed{N = 100} \tag{2}$$

1) $\rightarrow$ **Simuler un schéma d'evolution de l'actif S.**

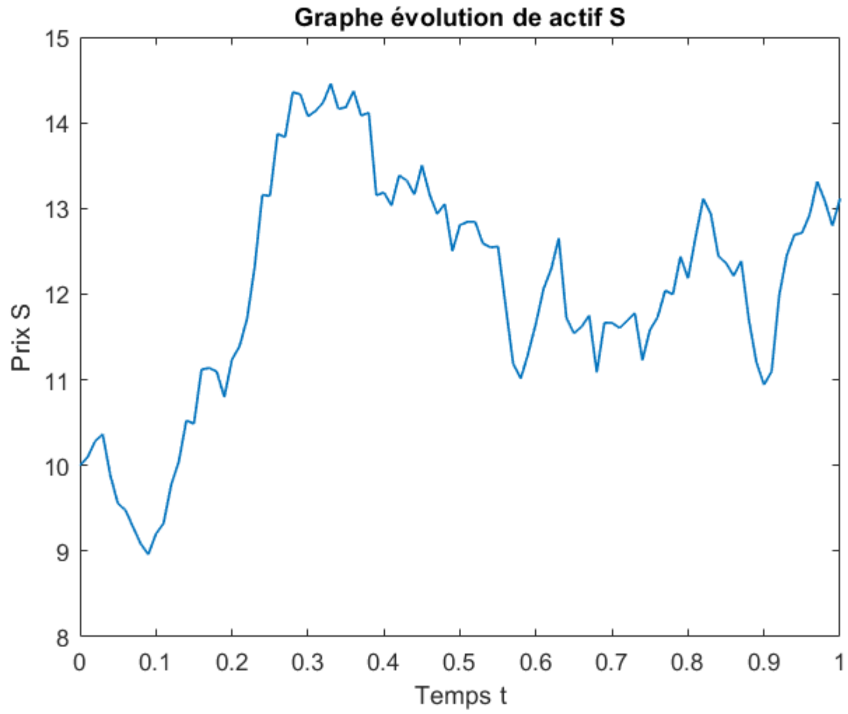


FIGURE 1 – Graphe d'evolution de l'actif S

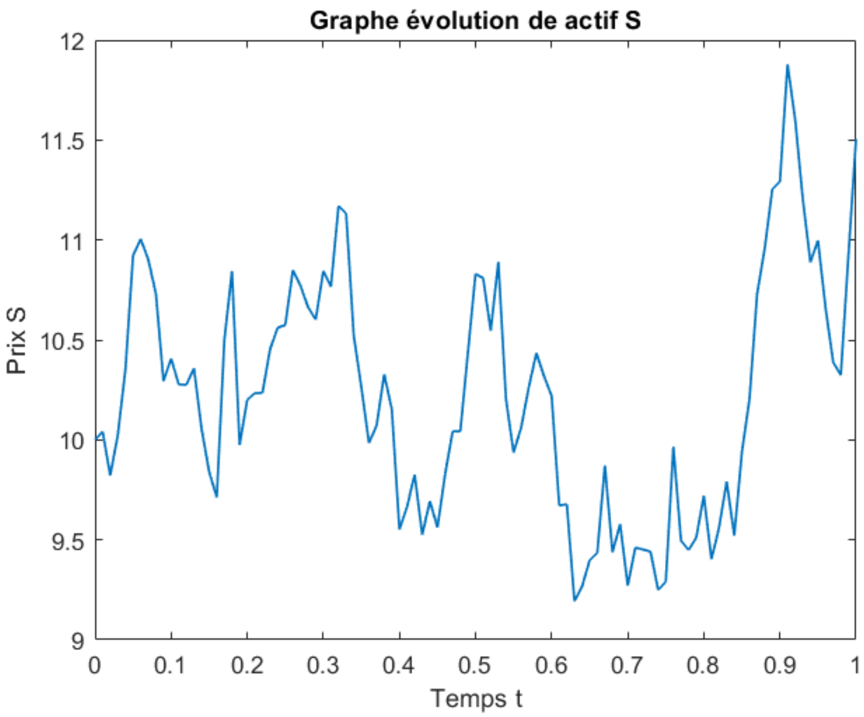


FIGURE 2 – Graphe d'evolution de l'actif S

Le code sous Matlab du graphe d'evolution de l'actif S :

```

Graphe_Prix_S(10)

function [] = Graphe_Prix_S(S0)

K=10;
T=1;
r=0.4;
sigma=0.3;
S(1)=S0;
N=100;
dt=T/N;
t=linspace(0,T,N+1);

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

figure;
plot(t,S,'LineWidth',1)
xlabel('Temps t')
ylabel('Prix S')
title('Graphe évolution de actif S')

end
```

Figure 3 – Code Sous MATLAB du fichier Code1.m

2) → **Simuler N_{mc} schémas d'evolution de l'actif S.**

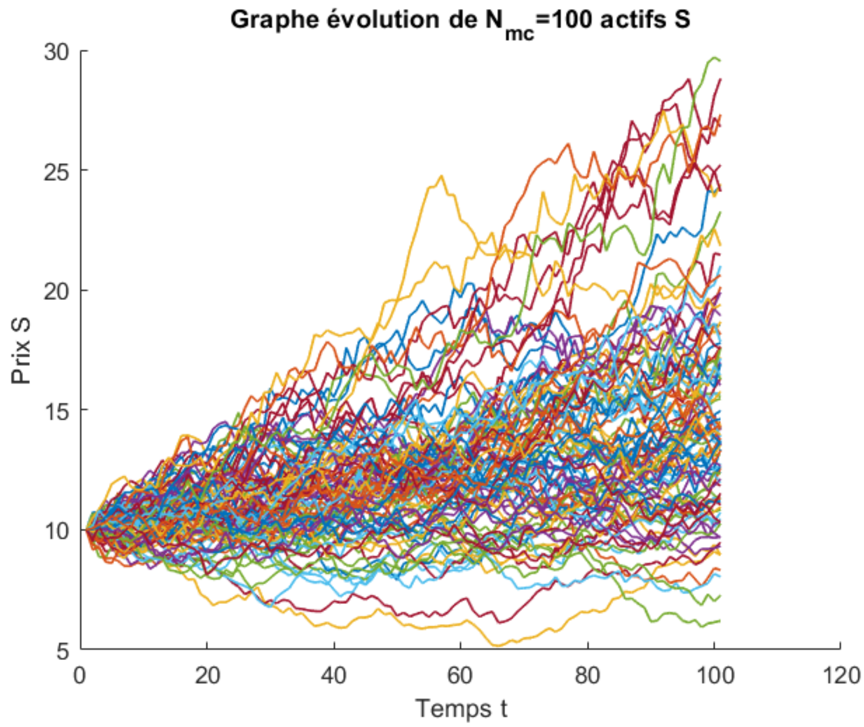


FIGURE 4 – Graphe d'evolution de $N_{mc} = 100$ actifs S

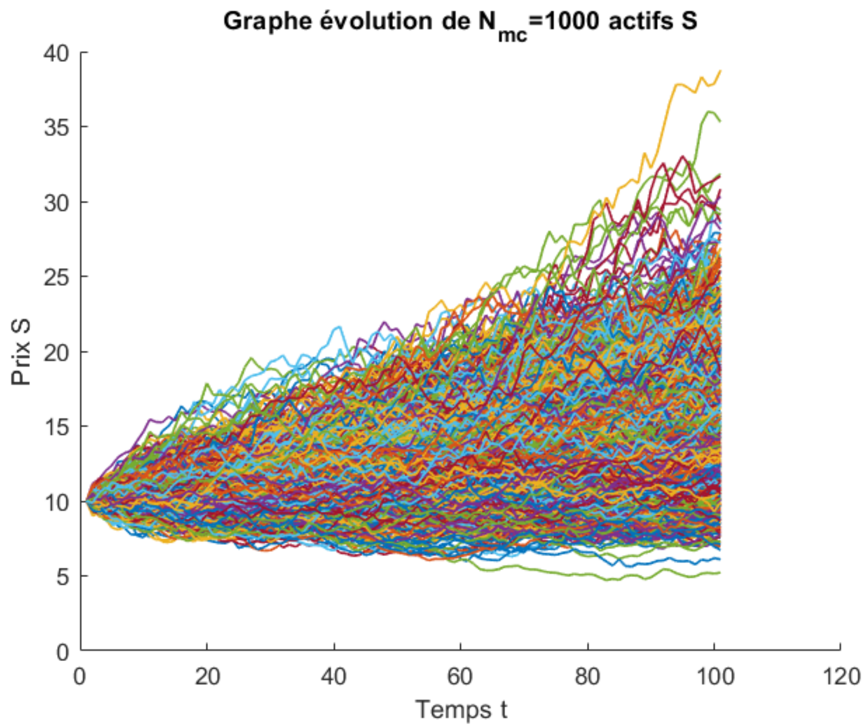


FIGURE 5 – Graphe d'evolution de $N_{mc} = 1000$ actifs S

Le code sous Matlab du graphe d'evolution de N_{mc} actifs S :

```
Graphe_Prix_S_MC(10,1000)

function [] = Graphe_Prix_S_MC(S0,Nmc)

K=10;
T=1;
r=0.4;
sigma=0.3;
S(1)=S0;
N=100;
dt=T/N;
t=linspace(0,T,N+1);

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

hold on
plot(S,'LineWidth',1)
xlabel('Temps t')
ylabel('Prix S')
title('Graphe évolution de N_{mc}=1000 actifs S')

end

end
```

FIGURE 6 – Code Sous MATLAB du fichier Code2.m

2 Partie II. Prix de l'option Asiatique à t = 0

Grace au Théoréme de Feynmann-Kac le prix de l'option Asiatique au moment t=0 est donné par l'ésperance conditionnelle :

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

Partie II.1 Fonction payoff de l'option Asiatique pour S_0 fixe

3) $\rightarrow$ Simulation du Payoff de l'option Asiatique :

```
>> Code3

ans =

    0.8361

>> Code3

ans =

    0.2193

>> Code3

ans =

    2.5084

>> Code3

ans =

    0
```

FIGURE 7 – Simulation du Payoff de l'option Asiatique

Le code sous Matlab du payoff de l'option asiatique :

```
Payoff_asiatique(10)

function [res] = Payoff_asiatique(S0)

K=10;
T=1;
r=0.4;
sigma=0.3;
S(1)=S0;
N=100;
dt=T/N;
sum=0;

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

res=max(sum/T-K,0);

end
```

FIGURE 8 – Code Sous MATLAB du fichier Code3.m

Partie II.2 Le prix de l'option Asiatique pour S_0 fixe

4) $\rightarrow$ Calcul du prix de l'option Asiatique pour S_0 fixe :

Pour $S_0 = 50$:

```
>> Code4

ans =

   34.6314

>> Code4

ans =

   34.5192

>> Code4
```

FIGURE 9 – Calcul du prix de l'option Asiatique pour S_0 fixe

5) $\rightarrow$ Calcul du prix de l'option Asiatique pour $S_0 = 10$:

```
>> Code4

ans =

    1.6368

>> Code4

ans =

    1.6364
```

FIGURE 10 – Calcul du prix de l'option Asiatique pour $S_0 = 10$

Le code sous Matlab du prix de l'option Asiatique :

```

Prix_Asiatique_S0_fixe(10,1000)

function [prix] = Prix_Asiatique_S0_fixe(S0,Nmc)

T=1;
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=1;
r=0.4;
sigma=0.3;
S(1)=S0;
N=100;
dt=T/N;
sum=0;

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

res=max(sum/T-K,0);

end
```

FIGURE 11 – Code Sous MATLAB du fichier Code4.m

Partie II.3 Graphe de l'option Asiatique

6) $\rightarrow$ Graphe $S_0 -> V(0, S_0)$:

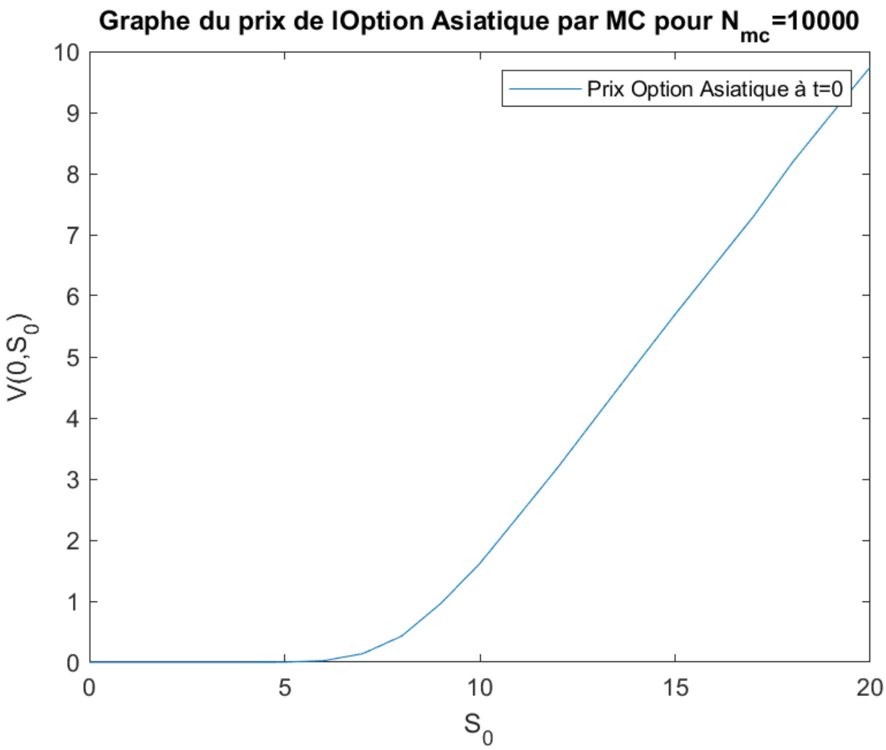


FIGURE 12 – Graphe du prix de l'option Asiatique

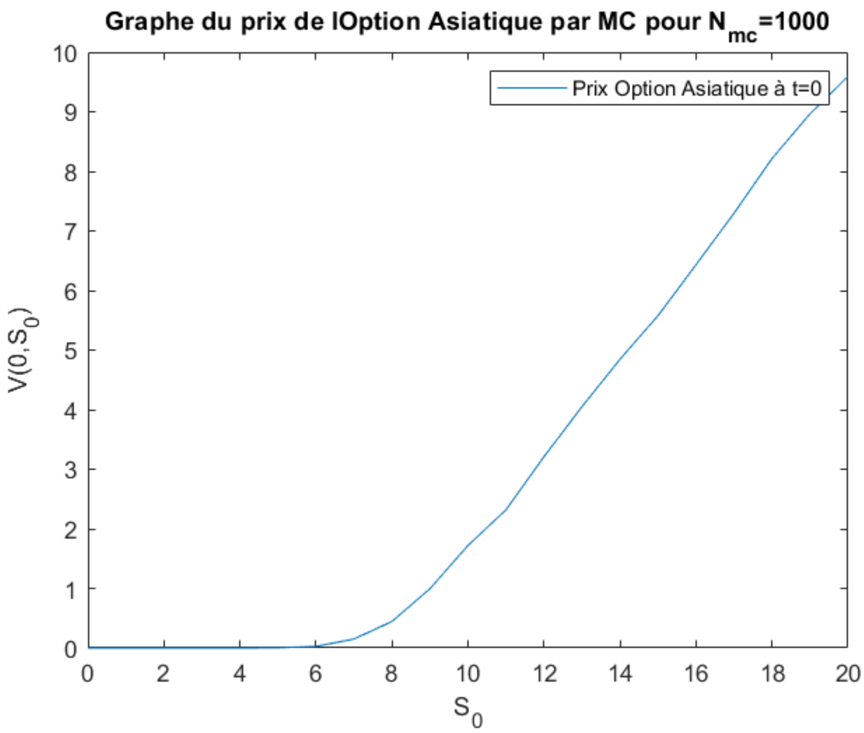


FIGURE 13 – Graphe du prix de l'option Asiatique

Le code sous Matlab du graphe du prix de l'option Asiatique :

```

Prix_Asiatique(10000)

function [] = Prix_Asiatique(Nmc)

L=20;
K=10;
for k=1:L+1
    S0(k)=k-1;
    prix_option(k)=Prix_Asiatique_S0_fixe(S0(k),Nmc);
end
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}=10000')
end

function [prix] = Prix_Asiatique_S0_fixe(S0,Nmc)

T=1;
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=1;
r=0.4;
sigma=0.3;
S(1)=S0;
N=100;
dt=T/N;
sum=0;

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

res=max(sum/T-K,0);

end
```

FIGURE 14 – Code Sous MATLAB du fichier Code5.m

7) → Les graphes de l'option Europeenne et l'option Asiatique :

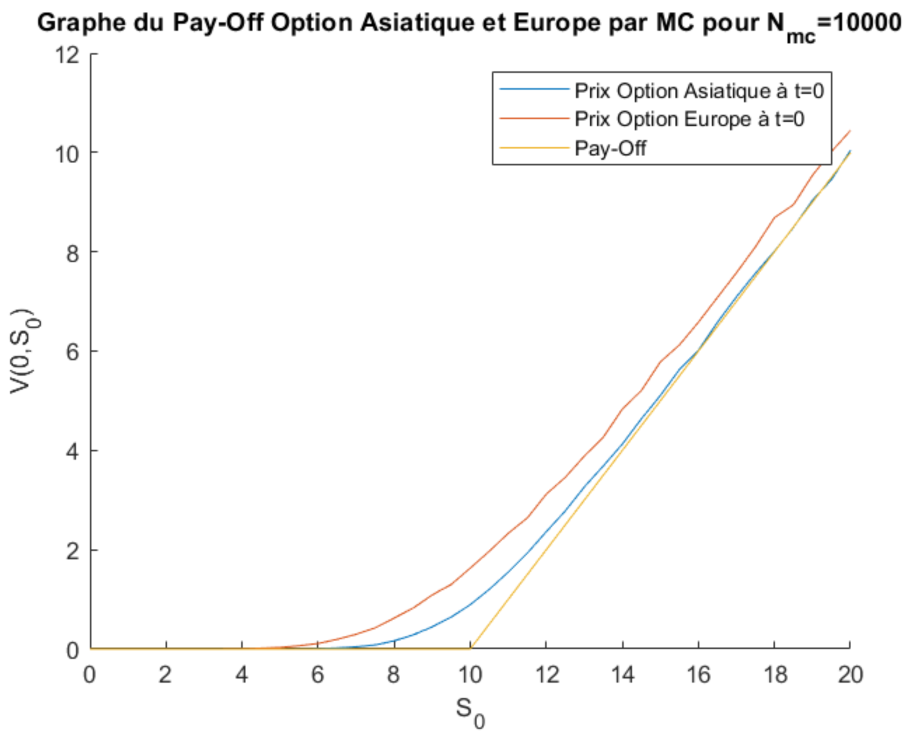


FIGURE 15 – Graphe du prix de l'option Asiatique

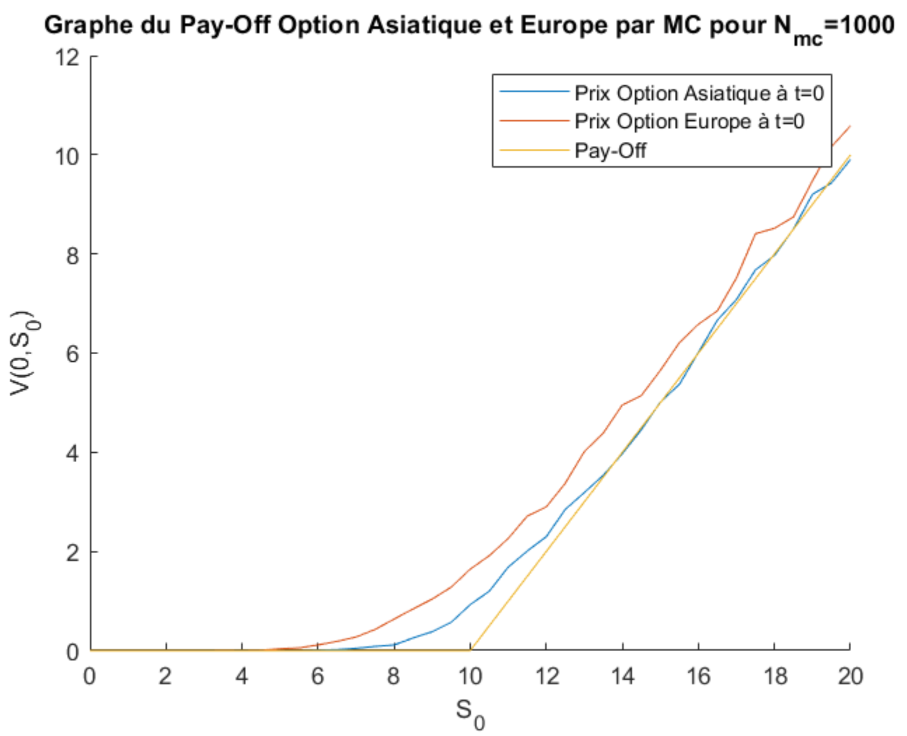


FIGURE 16 – Graphe du prix de l'option Asiatique

Le code sous Matlab des Les graphes de l'option Europeenne et l'option Asiatique :

```matlab

Graphe_Asie_europ(1000)

function [] = Graphe_Asie_europ(Nmc)

K=10;

for j=1:41
    S0(j)=0.5*(j-1);
    prix(j)=Prix_Asie(S0(j),Nmc);
    prix_eur(j)=V_estime1(S0(j),Nmc);
end

figure;
hold;
plot(S0,prix)
plot(S0,prix_eur)
plot(S0,Pay_Off(S0,K))
xlabel('S_0')
ylabel('V(0,S_0)')
legend('Prix Option Asiatique à t=0','Prix Option Europe à t=0','Pay-Off')
title('Graphe du Pay-Off Option Asiatique et Europe par MC pour N_{mc}=1000')

end

function [prix] = Prix_Asie(S0,Nmc)

T=0.5;
r=0.1;

for n=1:Nmc
    gain(n)=Gain_Asie(S0);
end

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

end

function [gain] = Gain_Asie(S0)

K=10;
A=Moyenne_S(S0);
gain=Pay_Off(A,K);

end


function [moyenne] = Moyenne_S(S0)

T=0.5;
N=100;
dt=T/(N+1);
integrale=0;
S=Prix_S(S0);

for i=0:N
    integrale=integrale+S(i+1)*dt;
end


moyenne=integrale/T;


end

function [prix] = Prix_S(S0)

T=0.5;
r=0.1;
sigma=0.5;
S(1)=S0;
N=100;
dt=T/(N+1);

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

prix=S;

end

function [prix] = V_estime1(S0,Nmc)

% Définition des constantes %

K=10;
T=0.5;
r=0.1;
sigma=0.5;
esperance=0;

for n=1:Nmc
    g=randn;
    esperance=esperance+Phi(K,r,T,sigma,S0,g);
end

prix=esperance/Nmc;

end

function [f] = Phi(K,r,T,sigma,S0,g)

f=exp(-r*T)*max(S0*exp((r-(sigma^2)/2)*T+sigma*sqrt(T)*g)-K,0);

end

function [f] = Pay_Off(S,K)
f = max(S-K,0);
end
```

Figure 17 – Code Sous MATLAB du fichier Code6.m

III 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 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.
→ Option Call Europ / Butterfly / Americ.
→ Options Call, Butterfly par Monte-Carlo.
→ Méthode de la réduction de la variance.
→ Méthode aux Différences Finies.
→ Conditions aux limites de Neumann.
→ Conditions aux limites de Dirichlet.
→ Programmation des solutions analytiques de Black et Sholes.
→ Volatilité locale.