E.I.S.T.I. Département Mathématiques

2 ème Année Spécialisation Génie Mathématique

Parcours MATH-FINANCE

Méthodes numériques avancée appliquée à la finance

TP 1: Résolution numérique de l'équation de Black et Scholes par la méthode explicite d'Euler

Le but de ce TP 1 est l'évaluation numérique du prix des options Européenne et Américaine par la méthode aux Différences Finies

Partie I Vanilla. Call. Conditions aux limites de Dirichlet

L'équation de Black et Scholes avec la condition finale et les conditions aux limites de Dirichlet s'ecrit de la forme:

$$\left\{\begin{array}{c} \frac{\partial V}{\partial t} - rV + rS\frac{\partial V}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} = 0 \\ V(t=T,S) = max(S-K,0) \\ V(t,S=0) = 0 \\ V(t,S=L) = L - Ke^{-r(T-t)} \end{array}\right. \tag{1}$$

Cette équation permet de trouver le prix $V(S,t)$ de l'option d'achat (ou call) d'une action qui vaut initialement S et qu'on pourra acheter au prix K dans un temps ultérieur T. $V(0,S)$ est le prix au temps $t=0$ de l'option d'achat de prix d'exercice K à l'échéance $T>0$, et d'actif S en $t=0$. On note σ la volatilité de l'action et r le taux d'intérêt.

Théorie

Discrétisons les variables: spatiale et temporelle:
En théorie la numération commence de 0.
x: $S_0 = 0, ...S_i, ...S_{N+1} = L$.
t: $t^0 = 0, ...t^n, ...t^{M+1} = T$.
Donc

$$\left\{\begin{array}{c} i = 0,1,2,...N+1 \\ n = 0,1,2,...M+1 \\ \Delta S = \frac{L}{N+1} \\ \Delta t = \frac{T}{M+1} \\ V(t_n, S_i) \equiv V_i^n \end{array}\right. \tag{2}$$

Conditions aux limites :

$$\begin{cases} V_0^n = 0 \\ V_{N+1}^n = L - Ke^{-r(T-t(n))} \\ n = M, ..., 1, 0 \end{cases} \tag{3}$$

Condition finale :

$$\{ \ V_i^{M+1} = max(S(i) - K, 0), i = 0, 1, 2..., N + 1 \tag{4}$$

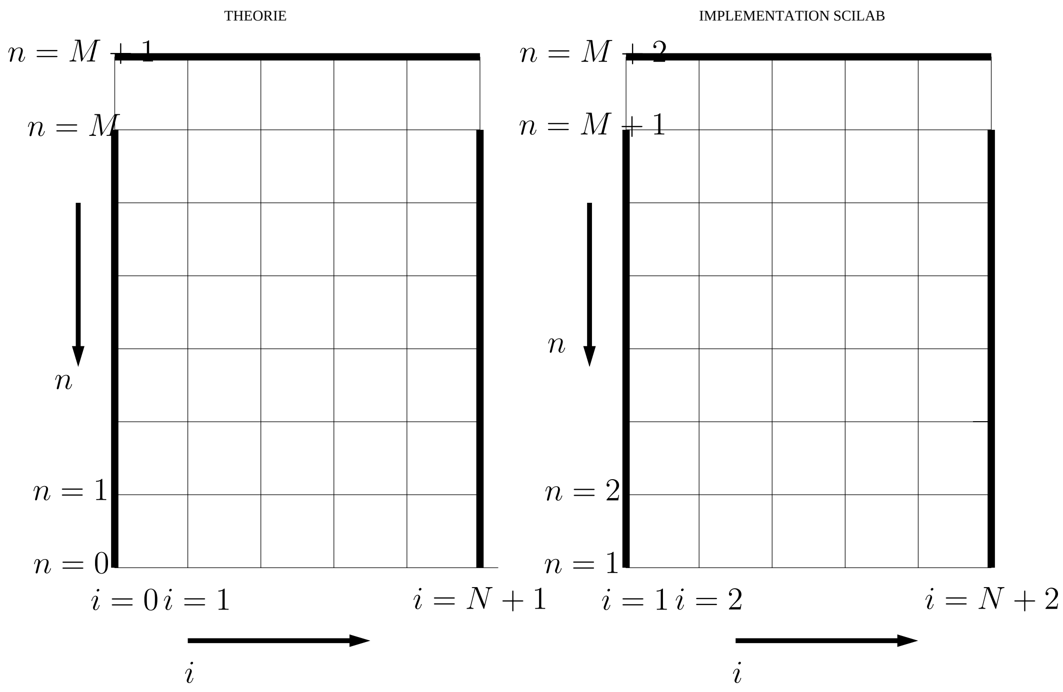
Pour discrétiser l'équation (1) on utilise:
pour $\frac{\partial V}{\partial t}$ la dérivée retrograde,
pour $\frac{\partial^2 V}{\partial S^2}$ la seconde dérivée, centrée,
pour $\frac{\partial V}{\partial S}$ la dérivée centrée.
On obtient

$$V_i^{n-1} = V_{i+1}^n \frac{\Delta t}{2}(\sigma^2 \frac{S(i)^2}{(\Delta S)^2} + r\frac{S(i)}{(\Delta S)}) + V_i^n(1 - \Delta t(\sigma^2 \frac{S(i)^2}{(\Delta S)^2} + r)) + V_{i-1}^n \frac{\Delta t}{2}(\sigma^2 \frac{S(i)^2}{(\Delta S)^2} - r\frac{S(i)}{(\Delta S)})$$

et on commence les calculs par $n = M + 1$.
Utilisez les valeurs suivantes:

$$\begin{cases} L = 20 \\ K = 10 \\ T = 0.5 \\ r = 0.1, \quad r = 0.4 \\ \sigma = 0.5 \\ N = 99 \\ M = 4999 \quad soit \quad \Delta t = T/(M + 1) = 10^{-4} \end{cases}$$

Implémentation numérique.



1) Si on programme en Matlab ou Scilab déplacer tout les indices fixes de 1.

- **Définir les vecteurs avec les vraies valeurs:**

  ◦ $S = (0 : N + 1) \cdot \Delta S \qquad ou \quad S = linspace(0, L, N + 2)$

Par conséquent les coordonnées portent les vraies valeurs :
$S(1) = 0,$
$S(2) = \Delta S,$
$S(3) = 2\Delta S, ...,$
$S(N + 2) = (N + 1)\Delta S \equiv L$
Il y a $N + 2$ composantes du vecteur S.

  ◦ $t = (0 : M + 1) \cdot \Delta t \qquad ou \quad t = linspace(0, T, M + 2)$

Par conséquent les coordonnées portent les vraies valeurs :
$t(1) = 0,$
$t(2) = \Delta t,$
$t(3) = 2\Delta t, ...,$
$t(M + 2) = (M + 1)\Delta t \equiv T$
Il y a $M + 2$ composantes du vecteur t

- **Programmation structurée**
  function[f]=condition-finale(S)
  $f = max(S - K, 0)$
  end

2)Implémentation (méthode 1) de la condition finale:

$$\left\{\begin{array}{c} for \quad i = 1, 2..., N+2 \\ V(M+2, i) = max(S(i) - K, 0) \\ end \end{array}\right. \tag{5}$$

ou
Implémentation (méthode 2) de la condition finale :

$$\left\{\begin{array}{c} for \quad i = 1, 2..., N+2 \\ V(M+2, i) = \text{condition\_finale}(S(i)) \\ end \end{array}\right. \tag{6}$$

Tracer condition finale:
figure;
plot(S, V(M+2,:))
xlabel('prix S')
title('condition finale')

3) Conditions aux limites :

$$\left\{\begin{array}{c} for \quad n = 1 : M+1 \\ V(n, 1) = 0 \\ V(n, N+2) = L - Ke^{-r(T-t(n))} \\ end \end{array}\right. \tag{7}$$

Tracer conditions aux limites:
figure;
plot(t, V(: , 1));
plot(t, V(: , N+2))
xlabel('temps t')
title('conditions aux limites')

4) Écrire le programme principale.

- **Programme principal**
  for n = M+2 : -1: 2
  for i = 2 : N+1
  V(n-1,i)=...
  end
  end

5) Visualisez la fonction $V(S, t)$ aux instants t différents: $t = T, t = T/2, t = 0$.

figure;
plot(S,V(M+2, :));
plot($S, V(T/(2 \cdot \Delta t)$ +1 , :));

plot(S,V(1, :));
title('Prix de loption t=T et T/2')

6) Tracer la fonction solution:
figure;
mesh(S,t,V);
xlabel('prix de lactif S')
ylabel('temps t')
zlabel('Prix de loption')
title('solution de lequation Black-Sholes');

Pour obtenir la surface colorée en Scilab on reduit le nombre de points tracés: on trace un point parmi cent points temporels.

for $j = 1 : 51$
$W(j, :) = V((j-1)\cdot 100+1, :);$
end
tnew=$(0 : 50)\cdot T/50$;
figure
surf(S,tnew,W);

7) Prenez $\Delta t = 10^{-3}$ et observez la divergence de l'algorithme explicite.

Partie II. Vanilla. Call. Conditions aux limites de Neumann

$$\left\{\begin{array}{c} \frac{\partial V}{\partial t} - rV + rS\frac{\partial V}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} = 0 \\ V(t=T,S) = max(S-K,0) \\ \frac{\partial V}{\partial S}(t, S=0) = 0 \\ \frac{\partial V}{\partial S}(t, S=L) = 1 \end{array}\right. \tag{8}$$

1) Implémenter les conditions aux limites de Neumann.
Dans le programme les conditions aux limites de Neumann sont placées entre les deux boucles.

for $n = M+2 : -1 : 1$
for $i = 2 : N+1$
Main programme: $V(n-1, i) =$
end for
$V(n-1, 1) = V(n-1, 2)$
$V(n-1, N+2) = V(n-1, N+1) + \Delta S$
end for

2) Présenter les graphes.

Partie III. Vanilla. Put. Conditions aux limites de Neumann

1) Implémenter les conditions aux limites de Neumann pour Put.
2) Présenter les graphes.

Partie IV. Put Américain.

Une personne qui possède une option américaine peut l'exercer à n'importe quel moment $t_n \in [0, T]$. La fonction pay-off s'écrit de la forme

$$Put_{pay-off} = max(K - S, 0)$$

1) Expliquer pourquoi on peut utiliser l'algorithme suivant:

$$V_i^{n-1} = \max(V_{i+1}^n \frac{\Delta t}{2}(\sigma^2 \frac{S(i)^2}{(\Delta S)^2} + r\frac{S(i)}{(\Delta S)}) + V_i^n(1 - \Delta t(\sigma^2 \frac{S(i)^2}{(\Delta S)^2} + r)) +$$

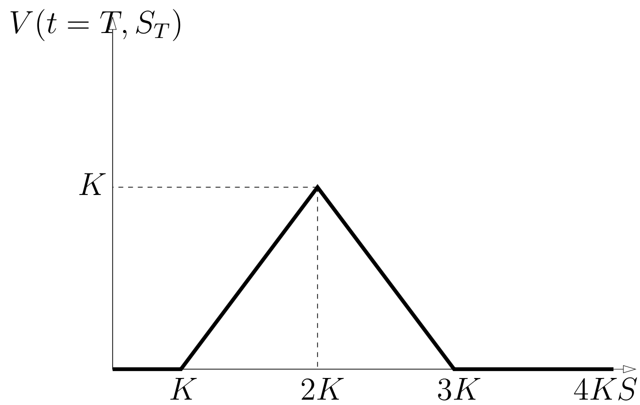
$$V_{i-1}^n \frac{\Delta t}{2}(\sigma^2 \frac{S(i)^2}{(\Delta S)^2} - r\frac{S(i)}{(\Delta S)}),\ \max(K - S(i), 0))$$

avec les conditions aux limites de Neumann pour le Put.

2) Tracer les graphes du Put Européen et Américain dans le meme systme des coordonnes.

Partie V. Option de Butterfly.

• Calculer le prix de l'option de Butterfly sachant la fonction Pay-Off predenté sur la figure:



Partie VI. Comparaison entre les solutions numérique et analytique

Vous connaissez la solution analytique de l'équation de Black et Scholes:

$$V_{analytique}(t,S) = SN(d_1) - Ke^{-r(T-t)}N(d_2)$$

$$d_{1,2} = \frac{\ln(S/K) + (r \pm \sigma^2/2)(T-t)}{\sigma\sqrt{T-t}}$$

1) Tracer le graphe:

$$S_i \longrightarrow V_{analytique}(t=0, S_i)$$

Pour cela appliquez l'approximation suivante de la fonction de repartítion de la loi Normale $N(x)$:

• si on programme en Matlab ou Scilab:
function [f]=$N(x)$
f=$1/2 * (1 + erf(x/\sqrt(2)))$;
end

• si on programme en Python ou C ou C++:

$$N(x) = \begin{cases} 1 - N'(x)(a_1\kappa + a_2\kappa^2 + a_3\kappa^3 + a_4\kappa^4 + a_5\kappa^5) & \text{si} \quad x \geq 0 \\ 1 - N(-x) & \text{si} \quad x < 0 \end{cases}$$

où

$$N'(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}, \qquad \kappa = \frac{1}{1+\gamma x}$$

Les coefficients sont: $\gamma = 0.2316419, a_1 = 0.319381530, a_2 = -0.356563782, a_3 = 1.781477937, a_4 = -1.821255978, a_5 = 1.330274429$

2) Tracer les graphes:

$$S_i \longrightarrow |V_{analytique}(t=0,S_i) - V_{conditions\_neumann}(t=0,S_i)|$$

et

$$S_i \longrightarrow |V_{analytique}(t=0,S_i) - V_{conditions\_dirichlet}(t=0,S_i)|.$$

3) Faire la conclusion.

Partie VII. Volatilit locale

Nous utilisons maintenant "Constant Elasticity of Variance Model" (CEV). La volatilité n'est pas une constante:

$$\sigma(t,S) = \frac{1}{\sqrt{S}} exp(-t/T)$$

Presenter les graphes.