Méthodes de Monte Carlo pour le pricing d'options

Mohamed Ben Alaya

14 février 2013

Nous allons tester les différentes méthodes probabilistes vu dans le cours en l'appliquant au calcul du call ou le put européen. On considère, pour $t \in [0,T]$, le modèle de Black & Scholes solution de l'EDS

$$dS_t = rS_tdt + \sigma S_tdW_t, \quad S_0 = s_0 > 0,$$

où $(W_t)_{0\leq t\leq T}$ désigne un mouvement Brownien standard et $(\mathcal{F}_t)_{0\leq t\leq T}$ sa filtration canonique.

- On sait que l'EDS ci-dessus admet une solution unique donnée par

$$S_t = s_0 exp\left((r - \frac{\sigma^2}{2})t + \sigma W_t\right).$$

  On note par $(S_t^{0,s_0})_{0\leq t\leq T}$ la solution de l'EDS précédente issue de s_0 à l'instant 0. Le prix d'une option européenne, de payoff $g(S_T)$ où g est une fonction numérique, s'écrit comme

$$\mathbb{E}\left(e^{-rT}g(S_T^{0,s_0})\right).$$

- On rappelle que pour le call, le payoff $g(x) = (x-K)_+$ et la valeur de l'option, à l'instant $t \in [0,T]$, est égale à $V(t, S_t^{0,s_0})$ avec

$$V(t,x) = xN(d_1) - Ke^{-r(T-t)}N(d_2),$$

  où

$$d_1 = \frac{\log(x/K) + (r + \sigma^2/2)(T-t)}{\sigma\sqrt{T-t}} \quad et \quad d_2 = d_1 - \sigma\sqrt{T-t}$$

  Pour le put on a une formulation analogue avec $V(t,x) = Ke^{-r(T-t)}N(-d_2) - xN(-d_1)$.

L'objectif de ce TP est d'illustrer numériquement dans le cadre du modèle de Black& Scholes les résultats du cours sur les schémas d'Euler et de Milshtein. On note T l'horizon de la simulation, $\delta = T/N$ (où $N \in \mathbb{N}^*$) le pas de discrétisation et pour $k \in \{0,\cdots,N\}, t_k = k\delta$ le k-ème instant de discrétisation.

1 Convergence forte

Exprimer en fonction de la valeur en t_k du sous-jacent S_{t_k} (resp. du schéma d'Euler $S^e_{t_k}$, resp. du schéma de Milshtein $S^m_{t_k}$) et de l'accroissement $\Delta W_{k+1} = W_{t_{k+1}} - W_{t_k}$ la valeur en t_{k+1} du sous-jacent $S_{t_{k+1}}$ (resp. du schéma d'Euler $S^e_{t_{k+1}}$, resp. du schéma de Milshtein $S^m_{t_{k+1}}$. Quel est le comportement de $\mathbb{E}((S_T - S^e_T)^2)$ en fonction de N ? Et celui de $\mathbb{E}((S_T - S^m_T)^2)$?

```
//paramètres
T=1;//horizon
sig=0.2;//volatilité
r=0.05;//intérêt
S0=100;//valeur initiale du sous-jacent
N=2;//nombre initial de pas de discrétisation
M=10000;//nombre de simulations indépendantes

erreul=[];//vecteur des erreurs fortes Euler
errmil=[];//vecteur des erreurs fortes Milshtein
liceul=[];//largeur des intervalles de confiance Euler
licmil=[];//largeur des intervalles de confiance Milshtein
Npas=[];// vecteur des nombres de pas

for j=1:5, //boucle sur le nombre de pas
  //paramètres utiles pour la discrétisation avec N pas

  //////////////////////////////////////////////////
  //À compléter avec le calcul des paramètres utiles
  //////////////////////////////////////////////////

  //variables de stockage
  someul=0;
  careul=0;
  sommil=0;
  carmil=0;


  for i=1:M, //simulations indépendantes
S=S0;
Se=S0;
Sm=S0;
for k=1:N, //boucle sur les pas de temps
  g=rand(1,'g');//génération d'une gaussienne centrée réduite

  ////////////////////////////////////////////////////////////
  //À compléter avec l'evolution du sous-jacent et des schémas
  ////////////////////////////////////////////////////////////

end;
someul=someul+(S-Se)^2;
careul=careul+(S-Se)^4;
sommil=sommil+(S-Sm)^2;
carmil=carmil+(S-Sm)^4;
```

```
  end;
  erreul=[erreul,someul/M];
  liceul=[liceul,1.96*sqrt((careul/M-(someul/M)^2)/M)];
  errmil=[errmil,sommil/M];
  licmil=[licmil,1.96*sqrt((carmil/M-(sommil/M)^2)/M)];
  Npas=[Npas,N];

  N=N*2;//multiplication du nombre N de pas par 2
end;

//Affichage des vecteurs d'erreurs et des demi-largeurs des IC à 95%
erreul
liceul
errmil
licmil


//representation graphique de (1/N,erreul) et (1/(N*N),errmil)
xbasc();
subplot(2,1,1);
plot2d(1../Npas',[erreul;erreul-liceul;erreul+liceul]');
subplot(2,1,2);
plot2d(1../Npas^2',[errmil;errmil-licmil;errmil+licmil]');
```

2 Convergence faible

On souhaite maintenant étudier la vitesse faible des schémas d'Euler et de Milshtein pour le calcul d'un Put européen d'échéance T dans le modèle de Black-Scholes, c'est-à-dire la dépendance en N des quantités $\mathbb{E}(e^{-rT}(K - S^e_T)+) - \mathbb{E}(e^{-rT}(K - S_T)+)$ et $\mathbb{E}(e^{-rT}(K - S^m_T)+) - \mathbb{E}(e^{-rT}(K - S_T)+)$. Pour cela, on va évaluer ces quantités d'une part en calculant $\mathbb{E}(e^{-rT}(K - S_T)+)$ par la formule de Black-Scholes et d'autre part en approchant cette espérance par un calcul Monte-Carlo utilisant les mêmes accroissements browniens que ceux qui servent à calculer $\mathbb{E}(e^{-rT}(K - S^e_T)+)$ et $\mathbb{E}(e^{-rT}(K - S^m_T)+)$. Quel est le comportement de $\mathbb{E}(e^{-rT}(K - S^e_T)+) - \mathbb{E}(e^{-rT}(K - S_T)+)$ en fonction de N ? Et celui de $\mathbb{E}(e^{-rT}(K - S^m_T)+) - \mathbb{E}(e^{-rT}(K - S_T)+)$?

```
//paramètres
T=1;//horizon
sig=0.2;//volatilité
r=0.05;//intérêt
S0=100;//valeur initiale du sous-jacent
K=110;// Strike du Put considéré
N=2;//nombre initial de pas de discrétisation
```

```
M=10000;//nombre de simulations indépendantes

//formule Black-Scholes pour le prix du Put
d1=(log(S0/K)+r*T)/(sig*sqrt(T))+sig*sqrt(T)/2;
d2=d1-sig*sqrt(T);
BS=K*exp(-r*T)*cdfnor("PQ",-d2,0,1)-S0*cdfnor("PQ",-d1,0,1);



//fonction de payoff du put
function y=put(S,K)
  y=0;
  if (S<K)
y=K-S;
  end;
endfunction;




erreul=[];//vecteur des erreurs faibles Euler
liceul=[];//largeur des intervalles de confiance 95% Euler
errmil=[];//vecteur des erreurs faibles Milshtein
licmil=[];//largeur des intervalles de confiance 95% Milshtein
conterreul=[];//vecteur des erreurs faibles Euler avec variable de contrôle
contliceul=[];//largeur des intervalles de confiance 95%
conterrmil=[];//vecteur des erreurs faibles Milshtein avec variable de contrôle
contlicmil=[];//largeur des intervalles de confiance 95%
Npas=[];// vecteur des nombres de pas



for j=1:5,//boucle sur le nombre de pas

  //variables de stockage
  puteul=0;
  careul=0;
  putmil=0;
  carmil=0;
  puteul=0;
  contputeul=0;
  contcareul=0;
  contputmil=0;
  contcarmil=0;
```

```
  //paramètres utiles pour la discrétisation avec N pas

  //////////////////////////////////////////////////
  //A compléter avec le calcul des paramètres utiles
  //////////////////////////////////////////////////

  for i=1:M,//simulations indépendantes
S=S0;
Se=S0;
Sm=S0;
for k=1:N,//boucle sur les pas de temps
  g=rand(1,'g');//génération d'une gaussienne centrée réduite

  ////////////////////////////////////////////////////////////
  //À compléter avec l'evolution du sous-jacent et des schémas
  ////////////////////////////////////////////////////////////

end;
//contribution de la trajectoire de S au Put
paymc=put(S,K);
//contribution de la trajectoire schéma d'Euler au Put
payeul=put(Se,K);
puteul=puteul+payeul;
careul=careul+payeul^2;

////////////////////////////////////////////////////////////
//À compléter en stockant dans contputeul (resp. contcareul)
//la somme (resp. somme des carrés) des différences entre payoff
//avec S et payoff avec Se
////////////////////////////////////////////////////////////

//contribution de la trajectoire schéma de Milshtein au Put

////////////////////////////////////////////////////////////
//Reprendre ce qui précède en remplaçant Euler par Milshtein
////////////////////////////////////////////////////////////

  end;
  erreul=[erreul,exp(-r*T)*puteul/M-BS];
  liceul=[liceul,1.96*exp(-r*T)*sqrt((careul/M-(puteul/M)^2)/M)];
  errmil=[errmil,exp(-r*T)*putmil/M-BS];
  licmil=[licmil,1.96*exp(-r*T)*sqrt((carmil/M-(putmil/M)^2)/M)];
  conterreul=[conterreul,exp(-r*T)*contputeul/M];
```

```
  contliceul=[contliceul,1.96*exp(-r*T)*sqrt((contcareul/M-(contputeul/ ...
  M)^2)/M)];
  conterrmil=[conterrmil,exp(-r*T)*contputmil/M];
  contlicmil=[contlicmil,1.96*exp(-r*T)*sqrt((contcarmil/M-(contputmil/ ...
  M)^2)/M)];
  Npas=[Npas,N];
  N=N*2;//multiplication du nombre N de pas par 2
end;

erreul
liceul
errmil
licmil
conterreul
contliceul
conterrmil
contlicmil

//representation graphique de (1/N,conterreul) et (1/N,conterrmil)
xbasc();
subplot(2,1,1);
plot2d(1../Npas',[conterreul;conterreul-contliceul;conterreul+contliceul]');
subplot(2,1,2);
plot2d(1../Npas',[conterrmil;conterrmil-contlicmil;conterrmil+contlicmil]');
```