

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

Valentin DE CRESPIEN DE BILLY

Matthias LANG

30.11.2021

N. d'étudiant : XXXXXXXX et 313411

Université Catholique de l'Ouest

Mathématiques financières

1 Calculer le prix du sous-jacent

$$dS_t = S_t(rdt + \sigma\sqrt{S_t}dW_t) \quad (1)$$

$$\Longleftrightarrow \frac{dS_t}{S_t} = rdt + \sigma\sqrt{S_t}dW_t \quad (2)$$

On prend l'équation 1 :

$$= dS_t = S_t rdt + \sigma S_t^{1.5} dW_t$$

Puis

$$\begin{aligned} d\langle S_t, S_t \rangle &= \langle dS_t, dS_t \rangle = \\ &= \langle S_t rdt + \sigma S_t^{1.5} dW_t, S_t rdt + \sigma S_t^{1.5} dW_t \rangle = \\ &= \langle \sigma S_t^{1.5} dW_t, \sigma S_t^{1.5} dW_t \rangle = \\ &= S_t^3 \sigma^2 \langle dW_t, dW_t \rangle = \\ &= S_t^3 \sigma^2 dt \end{aligned} \quad (3)$$

On pose : $X_t = \ln(S_t)$

$$\text{Formule d'Ito : } d\ln(S_t) = \frac{dS_t}{S_t} + \frac{1}{2} \frac{1}{S_t^2} d\langle S_t, S_t \rangle$$

Avec les équations 2 et 3 : (4)

$$\begin{aligned} d\ln(S_t) &= rdt + \sigma\sqrt{S_t}dW_t - \frac{1}{2}S_t\sigma^2dt = \\ &= (r - \frac{1}{2}S_t\sigma^2)dt + \sigma\sqrt{S_t}dW_t \end{aligned}$$

$$\begin{aligned} \ln\left(\frac{S_t}{S_0}\right) &= \ln(S_t) - \ln(S_0) = \int_0^t d\ln(S_u) = \\ &= \int_0^t (r - \frac{1}{2}S_u\sigma^2)du + \int_0^t \sigma\sqrt{S_u}dW_u \end{aligned} \quad (5)$$

...

Donc on ne peut pas facilement dériver une formule pour le prix qui dépend que des variables fixées, mais on peut le simuler pas à pas avec (1) :

$$\begin{aligned}
S_0 &\text{ soit connu} \\
dS_0 &= S_0(rdt + \sigma\sqrt{S_0}dW_0) \\
S_1 &\approx S_0 + dS_0 \\
dS_1 &= S_1(rdt + \sigma\sqrt{S_1}dW_1) \\
S_2 &\approx S_1 + dS_1 \\
&\dots
\end{aligned}
\tag{6}$$

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 sous risque neutre

```

```

sigma = 0.04/sqrt(S0); % Variance partie fixe

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 \n', starttime);
fprintf('%d -> Nombre de trajectoires \n', nt);
fprintf('%d -> Prix initial du sous jacent \n', S0)
fprintf('%0.5g -> Prix d''exercice de l''option \n', K);
35 fprintf(' . . . ')
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);
50 for i = 1:nt
    S(:,i) = brownmo(S0, r, sigma ,t0, T, n);
end
plot(S)

55 %% ~~~~~ calcul de X_t ~~~~~ %%

```

```

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

vecX_t = vecX_t/n;
65 X_t = mean(vecX_t(:,1));
vecC_inf = vecX_t-K;

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

70 C_inf = mean(vecC_inf);
%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;

75 %% ~~~~~ calcul de X_t_prim ~~~~~ %%

vecX_t_prim = S(1,:);
for i = 2:(n+1)
    vecX_t_prim = vecX_t_prim + S(i,:);
80 end

vecX_t_prim = vecX_t_prim/(n+1);
X_t_prim = mean(vecX_t_prim);
vecC_N = vecX_t_prim-K;

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

C_N = mean(vecC_N);
%C_N * exp(-rT) est une martingale donc

```

```

90 %  $E[\exp(-rT) * C_N] = C_N(S_0)$ 
C_N_0 = exp(-r*T)*C_N;

% Histogramm
%  $E_{\pi}(e^{-rT} (X_T - K)^+ / F_0) \sim 1/nt \sum \{C(T)\}$ 
95

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

100 duree= toc;
fprintf('%d trajectoires plotees\n', nt);
fprintf('Avec S0 = %d, K = %0.5g \n', S0, K);
fprintf('L''integrale par (t_0 - T) de X_t, ');
    fprintf('le prix estime C(T) = %0.5g \n', C_inf_0);
105 fprintf('La moyenne des X_t: C(T) = %0.5g \n', C_N_0);
fprintf('Fin en %0.5g\n', duree);

tiledlayout(2,1)
110 nexttile
hold on

x_ax = t; % x-axe
axis([0 T 0.8*min(min(S)) 1.5*max(max(S))]) %x-axe limits
115
plot(x_ax, S)

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

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

```

```

hold off

125
nexttile

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

130
%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)], "-k");
%plot([t0 T], [0 0], ":k"); % y=zero

135

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

140 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)
145        W(i) = W(i-1)+normrnd(0,1)*sqrt(delta);
    end
    S = X0 * exp( (mu-(sigma^2)/2) * (tseq-t0) + sigma*W );
end

150

%% a effacer, aine

function X_t = MoyMob(M, t_m)
    X_t = cumsum(M,1);
155    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
160 end

% Xn = MoyMob(S, n);

% calculer X_T'
165 % 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);

```

B Code VBA

```

Rem Attribute VBA_ModuleType=VBAModule
Option VBASupport 1
5 Sub Macro1()

    Dim T, n, nt, Nd As Integer
    Dim r, sigma, S0, t0 As Double

10 Dim i, j As Integer

    r = Range("A2").Value
    sigma = Range("A3").Value
15 T = Range("A4").Value
    n = Range("A5").Value
    nt = Range("A6").Value
    Nd = Range("A7").Value
    S0 = Range("A8").Value

```



```

20 t0 = Range("A9").Value

    Dim dt As Double
    dt = ((T - t0) / n)

25 Dim temps() As Double
    ReDim temps(n + 1)
    temps(0) = t0
    For j = 1 To n + 1
        temps(j) = temps(j - 1) + dt
30 Next

    Range("I2:I" & UBound(temps) + 1) = WorksheetFunction.
        Transpose(temps)

35 Dim S() As Double
    ReDim S(1 To n + 1, 1 To nt)
    Dim W As Double
    Dim x As Double

40 For j = 0 To nt - 1
    For i = 0 To n

        If i = 0 Then
            W = 0
45         Else
            W = W + Sqr(-2 * Log(Rnd())) * Cos(6.283185307 * Rnd())
                * Sqr(dt)
        End If

        x = S0 * Exp((r - (sigma ^ 2) / 2) * (temps(i) - t0) +
            sigma * W)

```

```

50 Cells(2 + i, 10 + j).Value = x ' copier s dans la
    worksheet
    'S(i, j) = x

Next
Next

55

'Cells(i, 3).Value =

60
'S = zeros(length(t),nt);
'for i = 1:nt
'    S(:,i) = brownmo(S0, r, sigma ,t0, T, n); %brownmo
    est definie en bas
'End

65
'function S = brownmo(X0, mu, sigma, t0, T, n) %x0

'    delta = (T-t0)/n;
'    W = zeros(1,n+1);
70 '    tseq = t0:((T-t0)/n):T;
'    for i = 2:(n+1)
'        W(i) = W(i-1)+normrnd(0,1)*sqrt(delta);
'    End
'    S = X0 * exp( (mu-(sigma^2)/2) * (tseq-t0) + sigma*W );

75
MsgBox "Simule pour " & nt & " trajectoires."

'Sheets("Dashboard").Activate
'Range("Parametres").Select
80 'Range("A13").Value = T

```

End Sub