NOM : Corrigé
PRENOM :

Date :
Groupe :

Calcul Stochastique et applications à la finance
Feuille-réponses du TP 3
Prix d'une option comme espérance conditionnelle du payoff

L'objet de cette séance est de calculer dans un modèle de Cox-Ross-Rubinstein à n étapes le prix d'une option Call de payoff $\varphi(S) = (S - K)^+$ non plus au moyen d'une récurrence rétrograde mais comme une espérance du payoff actualisé.

Comme précédemment, on modélise les prix par une marche aléatoire S_t définie par $S_0 = S0$ et $S_{t+\delta t} = S_t U_t$, avec $U_t \in \{\mathtt{up}, \mathtt{down}\}$. On introduit comme dans le TP1 la notation $S(i,j) = SS(i+1, j+1)$ pour représenter la valeur de l'actif S_t à l'instant $t = i\delta t$ s'il y a eu j up depuis l'instant $t = 0$. On pose à nouveau

$$n = 100 \;\; T = 1 \;\; \delta t = T/n \;\; up = e^{+\frac{\sigma}{\sqrt{n}}} \;\; down = e^{-\frac{\sigma}{\sqrt{n}}} \;\; \sigma = 0.4 \;\; S_0 = 140 \;\; r = 0.1 \;\; R = e^{r\delta t}.$$

Exercice 1. : On veut calculer la prime $c(0,0)$ d'un Call à la monnaie sur l'actif S_t. On a vu que cette prime peut se calculer en utilisant la formule $c(0,0) = \mathbb{E}(\varphi(S(n,J))/R^n$, où J est une variable aléatoire qui suit une loi binômiale : $\mathbb{P}(\{J = k\}) = \begin{pmatrix} n \\ k \end{pmatrix} p^k (1-p)^{n-k}$, avec $\begin{pmatrix} n \\ k \end{pmatrix} = \frac{n\,!}{k\,!(n-k)\,!}$.

1. Que retournent les commandes : `binomial(0.5,1)=` 0,5 0,5

   `binomial(0.5,2)=` 0,25 0,5 0,25

   `binomial(0.5,3)=` 0,125 0,375 0,375 0,125

2. De façon générale, consultez la doc en ligne pour déterminer ce que retourne la commande `binomial(p,n)` ; expliquez :

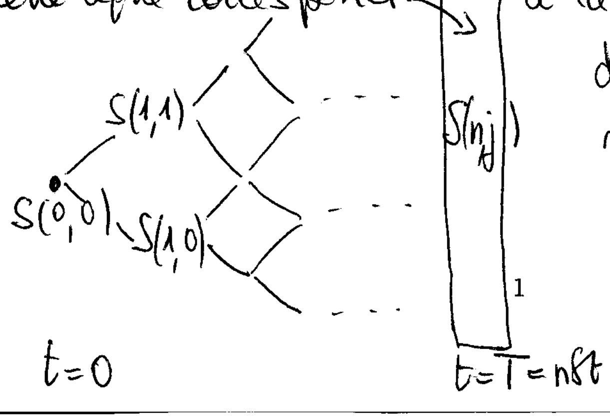
   Cette commande retourne un vecteur (ligne) dont les $n+1$ composantes sont les probabilités
   $$P(J=k) = C_n^k \, p^k (1-p)^{n-k}$$
   pour $k = 0, .., n$.

3. Qu'observez-vous en calculant `binomial(0.5,3)'` ?

   On obtient un vecteur ayant les mêmes composantes que binomial(0.5, 3) mais qui est un vecteur colonne (et non ligne). Le ' en Scilab calcule la transposée (de la matrice)

4. En utilisant le programme du TP1, recalculer les valeurs de de SS pour $i \in \{0, \ldots, n\}$ et $j \in \{0, \ldots, i\}$. Préciser à quoi correspond le vecteur `SS(1+n,1+0 :1+n)` .

   Ce vecteur est la dernière ligne de la matrice $(n+1) \times (n+1)$ calculée. Cette ligne correspond à la dernière colonne de l'arbre



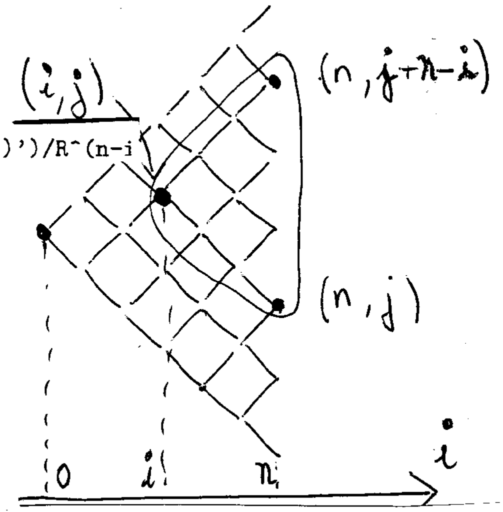
   du modèle CRR qui comporte $n+1$ noeuds, $S(n,j)$ pour $j = 0..n$ correspondant à $i = n$ et j quelconque entre 0 et n.

5. Introduire la fonction de payoff `Phi` comme au TP2. Préciser à quoi correspond `Phi(SS(1+n,1+0 :1+n))` puis le produit `Phi(SS(1+n,1+0 :1+n))*binomial(p,n)'`. En déduire la valeur de la prime du Call à la monnaie et la comparer avec la valeur de $CC(1,1)$ obtenue au TP2.

SS(1+n, 1+0:1+n) est un vecteur représentant les n+1 valeurs possibles de l'actif S à l'instant final $T=n\delta t$ et donc Phi(SS(1+n,1+0:1+n)) est le vecteur des n+1 valeurs du payoff. Le prix du Call étant l'espérance actualisée du payoff, on obtient cette espérance en effectuant le produit scalaire [Phi(SS(1+n,1+0:1+n))*binomial(p,n)'] qu'on actualise en divisant par R^n. On retrouve 28,39... comme au TP2

Exercice 2. : Le code suivant permet de calculer le prix d'un Call $C(i,j)$ à la monnaie à un instant $t \in \{O, \delta t, 2\delta t, ...\}$ quelconque et non plus seulement en $t = 0$. Epliquez pourquoi (on pourra dessiner les points de l'arbre CRR atteint à l'instant $T = n\delta t$ à partir d'un point (i,j) quelconque) puis calculer quelques valeurs de C.

```
C=zeros(n+1,n+1)
C(1+n, :)=Phi(SS(1+n, :));
fori=0 :n-1
    forj=0 :i
    C(1+i,1+j)=(Phi(SS(1+n,1+j :1+j+n-i))*binomial(p,n-i)')/R^(n-i
    end ;
end ;
```



Pour calculer le prix à l'instant $t=i\delta t$, après j "up", on se place au point (i,j) de l'arbre et on calcule l'espérance du payoff aux points que l'on peut atteindre : les points
(n,j), (n,j+1) ... (n,j+(n-i))
les probabilités conditionnelles étant binomial(p,n-i) et l'actualisation $1/R^{(n-i)}$

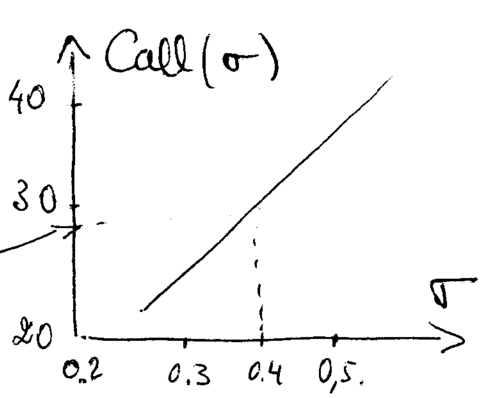
Exercice 3. : On se propose de tracer la courbe donnant la prime en fonction de la volatilité σ. Pour cela il faut reprendre le programme précédent donnant la prime C en remplaçant toutes les quantités dépendant de σ par des functions de σ. Par exemple au lieu de `up=exp(sigma/sqrt(n))` ; on devra poser

```
function up=up(sigma) ;
up=exp(sigma/sqrt(n)) ;
endfunction ;
```

et de même pour down, p, SS et C. Indiquer l'allure du graphe obtenu.

Si vous en avez le temps reprendre l'ensemble du TP, cette fois pour un Put.

Le tracé (voir le programme ci-joint) est celui d'une fonction strictement croissante. On retrouve 28.39 la valeur 28.39 trouvée pour $\sigma=0.4$.



Le prix du Call est donc une fonction strictement croissante de la volatilité de l'actif sous-jacent.