Corrigé

NOM :
PRENOM :

Date :
Groupe :

TP3 en 2011

Calcul Stochastique et applications à la finance
Feuille-réponses du TP 4
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

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

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 suivantes :

`binomial(0.5,1)=` 0.5 0.5

C'est la loi d'une v.a. de Bernoulli (binomiale avec $n=1$)

`binomial(0.5,2)=` 0.25 0.5 0.25

C'est la loi d'une v.a. $\mathcal{B}(0.5, 2)$

x	0	1	2
P_x	0.25	0.5	0.25

`binomial(0.5,3)=` 0.125 0.375 0.375 0.125

C'est la loi d'une v.a. $\mathcal{B}(0.5, 3)$

x	0	1	2	3
P_x	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 :

binomial(p, n) retourne le vecteur ligne des n+1 probabilités des n+1 valeurs {0, 1, .., n} prises par une v.a. binomiale $\mathcal{B}(n, p)$

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

L'apostrophe transpose le vecteur ligne en un vecteur colonne

0.125
0.375
0.375
0.125

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 constitué des valeurs de S_T dans un modèle CRR, de la plus petite (que des "down") à la plus grande (que des "up").

2.5641894 ... 7643.741

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.

Phi(SS(1+n, 1+0 : 1+n)) est le vecteur ligne des valeurs de $C_T = \varphi(S_T)$
La seconde expression est le produit scalaire du vecteur des valeurs de $C_T = \varphi(S_T)$
avec le vecteur des probabilités des valeurs de S_T.
C'est donc l'espérance $\mathbb{E}(\varphi(S_T)) = \sum_{j=0}^{n} \varphi(S(n,j))\,\mathbb{P}(\{J=j\})$

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.

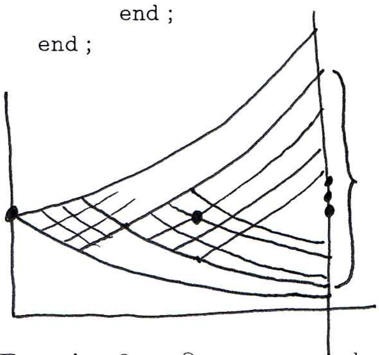
$CCC(i+1,j+1) = C(i,j) = \mathbb{E}(\varphi(S_T) \mid S_t = S(i,j))$ pour $t = i\delta t$

$\overset{(*)}{=} \mathbb{E}\left(\varphi\left(S(i,j)\,u^{J_n-J_i}\,d^{n-(J_n-J_i)}\right)\right)$

$= \sum_{k=0}^{n-i} \varphi(S(n,j+k)) \binom{n-i}{k} u^k d^{n-i-k}$

* car $J_n - J_i = \delta J_{i+1} + \dots + \delta J_n \sim \mathcal{B}(n-i, p)$

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



$C(0,0) = CCC(1,1) = 28,391627$ prime

$C(50,25) = CCC(51,26) = 18,934809$ $t = \frac{T}{2}$ $S_t = S_0$

$C(100,50) = CCC(101,51) = 0$ $S_T = S_0 = K$.

$S(n, j+k)$ des payoff

$C(100,51) = CCC(101,52) = 11,660189$

$C(100,52) = CCC(101,53) = 24,291522$

$C(100,100) = CCC(101,101) = 7503,741$ (!)

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.
Reprendre les exercices 2 et 3 pour un Put.

```
//Exercice 3                                                    ;
clear;
n=100;T=1;delta_t=T/n;S0=140;sigma=0.40;K=S0;r=0.1;R=exp(r*delta_t);
//on redéfinit up, down, p, S comme des fonctions de sigma
function up=up(sigma);
  up=exp(sigma/sqrt(n))
endfunction;
function down=down(sigma);
  down=exp(-sigma/sqrt(n))
endfunction;
function p=p(sigma);
  p=(R-down(sigma))./(up(sigma)-down(sigma))
endfunction;
function S=S(i,j,sigma);
  S=S0.*up(sigma).^(j).*down(sigma).^(i-(j));
endfunction;
function phi=phi(S);
  phi=max(S-K,0);
endfunction;
function psi=psi(S);
  psi=max(K-S,0);
endfunction;
//on trace le prix du Call noté Csig comme une fonction de sigma
//pour des valeurs de sigma comprises entre sigmin et sigmax
deltasig=0.05;sigmin=0.05;sigmax=0.8;
for sig=0:(sigmax-sigmin)/deltasig //sigma=sigmin+sig*deltasig
Csig(1+sig)=(phi(S(n,0:n,sigmin+sig*deltasig))*..
          binomial(p(sigmin+sig*deltasig),n)')/R^n;
Psig(1+sig)=(psi(S(n,0:n,sigmin+sig*deltasig))*..
          binomial(p(sigmin+sig*deltasig),n)')/R^n;
end;
plot2d(sigmin:deltasig:sigmax,Csig)
plot2d(sigmin:deltasig:sigmax,Psig)
```