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

Nous avons essayé d'atteindre une équation qui ne dèpend que des variables connues comme la formule de Black-Scholes. Cela n'a pas fonctionné.

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

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

On prend l'équation 1 :

$$= dS_t \;=\; S_t rdt + \sigma S_t^{1.5}dW_t \quad ; \text{Puis}$$

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

On pose : $X_t = ln(S_t)$

$$\text{Formule d'Ito : } dln(S_t) = \frac{dS_t}{S_t} + \frac{1}{2}\frac{-1}{S_t^2}d\langle S_t,\ S_t\rangle \tag{3}$$

$$\tag{4}$$

Avec les équations 2 et 3 :

$$dln(S_t) = rdt + \sigma\sqrt{S_t}dW_t - \frac{1}{2}S_t\sigma^2 dt = (r - \frac{1}{2}S_t\sigma^2)dt + \sigma\sqrt{S_t}dW_t \tag{5}$$

$$\begin{aligned}
ln(\frac{S_t}{S_0}) &= ln(S_t) - ln(S_0) = \int_0^t dln(S_u) = \\
&= \int_0^t (r - \frac{1}{2}S_t\sigma^2)du \ + \ \int_0^t \sigma\sqrt{S_t}dW_t \\
&= \dots
\end{aligned}$$

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) :

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

1.1 ecdf

Sieht Gaussian aus -> KI damit probieren, aber keine richtige Normalverteilung, also bootstrap !

1.2 Réduction de la variance du éstimateur

Les éstimateurs ont une variance telle que : $\hat{Var}(C) = \hat{\sigma_i^2}/n_t$, 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.

2 Réalisation numerique

Les algorithmes sont réalisées avec deux langues de programmation : Matlab et Visual Basic for Applications. Plusieurs graphiques sont y générés, vous les trouverez dans l'annexe **??**. En plus, avec le logiciel Excel nous avons crée un dashboard, voir une capture d'figure refX. Vous trouverez les scriptes et les images dans l'annexe, et avec la fiche de dashboard également dans le repository.

Appendices

Toutes les fiches se trouvent dans le repository en ligne :

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

A Graphiques

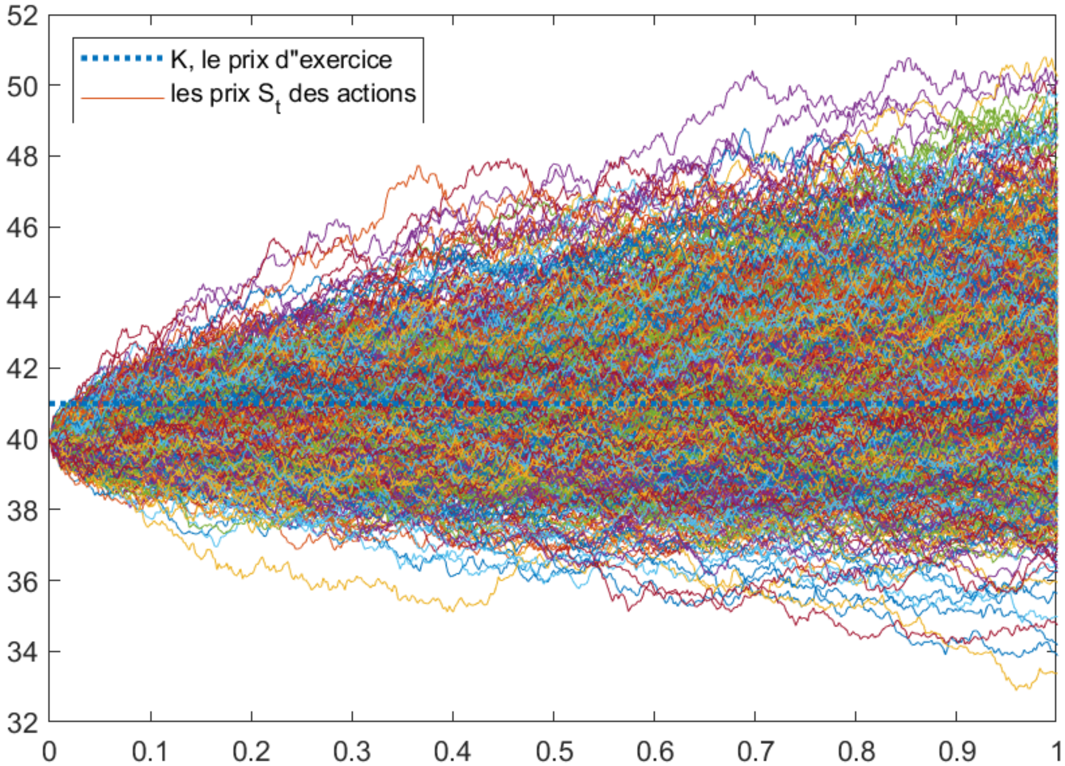


Figure 1 – Les graphes de tous trajectoires, plotés avec matlab

B Code Matlab

```
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ %
% Valentin DE CRESPIN DE BILLY                      UTF-8 %
% Matthias LANG                                30.11.2021 %
% exige:                                                  %
% - Statistics and Machine Learning Toolbox               %
% - Symbolic Math Toolbox                                 %
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ %
```

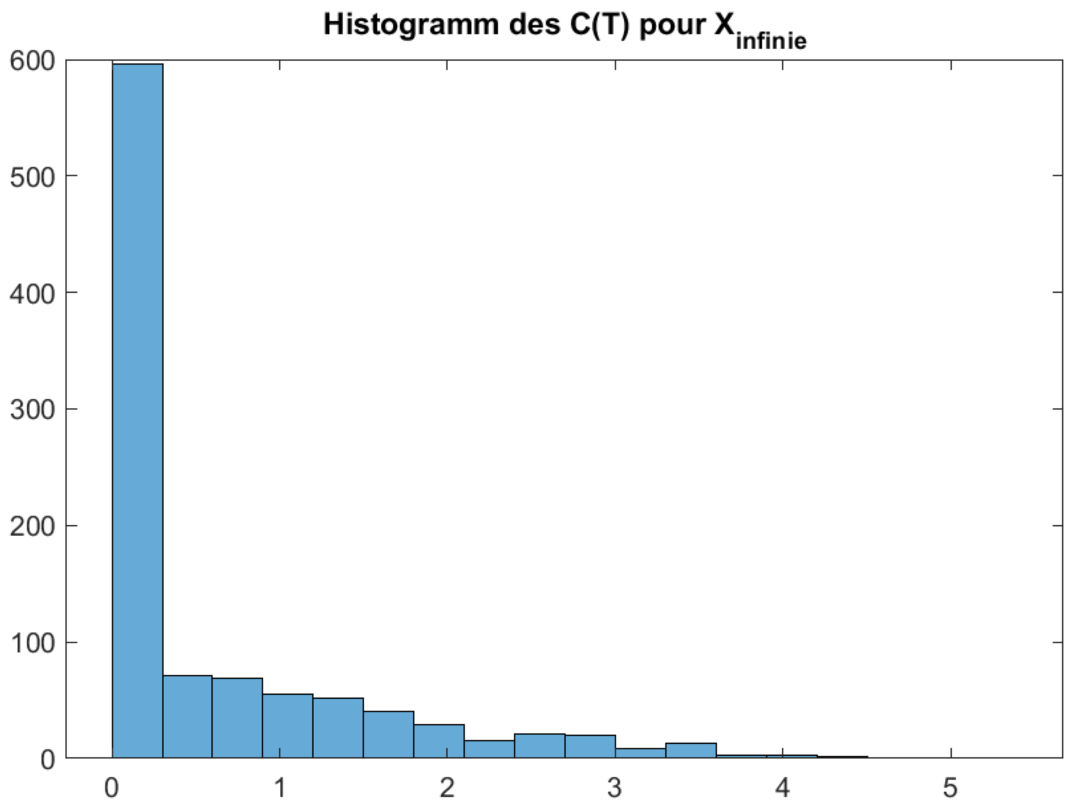


FIGURE 2 – Histogramme des simulations pour C_∞

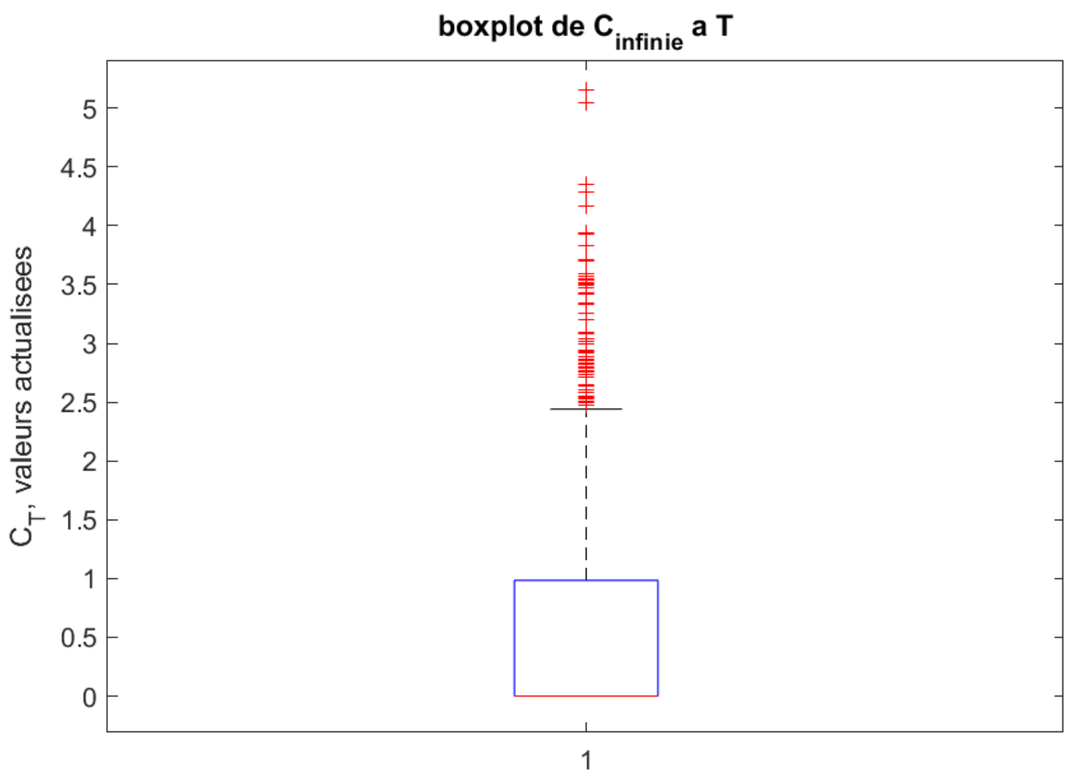


FIGURE 3 – Boxplot des simulations pour C_∞

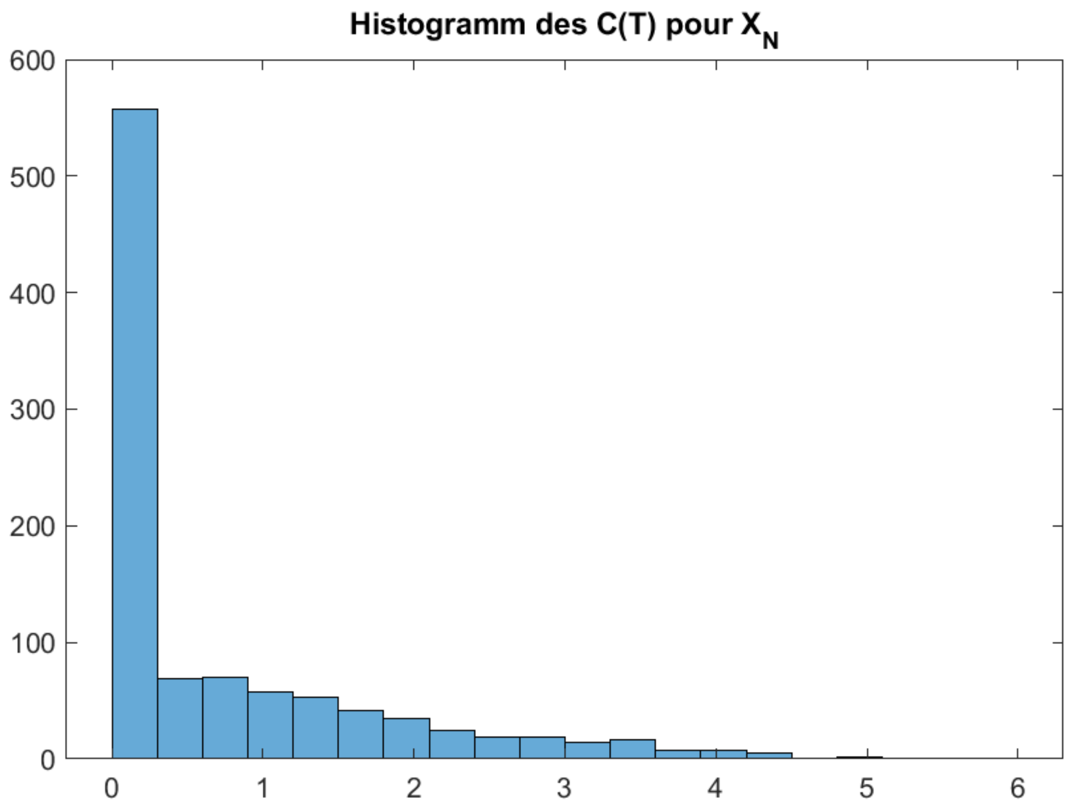


FIGURE 4 – Histogramme des simulations pour C_N

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

%% ~~~~~~~~~~~~~~~~~~~~ Parametres ~~~~~~~~~~~~~~~~~~~~~ %%

S0 = 40;                % Prix initial du sous jacent
r = 0.05;               % Taux d'interet sous risque neutre
sigma = 0.01;           % Variance partie fixe

t0 = 0;                 % Debut de la periode
n = 2^9;                % Nombre de intervalles
T = 5;                  % Fin de la periode
Nd = 8;                 % Nombre des sous-intervalles

nt = 10000;             % Nombre de trajectoires

alpha = 0.05;           % niveau au risque
```

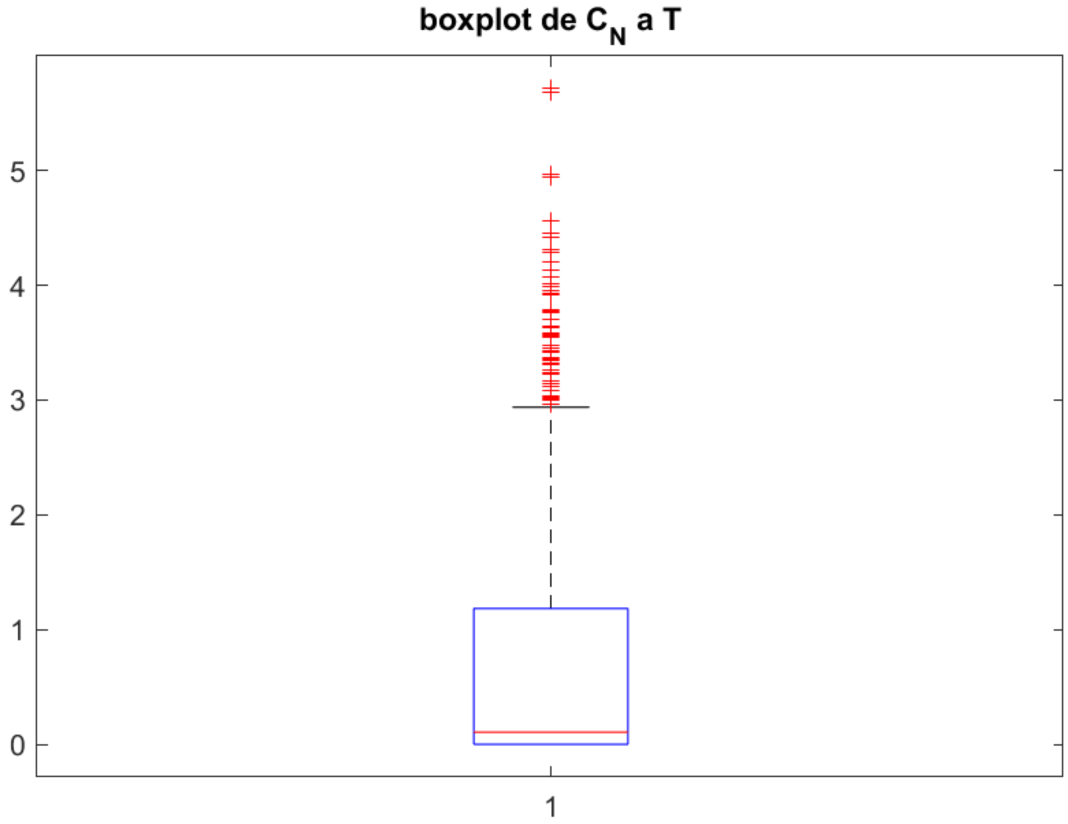


FIGURE 5 – Boxplot des simulations pour C_N

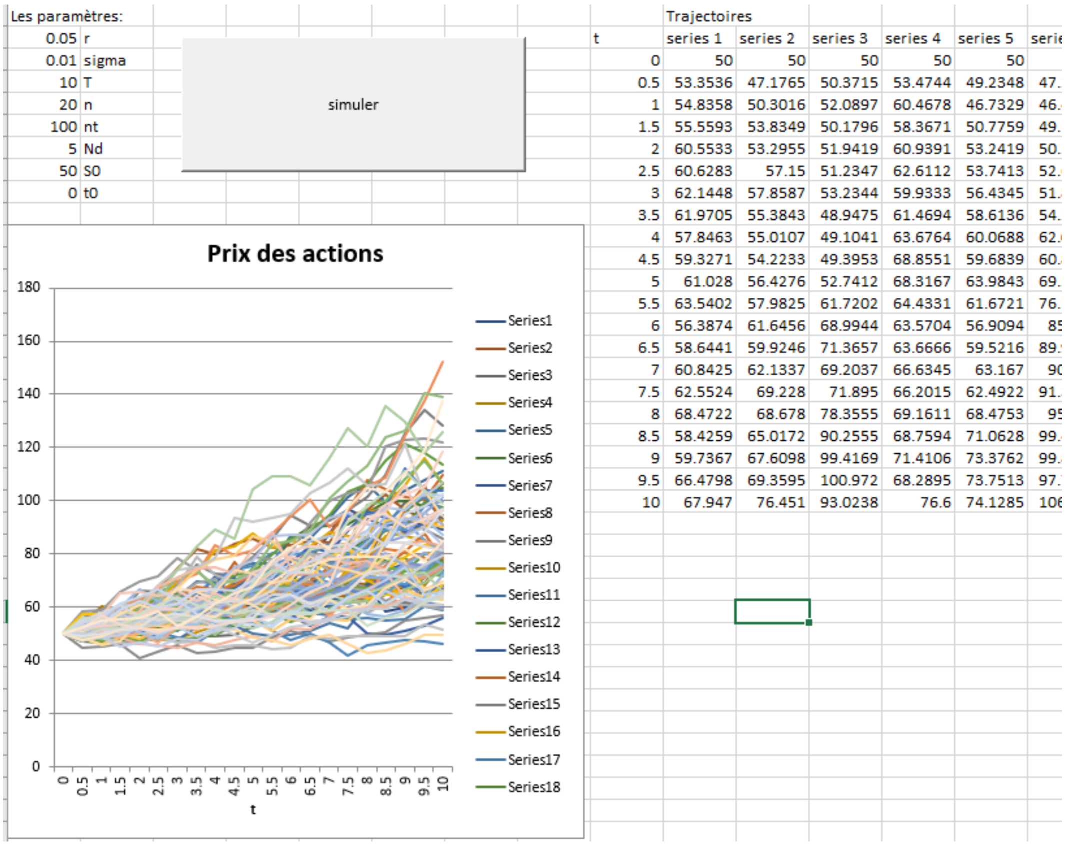
```

%% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ %%

if Nd > n/2-1
    warning("Le nombre de sous-intervalles est tres petit")
    fprintf('Il fallait Nd << n')
end

tic
starttime = datetime('now');
fprintf('\n ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ \n');
fprintf('La programme a demarre a %s \n', starttime);

% K basé sur le prix moyen d'une obligation sans risque
syms func(x)
```



Les paramètres:	
0.05	r
0.01	sigma
10	T
20	n
100	nt
5	Nd
50	S0
0	t0

	Trajectoires					
t	series 1	series 2	series 3	series 4	series 5	serie
0	50	50	50	50	50	
0.5	53.3536	47.1765	50.3715	53.4744	49.2348	47.
1	54.8358	50.3016	52.0897	60.4678	46.7329	46.
1.5	55.5593	53.8349	50.1796	58.3671	50.7759	49.
2	60.5533	53.2955	51.9419	60.9391	53.2419	50.
2.5	60.6283	57.15	51.2347	62.6112	53.7413	52.
3	62.1448	57.8587	53.2344	59.9333	56.4345	51.
3.5	61.9705	55.3843	48.9475	61.4694	58.6136	54.
4	57.8463	55.0107	49.1041	63.6764	60.0688	62.
4.5	59.3271	54.2233	49.3953	68.8551	59.6839	60.
5	61.028	56.4276	52.7412	68.3167	63.9843	69.
5.5	63.5402	57.9825	61.7202	64.4331	61.6721	76.
6	56.3874	61.6456	68.9944	63.5704	56.9094	85
6.5	58.6441	59.9246	71.3657	63.6666	59.5216	89.
7	60.8425	62.1337	69.2037	66.6345	63.167	90
7.5	62.5524	69.228	71.895	66.2015	62.4922	91.
8	68.4722	68.678	78.3555	69.1611	68.4753	95
8.5	58.4259	65.0172	90.2555	68.7594	71.0628	99.
9	59.7367	67.6098	99.4169	71.4106	73.3762	99.
9.5	66.4798	69.3595	100.972	68.2895	73.7513	97.
10	67.947	76.451	93.0238	76.6	74.1285	106

FIGURE 6 – le Excel dashboard

```
obligation(x) = S0*(1+r)^(x-t0);
K = double( int(obligation,t0,T)/(T-t0) );
bonds_T = obligation(T);

fprintf('%d -> Prix initial du sous jacent \n', S0)
fprintf('%0.5g -> Prix univers risque neutre a T\n',bonds_T)
fprintf('%0.5g -> Prix d''exercice de l''option \n', K);
fprintf('calculation en cours . . .\n')

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



%% ~~~~~~~~~~~~~~~ simuler pas a pas ~~~~~~~~~~~~~~~~~~~ %%

S = zeros(nt, n+1);
```

```
% Methode de Euler

S(:, 1) = S0;
S_anti = S;
VC = S;
for i = 2:(n+1)

    % ~~~~~~~~~~~~~~ Simulation des prix ~~~~~~~~~~~~~~~~ %

    dWt = normrnd(zeros(nt,1),sqrt(dt));
    dSi = S(:,i-1).* ...
          ( r*dt + sigma*sqrt(S(:,i-1)).*dWt );
    S(:,i) = S(:,i-1) + dSi;
    S(:,i) = S(:,i) .* (S(:,i) >= 0);

    % ~~~~~~~~~~~~~ variable antithetique ~~~~~~~~~~~~~~~ %

    dWt_a = -1*dWt;
    %dS_anti = S_anti(:,i-1).* ...
    %          ( r*dt + sigma*sqrt(S_anti(:,i-1)).*dWt_a );
    % sans biais (?):
    dS_anti = S_anti(:,i-1).* ...
             ( r*dt + sigma*sqrt(S(:,i-1)).*dWt_a );
    S_anti(:,i) = S_anti(:,i-1) + dS_anti;
    S_anti(:,i) = S_anti(:,i) .* (S_anti(:,i) >= 0);

    % ~~~~~~~~~~~~~ variable de controle ~~~~~~~~~~~~~~~~ %

    dWt_vc = normrnd(zeros(nt,1),sqrt(dt));
    dWt_vc = dWt_vc + dWt_vc ...
             .*sign(dWt).*(sign(dWt_vc) - sign(dWt));
    dSi = VC(:,i-1).* ...
          ( r*dt + sigma*sqrt(S(:,i-1)).*dWt_vc );
```

```
    VC(:,i)      =    VC(:,i-1) + dSi;
    VC(:,i) = VC(:,i) .* (VC(:,i) >= 0);
end

%% ~~~~~~~~~~~~~~ C_inf: calcul avec X_T ~~~~~~~~~~~~~~~ %%

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

C = (X - K) .* logical( X - K >= 0 );
C_0 = exp(-r*T)*C;

% avec la variable antithetique
X_a = (0.5*S0 + sum(S_anti(:,2:n),2) + 0.5*S_anti(:,n+1))/n;
% avec la variable de controle
X_vc = (0.5*S0 + sum(VC(:,2:n),2) + 0.5*VC(:,n+1))/n;

%% ~~~~~~~~~~~~ C_N: calcul avec X_T_prim ~~~~~~~~~~~~~~ %%

% 1/N * sum_1^N S_{kT/N}
% kT n'est necessairement pas un numero entier => 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_prim = sum(S(:,index),2)/Nd;

C_prim = (X_prim - K) .* logical( X_prim - K >= 0 );
C_0_prim = exp(-r*T)*C_prim;



%% ~~~~~~~~~~~~~~~~~~~ estimateurs ~~~~~~~~~~~~~~~~~~~~~ %%
```

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

% C
C_0_est = mean(C_0);
C_est_var = var(C_0)/nt; %/nt ?

X_mu = mean(X);

% C_N
C_0_prim_est = mean(C_0_prim);
C_prim_est_var = var(C_0_prim)/nt; %/nt?

X_prim_mu = mean(X_prim);



%% ~~~~~~~~~~~~~~ intervalle de confiance ~~~~~~~~~~~~~~ %%
%                  (seulement pour N=inf)                  %

v = nt/(nt-1)*var(X); % variance d'echantillonnage

%%% variable supposée normale
X_IC_gauss = [X_mu + sqrt(v/nt)*norminv(alpha/2) ...
              X_mu + sqrt(v/nt)*norminv(1-alpha/2) ];

% variable antithetique
X_ab_mu = mean([X;X_a]);
na = 2*nt;
va = na/(na-1)*var([X;X_a]);
X_a_IC_gauss = [X_ab_mu + sqrt(va/na)*norminv(alpha/2) ...
                X_ab_mu + sqrt(va/na)*norminv(1-alpha/2)];
```

```
% efficace ?
co = cov([X X_a]);
fprintf(['\nLa covariance entre X et la variable '...
        'antithetique est: %0.5g\n'], co(2,2))

%%% bootstrap pour C
sims = 10^3;
y = zeros(1, sims);
for i = 1:sims
    y(i) = mean(randsample(C_0_est,nt,true)) - C_0_est;
end

C_IC_boot =  [C_0_est + quantile(y,alpha/2) ...
              C_0_est + quantile(y,1-alpha/2) ];



%% ~~~~~~~~~~~~~~ variable de controle ~~~~~~~~~~~~~~~~~ %%
%                (seulement pour X_inf)                    %



% on pourrai utiliser au lieu de la var. antithetique la
% variable de controle suivante et vice-versa

% E(Y) ~= E(X) ~= E(Z) =~ mean(X_a)

EY_vc = mean(X_vc);
X_vc;

p = corr(X, X_vc);
% optimum: lambda =~ corr(X,Y)*(Var(X)/Var(Y))^.5
lambda = p*(var(X)/var(X_vc))^.5;
Z_vc = X - lambda * (X_vc - EY_vc);

na = nt;
```

```
va = na/(na-1)*var(Z_vc);

Z_IC_gauss = [EY_vc + sqrt(va/na)*norminv(alpha/2) ...
              EY_vc + sqrt(va/na)*norminv(1-alpha/2)];

% efficace ?
fprintf(['\nLa correlation entre X et la variable '...
        'de controle X_vc est: %0.5g\n'], p)



% utilisant X_a

EY_a = mean(X_a);
Y = 2*EY_a - X_a; % X_mu + EY_a - X_a ???

p = corr(X, Y);
%bien entendu, les deux sont au-peu-pres 1 correles

% optimum: lambda =~ corr(X,Y)*(Var(X)/Var(Y))^.5
lambda = p*(var(X)/var(Y))^.5;
Z_a = X - lambda * (Y - EY_a);

na = nt;
va = na/(na-1)*var(Z_a);

Z_a_IC_gauss = [EY_a + sqrt(va/na)*norminv(alpha/2) ...
                EY_a + sqrt(va/na)*norminv(1-alpha/2)];

% efficace ?
fprintf(['\nLa correlation entre X et la variable '...
        'de controle X_a est: %0.5g\n'], corr(X,X_a))

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

```

```
duree= toc;
fprintf('\n')
fprintf('%d trajectoires simules\n', nt);
fprintf('Fini en %0.5g secondes\n', duree);
fprintf('\n')

fprintf(' ~ Les estimateurs Monte-Carlo: ~ \n')

fprintf('L''estimateur du C_inf a t0 = %0.5g\n', ...
 C_0_est);
fprintf('Son ecart type = %0.5g\n', sqrt(C_est_var));



fprintf(['L''estimateur du C_N a t0, avec ' ...
    '%d sous-intervalles = %0.5g\n'], ...
    Nd, C_0_prim_est);
fprintf('Son ecart type = %0.5g\n', sqrt(C_prim_est_var));

fprintf('\n ~ Des intervalles de Confiance ~ \n');
fprintf('L''intervalle de confiance de X (normal):\n');
X_IC_gauss
fprintf('La meme intervalle avec var. antithetiques:\n');
X_a_IC_gauss
fprintf(['L''intervalle de confiance '...
        'de Z avec VC (normal):\n']);
Z_IC_gauss
fprintf(['L''intervalle de confiance '...
        'de Z avec X_a (normal):\n']);
Z_a_IC_gauss
fprintf('L''intervalle de confiance de C_0 (bootstrap):\n');
C_IC_boot



%% ~~~~~~~~~~~~~~~~~~~~~ graphes ~~~~~~~~~~~~~~~~~~~~~~~ %%
```

```
nt_a = 1; % graphes de S affiches
% 1:   graphes de S;
% 2-3: ecdf de C_inf et C_N;
% 4-5: boxplot des estimateurs
% 6:   deux graphiques qui demontrent une problematique
% 7:   intervalles de confiance

G = "g";
P = input(['\n' ...
    'Pour afficher n''importe quel graphique, tapez ' ...
    'son numero <1-8> ou [Enter]. \n' ...
    'Pour quitter tapez plusieures fois [Enter]:\n'] );

if isstring(P) || isempty(P)
    P = 1;
else
    if ~ismember(P,1:8)
        P = 1;
    end
end

while G~="q"
    disp("[Enter] pour continuer")
    switch P
    case 1
        fprintf('< 1: quelques premiers graphes de S >')
        figure(1)
        plot([t0 T],[K K], ':k', 'LineWidth',2)
        hold on
        plot(t, obligation(t))
        plot(t, S(1:nt_a,:),'b')
        plot(t, S_anti(1:nt_a,:), 'r:')
        plot(t, VC(1:nt_a,:),'g--')
```

```
        % probleme si nt < nt_a
        plot([t0 T],[K K], '--k', 'LineWidth',2)
        hold off
        % pour comparison, si j'epargne pour le taux r:
        %plot([t0 T], [S0 S0*(1+r)^(T-t0)],"--k"); %obl.
        %1% fplot(obligation, [t0 T], "-k");
        legend("K, le prix d''exercice", ...
               "obligation (sans risque)", ...
               "les prix S_t des actions",...
               "les variables antithetiques",...
               "les variables de controle",...
               "Location","northwest");

        if n*nt > 5000*5000; G="q"; end
        P=P+1; input('\n\n');

    case 2
        if n*nt > 5000*5000; G="q"; end

        fprintf(['< 2: fonction de distribution ' ...
            'cumulative estime'  ...
            '\n C(T) pour X_{infinie} de C_infinie >'])
        figure(1)
        % E_\pi (e^-rT (X_T - K)^+ / F_O) ~ 1/nt \sum{C(T)}
        %histogram( C_inf );
        ecdf( X );
        hold on
        plot([K K],[0 1], 'k')
        plot([min(X) max(X)], [.5 .5],':b')
        x_ax = [min(X):.1:max(X)];
        % probleme si max-min < .1
        nor = normcdf(x_ax,X_mu,sqrt(v));
        plot(x_ax,nor,':r')
        hold off
```

```
    legend("ecdf", "K", "P=50%", "cdf normal")
    title("ecdf X(T) pour X_{infinie}");

    P=P+1; input('\n\n');

case 3
    if n*nt > 5000*5000; G="q"; end

    fprintf(['< 3: fonction de distribution ' ...
        'cumulative estime'  ...
        '\n C(T) pour X_{infinie} de C_N >'])
    figure(1)
    ecdf( X_prim );
    hold on
    plot([K K],[0 1], 'k')
    plot([min(X_prim) max(X_prim)], [.5 .5],':b')
    hold off
    legend("ecdf", "K", "P=50%")
    title("ecdf X(T) pour X_{N}");

    P=P+1; input('\n\n');

case 4
    if n*nt > 5000*5000; G="q"; end

    fprintf(['< 4: boxplot de l''estimateur ' ...
            'C_{infinie} >'])
    figure(1)
    boxplot( C_0 );
    title('boxplot de C_{infinie} a T')
    ylabel('C_T, valeurs actualisees')

    P=P+1; input('\n\n');
```

```
case 5
    if n*nt > 5000*5000; G="q"; end

    fprintf('< 5: boxplot de l''estimateur C_{N} >')
    figure(1)
    boxplot ( C_0_prim );
    title('boxplot de C_{N} a T')

    P=P+1; input('\n\n');

case 6
    if n*nt > 5000*5000; G="q"; end

    fprintf('< 6: L''IC de la variable de controle')
    fprintf('\n suivant pour Z a aide de VC')

    plot(sort(Z_vc))
    hold on
    plot(sort(X))
    plot([1 na],[K K], '--k', 'LineWidth',1)
    hold off
    title("X vs variable de controle Z")
    legend("Z","X","K")

    input('\n... 6.5 < scatter >');

    scatter(X,X_vc);
    hold on;
    plot([min(X) max(X)],[min(X) max(X)],'-k');
    plot(X_mu,EY_vc,'*r','LineWidth',2);
    legend("X-X_{vc} en pair",...
           "X=X_{vc}",...
           "les moyennes");
    hold off
```

```
        xlabel("X")
        ylabel("X_{vc}")

        P=P+1; input('\n');

    case 7
        if n*nt > 5000*5000; G="q"; end

        fprintf('< 7: L''IC de la variable de controle ')
        fprintf('\n suivant pour Z a aide de X_a')

        plot(sort(Z_a))
        hold on
        plot(sort(X))
        plot([1 na],[K K], '--k', 'LineWidth',1)
        hold off
        title("X vs variable de controle Z")
        legend("Z","X","Z")

        input('\n... 7.5 < scatter >');

        scatter(X,Y);
        hold on;
        plot([min(X) max(X)],[min(X) max(X)],'-k');
        plot(X_mu,EY_a,'*r','LineWidth',2);
        legend("X-Y en pair","X=Y","les moyennes");
        hold off
        xlabel("X")
        ylabel("Y avec laquelle la v.c. est construite")

        P=P+1; input('\n');
    case 8

        fprintf('< 8: ICs (normales) >')
```

```
        plot([X_mu X_ab_mu EY_vc EY_a],[1 2 3 4], 'x')
        line([K K],[0 5],'Color','green','LineStyle','--')

        line(X_IC_gauss,[1 1])
        line(X_a_IC_gauss,[2 2])
        line(Z_IC_gauss,[3 3])
        line(Z_a_IC_gauss,[4 4])

        legend("estimateurs","K","Z_a","Z_{vc}","X_a","X")
        L1 = X_IC_gauss(2)-X_IC_gauss(1);
        L2 = X_mu - K;
        L = max(L1,L2);
        limf = X_IC_gauss + 0.9*L*[-1 1];
        %[X_mu*(1-limf) X_mu*(1+limf)]
        xlim(limf)
        ylim([0 5])
        yticks(1:4)

        yticklabels({'X','X_a', 'Z (vc)', 'Z (a)'})
        title('Intervalles de confiance (sauf C)')

        P=P+1;
    case 9
        if n*nt > 5000*5000; G="q"; end

        P=input(['\n ' ...
            'Pour afficher n''importe quel graphique, ' ...
            'tapez son numero <1-8> \n']);
        if ismember(P, 1:8)
            fprintf("Vous avez choisi: ")
        else
            G="q";
        end
    otherwise
```

```
        G="q";
    end
end

if n*nt > 5000*5000
    warning("Donnees trop grandes pour affichage");
end

fprintf("\n ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~\n")
fprintf(  " ~   MERCI POUR VOTRE ATTENTION    ~")
fprintf("\n ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~\n")
```

C Code VBA

```
Sub Macro1()

' parametres
Dim T, n, nt, Nd As Integer
Dim r, sigma, S0, t0 As Double

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)

' premier cellule de la table de trajectoires ~ t0
Dim Srow, Scol As Integer
```

```
Dim Scol_abc As String
Srow = 3
Scol_abc = "I"
Scol = 9

' worksheets
Dim sh_dash, sh_calc, sh_s As String
Dim sh_dash_o As Worksheet
sh_s = "Dashboard"
sh_dash = "Dashboard"
Set sh_dash_o = Worksheets(sh_dash)

' iteratives
Dim i, j As Integer

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

'effacer t et S() aines
With Worksheets(sh_s)
    Range(.Cells(Srow - 1, Scol), .Cells(Srow + 10000,
       Scol + 10000)).Delete
End With

' 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(Scol_abc & Srow & ":" & Scol_abc & UBound(temps) +
   1) = _
    WorksheetFunction.Transpose(temps)

'simuler et afficher S pas a pas
Cells(1 + 1, 9 + 0).Value = "t"
For j = 1 To nt
    x = S0
    i = 1
    Cells(2 + i, 9 + j).Value = x
    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
        End If
        S(i, j) = x
    Next
Next
'Range("J21:O100") = S()


' insert Chart
Worksheets(sh_s).Activate
Dim chartrange As Range
Set chartrange = Cells(Srow, Scol + 1)  'sans t
```

```vba
Set chartrange = chartrange.Resize(n + 1, nt)
MsgBox chartrange.Address

Worksheets(sh_dash).Activate
Dim Graphe As Object

'effacer graphes aines
For Each Graphe In ActiveSheet.ChartObjects
  Graphe.Delete
Next Graphe

Set Graphe = sh_dash_o.ChartObjects.Add( _
    Left:=Range("A11").Left, Width:=380, _
    Top:=Range("A11").Top, Height:=400)
With Graphe.Chart
    .SetSourceData chartrange
    .PlotBy = xlColumns 'echanger x et y axes
    .ChartType = xlLine
    .HasTitle = True
    .ChartTitle.Text = "Prix des actions"
    .FullSeriesCollection(1).XValues = _
        Range(Scol_abc & Srow & ":" & Scol_abc & UBound(
           temps) + 1)
    .Axes(xlCategory).HasTitle = True
    .Axes(xlCategory).AxisTitle.Text = "t"
End With


MsgBox "Simulation finie pour " & nt & " trajectoires."

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

```

```
End Sub
```