

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 :

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

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 $i \in \{1, \dots, N\}$, 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 reconnaît 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 :

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

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 $\text{price2}(N, r_N, h_N, b_N, s, f, k)$

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

if $k = N$ **then**

$\text{res} \leftarrow f(s)$

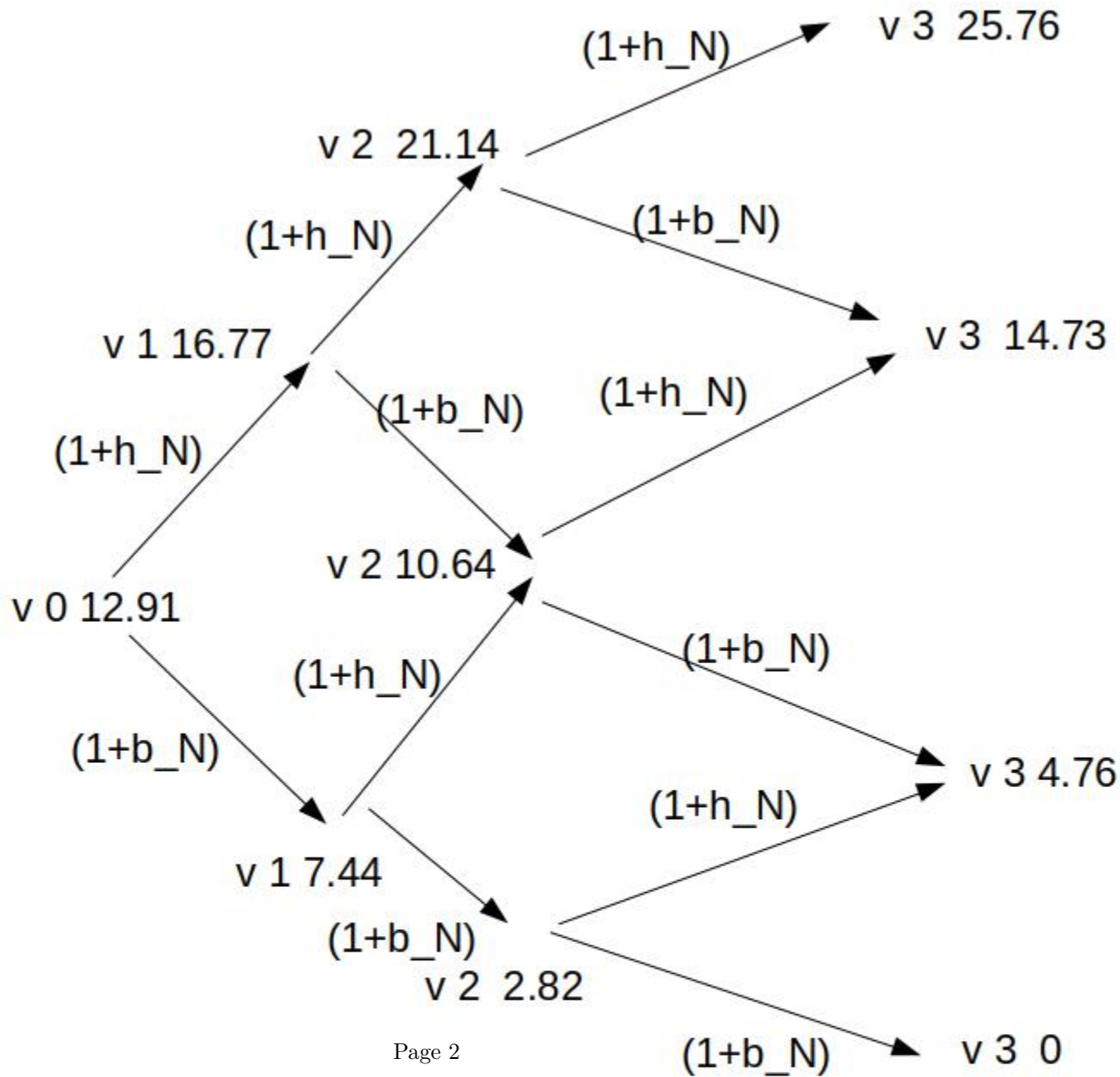
else

$q_N \leftarrow \frac{r_N - b_N}{h_N - b_N}$

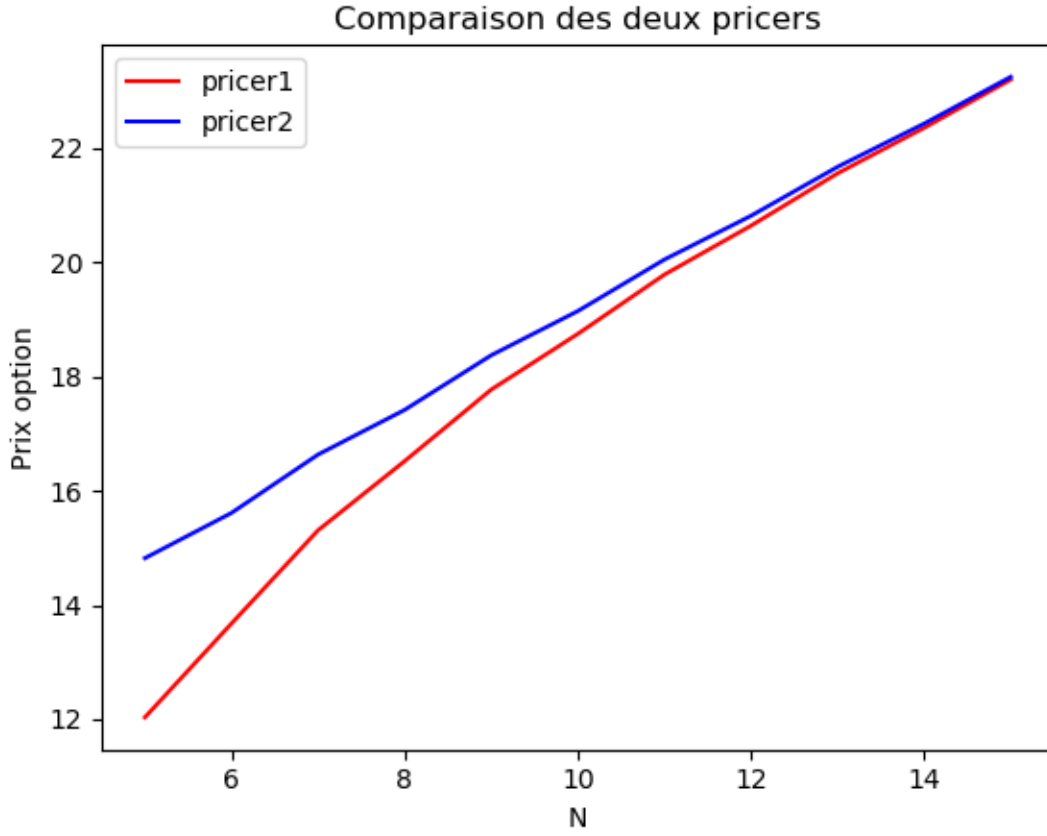
$\text{esp} \leftarrow \text{price2}(N, r_N, h_N, b_N, s(1 + h_N), f, k + 1)q_N + \text{price2}(N, r_N, h_N, b_N, s(1 + b_N), f, k + 1)(1 - q_N)$

$\text{res} \leftarrow \frac{\text{esp}}{1 + r_N}$

end if



7)



8) Le système à la date t_N vaut :

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

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

[1] – [2] donne :

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

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

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

9) Le système pour les autres dates t_k vaut :

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

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

Ce qui donne, par un raisonnement analogue :

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

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

10) Voir code python.

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

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

$$\iff d\ln(S_t) = (rdt + \sigma dB_t) - \frac{\sigma^2}{2}dt \quad \text{car} \quad dS_t = S_t(rdt + \sigma dB_t)$$

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

$$\iff \ln S_t - \ln s = (r - \frac{\sigma^2}{2})t + \sigma B_t \quad \text{car } B_0 = 0$$

Soit au final :

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

12)

Algorithm 3 price3(n, s, r, σ, T, f)

Ensure : $\text{res} = p_{(n)}^{\hat{}}$

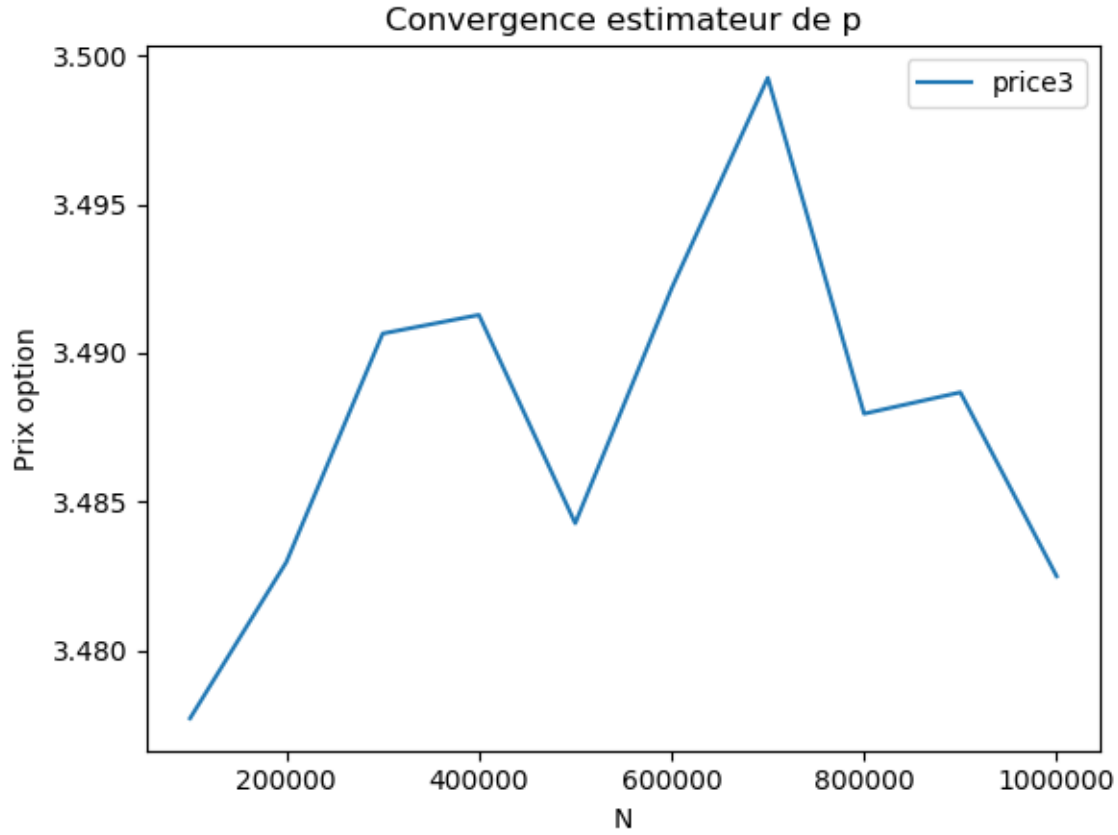
$\text{sum} \leftarrow 0$

for i **in** $1 : n$ **do**

$\text{sum} \leftarrow \text{sum} + \exp(-rT)f(s \exp((r - \frac{\sigma^2}{2})T + \sigma \sqrt{T} * \text{np.random.normal}()))$

$\text{res} \leftarrow \frac{\text{sum}}{n}$

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 est 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 $\epsilon' \sim N(0, 1)$ tel que pour tout T , on a $B_T = \sqrt{T}\epsilon'$. ϵ' 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)

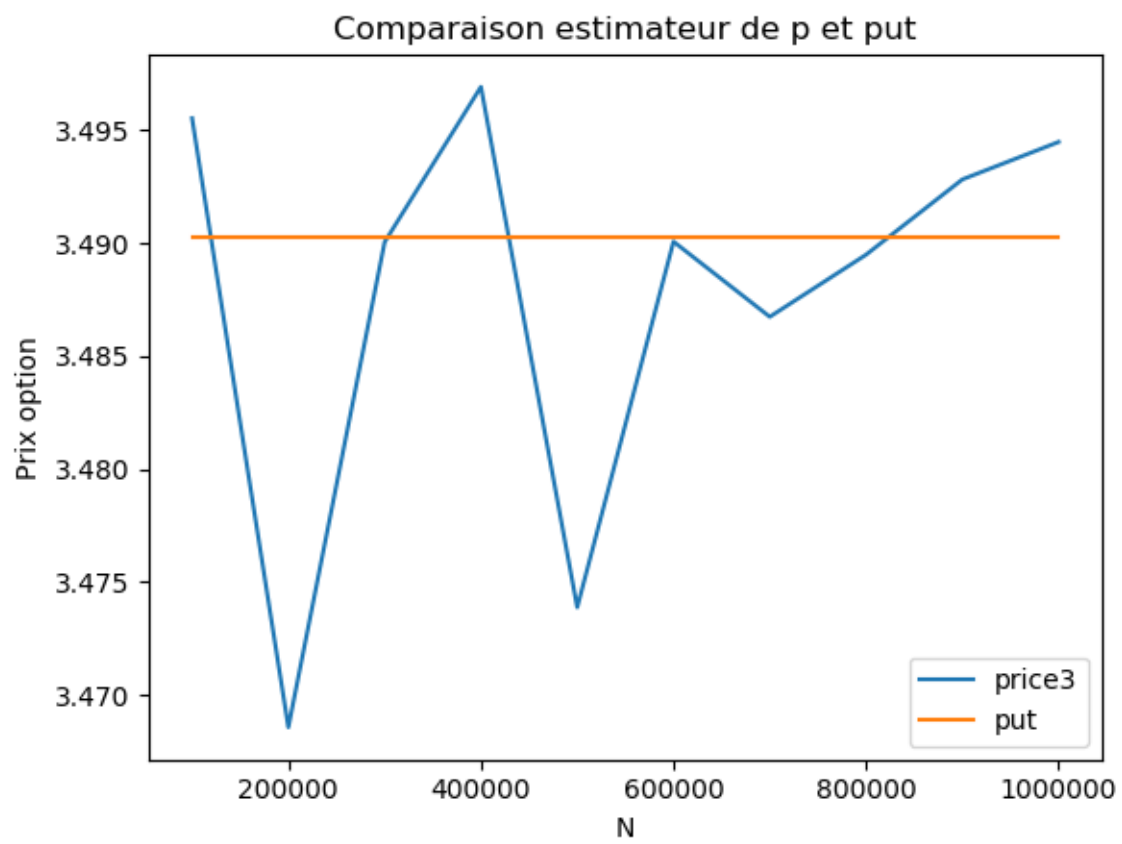
Ensure : res = p

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

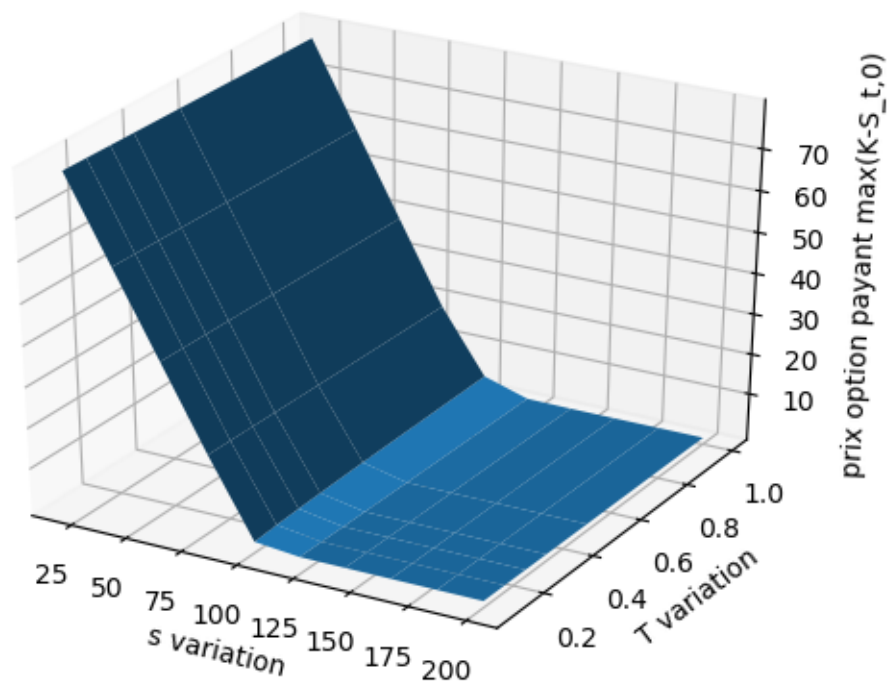
$p \leftarrow -s * \text{norm.cdf}(-d) + K \exp(-rT) * \text{norm.cdf}(-d + \sigma\sqrt{T})$

res $\leftarrow p$

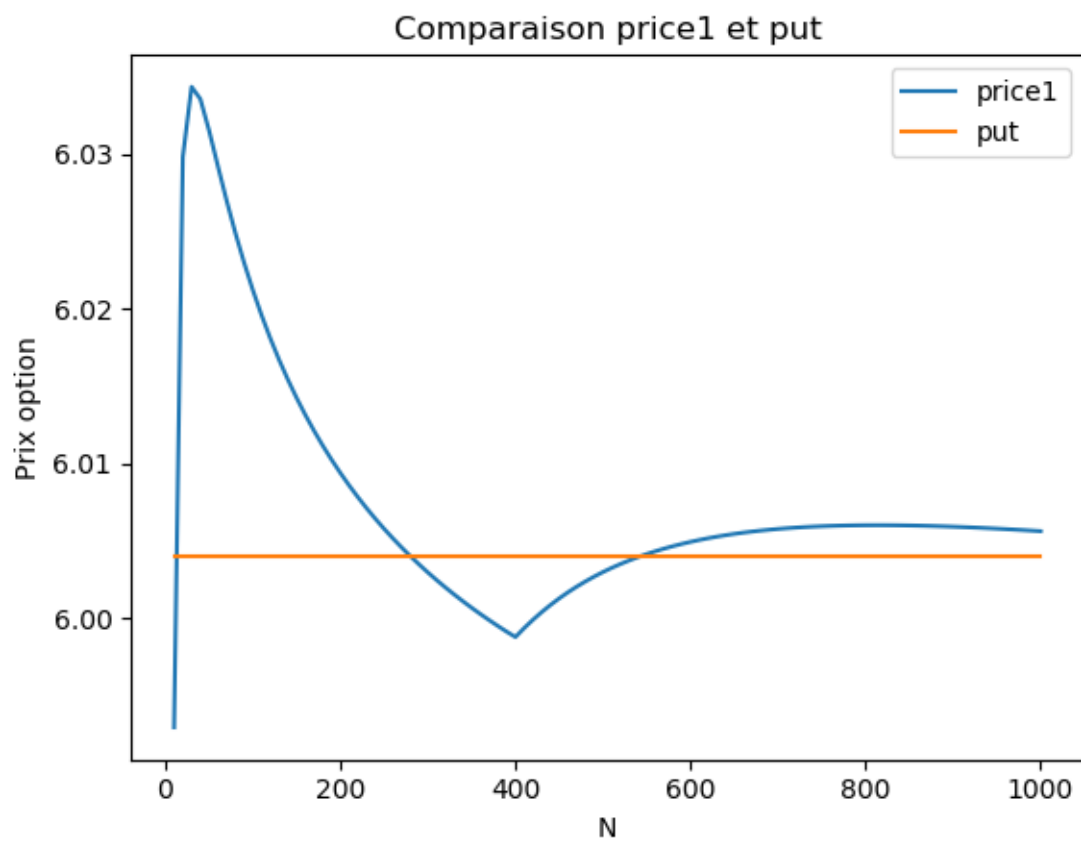
16) Voir code python.



17)



18)



19)

20)