

CODE MATLAB

```

1%% calcul du call et Put americain CRR
2clear;clc
3
4So=1;K=1;T=10;b=-0.02;h=0.03;r=0.01;p=(h-r)/(h-b)
;
5j_r=-T*log(1+h)/log((1+b)/(1+h));
6Co=0;
7for j=0:floor(j_r)
8    Co=Co+nchoosek(T,j)*p^j*(1-p)^(T-j)*(So*(1+b)
    ^j*(1+h)^(T-j)-K);
9end
10Co=Co/(1+r)^T;
11% utilisation de la relation de paritee call-put
    pour retrouver le prix du
12% put
13Po=Co-So+K*(1+r)^(-T);
14% nombre de simulation de la v.a
15N=1000;
16S1=0;% somme des Yi
17S2=0;% sommes des carres des Yi
18for i=1:N
19    % simulation d'une B(T,p)
20    X=binornd(T,p);Y=max(So*(1+b)^X*(1+h)^(T-X)-K
    ,0);
21    S1=S1+Y;S2=S2+Y^2;
22end
23% calcul d'esperance et actualisation
24Co_mc=(1+r)^(-T)*S1/N;
25% l'ecart-type de l'estimation
26S=sqrt((S2-S1^2/N)/(N-1));
27
28u=1+h;
29d=1+b;
30
31% utilisation d'une fonction recursive
32Pamo=putAmCRR(So,K,r,p,u,d,0,0,T);
33Camo=callAmCRR(So,K,r,p,u,d,0,0,T);
34
35% vecteur ouu on n'a que des hausses
36v1=[sj(So,0,0,u,d) sj(So,1,1,u,d) sj(So,2,2,u,d)
    sj(So,3,3,u,d) ....
37    sj(So,4,4,u,d) sj(So,5,5,u,d) sj(So,6,6,u,d)
    sj(So,7,7,u,d) ....
38    sj(So,8,8,u,d) sj(So,9,9,u,d) sj(So,10,10,u,d)
    ];
39v2=[sj(So,0,0,u,d) sj(So,1,0,u,d) sj(So,2,0,u,d)
    sj(So,3,0,u,d) ....
40    sj(So,4,0,u,d) sj(So,5,0,u,d) sj(So,6,0,u,d)
    sj(So,7,0,u,d) ....
41    sj(So,8,0,u,d) sj(So,9,0,u,d) sj(So,10,0,u,d)
    ];
42
43v3=[sj(So,0,0,u,d) sj(So,1,1,u,d) sj(So,2,1,u,d)
    sj(So,3,2,u,d) ....
44    sj(So,4,2,u,d) sj(So,5,3,u,d) sj(So,6,3,u,d)
    sj(So,7,4,u,d) ....
45    sj(So,8,4,u,d) sj(So,9,5,u,d) sj(So,10,5,u,d)
    ];
46
47v4=[sj(So,0,0,u,d) sj(So,1,0,u,d) sj(So,2,1,u,d)
    sj(So,3,1,u,d) ....
48    sj(So,4,2,u,d) sj(So,5,2,u,d) sj(So,6,2,u,d)
    sj(So,7,3,u,d) ....
49    sj(So,8,3,u,d) sj(So,9,4,u,d) sj(So,10,5,u,d)
    ];
50
51v5=[sj(So,0,0,u,d) sj(So,1,0,u,d) sj(So,2,1,u,d)
    sj(So,3,2,u,d) ....
52    sj(So,4,3,u,d) sj(So,5,4,u,d) sj(So,6,4,u,d)
    sj(So,7,5,u,d) ....
53    sj(So,8,5,u,d) sj(So,9,6,u,d) sj(So,10,6,u,d)
    ];
54
55[val1,pos1]=max(v1); % 10 naturellement, car il y
    a eu que des montees
56[val2,pos2]=max(v2); % 0 car le cours n'a fait
    que baisser
57[val3,pos3]=max(v3); % 9
58[val4,pos4]=max(v4); % 10
59[val5,pos5]=max(v5); % 10
60
61
62function [ pu ] = putAmCRR(So,K,r,p,u,d,n,j,T)
63% cette fonction calcule recursivement le prix d'
    un put americain en
64% utilisant la proposition du cours adaptee au
    put et de la programmation
65% dynamique.
66if(n==T)
67    pu=max(K-So*u^j*d^(T-j),0);
68else
69    pu1=(1+r)^(-1)*(p*putAmCRR(So,K,r,p,u,d,n+1,j
    ,T)+....
70    (1-p)*putAmCRR(So,K,r,p,u,d,n+1,j+1,T));
71    pu2=K-So*u^j*d^(n-j);
72    pu=max(pu1,pu2);
73end
74
75function [ call ] = callAmCRR(So,K,r,p,u,d,n,j,T)
76% cette fonction calcule recursivement le prix d'
    un call americain
77% en utilisant exactement la proposition du cours
    et de la programmation
78% dynamique
79
80if(n==T)
81    call=max(So*u^j*d^(T-j)-K,0);
82else
83    call1=(1+r)^(-1)*(p*callAmCRR(So,K,r,p,u,d,n
    +1,j,T)+(1-p)*callAmCRR(So,K,r,p,u,d,n+1,
    j+1,T));
84    call2=So*u^j*d^(n-j)-K;
85    call=max(call1,call2);
86end
87
88function [ s ] = sj( So,n,j,u,d)
89    % calcul de la valeur de l'actif s'il y a eu
90    % exactement j montee dans la trajectoire
91    s=So*u^j*d^(n-j);
92end

```