1) Soit $\mathbb{Q}$ une probabilité telle que sous cette probabilité on a $E_{\mathbb{Q}}[T_1^{(N)}] = 1 + r_N$. Ainsi : $E_{\mathbb{Q}}[T_1^{(N)}] = 1 + r_N = \sum_{x \in \{1+h_N, 1+b_N\}} x\mathbb{Q}(T_1^{(N)} = x) = (1 + h_N)q_N + (1 + b_N)(1 - q_N)$

Ainsi, on obtient :

$$\boxed{q_N = \frac{r_N - b_N}{h_N - b_N}}$$

On remarque que l'on a bien $q_N \in [0, 1]$.

2) Déterminons le prix de l'option $p_{(N)}$ de l'option qui paye $f(S_{t_N}^{(N)})$.

Pour $i \in \{1, .., N\}$, posons $X_i = \mathbb{1}_{T_i^{(N)} = 1+h_N}$. Il est facile de montrer que X_i suit une loi de Bernoulli de paramètre q_N. Posons $Y_N = \sum_{1 \leq i \leq N} X_i$. Les variables aléatoires $(T_i^{(N)})_{1 \leq i \leq N}$ étant indépendantes et identiquement distribuées, il en est de même pour les variables aléatoires $(X_i^{(N)})_{1 \leq i \leq N}$. On reconnait ainsi que Y_N est une loi binomiale de paramètres N et q_N. On remarque enfin que $P_{\mathbb{Q}}(S_{t_N}^{(N)} = s(1+h_N)^k(1+b_N)^{N-k}) = P_{\mathbb{Q}}(Y_i = k) = C_N^k q_N^k (1 - q_N)^{N-k}$. Ainsi, par le théorème de transfert :
$p_{(N)} = \frac{1}{(1+r_N)^N} \mathbb{E}_{\mathbb{Q}}[f(S_{t_N}^{(N)})] = \frac{1}{(1+r_N)^N} \sum_{k=0}^{N} f(s(1 + h_N)^k(1 + b_N)^{N-k}) P_{\mathbb{Q}}(S_{t_N}^{(N)} = s(1 + h_N)^k(1 + b_N)^{N-k})$

Ce qui donne :

$$\boxed{p_{(N)} = \frac{1}{(1 + r_N)^N} = \sum_{k=0}^{N} f(s(1 + h_N)^k(1 + b_N)^{N-k}) C_N^k q_N^k (1 - q_N)^{N-k}}$$

3)

Algorithm 1 price1(N, r_N, h_N, b_N, s, f)

Ensure : $res \times sum = p_{(N)}$

$q_N \leftarrow \frac{r_N - b_N}{h_N - b_N}$
$res \leftarrow \frac{1}{(1+r_N)^N}$
$sum \leftarrow 0$
for k in 0 :N **do**
  $sum \leftarrow sum + f(s(1 + h_N)^k(1 + b_N)^{(N-k)}) C_N^k q_N^k (1 - q_N)^{(N-k)}$
end for

---

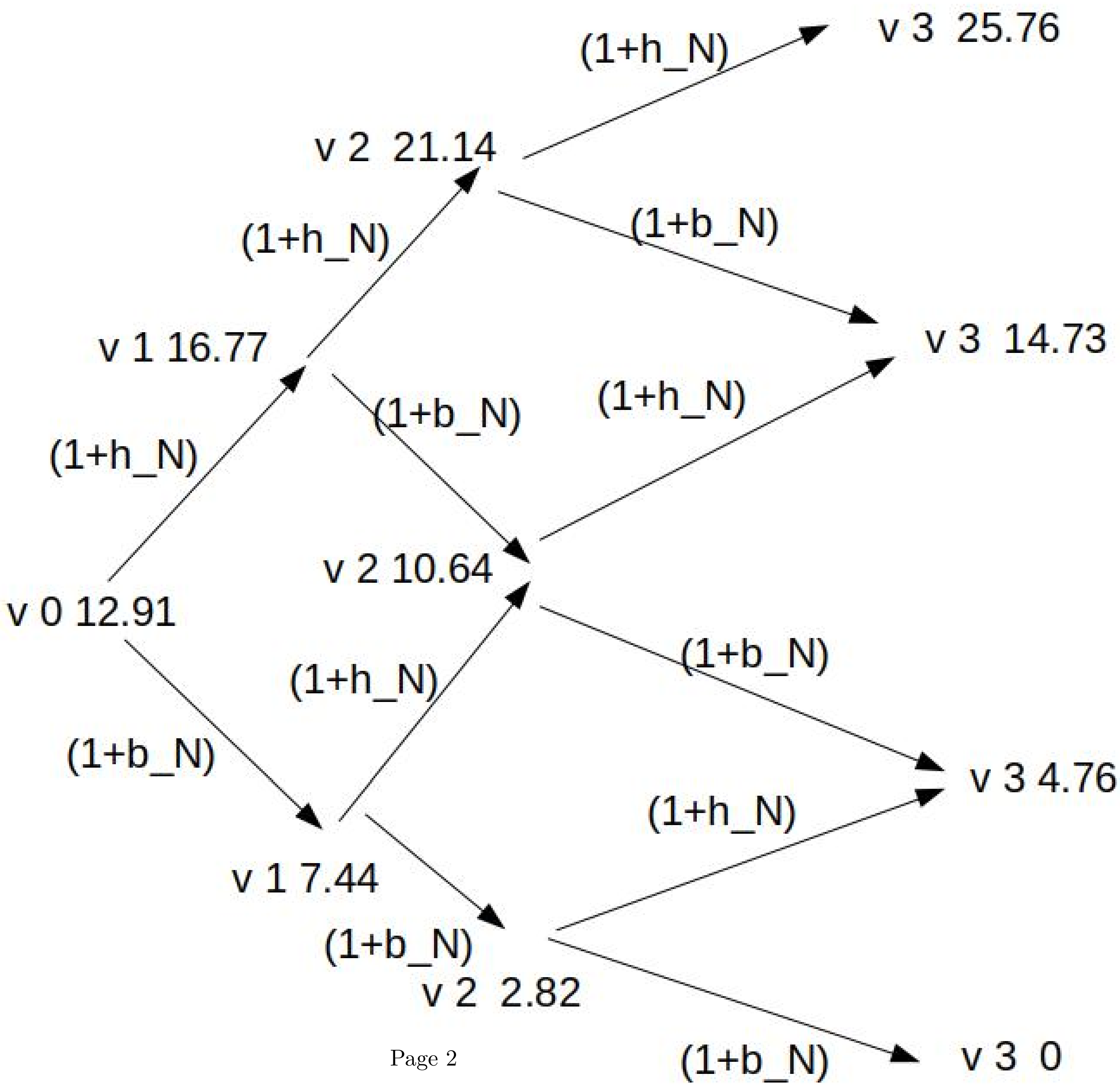
4) Voir code python.

5)On sait que $\mathbb{E}_{\mathbb{Q}}[v_{k+1}(S_{t_{k+1}}^{(N)}) | S_{t_k}^{(N)}] = v_{k+1}((1+h_N)S_{t_k}^{(N)})q_N + v_{k+1}((1+b_N)S_{t_k}^{(N)})(1-q_N)$. Ainsi, on appelle la fonction price2 avec $k = 0$:

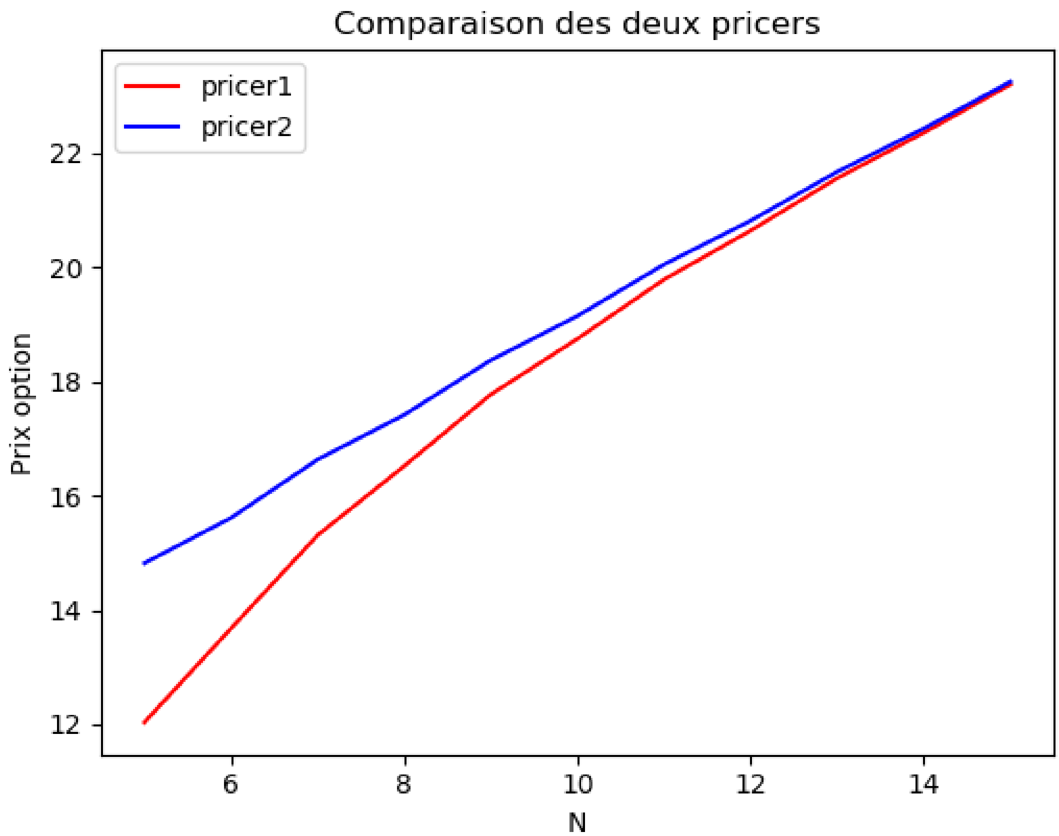
6)

Algorithm 2 price2$(N, r_N, h_N, b_N, s, f, k)$

Ensure : res $= p_{(N)}$
  if $k = N$ **then**
    $res \leftarrow f(s)$
  else
    $q_N \leftarrow \frac{r_N - b_N}{h_N - b_N}$
    $esp \leftarrow price2(N, r_N, h_N, b_N, s(1+h_N), f, k+1)q_N + price2(N, r_N, h_N, b_N, s(1+b_N), f, k+1)(1-q_N)$
    $res \leftarrow \frac{esp}{1+r_N}$
  end if



7)



8) Le système à la date t_N vaut :

$$\alpha_{N-1}(S^{(N)}_{t_{N-1}})(1+h_N)S^{(N)}_{t_{N-1}} + \beta_{N-1}(S^{(N)}_{t_{N-1}})S^0_{t_N} = f((1+h_N)S^{(N)}_{t_{N-1}}) \qquad [1]$$

$$\alpha_{N-1}(S^{(N)}_{t_{N-1}})(1+b_N)S^{(N)}_{t_{N-1}} + \beta_{N-1}(S^{(N)}_{t_{N-1}})S^0_{t_N} = f((1+b_N)S^{(N)}_{t_{N-1}}) \qquad [2]$$

$[1]-[2]$ donne :

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

$(1+h_N)[1]-(1+b_N)[2]$ donne :

$$\beta_{N-1}(S^{(N)}_{t_{N-1}}) = \frac{1}{S^0_{t_N}(h_N-b_N)}(f((1+b_N)S^{(N)}_{t_{N-1}})(1+h_N) - f((1+h_N)S^{(N)}_{t_{N-1}})(1+b_N))$$

9) Le système pour les autres dates t_k vaut :

$$\alpha_{k-1}(S^{(N)}_{t_{k-1}})(1+h_N)S_{t_{k-1}} + \beta_{k-1}(S^{(N)}_{t_{k-1}})S^0_{t_k} = v_k((1+h_N)S^{(N)}_{t_{k-1}}) \qquad [1]$$

$$\alpha_{k-1}(S^{(N)}_{t_{k-1}})(1+b_N)S_{t_{k-1}} + \beta_{k-1}(S^{(N)}_{t_{k-1}})S^0_{t_k} = v_k((1+b_N)S^{(N)}_{t_{k-1}}) \qquad [2]$$

Ce qui donne, par un raisonnement analogue :

$$\alpha_{k-1}(S^{(N)}_{t_{k-1}}) = \frac{v_k((1+h_N)S^{(N)}_{t_{k-1}}) - v_k((1+b_N)S^{(N)}_{t_{k-1}})}{(h_N-b_N)S^{(N)}_{t_{k-1}}}$$

$$\beta_{k-1}(S^{(N)}_{t_{k-1}}) = \frac{1}{S^0_{t_k}(h_N-b_N)}(v_k((1+b_N)S^{(N)}_{t_{k-1}}(1+h_N) - v_k((1+h_N)S^{(N)}_{t_{k-1}}(1+b_N))$$

10) Voir code pyton.

11) En appliquant la formule d'Ito à $ln(S_t)$, on obtient :

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

$$\iff \quad \mathrm{dln(S}_t) = (rdt + \sigma dB_t) - \frac{\sigma^2}{2}dt \qquad \text{car} \qquad dS_t = S_t(rdt + \sigma dB_t)$$

$$\iff \quad \mathrm{dln(S}_t) = (r - \frac{\sigma^2}{2})dt + \sigma dB_t$$

$$\iff \quad \mathrm{lnS}_t - lns = (r - \frac{\sigma^2}{2})t + \sigma B_t \qquad \text{car } B_0 = 0$$

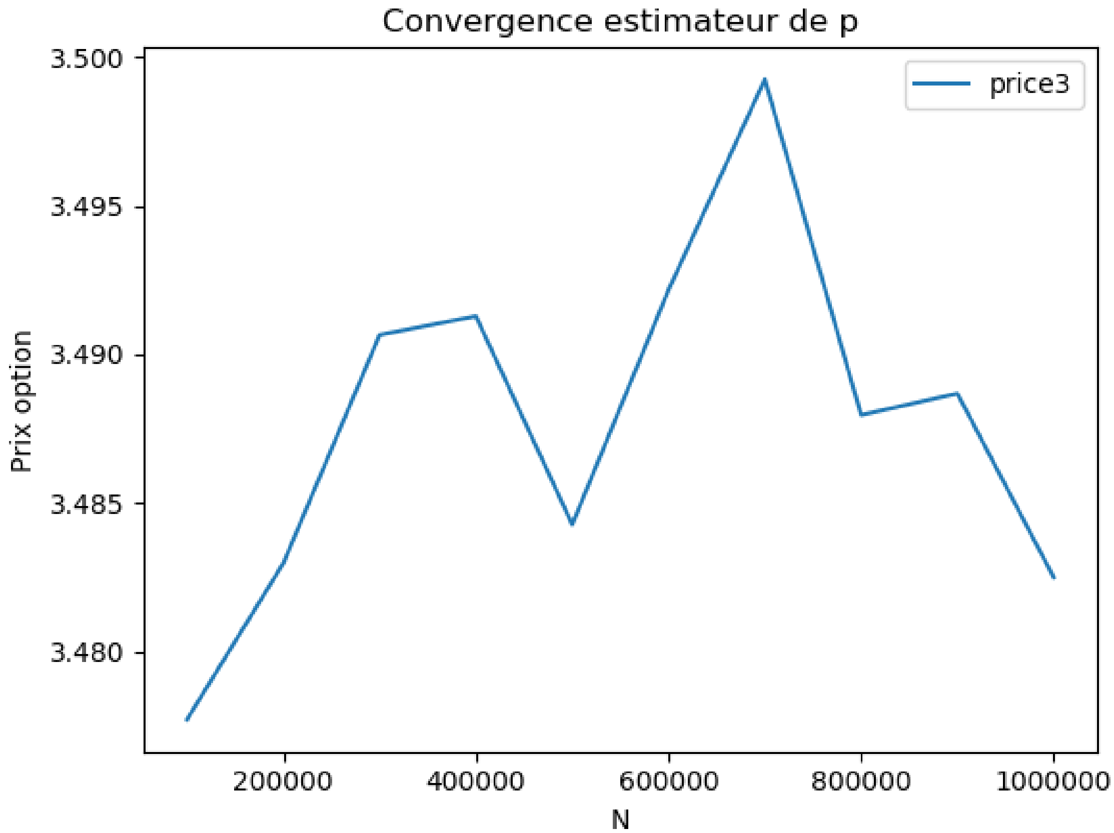
Soit au final :

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

12)

Algorithm 3 price3(n, s, r, σ, T, f)

Ensure : res = $p\hat{(n)}$
  $sum \leftarrow 0$
  for i in 1 : n **do**
    $sum \leftarrow sum + exp(-rT)f(s\exp((r - \frac{sigma^2}{2})T + sigma\sqrt{T} * np.random.normal()))$
    $res \leftarrow \frac{sum}{n}$
  end for



13)

14) Posons pour $1 \leq i \leq n, X_i = e^{-rT} f(s \exp((r - \frac{\sigma^2}{2})T + \sigma\sqrt{T}\epsilon_i)$. Pour montrer que la suite $(\hat{p}_{(n)})_{n\in\mathbb{N}}$

converge presque sûrement vers p, il suffit de montrer que E$[X_0]$=p. En effet, les $(\epsilon_i)_{1\leq i\leq n}$ étant indépendantes et identiquement distribuées, il en de même pour les $(X_i)_{1\leq i\leq n}$. E$[X_0]$=$p < +\infty$ permet de conclure, d'après la loi forte des grands nombres, que la suite $(\hat{p}_{(n)})_{n\in\mathbb{N}}$ converge presque sûrement vers p.

Or d'après les propriétés 1 et 2 du mouvement brownien énoncées dans le sujet, on sait que $B_t \sim N(0,t)$, donc que $\frac{B_t}{\sqrt{t}} \sim N(0,1)$. On en déduit qu'il existe $\varepsilon^{'} \sim N(0,1)$ tel que pour tout T, on a $B_T = \sqrt{T}\varepsilon^{'}$. $\varepsilon^{'}$ et les $(\epsilon_i)_{1\leq i\leq n}$ étant de même loi, on en déduit que E$[X_0]$=p.

15)

Algorithm 4 put(s, r, σ, T, K)

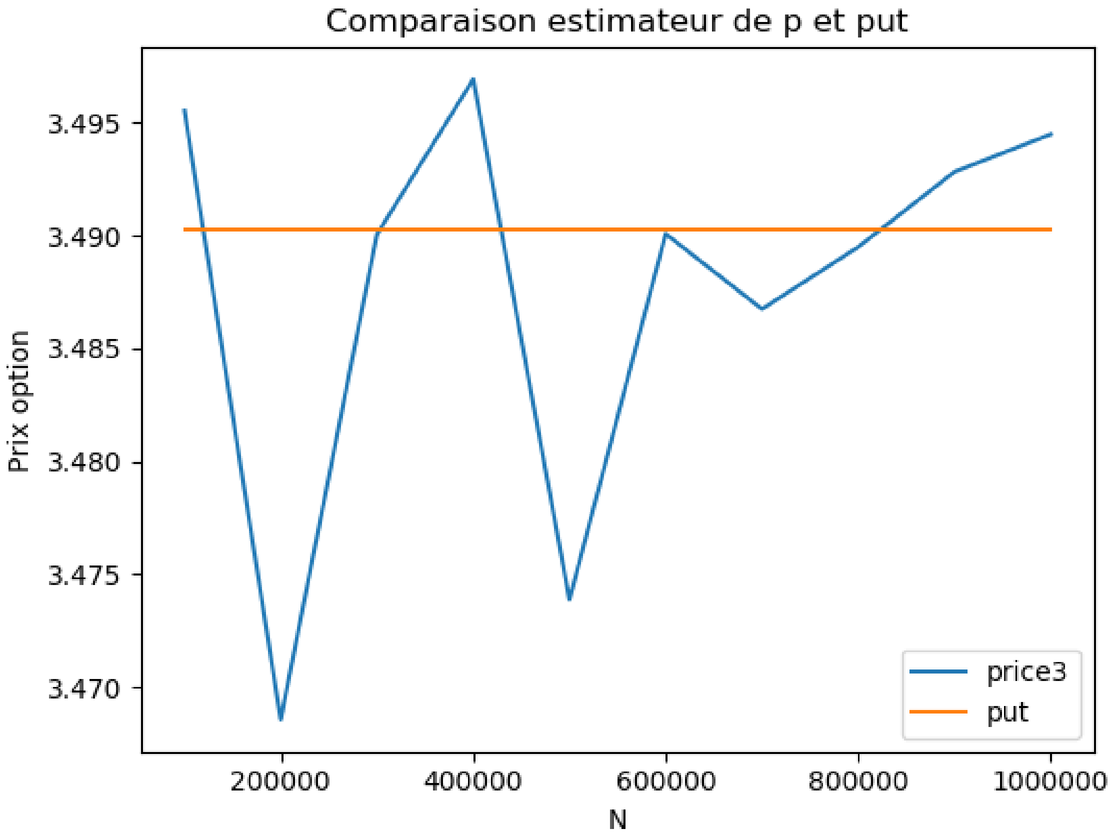
Ensure : res $= p$

$d \leftarrow (log(\frac{s}{K}) + (r + \frac{sigma^2}{2})T)/(sigma\sqrt{T})$

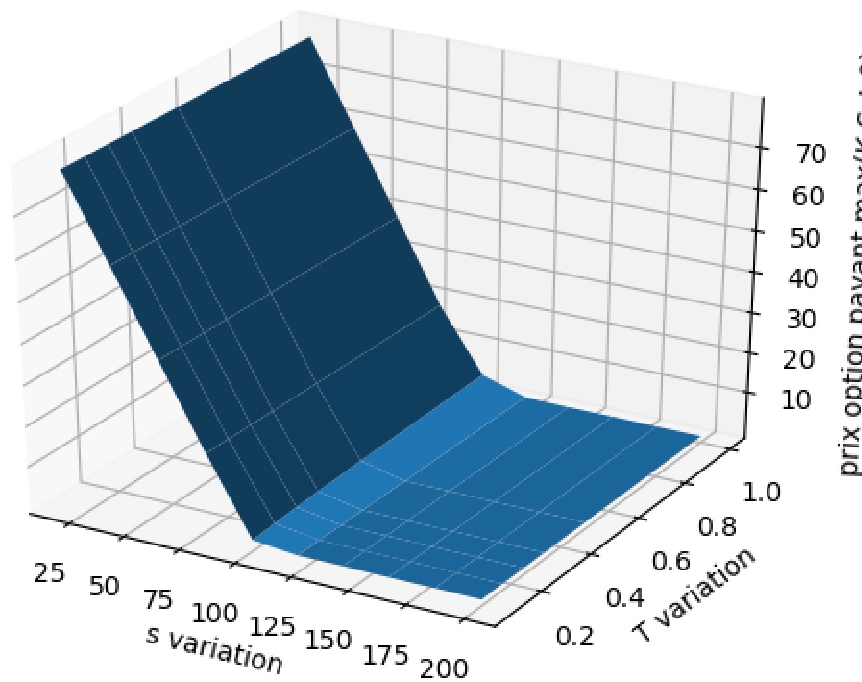
$p \leftarrow -s * norm.cdf(-d) + K\exp(-rT) * norm.cdf(-d + sigma\sqrt{T})$

$res \leftarrow p$

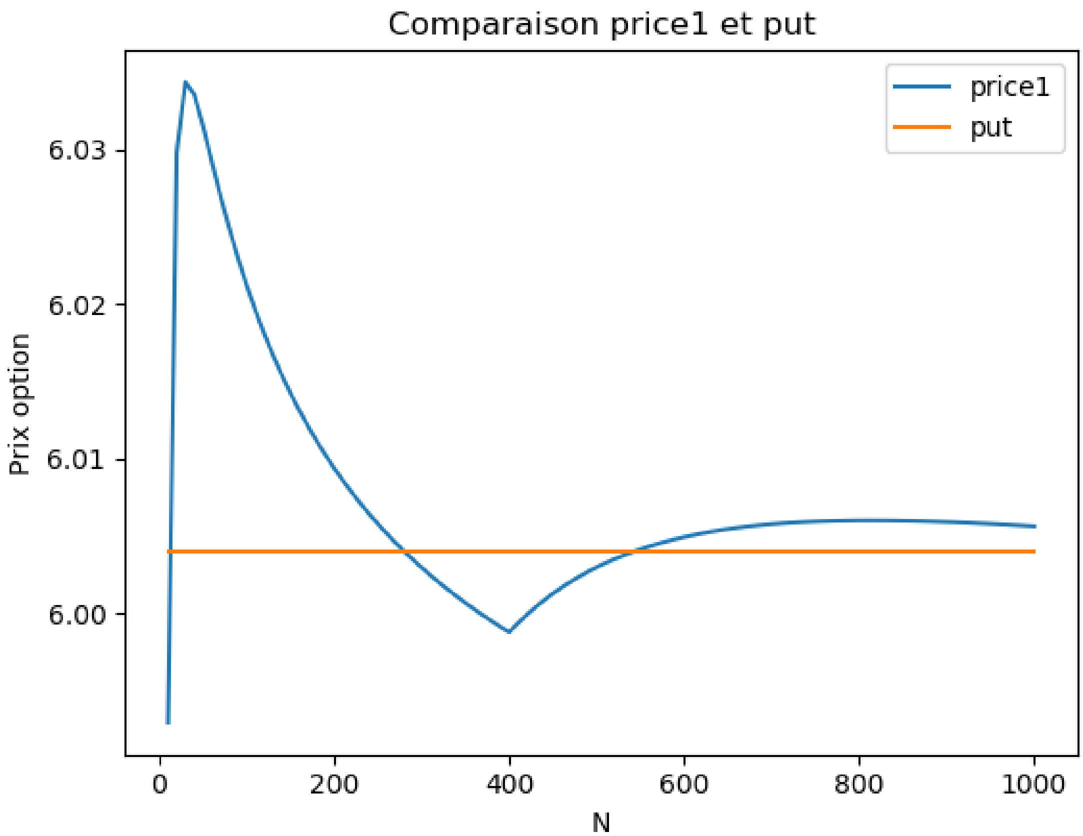
16) Voir code python.


Comparaison estimateur de p et put
Prix option
3.495
3.490
3.485
3.480
3.475
3.470
200000
400000
600000
800000
1000000
N
price3
put

17)


prix option payant max(K-S_t,0)
s variation
T variation
25
50
75
100
125
150
175
200
0.2
0.4
0.6
0.8
1.0
10
20
30
40
50
60
70

18)



19)

20) Le prix du put $P(t, S_t)$ vérifie l'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 :

$P(T, s) = max(K - s, 0) \forall s \in [0, L]$ [01]

$P(t, 0) = Ke^{r(t-T)} \forall t \in [0, T]$ [02]
$P(t, L) = 0 \forall t \in [0, T]$ [03]
Si l'on pose N (resp. M)le nombre de points en temps (resp. espace) à calculer, on pose $\Delta T = \frac{T}{N}$ (resp. $\Delta s = \frac{L}{M+1}$) puis $t_n = n\Delta T \forall n \in \{0, .., N\}$ (resp. $s_i = i\Delta s \forall i \in \{0, .., M+1\}$).

On approche alors $P(t, S_t)$ par $P(t_n, s_i)$. On pose $\mathbf{P}_n \in \mathbb{R}^M$ le vecteur colonne de composantes $(P_{n,1}, .., P_{n,M})$, où $P(t_n, s_i)$ est remplacé par $P_{n,i}$ dans le cas où les approximations ci-dessous deviennent des égalités.

Schéma implicite

Dans un schéma implicite, on a pour tout $(n, i) \in \{0, .., N-1\} \times \{1, .., M\}$:

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

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

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

Posons $A_i = -\frac{1}{2}\frac{\Delta T}{\Delta s^2}\sigma^2 s_i^2$, $B_i = \Delta T(r + \frac{\sigma^2 s_i^2}{\Delta s^2} - \frac{r s_i}{\Delta s} + \frac{1}{\Delta T})$ et $C_i = \Delta T(\frac{r s_i}{\Delta s} - \frac{1}{2}\frac{\sigma^2 s_i^2}{\Delta s^2})$
Dans un schéma explicite, on a pour tout $(n, i) \in \{0, .., N-1\} \times \{1, .., M\}$:

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

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

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