

Mini projet 1: Calcul du prix d'une option asiatique

Valentin DE CRESPIN DE BILLY

Matthias LANG

30.11.2021

N. d'étudiant : XXXXXXXX et 313411

Université Catholique de l'Ouest

Mathématiques financières

Appendices

A Code Matlab

```
% ~~~~~%
% UTF-8 %
% 30.11.2021 %
% Valentin DE CRESPIN DE BILLY %
5 % Matthias LANG %
% ~~~~~%

% ~~~~~ %
10 % ~~~~~ Mathematiques financieres: Mini-projet 1 ~~~~~ %
% ~~~~~ %

%% ~~~~~ Parametres ~~~~~ %%
15

S0 = 40; % Prix initial du sous jacent
N = 5; % Nombre des sous-intervalles % verifier que N << .
K = 50; % Prix d'exercice de l'option
20

r = 0.05; % Taux d'interet en univers risque neutre
sigma = 0.04; % Variance fixe de la mouvement brownien

t0 = 0; % Debut de la periode
25 n = 2^10; % Nombre de intervalles
T = 3; % Fin de la periode
nt = 100; % Nombre de trajectoires

30 starttime = datetime('now');
```

```

fprintf('La programme a demarre a %s', starttime);
fprintf('%d -> nombre de trajectoires', nt);
fprintf('%d -> Prix initial du sous jacent', S0)
fprintf('%0.5g -> Prix d exercice de l option', K);
35 tic

%% ~~~~~ Simulation ~~~~~ %%

40 syms func(x)
obligation(x) = S0*(1+r)^(x-t0);
%K = int(obligation,t0,T)/(T-t0);

45 dt = ((T-t0)/n);
t = t0:dt:T;

S = zeros(length(t),nt);
for i = 1:nt
50     S(:,i) = brownmo(S0, r, sigma ,t0, T, n);
end
plot(S)

%% ~~~~~ calcul de X_t ~~~~~ %%

55 vecX_t = zeros(1,nt);
for i = 1:nt
    for j = (1:n)
        vecX_t(i) = vecX_t(i) + (S(j,i)+S(j+1,i))/2;
60     end
end

vecX_t = vecX_t/n;
X_t = mean(vecX_t(:,1));

```

```

65 vecC_inf = vecX_t-K;

vecC_inf = vecC_inf .* ( vecC_inf >= 0 );

C_inf = mean(vecC_inf);
70 %C_inf * exp(-rT) est une martingale donc E[exp(-rT)*C_inf]= C_inf(S_0)
C_inf_0 = exp(-r*T)*C_inf;

%% ~~~~~ calcul de X_t_prim ~~~~~ %%

75 vecX_t_prim = S(1,:);
for i = 2:(n+1)
    vecX_t_prim = vecX_t_prim + S(i,:);
end
vecX_t_prim = vecX_t_prim/(n+1);
80 X_t_prim = mean(vecX_t_prim);
vecC_N = vecX_t_prim-K;

vecC_N = vecC_N .* ( vecC_N >= 0 );

85 C_N = mean(vecC_N);
%C_N * exp(-rT) est une martingale donc E[exp(-rT)*C_N]= C_N(S_0)
C_N_0 = exp(-r*T)*C_N;

90 %% Histogramm
% E_\pi (e^{-rT} (X_T - K)^+ / F_0) \sim 1/nt \sum \{C(T)\}

%% ~~~~~ Plot ~~~~~ %%

95 duree= toc;
fprintf('%d trajectoires plotees\n', nt);

```

```

fprintf('Avec S0 de %d, K = %0.5g \n', S0, K);
100 fprintf('L''integrable par (t_0 - T) de X_t, le prix estime C(T) = %0.5g \n
fprintf('La moyenne des X_t: C(T) = %0.5g \n', C_N_0);
fprintf('Fin en %0.5g\n', duree);

105 tiledlayout(2,1)
nexttile
hold on

x_ax = t; % x-axe
110 axis([0 T 0.8*min(min(S)) 1.5*max(max(S))]) %x-axe limits

plot(x_ax, S)

% pour comparison, si j'epargne pour le taux r:
115 %plot([t0 T], [S0 S0*(1+r)^(T-t0)], "--k"); % obligation
fplot(obligation, [t0 T], "-k");

legend("les prix S_t des actions", "sans risque");

120 hold off

nexttile

histogram( vecC_inf );
125 title("Histogramm des C(T) pour X_{infinie}");

%legend("", "l'estime pour X_inf", "l'estime pour X_N'")
%plot([C_chapeau C_chapeau], [-0.2*nt/sqrt(sigma*nt) nt/sqrt(sigma*nt)], "
%plot([t0 T], [0 0], ":k"); % y=zero

130

%% ~~~~~ fonctions ~~~~~ %%

```

```

135 function S = brownmo(X0, mu, sigma, t0, T, n) %x0
    delta = (T-t0)/n;
    W = zeros(1,n+1);
    tseq = t0:((T-t0)/n):T;
    for i = 2:(n+1)
140     W(i) = W(i-1)+normrnd(0,1)*sqrt(delta);
    end
    S = X0 * exp( (mu-(sigma^2)/2) * (tseq-t0) + sigma*W );
end

145 %% a effacer, aine

function X_t = MoyMob(M, t_m)
    X_t = cumsum(M,1);
150     t_m2= t_m+1;
    X_t = X_t(1:t_m2,:);
    for i = 1:t_m2
        X_t(i,:) = X_t(i,+)/i; %vectoriel?
    end
155 end

Xn = MoyMob(S, n);

% calculer X_T'
160 X_pr = zeros(N,nt);
for i = 1:N %vectoriel?
    X_pr(i,:) = S(floor(i*n/N),:);
end
X_pr = (1/N)*sum(X_pr, 1);

```