MPM2 Projet Mathématique 2017-2018

Sovathana SOUN
Afizullah RAHMANY

22 mai 2018

Table des matières

1 Introduction

Le but de ce projet est de modéliser un marché financier et de déterminer le prix et la couverture d'options européennes. Ce marché est caractérisé par deux actifs (un actif non-risqué qui désigne le prix à un instant t, supposé connu à l'instant initial appelé $\boldsymbol{S_t^0}$ et un actif risqué qui désigne le prix $\boldsymbol{S_t}$ à un instant t qui est une variable aléatoire).

L'exercice consiste à modéliser l'évolution du prix de l'actif risqué et d'en déduire la stratégie du vendeur de l'option pour s'assurer d'être couvert

Pour résoudre ce problème, nous allons faire notre modélisation dans le cas discret, puis dans le cas continue et enfin nous étudieront la convergence des ces deux modèles. Tout d'abord, nous choisirons le modèle de Cox-Ross-Rubinstein pour modéliser en premier temps l'évolution des prix de manière discrète avec une progression par arbre, Ensuite, nous utiliserons le modèle de Black-Scholes pour modéliser l'évolution des prix de manière continu.

2 Modèle de Cox-Ross-Rubinstein

Supposons que les prix des deux actifs sur le marché évoluent de manière discrète et ne changent qu'aux dates $(t_i)_{0\leq i\leq N}$ avec $t_0 = 0, t_N = T$ et $t_{i+1} - t_i = \delta$ pour $i \in \{0, .., N-1\}$ où $\delta = T/N$.
Le prix de l'actif sans risque à la date t_i pour $i \in \{0, \ldots, N\}$ est $S_{ti}^0 = (1 + r_N)^i$, où r_N est appelé est une constante positive appelée taux sans risque.
Le prix de l'actif risqué à l'instant initial est $S_0^N = s > 0$ et vaut $S_{ti}^N = T_i^N \times S_{ti-1}^N$ à l'instant t_i pour $i \in \{1..N\}$ où T_i^N est une variable aléatoire qui prend soit la valeur $(1 + h_N)$ soit la valeur $(1 + b_N)$ avec $b_N < r_N < h_N$. On supposera par la suite que les variables aléatoires $(T_i^N)_{1\leq i\leq N}$ sont i.i.d.

On note $\Omega := \ (\omega)_{1\leq i\leq l}$ l'ensemble des évènements possibles.

1. $l = N + 1$, c'est-à-dire que $S_{t_N}^{(N)}$ peut prendre $N + 1$ valeurs différentes.

$$S_{t_N}^{(N)} = s(1 + \ h_N)^{N-i}(1 + b_N)^i{}_{0\leq i\leq N}$$

Q est appelée la probabilité risque neutre. On note $q_N = Q(T_1^N = 1 + h_N)$.

$$E_Q\left[T_i^{(N)}\right] = 1 + r_N$$

$$\sum_{i=1}^{N} T_{1,i}^{(N)} Q(T_{1,i}^{(N)}) = 1 + r_N$$

$$q\,(1 + h_N) + (1 - q)(1 + b_N) = 1 + r_N$$

$$\Rightarrow q = \frac{r_N - b_N}{h_N - b_N}$$

2. Une fois que l'on connait la probabilité risque-neutre Q on peut déterminer le prix $P_{(N)}$ de l'option qui rapporte $f(S_t^N)$:

$$p_{(N)} = \frac{1}{(1 + r_N)^N} \times E_Q[f(S_{t_N}^{(N)})]$$

$$= \frac{1}{(1+r_N)^N} \times E_Q[f(s\prod_{i=1}^{N} T_i^N)]$$

$$= \frac{1}{(1+r_N)^N} \times \sum_{k=0}^{N} f(s\prod_{i=1}^{N} T_i^N) \times Q(\{\omega\}\, |T_k = \omega_k)$$

$$= \frac{1}{(1+r_N)^N} \times \sum_{k=0}^{N} f(s(1+h_N)^{N-k} \times (1+b_N)) \times q^{N-k}\,(1-q)^k \binom{k}{N}$$

2.1 Premier pricer

3. La fonction price1 prend en argument N, r_N , h_N, b_N, s ainsi que la fonction f et retourne le prix $p_{(N)}$.

Algorithm 1 fonction *price*1

function PRICE1$(N, r_N, h_n b_N, s, f)$
  soit $x = 0$
  $q = \frac{r_N - b_N}{h_N - b_N}$
  for k in range $(N+1)$ **do**
    $x = x + f\left(s(1+h_N)^k(1+b_N)^{n-k}\binom{k}{N}\right) q^N(1-q)^{n-k}$
  end for
  return $\frac{x}{(1+r)^N}$
end function

4. Ce premier pricer nous donne la valeur $p_{(N)} = 18.5687$ euros pour un actif partant d'un prix initial de 100 euros et dont la valeur peut varier de +/-62 euros (cas extrêmes) au bout de 10 intervalles de temps.

2.2 Deuxième pricer

5. Une autre méthode pour déterminer le prix $p_{(N)}$ est de déterminer le prix à chaque instant t_k avec $k \in \{0, .., N\}$ en commençant par le prix en t_N.

Étape 1 : à la date t_N, on définit la fonction v_N par

$$v_N(S_{t_N}^{(N)}) = f(S_{t_N}^{(N)})$$

Etape k : à la date t_k , $k \in \{0, \ldots, N-1\}$, le prix est donné par la fonction v_k qui est définie par récurrence

$$v_k(S_{t_k}^{(N)}) = \frac{1}{1+r_N} E_Q\left[\, v_{k+1}(S_{t_{k+1}}^{(N)}) \middle| S_{t_k}^{(N)} \right]$$

Le prix à la date initiale est alors $p_N = \ v_0(S_0^{(N)})$.

La fonction v est une fonction récursive qui prend en argument N, r_N, s, h_N, b_N, f, t avec t l'instant où on se situe et f la fonction donnant le gain et qui a pour cas terminale l'instant à la date t_N et qui retourne la valeur v_t suivant la trajectoire dans lequel on se place et qui est spécifié par la valeur de s.

La fonction price2 prend en argument N, r_N, h_N, b_N, f, s et qui évalue simplement la fonction

v en $t = 0$. Celle-ci remontre jusqu'à t = N - 1 et calcule donc le prix p_N.

$$
\begin{aligned}
v_k(S_{t_k}^{(N)}) = \tfrac{1}{1+r_N} E_Q \left[v_{k+1}(S_{t_{k+1}}^{(N)}) \middle| S_{Tt_k}^{(N)} \right] \\
&= \frac{1}{1+r_N} E_Q \left[\frac{1}{1+r_N} E_Q \left[v_{k+2}(S_{t_{k+2}}^{(N)}) \middle| S_{t_{k+1}}^{(N)} \middle| S_{t_k}^{(N)} \right] \right] \\
&= \frac{1}{(1+r_N)^2} E_Q \left[v_{k+2}(S_{t_{k+2}}^{(N)}) \middle| S_{t_{k+1}}^{(N)} \middle| S_{t_k}^{(N)} \right] \\
&= \frac{1}{(1+r_N)^{N-k}} E_Q \left[f(S_{t_N}^{(N)}) \middle| S_{t_k}^{(N)} \right]
\end{aligned}
$$

$$
v_k\left(S_{t_k}^{(N)}\right) = \frac{1}{(1+r_N)^{N-k}} \sum_{i=0}^{N-k} f\left(S_{t_k}^{(N)} (1+h_N)^{N-k-i} (1+b_N)^i\right) \begin{pmatrix} i \\ N-k \end{pmatrix} q^{N-k-i} q^i
$$

Algorithm 2 fonction *price*2

function PRICE2(N, r_N, h_n, b_N, s, f)
  function V(N, r_N, h_N, b_N, s, t)
    $q = \frac{r_N - b_N}{h_N - b_N}$
    if $t = N - 1$ **then**
      return $\{\frac{1}{1+r}\left(q \times f\left(s(1+h_N)\right) + \left((1-q) \times f(s(1+b_N))\right)\right)\}$
    else
      return $\{\frac{1}{1+r_N}\left(\left(q \times V\left(s(1+h_N), t+1\right)\right) + \left((1-q) \times V(s(1+b_N), t+1)\right)\right)\}$
    end if
  end function
  return $\{V(N, r_N, h_N, b_N, s, 0)\}$
end function

6. On a testé la fonction *price*2 pour $f(x) = \max(x - 95, 0)$ avec $s = 100$, $r_N = 0.02$, $h_N = 0.05$, $b_N = -0.05$, $N = 3$. Il est étonnant de constater que $f(x) = \max(x - 95, 0)$ alors que $s = 100$ qui signifie que si on l'action vaut 100 euros à la fin, cela nous donne quand même un gain de 5 euros au départ. On obtient $p_{(N)} = 10.7573$ euros pour un actif partant d'un prix initial de 100 euros et dont la valeur peut varier sur 3 intervalles de temps.

Voici l'arbre dont les noeuds correspondent aux valeurs $v_{(k)}$ avec la branche de gauche quand l'action diminue et la branche de droite quand l'action a augmenté au noeud précédent.

$v_0(S_0) = 10.76$

4.59 13.71

0 6.68 17.11

0 4.5 14.97 26.55

2.3 Comparaison

7. On va comparer les deux prix avec : $s = 100, r = 0.03, h = 0.05, b = -0.05, N = 5$ et $f(x) = \max(x - 100, 0)$.

On obtient avec la pricer 1 : $P_{(N)} \; = 14.067068522900389$ euros
On obtient avec la pricer 2 : $P_{(N)} \; = 14.067068522900382$ euros

On observe que les deux prix sont quasiment les mêmes à une erreur d'arrondie.

2.4 La couverture

La couverture est la manière d'investir notre argent (en tant que vendeur de l'option) à chaque instant $t_k, k \in \{0, ..., N-1\}$, afin d'avoir à la date T exactement $f(S_T^{(N)})$. Notons $\alpha_k(x)$ (resp. $\beta_k(x)$) le nombre d'actif risqué (resp. le nombre d'actif sans risque).

8. Pour les autres date t_k, le vendeur doit toujours avoir à la date t_k la valeur $v_k(S_{t_k}^N)$ en sa possession.
On donc avoir l'équation

$$\alpha_{N-1}(S_{t_{N-1}}^{(N)})(S_{t_N}^{(N)}) + \beta_{N-1}(S_{t_{N-1}}^{(N)})S_{t_N}^0 = f(S_{t_N}^{(N)})$$

Alors, on obtient :

$$\begin{cases} \alpha_{N-1}(S_{t_{N-1}}^{(N)})(1+h_N)(S_{t_{N-1}}^{(N)}) + \beta_{N-1}(S_{t_{N-1}}^{(N)})(1+r_N)^N = f\left((1+h_N)(S_{t_{N-1}}^{(N)})\right) \\ \alpha_{N-1}(S_{t_{N-1}}^{(N)})\,(1+b_N)\,(S_{t_{N-1}}^{(N)}) + \beta_{N-1}(S_{t_{N-1}}^{(N)})(1+r_N)^N = f\left((1+b_N)(S_{t_{N-1}}^{(N)})\right) \end{cases}$$

Ensuite,

$$\alpha_{N-1}(S_{t_{N-1}}^{(N)}) = \frac{f\left((1+h_N)(S_{t_{N-1}}^{(N)})\right) - f\left((1+b_N)(S_{t_{N-1}}^{(N)}))\right)}{(h_N - b_N)(S_t^{(N)}{}_{N-1})}$$

$$\beta_{N-1}(S_{t_{N-1}}^{(N)}) = \frac{f\left((1+h_N)(S_{t_{N-1}}^{(N)})\right) - \alpha_{N-1}(S_{t_{N-1}}^{(N)})(S_{t_N}^{(N)})}{S_{t_N}^0}$$

$$\beta_{N-1}(S_{t_{N-1}}^{(N)}) = \frac{(1+h_N)f\left((1+b_N)(S_{t_{N-1}}^{(N)})\right) - \; (1+b_N)f\left((1+h_N)(S_{t_{N-1}}^{(N)})\right)}{(h_N - b_N)(S_{t_N}^0)}$$

9. Pour le cas précédent, on a

$$\alpha_{k-1}(S_{t_{k-1}}^{(N)})(S_{tk}^{(N)}) + \beta_{k-1}(S_{t_{k-1}}^{(N)})(S_{tk}^0) = v_k(S_{t_k}^{(N)})$$

On obtient un système d'équations

$$\begin{cases} \alpha_{k-1}(S_{t_{k-1}}^{(N)})(1+h_N)(S_{t_{k-1}}^{(N)}) + \beta_{k-1}(S_{t_{k-1}}^{(N)})(1+r_N)^k = v_k\left((1+h_N)S_{t_{k-1}}^{(N)}\right) \\ \alpha_{k-1}\left(S_{tk-1}^{(N)}\right)(1+b) \times S_{tk-1}^{(N)} + \beta_{k-1}\left(S_{tk-1}^{(N)}\right)(1+r_N)^k = v_k\left((1+b_N)S_{t_{k-1}}^{(N)}\right) \end{cases}$$

avec une solution

$$\alpha_{k-1}\left(S_{t_{k-1}}^{(N)}\right) = \frac{v_k\left((1+h_N)(S_{t_{k-1}}^{(N)})\right) - v_k\left((1+b)(S_{t_{k-1}}^{(N)})\right)}{(h-b)S_{t_{k-1}}^{(N)}}$$

$$\beta_{k-1}\left(S_{t_{k-1}}^{(N)}\right) = \frac{v_k\left((1+h)(S_{t_{k-1}}^{(N)})\right) - \alpha_{N-1}(S_{t_{k-1}}^{(N)})(S_{t_k}^{(N)})}{S_{t_k}^0}$$

$$\beta_{k-1}(S_{t_{k-1}}^{(N)}) = \frac{(1+h_N)v_k\left((1+b)(S_{t_{k-1}}^{(N)})\right) - (1+b_N)v_k\left((1+h)(S_{t_{k-1}}^{(N)})\right)}{(h-b)S_{t_k}^0}$$

10. On considère $N=2, s=100, r_N=0.03, h_N=0.05, b_N=-0.05, f(x)=\max(x-100,0)$.
La couverture à la date 0 est

$$\alpha_0(S_0^N) = 0.7961, \beta_0(S_0^N) = -73.4282$$

La couverture à la date 1 est

$$\alpha_1(S_1^N) = \begin{pmatrix} 0.9762 \\ 0. \end{pmatrix}, \beta_1(S_1^N) = \begin{pmatrix} -91.7853 \\ 0. \end{pmatrix}$$

3 Modèle de Black-Scholes

3.1 Le modèle

Le prix S^0 de l'actif sans risque satisfait l'équation différentielle suivante

$$dS_t^0 = rS_t^0 dt$$

avec $S_0^0 = 1$.
Le prix S de l'actif risqué vérifie l'équation différentielle stochastique suivante

$$dS_t = S_t\left(rdt + \sigma dB_t\right)$$

avec $S_0 = s$, σ et r deux constantes strictement positives. B est un processus stochastique appelé mouvement brownien. Alors on obtien

$$S_t^0 = e^{rt}$$

11. On utilise la formule d'Ito pour trouver la solution à cette équation différentielle stochastique.

$$dg(S_t) = g'(S_t)dS_t + \frac{|\sigma S_t|^2}{2} g''(S_t)dt$$

On applique la formule d'Ito à $ln(S_t)$.

$$d(ln(S_t)) = (ln(S_t))'d(S_t) + \frac{|\sigma S_t|^2}{2}(ln(S_t))''dt$$

$$d(ln(S_t)) = \frac{dS_t}{S_t} - \frac{\sigma^2}{2}dt$$

$$d(ln(S_t)) = rdt + \sigma dB_t - \frac{\sigma^2}{2}dt$$

$$S_t = s \times \exp(\sigma B_t + (r - \frac{\sigma^2}{2})t)$$

3.2 Le pricer par la méthode de Monte-Carlo

La formule du prix d'une option qui rapporte $f(S_T)$ est donnée par

$$p := E\left[e^{-rT} f(S_T)\right]$$

Soit $(\xi)_{1\leq i\leq n}$ une suite de v.a i.i.d de loi $\aleph(0,1)$. On définit

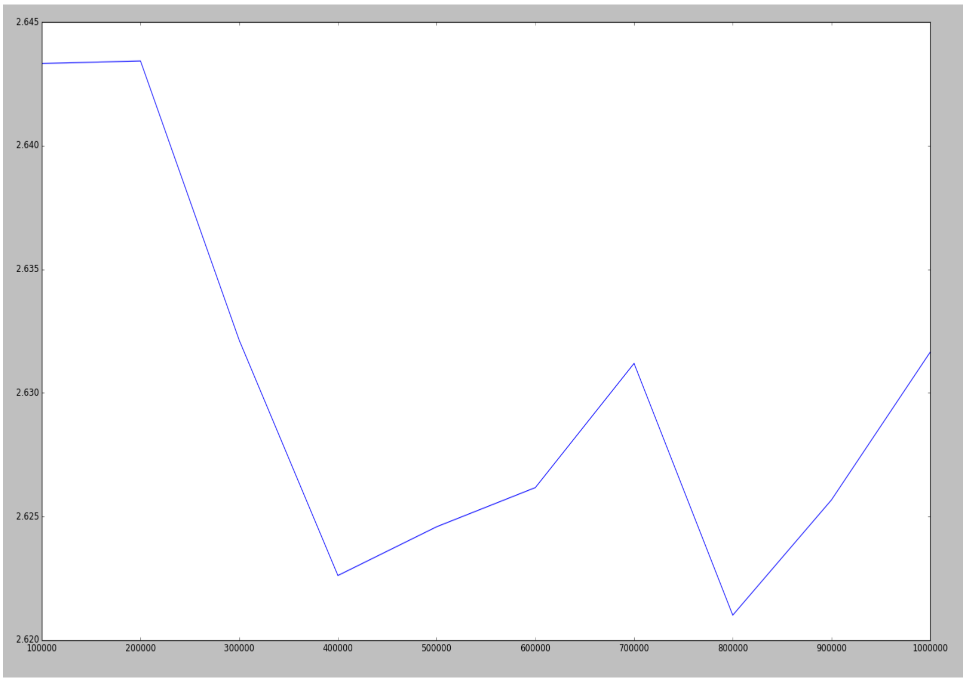
$$\hat{p}_{(n)} := \frac{1}{n}\sum_{i=1}^{n} e^{-rT} f\left(s \times exp\left((r - \frac{\sigma^2}{2})T + \sigma\sqrt{T}\xi_i\right)\right)$$

12. La fonction *price*3 prend en argument n, s, r, σ, T et f et renvoit $\hat{p}_{(n)}$.

Algorithm 3 fonction *price*3

function PRICE3$(N, r_N, h_n b_N, s, f)$
  soit $x = 0$
  $\xi_k \sim \mathcal{N}(0,\ 1)$
  for k in range $(1, N+1)$ **do**
    $x = x + \exp\left(-r_N T\right) \times f\left(s \times \exp\left((r_N - \frac{\sigma^2}{2})T + \sigma\sqrt{T}\xi_k\right)\right)$
  end for
  return $\{x/N\}$
end function

13. Le graphique pour le prix avec $r = 0.03, \sigma = 0.1, s = 100, T = 1, f(x) = \max(100-x, 0), n = 10^5 k$ pour $1 \leq k \leq 10$. On voit que le prix tourne autour de 2,6



14. D'après la loi forte de grande-nombre,

$$\frac{1}{n}\sum_{i=1}^{n} X_i \xrightarrow[n\to+\infty]{ps} E[X_1]$$

Alors, en utilisant cette loi nous allons montrer que

$$\hat{p}_{(n)} \xrightarrow[n\to+\infty]{ps} p$$

On a :

$$P\left(\left|\sqrt{n}\frac{\overline{X}-E[X]}{\sigma}\right| \leq 1,96\right) = 0.95$$

$$\left|\sqrt{n}\frac{\overline{X}-E[X]}{\sigma}\right| \leq 1.96$$

$$-1.96 \leq \sqrt{n}\frac{\overline{X}-E\left[X\right]}{\sigma} \leq 1,96$$

$$\overline{X} - 1,96\frac{\sigma}{\sqrt{N}} \leq E\left[X\right] \leq \overline{X} + 1,96\frac{\sigma}{\sqrt{N}}$$

On sait que l'amplitude de l'intervalle de confiance est inférieur à 10^{-2} : $2 \times 1,96\frac{\sigma}{\sqrt{N}} \leq 10^{-2}$
La valeur d'estimation de σ est :

$$\sigma \approx \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(X_i-\overline{X})^2}$$

On remplace $\overline{X} = \hat{p}, X = e^{-rT}f(S_T)$ On obtient :

$$N \geq \left(\frac{2\times 1.96}{10^{-2}}\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(X_i-\overline{X})^2}\right)^2$$

3.3 Le pricer par formule fermée

Le prix de l'option qui paye $\max(K - S_T, 0)$ en T est donné par la formule de Black-Scholes

$$p = -sF(-d) + Ke^{-rT}F(-d+\sigma\sqrt{T})$$

avec $d := \frac{1}{\sigma\sqrt{T}}[ln(\frac{s}{K}) + (r + \frac{\sigma^2}{2})T]$ et $F(x) = P(Y \leq x)$ où $Y \sim \mathcal{N}(0,\, 1)$.

15. La fonction put qui prend en argument s, r, σ, T, K et qui renvoit le prix de cette option.

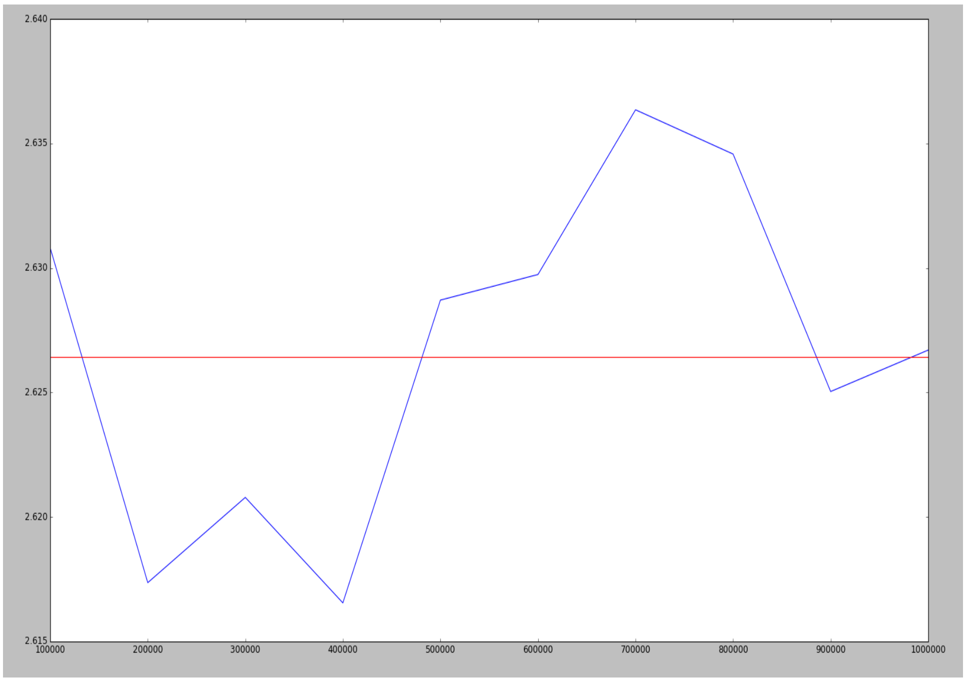
Algorithm 4 fonction *put*

1: **function** PUT(s, r, σ, T, K)
2: soit $d := \frac{1}{\sigma\sqrt{T}}[\ln(\frac{s}{K}) + (r + \frac{\sigma^2}{2})T]$
3: **function** F(X)
4: **return** $\{norm.cdf(x, 0, 1)\}$
5: **end function**
6: **return** $\{-sF(-d) + Ke^{-rT}F(-d+\sigma\sqrt{T})\}$
7: **end function**

16. On applique notre fonction avec $r = 0.04, \sigma = 0.1, s = 100, T = 1, K = 100$.

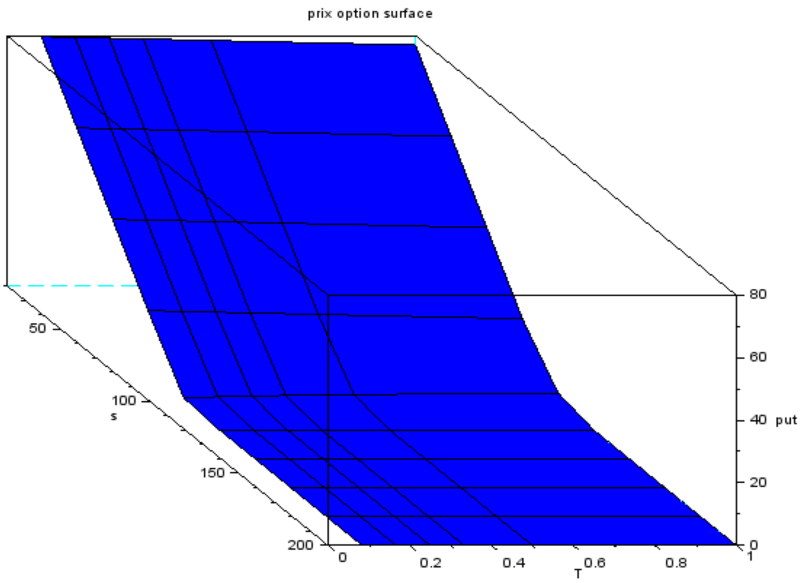
La réponse est 2.2574.

17. Le graphique du prix donné par la fonction *put* et $price3$.



On observe que la courbe de $price3$ est oscille autour d'une valeur de 2.6 ce qui est celle de l'asymptote horizontale. Le prix de $price3$ est en train de converger vers le prix renvoyé par put quand n tend vers l'infini.

18. Le graphic en 3D le prix de l'option de fonction *put* lorsque $r = 0.03, \sigma = 0.1, K = 100, s = 20k$ avec $1 \leq k \leq 10$ et $T \in \{\frac{1}{12}, \frac{1}{6}, \frac{1}{4}, \frac{1}{3}, \frac{1}{2}, 1\}$.



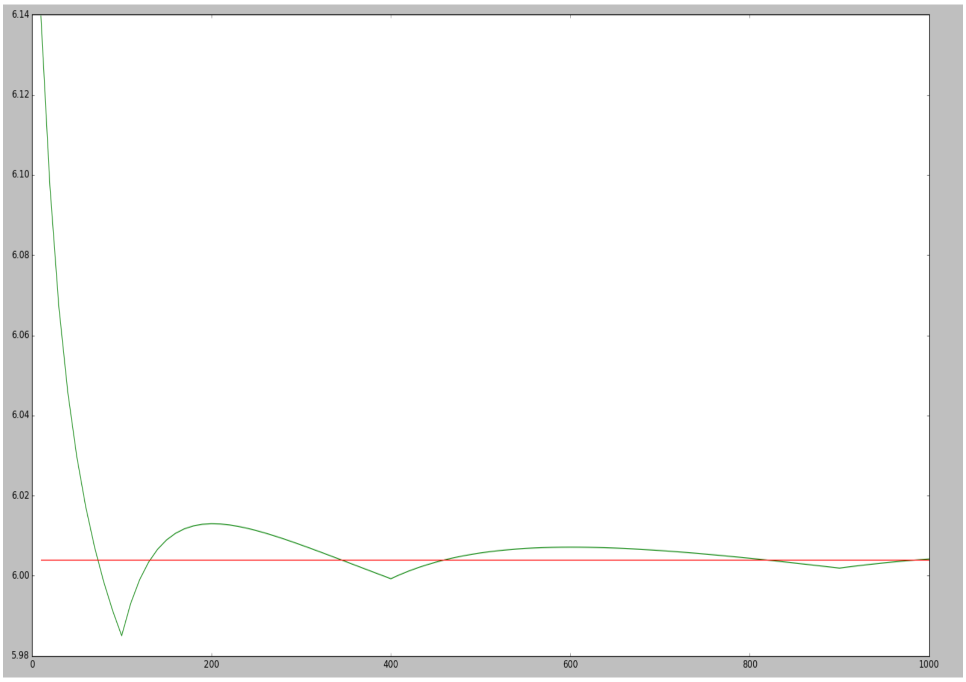
On observe que si on fixe s et que l'on fait varier t, t a un petit effet sur le prix $put()$, c'est à dire que le prix reste presque constante mais si on fixe T et que l'on varie s, le prix $put()$ devient décroissant rapidement et tend vers 0. Ainsi le prix de l'actif a un impact important sur le prix d'option.

4 Convergence des prix

On va s'intéresser à la convergence des prix donnés dans le cas du modèle de Cox-Ross-Rubinstein vers les prix donnés dans le cas du modèle de Black-Scholes.
On pose

$$r_N = \frac{rT}{N}, h_N = (1+r_N)exp\left(\sigma\sqrt{\frac{T}{N}}\right) - 1, b_N = (1+r_N)exp\left(-\sigma\sqrt{\frac{T}{N}}\right) - 1$$

19. Le graphique de price1 et put qui paye $\max(K - S_T, 0)$ pour $K = 100, s = 100, \sigma = 0.2, r = 0.04, T = 1$ avec $N = 10k$ pour $1 \leq k \leq 100$.



On observe que quand N tend vers infini, alors le price1 tend vers le prix de put.

5 EDP de Black-Scholes

Le prix du put $P(t, S_t)$ qui dépend du temps t et du prix d'actif risqué S_t à l'instant t vérifie une équation aux dérivées partiels (EDP) définie sur $[0, T] \times [0, L]$

$$\frac{\partial P}{\partial t} + rS\frac{\partial P}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 P}{\partial S^2} = rP$$

Les conditions sont :

$$P(T, s) = \max(K - s, 0), \forall s \in [0, L]$$
$$P(t, 0) = Ke^{r(t-T)}, \forall t \in [0, T]$$
$$P(t, L) = 0, \forall t \in [0, T]$$

On pose :

$$\triangle T = \frac{T}{N}, t_n = n\triangle T, \forall n \in \{0, ..., N\}$$
$$\triangle s = \frac{L}{M+1}, s_i = i\triangle s, \forall i \in \{0, ..., M+1\}$$

$P(t_n, s_i)$ est remplacé par P_i^n.
Posons $P_n \in \Re^M$ le vecteur colonne de composantes $(P_{n,1}, ..., P_{n,M})$.
Pour $\forall$(n,i)$\in$ $\{0, ..., N-1\} \times \{1, ..., M\}$, on approche les dérivées partiels intervenant dans l'EDP avec les 3 méthodes :

— différences finies explicites

$$\frac{\partial P}{\partial t}(t_n, s_i) \approx \frac{1}{\triangle T}(P(t_{n+1}, s_i) - P(t_n, s_i))$$
$$\frac{\partial P}{\partial S}(t_n, s_i) \approx \frac{1}{\triangle s}(P(t_n, s_i) - P(t_n, s_{i-1}))$$
$$\frac{\partial^2 P}{\partial S^2}(t_n, s_i) \approx \frac{1}{\triangle s^2}(P(t_n, S_{i+1}) - 2P(t_n, s_i) + P(t_n, s_{i-1}))$$

On va appliquer sur l'EDP.

$$\frac{1}{\triangle T}(P_{n+1}^i - P_n^i) + rS_i \frac{1}{\triangle s}(P_n^i - P_n^{i-1}) + \frac{1}{2}\sigma^2 S_i^2 \frac{1}{\triangle s^2}(P_n^{i+1} - 2P_n^i + P_n^{i-1}) = rP_n^i$$

$$\left(\frac{\sigma^2 S_i^2}{2\triangle s^2} - \frac{rS_i}{\triangle s}\right) P_n^{i-1} + \left(-\frac{1}{\triangle T} + \frac{rS_i}{\triangle s} - \frac{\sigma^2 S_i^2}{\triangle s^2} - r\right) P_n^i + \frac{\sigma^2 S_i^2}{2\triangle s^2} P_n^{i+1} = -\frac{1}{\triangle T} P_{n+1}^i$$
$$\left(\frac{\sigma^2 i^2}{2} - ri\right) P_n^{i-1} + \left(-\frac{1}{\triangle T} + r(i-1) - \sigma^2 i^2\right) P_n^i + \frac{\sigma^2 i^2}{2} P_n^{i+1} = -\frac{1}{\triangle T} P_{n+1}^i$$

Pour $i = 1$,

$$\left(\frac{\sigma^2}{2} - r\right) P_n^0 + \left(-\frac{1}{\triangle T} - \sigma^2\right) P_n^1 + \frac{\sigma^2}{2} P_n^2 = -\frac{1}{\triangle T} P_{n+1}^1$$

Pour $i = M$,

$$\left(\frac{M^2\sigma^2}{2} - rM\right) P_n^{M-1} + \left(-\frac{1}{\triangle T} - \sigma^2 M^2 + r(M-1)\right) P_n^M = -\frac{1}{\triangle T} P_{n+1}^M$$

Soit

$$a_i = \frac{\sigma^2 i^2}{2} - ri, b_i = -\frac{1}{\triangle T} + r(i-1) - \sigma^2 i^2, c_i = \frac{\sigma^2 i^2}{2}$$

L'équations matriciellle est

$$AP_n + B_n = -\frac{1}{\triangle T} P_{n+1}$$

où

$$A = \begin{pmatrix} b_1 & c_1 & 0 & \dots & \dots & 0 & 0 \\ a_2 & b_2 & c_2 & 0 & \dots & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & & \vdots \\ \vdots & \ddots & a_i & b_i & c_i & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \dots & \ddots & \ddots & \ddots & c_{M-1} \\ 0 & 0 & \dots & \dots & 0 & a_M & b_M \end{pmatrix}, B_n = \begin{pmatrix} \left(\frac{\sigma^2}{2} - r\right) Ke^{r(t_n - T)} \\ 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \\ 0 \end{pmatrix}$$

$$\Rightarrow P_n = A^{-1}\left(-\frac{1}{\triangle T} P_{n+1} - B_n\right)$$

— différences finies implicites

$$\frac{\partial P}{\partial t}(t_n, s_i) \approx \frac{1}{\triangle T}(P(t_n, s_i) - P(t_{n-1}, s_i))$$

$$\frac{\partial P}{\partial S}(t_n, s_i) \approx \frac{1}{\triangle s}(P(t_n, s_i) - P(t_n, s_{i-1}))$$

$$\frac{\partial^2 P}{\partial S^2}(t_n, s_i) \approx \frac{1}{\triangle s^2}(P(t_n, S_{i+1}) - 2P(t_n, s_i) + P(t_n, s_{i-1}))$$

Alors, on obtient :

$$\frac{1}{\triangle T}(P_n^i - P_{n-1}^i) + rS_i\frac{1}{\triangle s}(P_n^i - P_n^{i-1}) + \frac{1}{2}\sigma^2 S_i^2 \frac{1}{\triangle s^2}(P_n^{i+1} - 2P_n^i + P_n^{i-1}) = rP_n^i$$

$$\left(\frac{\sigma^2 S_i^2}{2\triangle s^2} - \frac{rS_i}{\triangle s}\right) P_n^{i-1} + \left(\frac{1}{\triangle T} + \frac{rS_i}{\triangle s} - \frac{\sigma^2 S_i^2}{\triangle s^2} - r\right) P_n^i + \frac{\sigma^2 S_i^2}{2\triangle s^2} P_n^{i+1} = \frac{1}{\triangle T} P_{n-1}^i$$

$$\left(\frac{\sigma^2 i^2}{2} - ri\right) P_n^{i-1} + \left(\frac{1}{\triangle T} + r(i-1) - \sigma^2 i^2\right) P_n^i + \frac{\sigma^2 i^2}{2} P_n^{i+1} = \frac{1}{\triangle T} P_{n-1}^i$$

Pour $i = 1$,

$$\left(\frac{\sigma^2}{2} - r\right) P_n^0 + \left(\frac{1}{\triangle T} - \sigma^2\right) P_n^1 + \frac{\sigma^2}{2} P_n^2 = \frac{1}{\triangle T} P_{n-1}^1$$

Pour $i = M$,

$$\left(\frac{\sigma^2 M^2}{2} - rM\right) P_n^{M-1} + \left(\frac{1}{\triangle T} + r(M-1) - \sigma^2 M^2\right) P_n^M = \frac{1}{\triangle T} P_{n-1}^M$$

Soit

$$a_i = \frac{\sigma^2 i^2}{2} - ri, b_i = \frac{1}{\triangle T} + r(i-1) - \sigma^2 i^2, c_i = \frac{\sigma^2 i^2}{2}$$

L'équation matriciellle est

$$AP_n + B_n = \frac{1}{\triangle T} P_{n-1}$$

où

$$A = \begin{pmatrix} b_1 & c_1 & 0 & \dots & \dots & 0 & 0 \\ a_2 & b_2 & c_2 & 0 & \dots & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & & \vdots \\ \vdots & \ddots & a_i & b_i & c_i & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & \dots & \ddots & \ddots & \ddots & c_{M-1} \\ 0 & 0 & \dots & \dots & 0 & a_M & b_M \end{pmatrix}, B_n = \begin{pmatrix} \left(\frac{\sigma^2}{2} - r\right) K e^{r(t_n - T)} \\ 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \\ 0 \end{pmatrix}$$

$$\Rightarrow P_{n-1} = \triangle T(AP_n - B_n)$$

— méthode de Crank-Nicholson

$$\frac{\partial P}{\partial t}(t_n, s_i) \approx \frac{1}{\triangle T}(P(t_{n+1}, s_i) - P(t_n, s_i))$$

$$\frac{\partial P}{\partial S}(t_n, s_i) \approx \frac{1}{2}\left(\frac{P(t_{n+1}, s_{i+1}) - P(t_{n+1}, s_{i-1})}{2\triangle s} + \frac{P(t_n, s_{i+1}) - P(t_n, s_{i-1})}{2\triangle s}\right)$$

$$\frac{\partial^2 P}{\partial S^2}(t_n, s_i) \approx \frac{1}{2\triangle s^2}((P(t_{n+1}, s_{i+1}) - 2P(t_{n+1}, s_i) + P(t_{n+1}, s_{i-1})) + (P(t_n, s_{i+1}) - 2P(t_n, s_i) + P(t_n, s_{i-1})))$$

Alors on obtient :

$$\frac{1}{\triangle T}(P_{n+1}^{i}-P_{n}^{i})+\frac{rS_i}{2}\left(\frac{P_{n+1}^{i+1}-P_{n+1}^{i-1}}{2\triangle s}+\frac{P_{n}^{i+1}-P_{n}^{i-1}}{2\triangle s}\right)+\frac{\sigma^2 S_i^2}{2\triangle s^2}((P_{n+1}^{i+1}-2P_{n+1}^{i})+P_{n+1}^{i-1})+(P_{n}^{i+1}-2P_{n}^{i}+P_{n}^{i-1}))$$

$$= rP_n^i$$

$$\left(-\frac{ri}{4}+\frac{\sigma^2 i^2}{4}\right)P_n^{i-1}-\left(\frac{1}{\triangle T}+r+\frac{\sigma^2 i^2}{2}\right)P_n^i+\left(\frac{ri}{4}+\frac{\sigma^2 i^2}{4}\right)P_n^{i+1}$$

$$=-\left(-\frac{ri}{4}+\frac{\sigma^2 i^2}{4}\right)P_{n+1}^{i-1}+\left(-\frac{1}{\triangle T}+\frac{\sigma^2 i^2}{2}\right)P_{n+1}^i-\left(\frac{ri}{4}+\frac{\sigma^2 i^2}{4}\right)P_{n+1}^{i+1}$$

Pour $i=1$,

$$\left(-\frac{r}{4}+\frac{\sigma^2}{4}\right)P_n^{0}-\left(\frac{1}{\triangle T}+r+\frac{\sigma^2}{2}\right)P_n^1+\left(\frac{r}{4}+\frac{\sigma^2}{4}\right)P_n^{2}$$

$$=-\left(-\frac{r}{4}+\frac{\sigma^2}{4}\right)P_{n+1}^{0}+\left(-\frac{1}{\triangle T}+\frac{\sigma^2}{2}\right)P_{n+1}^1-\left(\frac{r}{4}+\frac{\sigma^2}{4}\right)P_{n+1}^{2}$$

Pour $i=M$,

$$\left(-\frac{rM}{4}+\frac{\sigma^2 M^2}{4}\right)P_n^{M-1}-\left(\frac{1}{\triangle T}+r+\frac{\sigma^2 i^2}{2}\right)P_n^M$$

$$=-\left(-\frac{rM}{4}+\frac{\sigma^2 M^2}{4}\right)P_{n+1}^{M-1}+\left(-\frac{1}{\triangle T}+\frac{\sigma^2 M^2}{2}\right)P_{n+1}^M$$

Soit

$$a_i=-\frac{ri}{4}+\frac{\sigma^2 i^2}{4}, b_i=\frac{1}{\triangle T}+r+\frac{\sigma^2 i^2}{2}$$

$$c_i=\frac{ri}{4}+\frac{\sigma^2 i^2}{4}, d_i=-\frac{1}{\triangle T}+\frac{\sigma^2 i^2}{2}$$

L'équation matricielle est

$$u_n+AP_n=-u_{n+1}+BP_{n+1}$$

avec

$$A=\begin{pmatrix} b_1 & c_1 & 0 & \dots & \dots & 0 & 0\\ a_2 & b_2 & c_2 & 0 & \dots & 0 & 0\\ 0 & \ddots & \ddots & \ddots & \ddots & & \vdots\\ \vdots & \ddots & a_i & b_i & c_i & \ddots & \vdots\\ \vdots & & \ddots & \ddots & \ddots & \ddots & 0\\ 0 & 0 & \dots & \ddots & \ddots & \ddots & c_{M-1}\\ 0 & 0 & \dots & \dots & 0 & a_M & b_M \end{pmatrix}, B=\begin{pmatrix} d_1 & -c_1 & 0 & \dots & \dots & 0 & 0\\ -a_2 & d_2 & -c_2 & 0 & \dots & 0 & 0\\ 0 & \ddots & \ddots & \ddots & \ddots & & \vdots\\ \vdots & \ddots & -a_i & d_i & -c_i & \ddots & \vdots\\ \vdots & & \ddots & \ddots & \ddots & \ddots & 0\\ 0 & 0 & \dots & \ddots & \ddots & \ddots & -c_{M-1}\\ 0 & 0 & \dots & \dots & 0 & -a_M & d_M \end{pmatrix}$$

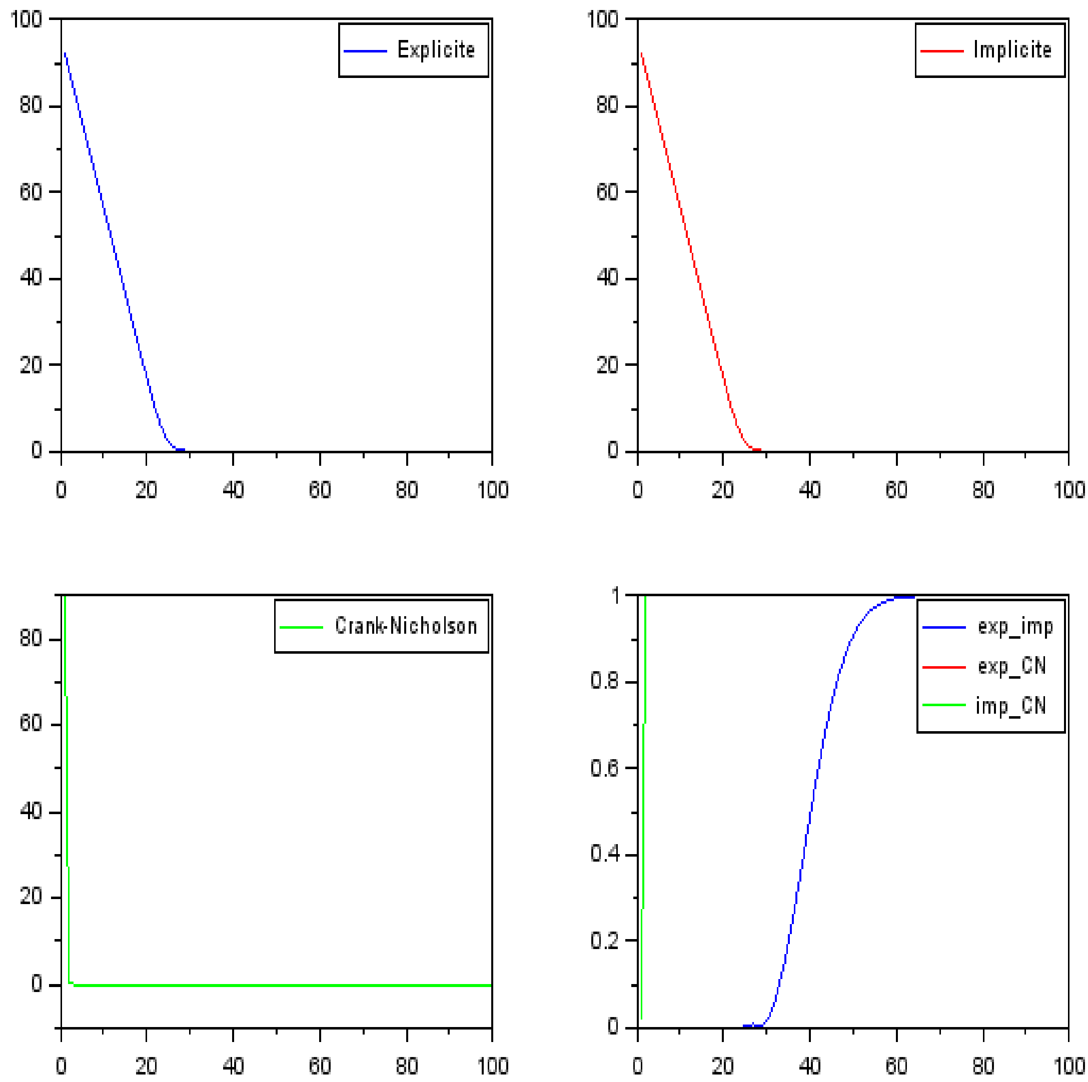
$$u_n=\begin{pmatrix} \left(\frac{\sigma^2}{4}-\frac{r}{4}\right)Ke^{r(t_n-T)}\\ 0\\ 0\\ \vdots\\ \vdots\\ 0\\ 0 \end{pmatrix}$$

$$\Rightarrow P_n=A^{-1}(-u_{n+1}-u_n+BP_{n+1})$$

Comme on connait $P_N=\max(K-s_i,0)_{1\le i\le M}$, on peut déterminer $P_{N-1},...,P_0$ itérativement.

En fait, on peut résoudre l'équation matricielle en utilisant la méthode de fatorisation LU. Comme dans Python, il y a la fonction de l'inversion, on utilise cette fonction au lieu de la factorisation LU. Quand on utilise la fonction inv(), on a le problème tant que la taille de matrix est très grande. Alors on fait la teste avec M = 100.

20. Les trois graphiques de la courbe P_0 et le graphique des erreurs relatives de la courbe avec $K = 100, r = 0.04, \sigma = 0.1, T = 1, L = 4K$ et $M = 100$ et $N = 100$.



On observe que la méthode de différence finie explicite et implicite a la même graphique. Le prix de put est décroissant.

6 Conclusion

Après la réalisation de ce projet, nous avons compris comment déterminer le prix d'une option de vente et une couverture en utilisant les deux modèles : Cox-Ross-Rubinstein de manière continu et Black-Scholes de manière discrète. D'autre part, nous avons compris comment résoudre l'EDP en utilisant les méthode de différence finie explicite et implicite et la méthode de Crank-Nicholson.