ensiie
Paris | Évry-Courcouronnes

Rapport du Projet Mathématique

12 Avril 2022

Adib HABBOU
Adel KEBLI
Rabab KHATIB

Parte 1 :
Modèle de Cox-Ross-Rubinstein

Table des matières

1 Modèle de Cox-Ross-Rubinstein

1.1 Question 1 - Probabilité risque-neutre q_N

On exprime dans un premier temps $\mathbb{E}_{\mathbb{Q}}[T_1^{(N)}]$:

$$\mathbb{E}_{\mathbb{Q}}[T_1^{(N)}] = (1+h_N) \times \mathbb{Q}(T_1^{(N)} = 1+h_N) + (1+b_N) \times \mathbb{Q}(T_1^{(N)} = 1+b_N)$$

Sachant que $\mathbb{E}_{\mathbb{Q}}[T_1^{(N)}] = 1 + r_N$:

$$\begin{aligned} 1+r_N &= (1+h_N)q_N + (1+b_N)(1-q_N) \\ 1+r_N &= (1+h_N)q_N + 1 + b_N - (1+b_N)q_N \\ r_N - b_N &= (h_N - b_N)q_N \end{aligned}$$

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

1.2 Question 2 - $Prix_{Bin}^{(N)}$

Nous avons $Prix_{Bin}^{(N)} = \frac{1}{(1+r_N)^N}\, \mathbb{E}_{\mathbb{Q}}[f(S_{t_N}^{(N)})]$. Donc :

$$\begin{aligned} Prix_{Bin}^{(N)} &= \frac{1}{(1+r_N)^N}\, \mathbb{E}_{\mathbb{Q}}[f(S_{t_N}^{(N)})] \\ Prix_{Bin}^{(N)} &= \frac{1}{(1+r_N)^N}\, \mathbb{E}_{\mathbb{Q}}[f(s.\prod_{i=1}^{N} T_i^{(N)})] \\ Prix_{Bin}^{(N)} &= \frac{1}{(1+r_N)^N} \mathbb{E}_{\mathbb{Q}}[f(s.(1+h_N)^X.(1+b_N)^{N-X})] \end{aligned}$$

Pour tout $i \in \mathbb{N}_N$, $T_i^{(N)}$ vaut soit $(1+h_N)$ soit $(1+b_N)$. Les $T_i^{(N)}$ sont des variables aléatoires pour lesquelles on pose le succès à $(1+h_N)$. Soit X le nombre de fois qu'on obtient $T_i^{(N)} = (1+h_N)$: comme on répétera N fois l'expérience du choix de X avec à chaque fois deux issues possibles, on aura $X \sim \mathcal{B}(q_N, N)$, avec $q_N = \mathbb{Q}(T_1^{(N)} = 1+h_N)$ la probabilité risque-neutre. Donc, $\forall k \in \mathbb{N}_N$, $\mathbb{P}(X = k) = \begin{pmatrix} N \\ k \end{pmatrix} q_N^k (1-q_N)^{N-k}$. Ce qui nous donne :

$$\boxed{Prix_{Bin}^{(N)} = \frac{1}{(1+r_N)^N} \sum_{k=0}^{N} \binom{N}{k} q_N^k (1-q_N)^{N-k} f(s.(1+h_N)^k.(1+b_N)^{N-k})}$$

1.3 Question 3 - fonction pricer_1

Nous nous appuyions sur trois fonctions auxiliaires :
— une fonction `qN` implémentant la relation trouvée à la question 1 ;
— une fonction `fact` calculant la factorielle d'un entier n positif ou nul ;
— une fonction `coeff_binom` calculant le coefficient binomiale.
La fonction `pricer_1` va traduire l'expression de $Prix_{Bin}^{(N)}$ trouvée à la question 2, en calculant d'abord la somme dans la formule, puis en renvoyant enfin $Prix_{Bin}^{(N)}$.

1.4 Question 4 - test pricer_1

On code la fonction `f` : `x` $\mapsto$ `max(x - 110, 0)`.

Pour les paramètres donnés, à savoir s = 100, h_N = 0.05, b_N =-0.05, r_N =0.02 et N=20, la fonction `pricer_1` renvoie :

$$\boxed{Prix_{Bin}^{(N)} = 26.616941360258558 \text{ €}}$$

1.5 Question 5 - fonction pricer_2

Algorithme

À l'étape 1, on a : $v_N(S_{t_N}^{(N)}) = f(S_{t_N}^{(N)})$

À l'étape k, on a donc : $v_k(S_{t_k}^{(N)}) = \frac{1}{1+r_N}\mathbb{E}_{\mathbb{Q}}[v_{k+1}(S_{t_{k+1}}^{(N)})|S_{t_k}^{(N)}]$

On fixe le prix de l'actif risqué $S_{t_k}^{(N)}$ tel que $S_{t_k}^{(N)} = S$. On obtient alors :

$$v_k(S_{t_k}^{(N)}) = \frac{1}{1+r_N}\mathbb{E}_{\mathbb{Q}}[v_{k+1}(S_{t_{k+1}}^{(N)})|S_{t_k}^{(N)} = S]$$
$$v_k(S_{t_k}^{(N)}) = \frac{1}{1+r_N}\mathbb{E}_{\mathbb{Q}}[v_{k+1}(S_{t_{k+1}}^{(N)})]$$
$$v_k(S_{t_k}^{(N)}) = \frac{1}{1+r_N}\mathbb{E}_{\mathbb{Q}}[v_{k+1}(S \times T_{k+1})]$$

Par le théorème de transfert, on a donc :

$$v_k(S_{t_k}^{(N)}) = (\frac{1-q_N}{1+r_N} \times v_{k+1}(S(1+b_N)) + \frac{q_N}{1+r_N} \times v_{k+1}(S(1+h_N)))$$

Fonction pricer_2

Pour implémenter la fonction `pricer_2`, on remplit tout d'abord une liste avec touts les éléments initiaux $v_N(S_{t_N}^{(N)}) = f(S_{t_N}^{(N)})$. Comme nous avons une relation entre des éléments qui se "suivent" (v_k dépendant de v_{k+1} d'après l'expression donnée par l'algorithme), pour remonter à $v_0(S_{t_0}^{(N)})$, il nous suffit de parcourir l'arbre de probabilité des valeurs de droite à gauche. En parcourant cet arbre, pour remonter aux valeurs de l'étape précédente, on applique la formule ci-dessus à chaque itération : ainsi on obtient le nouveau v_k qu'on stocke d'abord dans une liste `w`, avant de tout copier dans la liste `v`. Ceci nous permettra de ne pas perdre les valeurs de l'étape N-1. Une fois parcourue l'entièreté de l'arbre, on se retrouve à la racine qui contient $v_0(S_{t_0}^{(N)}) = Prix_{Bin}^{(N)}$, la valeur souhaitée.

```python
def pricer_2 (N, rN, hN, bN, s, f) :
  v = []
  for k in range (N + 1) :
    v.append(f(s * ((1 + hN)**k) * (1 + bN)**(N - k)))
  q = qN(rN, hN, bN)
  v0 = (1 - q) / (1 + rN)
  v1 = q / (1 + rN)
  while (len(v) != 1) :
    w = []
    for i in range (len(v) - 1) :
      w.append(v0 * v[i] + v1 * v[i + 1])
    v = w.copy()
  return v[0]
```

Script Python de la fonction pricer_2

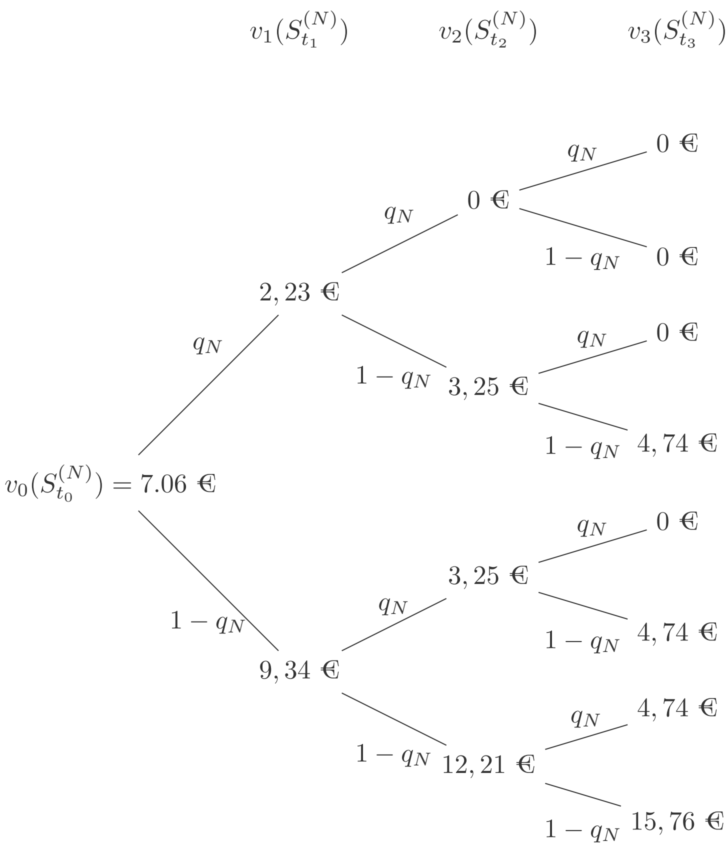
1.6 Question 6 - test pricer_2

On code la fonction `g` : `x` $\mapsto$ `max(x - 100, 0)`.

La fonction `pricer_2` nous permet d'obtenir :

$$\boxed{Prix_{Bin}^{(N)} = 7.063436197239376 \text{ €}}$$

Arbre des valeurs de v_k



1.7 Question 7 - comparaison pricer_1 et pricer_2

On utilise la fonction `randint` pour générer un N aléatoire entre 5 et 15. On renvoie la valeur absolue de la différence des valeurs obtenues avec `pricer_1` et `pricer_2`. On enregistre toutes les valeurs obtenues dans un `data frame` et on remarque que les différences sont bien proches de 0 : les deux pricers permettent pratiquement la même précision.

N	5	6	7	8	9	10
comparaison	8.881784e-16	8.881784e-16	8.881784e-16	0.0	0.0	8.881784e-16

N	11	12	13	14	15
comparaison	8.881784e-16	3.552714e-15	0.0	1.776357e-15	1.776357e-15

Data frames de comparaison selon la valeur de N

1.8 Question 8 - Système d'équations pour f

On a le système d'équations suivant :

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

Ce qui se résume à :

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

Donc, en soustrayant la deuxième ligne à la première et en isolant le $\alpha_{N-1}(S_{t_{N-1}}^{(N)})$, on obtient :

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

De même, on multiplie la deuxième équation par $(1+h_N)$ puis on la divise par $(1+b_N)$. En soustrayant la deuxième équation à la première, on obtient :

$$\beta_{N-1}(S_{t_{N-1}}^{(N)})\ (1+r_N)^N\ (1 - \tfrac{1+h_N}{1-b_N}) = f((1+h_N)S_{t_{N-1}}^{(N)}) - \tfrac{1+h_N}{1-b_N}\ f((1+b_N)S_{t_{N-1}}^{(N)})$$

On en déduit que :

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

1.9 Question 9 - Système d'équations pour v_k

Pour tout $k \in \{0, ..., N-1\}$, on a le système suivant :

$$\begin{cases} v_k(S_{t_k}^{(N)}) & = \alpha_{k-1}(S_{t_{k-1}}^{(N)})S_{t_k}^{(N)} + \beta_{k-1}(S_{t_{k-1}}^{(N)})(1+r_N)^k \\ S_{t_k}^{(N)} & = (1+h_k)S_{t_{k-1}}^{(N)} \\ S_{t_k}^{(N)} & = (1+b_k)S_{t_{k-1}}^{(N)} \end{cases}$$

De manière analogue, pour tout $k \in \{0, ..., N-1\}$, on peut retrouver les expressions suivantes :

$$\boxed{\alpha_k(S_{t_k}^{(N)}) \;=\; \frac{v_{k+1}((1+h_N)S_{t_k}^{(N)}) - v_{k+1}((1+b_N)S_{t_k}^{(N)})}{S_{t_k}^{(N)}\,(h_N - b_N)}}$$

$$\boxed{\beta_k(S_{t_k}^{(N)}) = \frac{1}{(1+r_N)^{k+1}\,(b_N-h_N)}\ (\,(1+b_N)\,v_{k+1}((1+h_N)S_{t_k}^{(N)})\ -\ (1+h_N)\,v_{k+1}((1+b_N)S_{t_k}^{(N)})\,)}$$

1.10 Question 10 - Couverture à la date 0 et à la date 1

Les fonctions `alpha` et `beta` appliquent les formules trouvées aux questions précédentes.

On effectue ensuite le calcul pour la date 0 avec $S_{t_0} = s$ puis pour la date 1 avec $S_{t_1} = s \times T_1$.

On utilise donc une fonction qui calcule T_1 avec la formule suivante :

$$T_1 = q_N \times (1 + h_N) + (1 - q_N) \times (1 + b_N)$$

Nombre d'actifs, risqués et non, à la date 0 : $\begin{cases} \alpha_0(S_{t_0}^{(N)}) & = 0.5 \\ \beta_0(S_{t_0}^{(N)}) & = -44.77330568385333 \end{cases}$

Nombre d'actifs, risqués et non, à la date 1 : $\begin{cases} \alpha_1(S_{t_1}^{(N)}) & = 0.7912621359223306 \\ \beta_1(S_{t_1}^{(N)}) & = -91.78527665189932 \end{cases}$

Nous remarquons que β peut prendre une valeur négatif : étant β le nombre d'actifs sans risque, ceci peut correspondre à une dette contractée pour pouvoir acheter α actifs risqués, que dans ce cas ci sont positifs.