NOM : 2011-2012.
PRENOM : Groupe : .

TP1 : Marche de Wiener, modèle CRR et prix d'un Call

Exercice 1. : La marche de Wiener On considère la marche aléatoire W_t définie par $W_0 = 0$ et $W_{t+\delta t} = W_t \mp \sqrt{\delta t}$ (marche de Wiener). On désigne par i le nombre de pas de temps, $t = i\delta t$ et par j le nombre de *up*, on peut ranger les valeurs prises par la marche W_t dans une matrice triangulaire (attention, la numérotation des lignes et des colonnes par Scilab commence à 1 et non 0 !) que l'on note WW avec $W(i,j) = WW(i+1, j+1)$, pour $i = 0, \ldots, n$ et $j = 0, \ldots, i$.

Saisir le programme suivant puis l'exécuter :

```
n=100;T=1;delta_t=T/n;
WW=zeros(n+1,n+1);WW(1,1)=0;
For i=1:n WW(i+1,1)=WW(i,1)-sqrt(delta_t);
      For j=1:i WW(i+1,j+1)=WW(i,j)+sqrt(delta_t);
      end;
end;
```

Calculer les valeurs de WW en fonction de δt et indiquer, à titre d'exemple, les 3 suivantes :

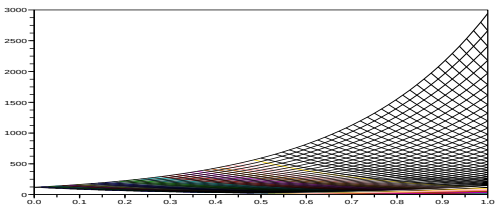
$$\boxed{WW(4,1) = ...\sqrt{\delta t}, \quad WW(5,4) = ...\sqrt{\delta t}, \quad WW(2,2) = ...\sqrt{\delta t}.}$$

Calculer les différentes valeurs de $W(i,j)$ pour $i = 1$, $i = 2$ et $i = 3$. Esquisser l'arbre de Wiener.

Exercice 2. :La marche CRR Considérons à présent le modèle de Cox-Ross-Rubinstein, c'est-à-dire la marche aléatoire S_t définie par $S_0 = S0$ et $S_{t+\delta t} = S_t U_t$, avec $U_t \in \{\texttt{up},\texttt{down}\}$. On suppose que $up = e^{+\frac{\sigma}{\sqrt{n}}}$ et $down = e^{-\frac{\sigma}{\sqrt{n}}}$. Introduire ces deux quantités dans Scilab puis, en supposant, comme précédemment que $S(i,j) = SS(i+1, j+1)$, calculer les valeurs prises par SS et donc par la marche S_t en prenant $\sigma = 0.3$ et $S0 = 100$. Indiquer quelques unes des valeurs de $SS(i,j)$ que vous trouvez, puis toutes les valeurs de $S(i,j)$, pour $i = 1$, $i = 2$ et $i = 3$.

Exercice 3. Tracé de l'arbre CRR : On veut à présent tracer l'arbre CRR joignant les points $(i * \delta t, S(i,j))$ à $((i+1) * \delta t, S(i+1, j + \delta J_{i+1}))$, avec $\delta J_{i+1} \in \{0,1\}$:

Arbre de Cox-Ross-Rubinstein :



On utilisera pour cela l'instruction suivante `plot2d(Abs,Ord)` qui trace les lignes polygonales (ouvertes) dont les abscisse et les ordonnées des sommets figurent dans les colonnes homologues des matrices `Abs` et `Ord`.

Le code suivant utilise un algorithme qui parcourt tous les arêtes de l'arbre CRR une fois et une seule et qui se prête donc bien à la syntaxe de `plot2d`.

```
   Abs=zeros(n+1,n+1); // le dessin comportera n+1 lignes polygonales
Ord=zeros(n+1,n+1); // ayant chacune n cotes, donc n+1 sommets
for k=0 :n // k numerote les lignes polygonales
    for l=0 :n-k // premieres moitie : depart en (t,S(t,k)) avec t=k*delta_t
        Abs(l+1,k+1)=(k+l)*delta_t ; // les abscisses croissent
        Ord(l+1,k+1)=SS(k+l+1,k+1); // et les ordonnees decroissent : j=k=Cste
    end ; // on arrive en (T,S(T,k)) avec T=n*delta_t
    for l=1 :k // on repart de ce point
        Abs(n-k+l+1,k+1)=(n-l)*delta_t ; // avec les abscisses qui diminuent
        Ord(n-k+l+1,k+1)=SS(n-l+1,k-l+1); // et les ordonnees aussi
    end ; // la lignes n^o k abouti en (t,S(t,0)) avec t=(n-k)*delta_t
end ;
```

Tracer l'arbre CRR pour diverses valeurs de n. Quelle est la plus grande valeur atteinte et quelle est la plus petite ?

Utilisez les commentaires du code pour comprendre l'algorithme de parcours de l'arbre, pour une petite valeur de n, $n = 3$ par exemple. Combien de lignes polygonales sont-elles calculées ? Combien y-a-t-il de sommets dans chaque ligne polygonale ? Expliquez.

Exercice 4. : reprendre le code pour tracer cette fois l'arbre de la marche de Wiener. Esquissez le dessin obtenu.

La marche CRR est un modèle pour des cours d'actions que l'on préfère en général à la marche de Wiener. Expliquer pourquoi.

Exercice 5. : Calcul du prix d'un Call On a vu que le prix $c(s, S_t)$ d'un portefeuille de couverture d'une option ayant pour pay-off φ peut se calculer par récurence rétrograde : On commence par la plus grande valeur $i = n$ car $c(T, S_T) = \varphi(S_T)$ puis 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 de calcul. Sous `scilab`, pour le même raisons que pour S, on posera `CC(i+1,j+1)=`$C(i, j)$.

Compléter le code scilab ci-dessous calculant les valeurs `CC(i,j)` d'une option Call d'échéance $T = 1$ dans un modèle à $n = 100$ étapes (3 lignes incomplètes), puis, calculer les valeurs de CC.

```
function phi=phi(S);
phi=.........;
endfunction;
CC=zeros(n+1,n+1);
for j=1 :n+1
CC=(n+1,j)=.....................................;
end;
for i=n :-1 :1
for j=1 :i
CC(i,j)=.................................................;
end;
end;
```

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

Pour $n = 40$ puis pour $n = 150$ la prime serait

Si vous changez la volatilité, quelle conséquence observez-vous sur la prime du Call ? Expliquer.