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

Valentin DE CRESPIN DE BILLY

Matthias LANG

30.11.2021

N. d'étudiant : 247067 et 313411

Université Catholique de l'Ouest

Mathématiques financières

1. Calculer le prix du sous-jacent

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

$$\iff \frac{dS_t}{S_t} = rdt + \sigma\sqrt{S_t}dW_t \tag{2}$$

On prend l'équation 1 :

$$\begin{gathered}
= dS_t \;=\; S_t r dt + \sigma S_t^{1.5} dW_t \\
\text{Puis} \\
d\langle S_t,\quad S_t\rangle = \langle dS_t,\quad dS_t\rangle \;= \\
= \langle S_t r dt + \sigma S_t^{1.5} dW_t,\quad S_t r dt + \sigma S_t^{1.5} dW_t\rangle \;= \\
= \langle \sigma S_t^{1.5} dW_t,\quad \sigma S_t^{1.5} dW_t\rangle \;= \\
= S_t^3\sigma^2\langle dW_t,\quad dW_t\rangle \;= \\
= S_t^3\sigma^2 dt
\end{gathered} \tag{3}$$

$$\begin{gathered}
\text{On pose : } X_t = ln(S_t) \\
\text{Formule d'Ito : } dln(S_t) = \frac{dS_t}{S_t} + \frac{1}{2}\frac{-1}{S_t^2}d\langle S_t,\; S_t\rangle \\
\text{Avec les équations 2 et 3 :} \\
dln(S_t) = rdt + \sigma\sqrt{S_t}dW_t - \frac{1}{2}S_t\sigma^2 dt \;= \\
= (r - \frac{1}{2}S_t\sigma^2)dt + \sigma\sqrt{S_t}dW_t
\end{gathered} \tag{4}$$

$$\begin{gathered}
ln(\frac{S_t}{S_0}) = ln(S_t) - ln(S_0) = \int_0^t dln(S_u) = \\
= \int_0^t (r - \frac{1}{2}S_t\sigma^2)du \;+\; \int_0^t \sigma\sqrt{S_t}dW_t \\
\ldots
\end{gathered} \tag{5}$$

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

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

1.1. Réduction de la variance du éstimateur

Les éstimateurs ont une variance telle que : $V\hat{a}r(C) = \hat{\sigma_i^2}/n_t$, oú n_t est le nombre des observations et $\hat{\sigma_i^2}$ est la variance estimée de la population, qui est égal à la variance de l'échantillon.

Supposons que nous ne connaissions ni les paramètres ni la règle à partir desquels les prix sont établis. Nous ne pouvons donc pas augmenter le nombre d'observations pour améliorer l'estimateur. Quelle autre possibilité existe-t-il pour réduire sa variance ?

Avec les techniques de bootstrap on pourrait répliquer les données. Mais on risque de introduir un biais. Si on utilise une variable de contrôle on n'invente pas des nouvelles données, ni risque-t-on de changer l'ésperance.

Appendices

Toutes les fiches se trouvent dans le repository en ligne :

`https://github.com/matthias-10/UCO_actuariat_mini-projet`

A. Graphiques

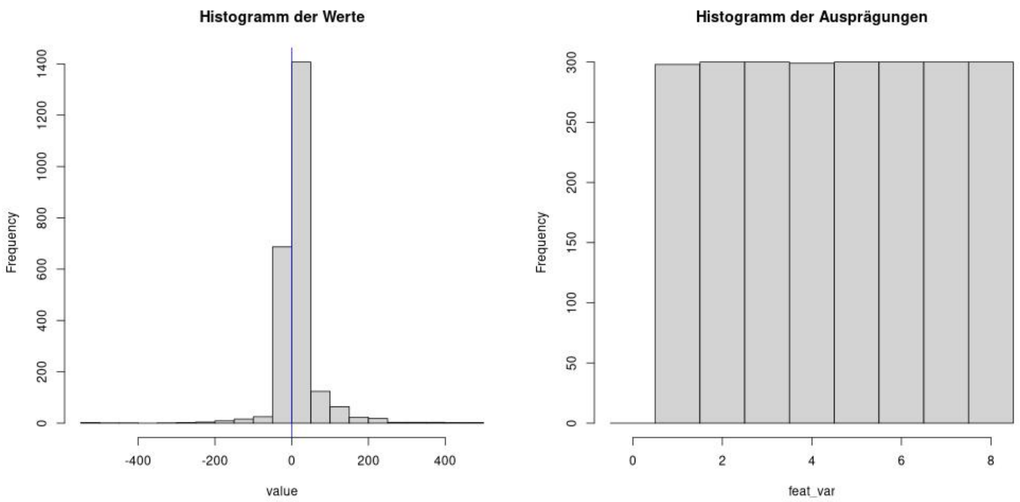

FIGURE 1 – Histogramme d'un èstimateur'

B. Code Matlab

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


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

```
%% ~~~~~~~~~~~~~~~~~~~~ Parametres ~~~~~~~~~~~~~~~~~~~~ %%

S0 = 40;                   % Prix initial du sous jacent
K = 0;                    % Prix d'exercice de l'option

r = 0.05;                  % Taux d'interet sous risque neutre
sigma = 0.01/sqrt(S0);     % Variance partie fixe

Nd = 5;                     % Nombre des sous-intervalles
 % verifier que Nd << n => a faire: ecrire un test
t0 = 0;                    % Debut de la periode
n = 2^9;                   % Nombre de intervalles
T = 1;                     % Fin de la periode
nt = 1000;                 % Nombre de trajectoires


starttime = datetime('now');
fprintf('\n ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ \n');
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)

%1% syms func(x) %1% requires Symbolic Math Toolbox.
%1% obligation(x) = S0*(1+r)^(x-t0);

%K = int(obligation,t0,T)/(T-t0);
%1% bonds_T = obligation(T);
%1% fprintf('%0.5g -> Prix d''une obligation a T\n', bonds_T)
fprintf('%0.5g -> Prix d''exercice de l''option \n', K);
fprintf(' . . . \n\n')
tic



%% ~~~~~~~~~~~~~~~~~~~~ Simulation ~~~~~~~~~~~~~~~~~~~~ %%
```

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

S = zeros(n+1,nt);
S(1,:) = S0;

% Simulation pas a pas
for i = 2:(n+1)
    dW_t = normrnd(zeros(1,nt),sqrt(dt));
    dSi = S(i-1,:).*( r*dt + sigma*sqrt(S(i-1,:)).*dW_t );
    S(i,:) = S(i-1,:) + dSi;
end


%% ~~~~~~~~~~~~~~~ prix de l'option C ~~~~~~~~~~~~~~~~~~ %%
%size(S)
% >ans =      513        1000
C_inf = zeros(1,nt);
C_val = zeros(1,nt);
C_mat = zeros(1,nt);
for j = 1:nt
    S_vec = S(:,j);
    %% ~~~~~~~~~~~~~~~ calcul avec X_t ~~~~~~~~~~~~~~~~~ %%

    %%%%
    % debogage: C_inf = 0, sauf que la premiere et derniere valeur de C_in

    % integral: l'aire de t0 a T sous S
    X_T = 0.5*S0 + sum(S_vec(2:n,:),1) + 0.5*S_vec(n+1,:);
    X_T = X_T/n; %ou (n+1)?

    C_inf_j = X_T - K .* ( X_T - K >= 0 );
    C_inf_0 = exp(-r*T)*C_inf_j;
```

```
% ~ Estimateur ~
% C_inf * exp(-rT) est une martingale donc
% E[exp(-rT)*C_inf]= C_inf(S_0)

C_inf(j)=C_inf_0;



%% ~~~~~~~~~ calcul avec X_t_prim (Valentin)~~~~~~~~ %%

X_t_prim = sum(S_vec,1)/(n+1);
%X_t_prim = mean(vecX_t_prim);
C_N_j = X_t_prim - K .* ( X_t_prim - K >= 0 );

% 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_j;

C_val(j)=C_N_0;


%% ~~~~~~~~~~ calcul avec X_t_prim (Matthias)~~~~~~~ %%

%1/N * sum_1^N S_{kT/N}
% => kT n'est pas un numero entier, il faut arrondir

index = fliplr(1:n);
warn_id = 'MATLAB:colon:nonIntegerIndex';
warning('off', warn_id);
% ^supprime Warning a cause de arrondir:
index = index(1:(n/Nd):end);
X_t_matthias = sum(S_vec(index,:),1)/Nd;

C_N_j = X_t_matthias - K .* ( X_t_matthias - K >= 0 );
```

```
    % 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_j;

    C_mat(j)=C_N_0;

end

%% ~~~~~~~~~~~~ affichage des estimateurs ~~~~~~~~~~~~~~ %%

% C_inf
C_inf_est = mean(C_inf);
C_inf_est_var = var(C_inf);

fprintf('L''estimateur du C_inf a t0 = %0.5g\n', ...
 C_inf_est);
fprintf('Son ecart type = %0.5g\n', sqrt(C_inf_est_var));

fprintf('\n methodes differentes pour C_N, \n premier Valentin, puis Matth
% Valentin C_inf
C_N_est_val = mean(C_val);
C_N_est_var_val = var(C_val);

fprintf('L''estimateur du C_N a t0 = %0.5g\n', ...
 C_N_est_val);
fprintf('Son ecart type = %0.5g\n', sqrt(C_N_est_var_val));

% Matthias C_inf
C_N_est_mat = mean(C_mat);
C_N_est_var_mat = var(C_mat);

fprintf('L''estimateur du C_N a t0 = %0.5g\n', ...
 C_N_est_mat);
```

```
fprintf('Son ecart type = %0.5g\n', sqrt(C_N_est_var_mat));


duree= toc;
fprintf('\n')
fprintf('%d trajectoires simules\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);
fprintf('La moyenne des X_t: C(T) = %0.5g \n', C_N_0);
fprintf('Fini en %0.5g\n', duree);


%% ~~~~~~~~~~~~~~~~~~~~~ graphes ~~~~~~~~~~~~~~~~~~~~~~~ %%

% 1:   graphe de S;
% 2-3: histogrammes de C_inf et C_N;
% 4-5:   boxplot des estimateurs

fprintf('\n 1: graphe de S \n')
input('Tapez [Enter] pour afficher le graphe\n')

%axis([0 T 0.8*min(min(S)) 1.5*max(max(S))]) %x-axe limits

plot(t, S)

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

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


fprintf('\n 2: histogramme de C_inf \n')
```

```
input('Tapez [Enter] pour afficher le graphe\n')

% E_\pi (e^-rT (X_T - K)^+ / F_O) ~ 1/nt \sum {C(T)}
histogram( C_inf );
title("Histogramm des C(T) pour X_{infinie}");


fprintf('\n 3: histogramme de C_N \n')
input('Tapez [Enter] pour afficher le graphe\n')

histogram( C_mat );
title("Histogramm des C(T) pour X_{N}");


fprintf('\n 4: boxplot des estimateurs \n')
input('Tapez [Enter] pour afficher le graphe\n')

%tiledlayout(1,2)
%nexttile
%hold on

boxplot( C_inf );
title('boxplot de C_{infinie} a T')
ylabel('C_T, valeurs actualisees')

%hold off
%nexttile
%hold on

boxplot ( C_mat );
title('boxplot de C_{N} a T')

%hold off
```

C. Code VBA

```
Sub Macro1()

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

Dim i, j As Integer


r = Range("A2").Value
sigma = Range("A3").Value
T = Range("A4").Value
n = Range("A5").Value
nt = Range("A6").Value
Nd = Range("A7").Value
S0 = Range("A8").Value
t0 = Range("A9").Value

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

' afficher t
Dim temps() As Double
ReDim temps(n + 2)
temps(0) = t0
For j = 1 To n + 2
    temps(j) = temps(j - 1) + dt
Next

Range("I3:I" & UBound(temps) + 1) = WorksheetFunction.
   Transpose(temps)

Dim S() As Double
```

```vb
ReDim S(1 To n + 1, 1 To nt)
Dim dW As Double
Dim dS As Double
Dim x As Double

'effacer S() aine

'simuler S pas a pas
For j = 1 To nt
    x = S0
    i = 1
    Cells(2 + i, 9 + j).Value = x 'S(i, j) ' copier s dans
        la worksheet
    Cells(1 + i, 9 + j).Value = "series " & j
    For i = 1 To n + 1
        If i > 1 Then
            dW = Sqr(-2 * Log(Rnd())) * Cos(6.283185307 *
               Rnd()) * Sqr(dt)
            'dS = S(i - 1, j) * (r * dt + sigma * Sqr(S(i
               - 1, j)) * dW)
            'aine = Cells(1 + i, 10 + j).Value
            dS = x * (r * dt + sigma * Sqr(x) * dW)
            'S(i, j) = S(i - 1, j) + dS
            x = x + dS 'S(i - 1, j) + dS
            Cells(2 + i, 9 + j).Value = x  'S(i, j) '
               copier s dans la worksheet
        End If
        S(i, j) = x
    Next
Next

'Range("J21:O100") = S()
```

```
' insert Chart
'Dim Cht As Chart
'Set Cht = Charts.Add
'With Cht
'     .SetSourceData Source:=Sheets("Sheet1").Range("A1:B6
   ")
'     .ChartType = xl3DArea
'End With


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

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

End Sub
```