Université de Nice
Département de Mathématiques
NOM :
PRENOM :

L3MASS, année 2019-2020
Calcul stochastique et Finance
Date : .
Salle : .

Calcul Stochastique et applications à la finance
Feuille-réponses du TP 2
Calcul du prix d'un Call et d'un Put

L'objet de cette séance est de calculer dans un modèle de Cox-Ross-Rubinstein à n étapes le prix d'une option Call et celui d'une option Put de pay-off respectifs $\varphi(S) = (S-K)^+$ et $\psi(S) = (K-S)^+$. Dans un premier temps nous considèrons les Call et les Put à *la monnaie*, c'est-à-dire que nous supposons leur prix d'exercice K égal à 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`$\}$.

1. Définir, comme dans le TP1, la fonction $S(i,j)$ pour $i \in \{0,\dots,n\}$ et $j \in \{0,\dots,i\}$. On prendra $n = 100$, $\sigma = 0.3$ et $S_0 = 140$. Indiquer quelle valeur vous trouvez pour S à l'instant $t = 10\delta t$ s'il n'y a eu que 4 `down`. Expliquer.

2. Cette valeur de S dépend-elle du choix de n ? Expliquer.

3. On rappelle que le prix d'une option ayant pour date d'exercice T et pour payoff $\varphi(S_T)$ se calcule par récurrence rétrograde. Donc la première idée serait de définir, comme pour S une fonction $C(i,j)$. Nous verrons en fin de séance pourquoi ce n'est pas la bonne idée. Pour le Call, nous allons le coder dans Scilab comme une matrice triangulaire notée CC telle que `CC(1+i,1+j)`=$C(i,j)$ de taille $(n+1) \times (n+1)$.
```
CC=zeros(n+1,n+1) ;
for j=0:n CC(1+n,1+j)=max(S(n,j)-K,0) ;
end ;
for i=n-1:-1:0
      for j=0:i CC(1+i,1+j)=(p*CC(1+i+1,1+j+1)+(1-p)*CC(1+i+1,1+j))/R ;
      end ;
end ;
```
   Choisir K pour un Call *à la monnaie*, introduire les constantes $R = \exp(r\delta t)$ pour $r = 0.1$ et $p = (R-d)/(u-d)$ puis exécuter ce code pour calculer les valeurs `CC(i,j)`. Quelle est la prime du Call (sa valeur à l'instant de souscription du contrat) ? Vérifier, sur quelques valeurs, que le payoff du Call est bien celui que vous attendiez.

4. Quel est le prix du Call à la monnaie à la date $t = 4\delta t$ si le cours de l'actif sous jacent n'a fait qu'augmenter depuis la date de souscription du contrat ? Même question si le cours a baissé une fois exactement ?

5. En vous inspirant du code précédent, que vous dupliquerez puis modifierez, calculer cette fois le prix $PP(1+i, 1+j)$ d'un Put à la monnaie. Expliquer les lignes que vous avez du changer dans votre code.

6. Vérifier, en faisant quelques expériences, que le prix du Put est une fonction croissante de K. Pouvez-vous l'expliquez ? Qu'en est-il du Call ?

7. Vérifier, en faisant quelques expériences, que le prix du Call est une fonction croissante de σ. Pouvez-vous l'expliquez ? Qu'en est-il du Put ?

8. Revenons à l'idée de calculer une fonction $C(i, j)$, notée ci-dessous `Callrec`, (et non une matrice `CC`) qui sera donc telle que, pour $i = n$, $C(n, j) = \max(S(n, j) - K, 0)$, puis pour $0 \leq i \leq n-1$ $C(i, j) = (pC(i+1, j+1) + (1-p)C(i+1, j))/R$.

```
function c=Callrec(i,j);
      if i==n then
            c=max (S(n,j)-K,0)
      else
            c=p*Callrec(i+1,j+1)+(1-p)*Callrec(i+1,j);
      end;
endfunction;
disp(Callrec(0,0),"la prime vaut",n,"Pour n=")
```

ATTENTION : avant d'exécuter ce code, choisir $n = 10$, puis **recalculer**, pour cette valeur de n, la fonction S, et les constantes δt, *down*, *up*, R et p. Quelle valeur trouvez-vous pour le Call ? Comparez avec les valeurs données par CC. **Augmenter progressivement** n **de 10 à 20**. Qu'observez-vous ? Pouvez-vous l'expliquer ?(Indication : estimez, en fonction de n combien de fois le programme fait-t-il appel à une valeur de C pour calculer la prime $C(0, 0)$)