NOM :
PRENOM : Corrigé

Date :
Groupe :

**Calcul stochastique : feuille de réponses du TP 4**
**Frontière d'exercice du Put américain**

On reprend les notations des TP précédents, avec les constantes suivantes $n = 50$, $T = 1$, $\sigma = 0.4$, $S_0 = 100$ et $r = 0.25$.

**Exercice 1.** : Exécuter avec ces nouvelles constantes le programme définissant sous Scilab $SS$, $CC$ et $PP$ (reprendre le programme définissant ces options par récurrence retrograde). Calculer la plus grande et la plus petite valeur de $S_T$ et, pour chacune d'elle, la valeur correspondante $C(T, S_T)$ d'un Call Européen à la monnaie, puis faites de même pour un Put Européen à la monnaie.

Les valeurs minimales et maximale sont SS(51, 1) = 5,9105747 et SS(51,51) = 1691,8829. La première correspond à 50 down (sur 50 mouvements (donc pas de up)) et la seconde à 50 up.

Les valeurs correspondantes pour le call sont CC(51,1) = 0 et CC(51,51) = 1591,8829 (qui s'expliquent facilement puisque en ces points C est égal au payoff)

De même, les valeurs correspondante pour le put sont
PP(51, 1) = 94,089425
PP(51,51) = 0.

**Exercice 2.** : Calculer le prix de ces 2 options à l'instant $t = 0$. Recommencer pour une autre valeur de $r$ et expliquer la différence de prix obsedrvée. Revenir à $r = 0,25$ pour les questions suivantes.

Pour la prime, on trouve, lorsque r = 0,25,
PP(1,1) = 5,5743234
et CC(1,1) = 27,694245

Par contre lorsqu'on change de valeur de r, cette prime n'évolue pas de la même façon : Pour r = 0,05, on a
PP(1,1) = 13,068218
et CC(1,1) = 17,945275

donc le put est devenu plus cher alors que le Call est devenu moins cher. La formule donnant la prime comporte r dans son coef. d'actualisation mais aussi dans la valeur de la probabilité neutre $p = \frac{R-d}{u-d}$.

**Exercice 3.** : On a vu que pour obtenir le prix d'une option Américaine, il suffit de remplacer, dans la fomule du prix de l'option Européenne correspondante, la relation de récurrence $C_t = e^{-\delta t}\mathbb{E}(C_{t+\delta t}/\mathcal{F}_t)$ par la relation $C_t^{Amer} = \text{Max}\{\phi(S_t), e^{-\delta t}\mathbb{E}(C_{t+\delta t}^{Amer}/\mathcal{F}_t)\}$. Utiliser cette propriété pour calculer le prix d'une option américaine de même pay off final que celle de la question précedente. Indiquer la valeur trouvée puis la comparer avec celle du Call Européen. Commenter.

Dans le programme donnant CC (par récurrence), on remplace la ligne

CC(i,j) = (p*CC(i+1,j+1) + (1-p)*CC(i+1,j))/R;

par la ligne

CCA(i,j) = max[phi(SS(i,j)), (p*CCA(i+1,j+1)+(1-p)*CCA(i+1,j))/R];

Après exécution du programme, on trouve

CCA(1,1) = 27,694245

On constate que CCA(1,1) = CC(1,1), autrement dit le prix du Call américain est égal au prix du Call européen de même pay off et même date d'exercice ;
Ce n'est pas étonnant : c'est un théorème montré en cours.

**Exercice 4.** : Reprendre la question précédente pour un Put Américain.

Pour calculer PPA(i,j), on ne change que le pay off phi que l'on remplace par psi :

```
function psi = psi(S);
psi = max(K-S, 0);
end function;
```

Sinon, on reprend le programme de CCA en changeant pour PPA uniquement le nom. On trouve

PPA(1,1) = 8,376521 qui est donc supérieur à PP(1,1).

Le Put américain est donc supérieur au Put européen ce qui est normal puisque le contrat américain donne plus de droit à son détenteur que l'analogue européen (et conforme à la théorie)

**Exercice 5.** : Reprendre la question précédente avec $r = 0$ cette fois.

Si l'on exécute à nouveau le programme pour $r=0$, on constate cette fois que

$PPA(1,1) \underset{(r=0)}{=} 15,772894$

qui est donc identique à $PP(1,1) \underset{\text{pour } r=0}{=} 15.772894$

On s'attend à ce résultat car on sait qu'il n'y a de différence entre Put américain et européen que lorsque $r \neq 0$.

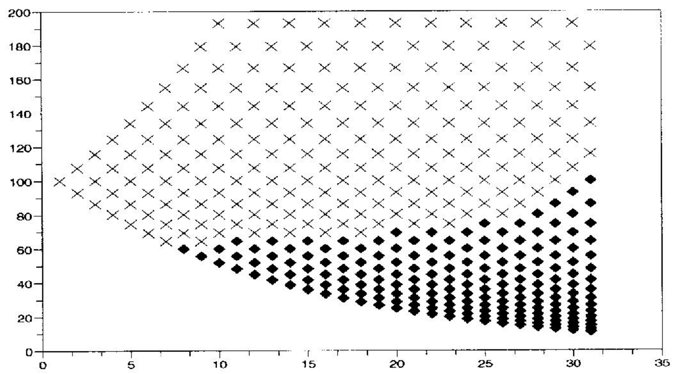
**Exercice 6.** : Afin de tracer la frontière d'exercice du Put Américain, définir une matrice $EPA(i,j)$ (pour "Exercice du Put Américain") qui vaut 1 aux points $(i,j)$ de l'arbre de Cox situés en dessous de la frontière d'exercice et 0 aux points situés au dessus. On pourra initialiser cette matrice par `EPA=-ones(n+1,n+1)` pour remplir les points inintéressants de la matrice par des $-1$. A noter que la commande

`plot2d(i,SS(i,j),-2*EPA(i,j)-2)`

permet de tracer les points de l'arbre $(i, S(i,j))$ en utilisant un symbol différent selon que `EPA(i,j)` vaut 1 ou 0 (voir l'aide en ligne pour la syntaxe de la commande `Plot2d`).

Pour tracer la frontière d'exercice, il suffit alors de repérer pour chaque $i$, s'il y a un point $(i, S(i,j))$ pour lequel EPA vaut 1 et, dans ce cas, de choisir celui d'ordonnée maximale.

Tracer la frontière d'exercice et étudier comment sa forme évolue en fonction de $r$. Expliquer ce que vous observez.




ici $r = 0,05$

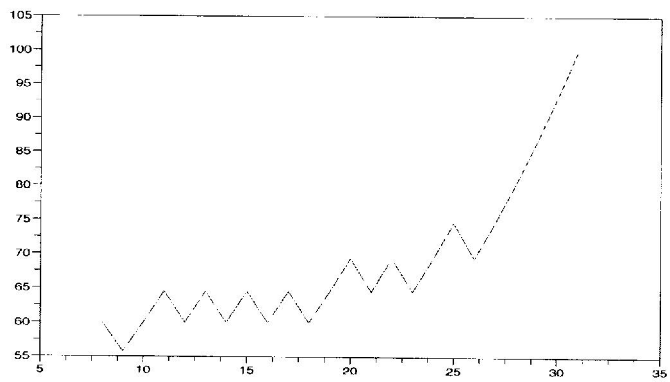
Le programme trace avec une croix x les points $(i,j)$ pour lesquel $PPA_t = e^{-r\delta t}\,\mathbb{E}(PPA_{t+\delta t}/\mathcal{F}_t)$ (le max est égal à son 2e terme) et avec un losange ◆ ceux pour lequel $PPA = psi$ (le max est égal à son 1er terme). La ligne de séparation entre ces deux régions est la frontière d'exercice.

Si l'on augmente $r$ la frontière a tendance à "s'aplatir" et au contraire si l'on diminue encore $r$ ($r=0,01$ ou $r=0,001$) elle se redresse $\xrightarrow{\quad}$ $r \to 0$ et se rapproche de la verticale (partie de l'arbre au temps final)

```
///////Definition de la matrice EPA (Exercice du Put Americian)///////
EPA=-ones(n+1,n+1);
for j=1 :n+1
      if SS(n+1,j)<K+1 then EPA(n+1,j)=1;
      else EPA(n+1,j)=0;
      end
end;
for i=n :-1 :1
      for j=1 :i
            if psi(SS(i,j)) > (p*PPAmer(i+1,j+1)+(1-p)*PPAmer(i+1,j))/R
            then EPA(i,j)=1;
            else EPA(i,j)=0;
            end
      end
end;
////////Tracé de l'arbre pour les petites valeurs de i////////////////
for i=1 :50
      for j=1 :i
            if SS(i,j)<2*S0
            plot2d(i,SS(i,j),-2*EPA(i,j)-2)
            end;
      end;
end;
//////////////////////Tracé de la frontière d'exercice///////////////
ii=0;
for i=1 :n+1
       if max(EPA(i,1 :i))==1 then ii=ii+1;Frontiere_t(ii)=i;
      Frontiere_S(ii)=SS(i,sum(EPA(i,1 :i)));
      end;
end;
plot(Frontiere_t,Frontiere_S,'r-')
```

si = -4, alors "◆"
si = -2, alors "X"

tracé de la figure précédente




Ce programme repère, pour chaque i le plus haut point (s'il existe) où EPA = 1 (sur la figure précédente: "◆") Il joint entre eux les points ainsi repérés pour tracer une ligne polygonale représentant la frontière entre la région de "X" et la région de "◆".