Projet Mathématique 2022

[redacted], [redacted], [redacted]

Avril 2022

1 Modèle de Cox-Ross-Rubinstein (binomial)

1) On sait que $E_{\mathbb{Q}}[T_1^{(N)}] = 1 + r_N$, de plus $q_N = \mathbb{Q}(T_1^{(N)} = 1 + h_N)$, notons de plus que la variable $T_1^{(N)}$ ne prend que les valeurs $1 + h_N$ et $1 + b_N$, en regroupant les informations on obtient :

$$\begin{aligned} E_{\mathbb{Q}}[T_1^{(N)}] &= (1 + h_N)\mathbb{Q}(T_1^{(N)} = 1 + h_N) + (1 + b_N)\mathbb{Q}(T_1^{(N)} = 1 + b_N) \\ &= (1 + h_N)q_N + (1 + b_N)(1 - q_N) \end{aligned}$$

Comme $E_{\mathbb{Q}}[T_1^{(N)}] = 1 + r_N, 1 + r_N = (1 + h_N)q_N + (1 + b_N)(1 - q_N)$ on obtient donc $\boxed{\mathrm{q}_N = \frac{r_N - b_N}{h_N - b_N}}$

2) On a

$$\begin{aligned} E_{\mathbb{Q}}[f(S_{t_N}^{(N)})] &= E_{\mathbb{Q}}[f(S_{t_{N-1}}^{(N)} T_N^{(N)})] \\ &= E_{\mathbb{Q}}[f(S_0^{(N)} T_1^{(N)} ... T_1^{(N)})] \\ &= E_{\mathbb{Q}}[f(sT_1^{(N)} ... T_N^{(N)})] \end{aligned}$$

On pose $X = sT_1^{(N)} ... T_1^{(N)}$, sachant que $E[f(X)] = \sum_{k \in support} f(k)\mathbb{P}(X = k)$ et que tous les $T_i^{(N)}$ sont i.i.d et ne prennent que la valeur $1 + h_N$ et $1 + b_N$, la variable X peut prendre la valeur $s(1 + h_N)^k(1 + b_N)^{N-k}$, $\binom{N}{k}$ fois le tout avec une probabilité de $q_N^k(1 - q_N)^{N-k}$ avec $k = 0, 1, ..., N$, d'où

$$E_{\mathbb{Q}}[f(S_{t_N}^{(N)})] = \sum_{k=1}^{N} \binom{N}{k} f(s(1 + h_N)^k(1 + b_N)^{N-k}) q_N^k (1 - q_N)^{N-k}$$

puis sachant que $Prix_{Bin}^N = \frac{1}{(1+r_N)^N} E_{\mathbb{Q}}[f(S_{t_N}^{(N)})]$, on obtient

$$\boxed{\mathrm{Prix}_{Bin}^N = \frac{1}{(1+r_N)^N} \sum_{k=1}^{N} \binom{N}{k} f(s(1 + h_N)^k(1 + b_N)^{N-k}) q_N^k (1 - q_N)^{N-k}}$$

3) On implémente la fonction $pricer_1$ de la façon suivante

Algorithm 1 algorithme pour pricer_1

Require: N, r_N , h_N , b_N , s, f
Ensure: $Prix_{Bin}^N$
  $q_N \leftarrow \frac{r_N - b_N}{h_N - b_N}$
  $sum \leftarrow 0$
  for k `from 0 to` N **do**
    $sum \leftarrow sum + \binom{N}{k} f(s(1+h_N)^k(1+b_N)^{N-k}) q_N^k (1-q_N)^{N-k}$
  end for
  $Prix_{Bin}^N \leftarrow \frac{sum}{(1+r_N)^N}$
  return $Prix_{Bin}^N$

Remarquons que l'on implémente aussi la fonction *binom* afin d'obtenir les coefficients binomiaux pour pouvoir utiliser cet algorithme, de même on implémente la fonction factorielle.

4) La valeur retournée est 26.616941360258558

5) On a pour $k = 0, 1, ..., N-1$

$$\begin{aligned} v_k(S_{t_k}^{(N)}) &= \frac{1}{1+r_N} E_{\mathbb{Q}}[v_{k+1}(S_{t_k+1}^{(N)}) | S_{t_k}^{(N)}] \\ &= \frac{1}{1+r_N} E_{\mathbb{Q}}[v_{k+1}(T_{k+1}^{(N)} S_{t_k}^{(N)}) | S_{t_k}^{(N)}] \\ &= \frac{1}{1+r_N} [v_{k+1}((1+h_N) S_{t_k}^{(N)}) q_N + v_{k+1}((1+b_N) S_{t_k}^{(N)})(1-q_N)] \qquad (*) \end{aligned}$$

(*) $T_{k+1}^{(N)}$ et $S_{t_k}^{(N)}$ sont indépendantes.

On pose $V_k^{'} = v_k(S_{t_k}^{(N)})$ le vecteur contenant toutes les valeurs possibles de $v_k(S_{t_k}^{(N)})$ en fonction des valeurs prises par $S_{t_k}^{(N)}$. Regardons $V_k^{'}$ pour les premières valeurs de k.

$$\begin{array}{cccc} V_0^{'} & V_1^{'} & V_2^{'} & V_3^{'} \\ v_0(s) & \begin{pmatrix} v_1(s(1+h_N)) \\ v_1(s(1+b_N)) \end{pmatrix} & \begin{pmatrix} v_2(s(1+h_N)^2) \\ v_2(s(1+h_N)(1+b_N)) \\ v_2(s(1+b_N)(1+h_N)) \\ v_2(s(1+b_N)^2) \end{pmatrix} & \begin{pmatrix} v_3(s(1+h_N)^3) \\ v_3(s(1+h_N)^2(1+b_N)) \\ v_3(s(1+h_N)(1+b_N)(1+h_N)) \\ v_3(s(1+h_N)(1+b_N)^2) \\ v_3(s(1+b_N)(1+h_N)^2) \\ v_3(s(1+b_N)(1+h_N)(1+b_N) \\ v_3(s(1+b_N)^2(1+h_N)) \\ v_3(s(1+b_N)^3 \end{pmatrix} \end{array}$$

Réécrivons ces quatre vecteurs autrement : posons $s_{10} = s(1 + h_N)$, $s_{11} = s(1 + b_N)$, $s_{20} = s(1 + h_N)^2$, $s_{21} = s(1 + h_N)(1 + b_N)$, $s_{22} = s(1 + b_N)^2$, $s_{30} = s(1 + h_N)^3$, $s_{31} = s(1 + h_N)^2(1 + b_N)$, $s_{32} = s(1 + b_N)^2(1 + h_N)$, et $s_{33} = s(1 + b_N)^3$, alors on obtient :

$$
\begin{array}{cccc}
V_0^{'} & V_1^{'} & V_2^{'} & V_3^{'} \\
v_0(s) & \begin{pmatrix} v_1(s_{10}) \\ v_1(s_{11}) \end{pmatrix} & \begin{pmatrix} v_2(s_{20}) \\ v_2(s_{21}) \\ v_2(s_{21}) \\ v_2(s_{22}) \end{pmatrix} & \begin{pmatrix} v_3(s_{30}) \\ v_3(s_{31}) \\ v_3(s_{31}) \\ v_3(s_{32}) \\ v_3(s_{31}) \\ v_3(s_{32}) \\ v_3(s_{32}) \\ v_3(s_{33}) \end{pmatrix}
\end{array}
$$

A partir de chaque $V_i^{'}$ on peut éliminer chaque doublon de façon à obtenir les vecteurs suivants

$$
\begin{array}{cccc}
V_0 & V_1 & V_2 & V_3 \\
v_0(s) & \begin{pmatrix} v_1(s_{10}) \\ v_1(s_{11}) \end{pmatrix} & \begin{pmatrix} v_2(s_{20}) \\ v_2(s_{21}) \\ v_2(s_{22}) \end{pmatrix} & \begin{pmatrix} v_3(s_{30}) \\ v_3(s_{31}) \\ v_3(s_{32}) \\ v_3(s_{33}) \end{pmatrix}
\end{array}
$$

En comparant avec les vecteurs précédents on peut déduire la formule récursive suivante :

$$\forall k \in [0; N-1], \forall i \in [0; k], V_k[i] = \frac{1}{1 + r_N}(q_N V_{k+1}[i] + (1 - q_N)V_{k+1}[i + 1])$$

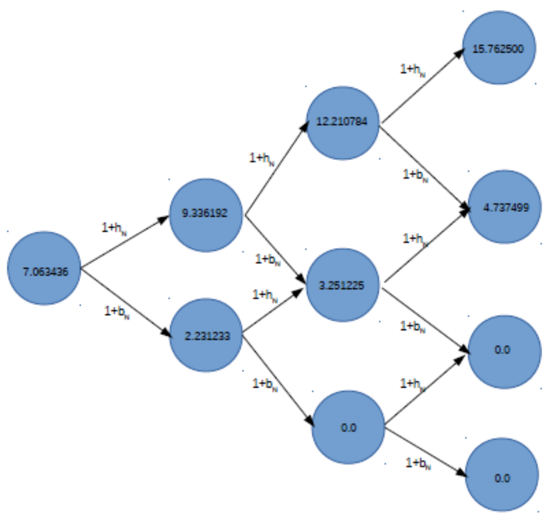
Et ce en partant de $V_N = \begin{pmatrix} f(s(1 + h_N)^N) \\ f(s(1 + h_N)^{N-1}(1 + b_N) \\ \ldots \\ f(s(1 + b_N)^N) \end{pmatrix}$

D'où l'algorithme suivant pour la fonction $pricer_2$:

Algorithm 2 algorithme pour pricer$_2$

Require: N, r_N , h_N , b_N , s, f
Ensure: $Prix^N_{Bin}$
  $q_N \leftarrow \frac{r_N - b_N}{h_N - b_N}$
  $V \leftarrow []$
  $V_N \leftarrow []$
  for i `from` 0 `to` N **do**
    `add` $f(s((1+h_N)^(N-i))((1+b_N)^i))$ `to` V_N
  end for
  `add` V_N `to` V
  for j `from` 0 `to` $N-1$ **do**
    $V_{N-1-j} \leftarrow []$
    $V^{lj} \leftarrow$ `last element of` V
    for k `from` 0 `to` $N-1-j$ **do**
      `add` $\frac{q_N V^{'}[k] + (1-q_N) V^{'}[k+1]}{(1+r_N)}$ `to` V_{N-1-j}
    end for
    `add` V_{N-1-j} `to` V
  end for
  $V^l \leftarrow$ `last element of` V
  $Prix^N_{Bin} \leftarrow V^l[0]$
  return $Prix^N_{Bin}$

6) On obtient un pricer égal à $\boxed{7.063436197239379}$.
L'arbre binaire correspondant est :



7) Avec $N = 9$, on obtient par exemple :
$pricer_1 = 11.041666344501927$
$pricer_2 = 11.041666344501921$
soit un écart de l'ordre de 10^{-15}

8) Le portefeuille doit vérifier

$$\alpha_{N-1}(S^{(N)}_{t_{N-1}})S^{(N)}_{t_N} + \beta_{N-1}(S^{(N)}_{t_{N-1}})(1+r_N)^N = f(S^{(N)}_{t_N})$$

avec $S^{(N)}_{t_N} = (1+h_N)S^{(N)}_{t_{N-1}}$ ou $(1+b_N)S^{(N)}_{t_{N-1}}$, l'équation devant être vérifiée dans les deux cas, d'où le système d'équations suivant :

$$\begin{cases} \alpha_{N-1}(S^{(N)}_{t_{N-1}})S^{(N)}_{t_{N-1}}(1+h_N) + \beta_{N-1}(S^{(N)}_{t_{N-1}})(1+r_N)^N = f(S^{(N)}_{t_{N-1}}(1+h_N)) & \text{(i)} \\ \alpha_{N-1}(S^{(N)}_{t_{N-1}})S^{(N)}_{t_{N-1}}(1+b_N) + \beta_{N-1}(S^{(N)}_{t_{N-1}})(1+r_N)^N = f(S^{(N)}_{t_{N-1}}(1+b_N)) & \text{(ii)} \end{cases}$$

Résolvons le :

$$\frac{(i)-(ii)}{S^{(N)}_{t_{N-1}}(h_N - b_N)} \Rightarrow \boxed{\alpha_{N-1}(S^{(N)}_{t_{N-1}}) = \frac{f(S^{(N)}_{t_{N-1}}(1+h_N)) - f(S^{(N)}_{t_{N-1}}(1+b_N))}{S^{(N)}_{t_{N-1}}(h_N - b_N)}}$$

Maintenant réinjectons le résultat obtenu dans (i) :

$$\begin{aligned}
\beta_{N-1}(S^{(N)}_{t_{N-1}}) &= \frac{f(S^{(N)}_{t_{N-1}}(1+h_N)) - \alpha_{N-1}(S^{(N)}_{t_{N-1}})S^{(N)}_{t_{N-1}}(1+h_N)}{(1+r_N)^N} \\
&= \frac{f(S^{(N)}_{t_{N-1}}(1+h_N)) - \frac{\cancel{S^{(N)}_{t_{N-1}}}(1+h_N)f(S^{(N)}_{t_{N-1}}(1+h_N)) - \cancel{S^{(N)}_{t_{N-1}}}(1+h_N)f(S^{(N)}_{t_{N-1}}(1+b_N))}{\cancel{S^{(N)}_{t_{N-1}}}(h_N - b_N)}}{(1+r_N)^N} \\
&= \frac{\cancel{f((1+h_N)S^{(N)}_{t_{N-1}})h_N} - f((1+h_N)S^{(N)}_{t_{N-1}})b_N - f((1+h_N)S^{(N)}_{t_{N-1}}) - \cancel{f((1+h_N)S^{(N)}_{t_{N-1}})h_N}}{(1+r_N)^N(h_N - b_N)} \\
&\qquad + \frac{f((1+b_N)S^{(N)}_{t_{N-1}}) + f((1+b_N)S^{(N)}_{t_{N-1}})b_N}{(1+r_N)^N(h_N - b_N)}
\end{aligned}$$

$$\boxed{\beta_{N-1}(S^{(N)}_{t_{N-1}}) = \frac{f((1+b_N)S^{(N)}_{t_{N-1}})(1+h_N) - f((1+h_N)S^{(N)}_{t_{N-1}})(1+b_N)}{(h_N - b_N)(1+r_N)^N}}$$

9) Le portefeuille doit vérifier

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

avec $S_{t_k}^{(N)} = (1+h_N)S_{t_{k-1}}^{(N)}$ ou $(1+b_N)S_{t_{k-1}}^{(N)}$, l'équation devant être vérifiée dans les deux cas, d'où le système d'équations suivant :

$$\begin{cases} \alpha_{k-1}(S_{t_{k-1}}^{(N)})S_{t_{k-1}}^{(N)}(1+h_N) + \beta_{k-1}(S_{t_{k-1}}^{(N)})(1+r_N)^k = v_k(S_{t_{k-1}}^{(N)}(1+h_N)) & \text{(i)} \\ \alpha_{k-1}(S_{t_{k-1}}^{(N)})S_{t_{k-1}}^{(N)}(1+b_N) + \beta_{k-1}(S_{t_{k-1}}^{(N)})(1+r_N)^k = v_k(S_{t_{k-1}}^{(N)}(1+b_N)) & \text{(ii)} \end{cases}$$

Remarquons qu'il s'agit de faire le même calcul qu'à la question précédente (les indices en N sont remplacés par k, de même f est remplacée par v_k). En faisant donc $\frac{(i)-(ii)}{S_{t_{k-1}}^{(N)}(h_N-b_N)}$ puis en réinjectant le résultat obtenu dans (ii) on obtient :

$$\boxed{\begin{cases} \alpha_{k-1}(S_{t_{k-1}}^{(N)}) = \frac{v_k(S_{t_{k-1}}^{(N)}(1+h_N))-v_k(S_{t_{k-1}}^{(N)}(1+b_N))}{S_{t_{k-1}}^{(N)}(h_N-b_N)} \\ \beta_{k-1}(S_{t_{k-1}}^{(N)}) = \frac{v_k((1+b_N)S_{t_{k-1}}^{(N)})(1+h_N)-v_k((1+h_N)S_{t_{k-1}}^{(N)})(1+b_N)}{(h_N-b_N)(1+r_N)^k} \end{cases}}$$

10) Commençons à la date 0 :

$$\begin{cases} \alpha_0 = \frac{v_1((1+h_N)s)-v_1((1+b_N)s)}{s(h_N-b_N)} \\ \beta_0 = \frac{v_1((1+b_N)s)(1+h_N)-v_1((1+h_N)s)(1+b_N)}{(h_N-b_N)(1+r_N)} \end{cases}$$

Or,

$$v_k(S_{t_k}^{(N)}) = \frac{1}{1+r_N}[v_{k+1}((1+h_N)S_{t_k}^{(N)})q_N + v_{k+1}((1+b_N)S_{t_k}^{(N)})(1-q_N)], \qquad k = 0,1,...,N-1$$

donc

$$v_1(S_{t_1}^{(N)}) = \frac{1}{1+r_N}[v_2((1+h_N)S_{t_1}^{(N)})q_N + v_2((1+b_N)S_{t_1}^{(N)})(1-q_N)]$$

puis comme $v_2(S_{t_2}^{(2)}) = f(S_{t_2}^{(2)})$ on obtient :

$$\begin{cases} v_1(s(1+h_N)) = \frac{1}{1+r_N}[f(s(1+h_N)^2)q_N + f(s(1+b_N)(1+h_N))(1-q_N)] \\ v_1(s(1+b_N)) = \frac{1}{1+r_N}[f(s(1+h_N)(1+b_N)q_N + f(s(1+b_N)^2)(1-q_N)] \end{cases}$$

Ce qui permet de trouver $\boxed{\alpha_0 = 0.7961165048543688, \beta_0 = -73.42822132151944}$.

Ensuite on a pour la date 1 :

$$\begin{cases} \alpha_1 = \frac{f((1+h_N)S_{t_1}^{(2)})-f((1+b_N)S_{t_1}^{(2)})}{S_{t_1}^{(2)}(h_N-b_N)} \\ \beta_1 = \frac{f((1+b_N)S_{t_1}^{(2)})(1+h_N)-f((1+h_N)S_{t_1}^{(2)})(1+b_N)}{(h_N-b_N)(1+r_N)^2} \end{cases}$$

Avec $S_{t_1}^{(2)} = s(1+h_N)$ ou $s(1+b_N)$.

En distinguant les deux cas on obtient donc :

Si $S_{t_1}^{(2)} = s(1+h_N)$, $\boxed{\alpha_1 = 0.9761904761904762, \beta_1 = -91.78527665189932}$.

Si $S_{t_1}^{(2)} = s(1+b_N)$, $\boxed{\alpha_1 = 0, \beta_1 = 0}$.

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

En considérant g = ln , on a :

$d(ln(S_t)) = \frac{dS_t}{S_t} - \frac{\sigma {S_t}^2}{2}\frac{1}{S_t^2}dt$

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

Or, $dS_t = S_t(rdt + \sigma^2 dB_t)$

D'où $dln(S_t) = rdt + \sigma dB_t - \frac{\sigma^2 dt}{2}$

En remarquant que $dln(S_t) = \frac{dS_t}{S_t}$, on obtient :

$\frac{dS_t}{dt} = S_t(r - \frac{\sigma^2}{2} + \frac{\sigma dB_t}{dt})$

On résout donc une EDO d'ordre 1 avec $S_0 = s$:

$\boxed{S_t(t) = se^{(r-\sigma^2)t+\sigma B_t}}$

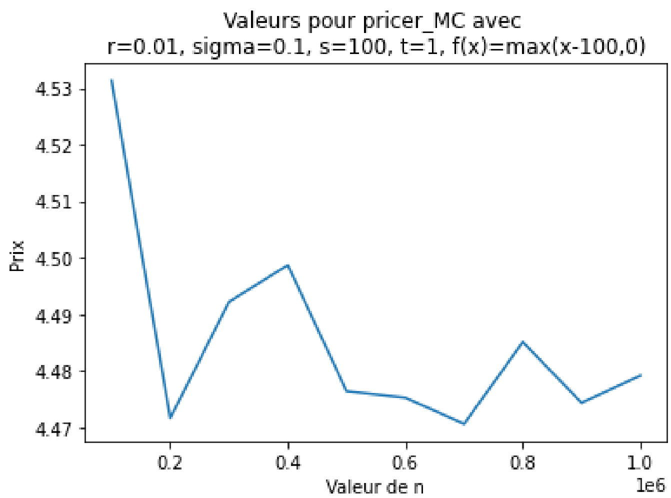
12) On implémente la fonction pricer_MC de la façon suivante

Algorithm 3 Algorithme pour pricer_MC

Require: n, s , r , σ , T, f
Ensure: $Prix_{MC}^{(n)}$
  $(\xi_i)_{1\leq i\leq n} \sim \mathcal{N}(0,1)$
  $sum \leftarrow 0$
  for k `from` 0 `to` $n-1$ **do**
    $sum \leftarrow sum + f(s * exp((r - \frac{\sigma^2}{2})T + \sigma\sqrt{T}\xi_i)$
  end for
  return $e^{-rT} * \frac{sum}{n}$

---

13) Avec $r = 0.01$, $\sigma = 0.1$, $s = 100$, $T = 1$, $f(x) = max(x - 100, 0)$, et $n = 10^5 k$ avec $1 \leq k \leq 10$, on obtient le graphique suivant pour pricer_MC :



15) On implémente la fonction put_BS de la façon suivante

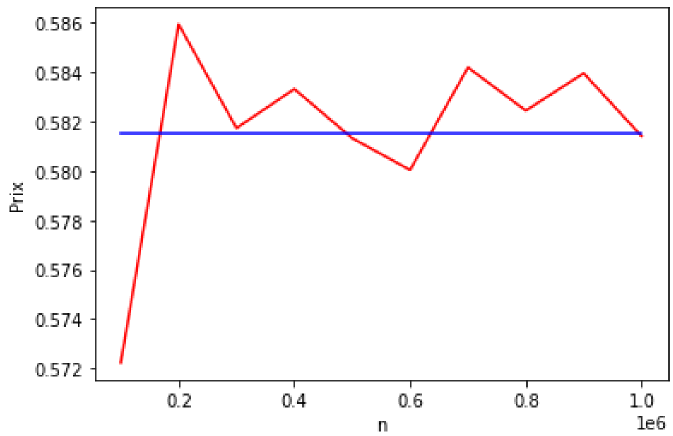
Algorithm 4 Algorithme pour put_BS

Require: s , r , σ , T, K
Ensure: $Prix_{BS}$
  $d1 \leftarrow \frac{1}{\sigma\sqrt{T}}[log(\frac{s}{K}) + (r + \frac{\sigma^2}{2})T]$
  $d2 \leftarrow d1 - \sigma\sqrt{T}$
  $Y \sim \mathcal{N}(0,1)$
  return $-s\mathbb{P}(Y \leq -d1) + Ke^{-rT}\mathbb{P}(Y \leq -d2)$

---

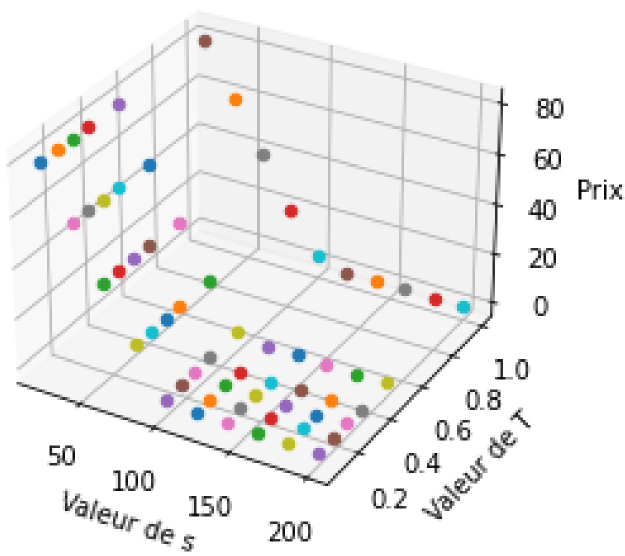
16) Avec $r = 0.01$, $\sigma = 0.1$, $s = 100$, $T = 1$ et $K = 90$, on obtient :
$put\_BS = 0.5815000751362422$

17) Avec $r = 0.01$, $\sigma = 0.1$, $s = 100$, $T = 1$, $f(x) = max(90 - x, 0)$, et $n = 10^5 k$ avec $1 \leq k \leq 10$, on obtient le graphique suivant, avec pricer_MC en rouge et put_BS en bleu :



On remarque que les valeurs pour de pricer_MC sont proches de celle de put_BS (qui est constante).

18) On trace un graphique en 3 dimensions représentant le prix de l'option avec $r = 0.01$, $\sigma = 0.1$, $K = 100$, $s = 20k$ 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 remarque que la valeur de T fait très peu varier le prix et que le prix diminue lorsque s augmente entre 0 et 100, mais rest e stable pour des valeurs de s entre 100 et 200.

20)

$$\frac{\partial P}{\partial t}(t,x) = \frac{1}{2}\sigma^2\frac{\partial^2 P}{\partial x^2}(t,x) + (r - \frac{\sigma^2}{2})\frac{\partial P}{\partial t}(t,x) - rp(t,x)$$

On pose

$$P(t_m, x_j) = P_{m,j}$$

$$\begin{pmatrix} max(Ke^{-r_{min}}, 0) \\ max(Ke^{-x_{min}-R}, 0) \\ \cdots \\ max(Ke^{-x_{min}-(N-2)h}, 0) \end{pmatrix}$$

On pose M$=1000$
$\Delta T = \frac{T}{M}$
h $=$ (x$_{max} - x_{min})/N$
Euler implicite :

$$P_{m+1,j} = P_{m,j} + \Delta T(\frac{\sigma^2}{2h^2}(P_{m+1,j+1} - 2P_{m+1,j} + P_{m+1,j-1}) + \frac{(r - \frac{\sigma^2}{2})(P_{m+1,j+1} - P_{m+1,j+1})}{2h} - rP_{m+1,j})$$

$$P_{m+1,j} = P_{m,j} + (\frac{\Delta T\sigma^2}{2h^2} - (r - \frac{\sigma^2}{2})\frac{\Delta T}{2h})P_{m+1,j-1}$$

$$-(r\Delta T + \frac{\Delta T\sigma^2}{h^2}P_{m+1,j-1} + (\frac{\Delta T\sigma^2}{2h^2} + (r - \frac{\sigma^2}{2})\frac{\Delta T}{2h})P_{m+1,j+1}$$

$$d = e^{x_{min} - 2r\Delta T} - e^{x_{min} + r\Delta}$$

$$a = 1 + r\Delta T + \frac{\sigma^2\Delta T}{h^2}$$

$$c = -\frac{\sigma^2\Delta T}{2h^2} + \frac{(r - \frac{\sigma^2}{2})\Delta T}{2h}$$

$$b = -\frac{\sigma^2\Delta T}{2h^2} - \frac{(r - \frac{\sigma^2}{2})\Delta T}{2h}$$

B$=\begin{pmatrix} e^{r\Delta T} & 0 & \cdots & & & 0 \\ c & a & b & 0 & \cdots & 0 \\ 0 & c & a & b & 0 & \cdots 0 \\ \cdots & & & & & \\ 0 & \cdots & c & & a & b \\ 0 & \cdots & & & 0 & 1 \end{pmatrix}$

$$C = \begin{pmatrix} d \\ 0 \\ \dots \\ 0 \end{pmatrix}$$

Alors

$$P_{m+1} = AP_m + C$$

Donc appliquer successivement la formule pour m=0,....,M-2

Cranck Nicholson :

$$P_{m+1,j} = P_{m,j} + \frac{1}{2}(\frac{\Delta T\sigma^2}{2h^2} - \frac{1}{2}(r - \frac{\sigma^2}{2})\frac{\Delta T}{2h})P_{m+1,j-1} - (r\Delta T + \frac{\Delta T\sigma^2}{h^2}P_{m+1,j-1}$$

$$+\frac{1}{2}(\frac{\Delta T\sigma^2}{2h^2} + (r - \frac{\sigma^2}{2})\frac{\Delta T}{2h})P_{m+1,j+1}(1 - r\Delta T - \frac{\sigma^2\Delta T}{h^2})P_{m,j}$$

$$+\frac{1}{2}(\frac{\sigma^2\Delta T}{h^2} - \frac{(r - \frac{\sigma^2}{2})\Delta T}{2h})P_{m,j+1} + \frac{1}{2}(\frac{\sigma^2\Delta T}{h^2} - \frac{(r - \frac{\sigma^2}{2})\Delta T}{2h})P_{m,j-1}$$

On pose

$d = e^{x_{min} - r\Delta T} - e^{x_{min}}$

$a = \frac{-(\frac{\Delta T\sigma^2}{2h^2} - (r - \frac{\sigma^2}{2})\frac{\Delta T}{2h})}{2}$

$b = 1 + \frac{r\Delta T + \frac{\Delta T\sigma^2}{h^2}}{2}$

$c = \frac{-(\frac{\Delta T\sigma^2}{2h^2} + (r - \frac{\sigma^2}{2})\frac{\Delta T}{2h})}{2}$

$a' = -a$

$b' = 1 - \frac{r\Delta T}{2} - \frac{\sigma^2\Delta T}{h^2}$

$c' = c$

$$A = \begin{pmatrix} 1 & 0 & \cdots & & & 0 \\ a & b & c & 0 & \cdots & 0 \\ 0 & a & b & c & 0 & \cdots 0 \\ \cdots & & & & & \\ 0 & \cdots & a & & b & c \\ 0 & \cdots & & & 0 & 1 \end{pmatrix}$$

$$\text{B}= \begin{pmatrix} e^{-r\Delta T} & 0 & \cdots & & & 0 \\ a' & b' & c' & 0 & \cdots & 0 \\ 0 & a' & b' & c' & 0 & \cdots 0 \\ \cdots & & & & & \\ 0 & \cdots & a' & & b' & c' \\ 0 & \cdots & & & 0 & 1 \end{pmatrix}$$

$$C = \begin{pmatrix} d \\ 0 \\ \cdots \\ 0 \end{pmatrix}$$

Alors

$$P_{m+1} = AP_m + C$$

Donc on applique successivement la formule pour m=0,....,M-2

Euler explicite

$$P_{m+1,j} = P_{m,j}+\Delta T(\frac{\sigma^2}{2h^2}(P_{m,j+1}-2P_{m,j}+P_{m,j-1})+(r-\frac{\sigma^2}{2})\frac{(P_{m,j+1}-P_{m,j-1})}{2h}-r*P_{m,j})$$

$$P_{m+1,j} = (1-r\Delta T-\frac{\sigma^2\Delta T}{h^2})P_{m,j}+(\frac{\sigma^2\Delta T}{h^2}-\frac{(r-\frac{\sigma^2}{2})\Delta T}{2h})P_{m,j+1}+(\frac{\sigma^2\Delta T}{h^2}-\frac{(r-\frac{\sigma^2}{2})\Delta T}{2h})P_{m,j-1}$$

$$d = e^{x_{min}-r\Delta T} - e^{x_{min}}$$

$$a = 1 - r\Delta T - \frac{\sigma^2\Delta T}{h^2}$$

$$b = \frac{\sigma^2\Delta T}{h^2} - \frac{(r-\frac{\sigma^2}{2})\Delta T}{2h}$$

$$c = \frac{\sigma^2\Delta T}{h^2} + \frac{(r-\frac{\sigma^2}{2})\Delta T}{2h}$$

$$A = \begin{pmatrix} e^{-r\Delta T} & 0 & \cdots & & & 0 \\ b & a & c & 0 & \cdots & 0 \\ 0 & b & a & c & 0 & \cdots 0 \\ \cdots & & & & & \\ 0 & \cdots & b & & a & c \\ 0 & \cdots & & & & 0 \end{pmatrix}$$

$$C = \begin{pmatrix} d \\ 0 \\ \cdots \\ 0 \end{pmatrix}$$

Alors

$$P_{m+1} = AP_m + C$$

Donc appliquer successivement la formule pour m=0,....,M-2