

Fiche de TP 1 :  
**Le filet  $\varphi(t, S)$  et le delta dans un modèle binomial**

On considère le modèle de Cox-Ross-Rubinstein  $S_{t+\delta t} = S_t u_{\delta t + \delta t}$ , où  $u_s \in \{u, d\}$ , avec  $u = e^{+r\delta t}$  et  $d = e^{-r\delta t} (= \frac{1}{u})$ .

1. Le programme ci-dessous permet de tracer l'arbre binomial  $(t, S_t)$  pour un modèle de Cox-Ross-Rubinstein, de valeur initiale  $S_0 = 140$  et de volatilité  $\sigma = 0,4$ .  
Pour vous familiariser avec quelques instructions Maple, étudiez ce programme en vous aidant du "Help" en ligne puis exécutez-le.
2. En vous inspirant du programme précédent, tracez le graphe de la fonction  $(t, S_t) \mapsto \varphi(t, S_t) := C_t$ , construite au dessus de l'arbre binomial précédent, pour un call à la monnaie ( $K = S_0$ ), défini par récurrence descendante à partir de sa valeur finale  $C_T = (S_T - K)^+$ , par la formule d'évaluation  $C_{t-\delta t} = e^{-r\delta t} \mathbb{P}_{t-\delta t}^*(C_t)$ , pour la probabilité  $\mathbb{P}^*$  pour laquelle le sous-jacent actualisé est une martingale, caractérisée par  $p = \frac{R-d}{u-d}$ , avec  $R = e^{r\delta t}$ ; (on suppose ici que  $r = 0$ ).
3. Même exercice pour un put  $P_t$  sur le même sous-jacent de prix d'exercice  $K = 130$ .
4. Représenter sur un même dessin les courbes  $s \mapsto \varphi(t, s)$  pour quelques valeurs de  $t$  précédant l'instant final,  $t = T, t = T - \delta t, t = T - 2\delta t, \dots$  pour mettre en évidence l'évolution du prix de l'option peu avant la date d'exercice.
5. Même exercice pour les courbes  $S_t \rightarrow \Delta_t(t, S_t)$  donnant le delta de couverture de l'option.

Exercice 1 : calcul et tracé de l'arbre du sous jacent du modele Cox-Ross-Rubinstein

```
> restart;
> n:=10:T:=1:delta_t:=evalf(T/n):
> S0:=140:sigma:=0.4:
> up:=exp(sigma*sqrt(delta_t)):
> down:=exp(-sigma*sqrt(delta_t)):
> S:=proc(i,j) option remember:
> if i=0 then S0:
> elif j=0 then evalf(S(i-1,0)*down):
> else evalf(S(i-1,j-1)*up) fi: end:
> with(plottools):
> arbre:=proc(i,j) option remember:
> if i<n then line([i,S(i,j)], [i+1,S(i+1,j)]), arbre(i+1,j),
> line([i,S(i,j)], [i+1,S(i+1,j+1)]), arbre(i+1, j +1)
> fi: end:
> plots[display](arbre(0,0));
```