NOM : Corrigé
PRENOM :

Finance Stochastique

Date : 8 février 2012

Groupe :

Finance mathématique : Feuille de réponses du TP 3
Calcul de prix d'options Call et Put

L'objet de cet exercice est de calculer dans un modèle de Cox-Ross-Rubinstein à n étapes le prix d'une option Call et d'une option Put de fonction de pay-off respective $\varphi(S) = (S - K)^+$ et $\varphi(S) = (K - S)^+$. Dans un premier temps nous considérons les Call et les Put à *la monnaie*, c'est-à-dire que l'on suppose que leur prix d'exercice vaut $K = S_0$. Rappelons que dans le modèle de Cox-Ross-Rubinstein, les prix de l'actif sous-jacent sont modélisés par une marche aléatoire S_t définie par $S_0 = S0$ et $S_{t+\delta t} = S_t U_t$, avec $U_t \in$ {up,down}, avec up< 1 <down dépendent de n. On pose à nouveau

$$n = 100 \quad T = 1 \quad \delta t = T/n \quad up = e^{+\frac{\sigma}{\sqrt{n}}} \quad down = e^{-\frac{\sigma}{\sqrt{n}}}$$

et on introduit comme dans le TP1 la notation $S(i,j) = SS(i+1,j+1)$, les valeurs des constantes étant cette fois $\sigma = 0.4$ et $S_0 = 140$.

Exercice 1. : Recalculer les valeurs de SS pour $i \in \{0,\dots,n\}$ et $j \in \{0,\dots,i\}$ et indiquer les valeurs obtenues après 15 pas de temps lorsqu'il n'y a eu aucun *up* puis lorsqu'il n'y a eu aucun *down*.

On recalcule avec Scilab la matrice SS avec les constantes ci-dessus. On trouve
SS(16,1) = 76,833629 (15 pas de temps, 0 up)
SS(16,16) = 255,09663 (15 pas de temps, 15 up)

Trouve-t-on les mêmes valeurs pour SS lorsque n ne vaut plus que $n = 50$? Pourquoi ?

Non, si on reprend le calcul de SS avec n = 50, les valeurs changent (SS(16,1) = 59,92..) et (SS(16,16) = 327,06..)
Ceci se comprend car up et down dépendent de δt et donc de n, ce qui influe sur SS.

Que trouve-t-on pour SS lorsque la volatilité est nulle ? Expliquez pourquoi.

Lorsque σ = 0, up = e^0 = 1 = down. On voit donc que SS ≡ S_0 pour tout i et j.

Exercice 2. : Afin de remplacer la fonction SS par la fonction $S(i,j)$, où i est bien le nombre de pas de temps et j le nombre de *up*, on pose

```
function S=S(i,j) ; S=SS(i+1,j+1) endfunction;
```

Indiquer les valeurs de $S(i,0)$ pour $i = 0\dots4$.

On trouve S(0,0) = 140 S(1,0) = 132,30.. S(2,0) = 125,02..
S(3,0) = 118,14.. et S(4,0) = 111,65..

Exercice 3. : **Calcul de la prime d'un Call et d'un Put** On a vu que le prix $c(t,S_t)$ d'un portefeuille de couverture d'une option de pay-off φ peut se calculer par récurrence rétrograde : on commence par la plus grande valeur $i = n$ car $c(T,S_T) = \varphi(S_T)$ puis, pour les $i < n$, on procède de la façon suivante : en notant $C(i,j) := c(i\delta t, S(i,j))$, on a $C(n,j) = \varphi(S(n,j))$, puis $C(i,j) = pC(i+1,j+1) + (1-p)C(i+1,j)$, où p est la probabilité risque neutre. Sous scilab, pour le même raisons que pour S, on pose CC(i+1,j+1)=$C(i,j)$.

Le code scilab ci-dessous calcule les valeurs CC(i,j) d'une option Call d'échéance $T = 1$ dans un modèle à $n = 100$ étapes.

```
function phi=phi(S);
phi=max(S-K,0);
endfunction;
CC=zeros(n+1,n+1);
for j=0 :n
CC(n+1,j+1)=phi(SS(n+1,j+1));
end;
for i=n-1 :-1 :0
for j=0 :i
CC(i+1,j+1)=(p*CC((i+1)+1,(j+1)+1)+(1-p)*CC((i+1)+1,j+1))/R;
end;
end;
```

Quelle valeur trouvez vous pour la *prime* $c(0,0)$ d'un Call à la monnaie pour $n = 100$?

Pour la calculer, à l'aide de ce code, il faut choisir r (on choisit $r = 0,5$), calculer R et calculer la probabilité $p = \frac{R-d}{u-d}$.
On pose aussi $K = S_0$.
On obtient $CC(1,1) = 57,211899$.

Recommencer le calcul pour deux autres valeurs de K (par exemple $K = S_0 - 10$ et $K = S_0 + 10$). Qu'observez-vous ? Comment le prix du Call varie-t-il avec K ? Expliquez.

On observe que pour $K = S_0 - 10$, le prix est $62,529..$ et pour $K = S_0 + 10$, il vaut $52,171..$ On constate que le prix semble décroissant par rapport à K.
Cela s'explique car ce prix vaut $e^{-rT}\mathbb{E}((S_T - K)^+)$ qui est bien une fonction décroissante de K.

Reprendre les deux questions précédente pour un Put à la monnaie.

Pour le Put à la monnaie, on trouve $PP(1,1) = 2,1262$
On constate que le Put est une fonction croissante de K ce qui s'explique puisqu'il vaut $e^{-rT}\mathbb{E}((K - S_T)^+)$.

Exercice 4. :

Vérifier expérimentalement la relation de parité Call-Put à l'instant $t = 0$:

$$C_t - P_t = S_t - Ke^{-r(T-t)}.$$

Si on calcule $C_0 - P_0$, on trouve $57,2119 - 2,1262$ (à peu près) soit $55,0857$.
Si on calcule $S_0 - Ke^{-r}$ on trouve aussi $55,0857$
La relation de parité Call-Put est donc bien satisfaite ici.

Exercice 5. :

On veut tracer la courbe donnant la prime en fonction de la volatilité σ. Pour cela il convient de redéfinir les quantités up, down et p, puis de même SS et CC (ou PP) comme des fonctions de σ. Tracer cette courbe et la reproduire approximativement ci-dessous. Puis expliquez ce que vous observez.

voir le code au TP4