Projet modèles financiers en temps discret : Modèle binomial Cox-Ross-Rubinstein et sa convergence vers le modèle Black-Scholes

Fait par Mamadou SOW et Jérémy LIABEUF

21 octobre 2024

ISFA
Institut de Science Financière et d'Assurances
GRANDE ÉCOLE D'ACTUARIAT & GESTION DES RISQUES

Table des matières

1 Présentation du modèle binomial **1**
1.1 Modèle binomial à une période 1
1.2 Modèle binomial à n périodes 2

2 Modèle Cox, Ross et Rubinstein (CRR) **4**
2.1 Description du modèle 4
2.2 Modélisation probabiliste du marché 5

3 Modèle de Black-Scholes **6**
3.1 Présentation du modèle 6
3.2 La formule de Black-Scholes dans le cas d'un call européen : 7
3.3 Convergence du modèle CRR vers le modèle de Black-Scholes . . 7
3.3.1 limite du prix CRR 7
3.3.2 Convergence vers Black-Scholes : 8
3.3.3 Vitesse de convergence 9

4 Application Numerique **10**
4.1 Algorithme CCR sur Python 10
4.1.1 Élaboration de l'arbre binomial 10
4.1.2 Élaboration des prix de l'actif sous-jacent 11
4.1.3 Retropropagation des prix des options 11
4.1.4 Affichage du prix de l'action call européen 12
4.2 Algorithme BS sur Python 12
4.3 Mise en place d'une simulation pour calculer le prix d'option d'achat 13
4.3.1 Simulation du premier modèle 13
4.3.2 Simulation du second modèle 15

5 Autres applications **16**
5.1 Application sur les prix de vente (put) européen 16
5.2 Application sur les options composés 16

Introduction

Le modèle binomial est un modèle de base qui représente, par un simple graphe, les différentes trajectoires d'évolution d'un actif financier ou d'un produit dérivé pendant la durée de validité de ce dernier. Dans ce premier chapitre, on va présenté les circonstances qui ont permis l'utilisation de ce modèle à des fins d'évaluation de risques financiers, ainsi que ses avantages. Selon l'approche générale (Korn et Müller, 2010), l'utilisation du modèle binomial survient dans le contexte d'absence d'arbitrage et dans un marché complet (c'est à dire quand tous les actifs peuvent être répliqués). En disposant d'un portefeuille contenant trois actifs dont chaque actif peut être répliqué par les deux autres, on présume aussi que la valeur future du portefeuille peut prendre seulement deux états, soit dans une situation de hausse ou bien une situation de baisse des prix. Dans cette optique, l'utilisation du modèle binomial est adéquate et reflète bien une grande partie de la réalité.
Dans ce rapport, nous allons d'abord présenter le modèle binomial. Ensuite, nous parlerons de l'évaluation des options grâce au modèle binomial par la méthode de Cox-Ross-Rubinstein. Nous présenterons les calculs associés à l'option call européenne. Enfin, nous verrons sa convergence vers le modèle de Black-Scholes.

1 Présentation du modèle binomial

1.1 Modèle binomial à une période

Considérons un portefeuille contenant un seul actif risqué négocié sur une période de temps qui commence à l'instant 0 et qui prend fin à l'instant T. Le prix de cet actif est de S_0 au début de la période et de S_r à la fin de cette même période. Par la suite, supposons que le prix S_r à l'instant T dépend du lancer d'une pièce de monnaie. On désigne par $S_r(U)$ le prix de notre actif si le résultat du lancer de cette pièce est un succès et $S_r(D)$ s'il s'agit d'un échec. Aussi, on admet que pour l'expérience aléatoire considérée, la probabilité d'un succès est p et celle d'un échec est $1-p$. Notons que p est strictement positive et qu'on doit obligatoirement avoir $p \neq \frac{1}{2}$. Finalement, toutes les valeurs $S_i(x)$, $x = U, D$ et $i = 0, T$, sont strictement positives. La figure 1.1 schématise ce qui précède.

$$S_0$$

$$S_T(U) \qquad S_T(D)$$

En se plaçant maintenant à l'instant T, on sera en mesure de connaître le résultat du lancer de la pièce de monnaie. Par conséquent, les prix $S_r(U)$ et $S_r(D)$ seront connus aussi. Cependant, il sera difficile de déterminer la valeur initiale S_0, d'ailleurs, c'est de ce contexte que découle son aspect aléatoire puisqu'elle dépend du résultat de l'expérience aléatoire.

On introduit dans ce qui suit les paramètres u, d et r. Les paramètres u, d et r sont introduits dans ce qui suit. D'une part, les deux premiers paramètres découlent de la considération de l'impact de notre expérience aléatoire sur le prix de notre actif, alors que r est le taux d'intérêt en vigueur. D'autre part, on présume que $u > 1$, $d < 1$ et $0 < d < (1 + r) < u$. Il est crucial de noter que la dernière inégalité exprime l'absence d'arbitrage qui se traduit par la possibilité de réaliser un bénéfice considérable à l'instant T tout en partant d'une richesse de valeur 0 à l'instant 0.

Précisons maintenant cette pensée, si $d > (1+r)$, un négociateur peut acheter un actif négocié à l'instant 0 par l'argent emprunté. Par conséquent, au pire des cas, à l'instant T , le prix de l'actif réalise un profit qui couvre suffisamment le montant d'argent emprunté et le montant des intérêts chargés, d'où la présence d'arbitrage. Autrement, si $U < (1+r)$, le négociateur n'a plus d'intérêt à investir sur l'actif risqué à l'instant O. En effet, à l'instant T et pour la même période, le profit réalisé de l'investissement sur l'actif risqué sera considérablement inférieur à celui d'un actif sans risque, ce dernier sera garanti et plus important. Là aussi, il y a une opportunité d'arbitrage.

Par la suite, on suppose que $c_l = t$, et qu'on dispose d'une option d'achat européenne dont le prix d'exercice à la date de maturité T est K. De ce qui précède, on prétend avoir $S_r(D) < I < S_r(U)$ afin de s'assurer qu'il n'y ait pas d'opportunité d'arbitrage.

Soit :

$$\begin{cases} S_T(U) = uS_0 \\ S_T(D) = dS_0 \end{cases}$$

Il en résulte qu'à la date de maturité T, la valeur de l'option, notée V, est exprimée par :

$$V = (S_T - K)^+$$

1.2 Modèle binomial à n périodes

On considère maintenant les mêmes hypothèses décrites dans la section précédente. Par contre, on dispose cette fois de plusieurs dates d'échéances, soit les instants t_1, t_2, ..., t_n. En rappelant que $d = r$, on en déduit que deux résultats successifs et différents, pendant deux périodes consécutives, s'annulent et n'ont pas d'impact sur le prix d'un actif. En effet, le prix de l'actif ne change pas après deux périodes successives si le résultat du dernier lancer de la pièce de monnaie est un succès précédé par un échec ou un échec précédé par un succès ($S_i(UU...U) = S_{i+1}(UU...UD)$ ou $S_{i-1}(UU...U) = S_{i+1}(UU...UUD), i = 1, ..., n-1$). De ce fait, on aura à chaque instant t_i, $i+1$ valeurs possibles pour le portefeuille. La Figure suivante illustre ce qui précède.

En effet, pour i = 2, on a d'une part :

$$S_2(UD) = udS_0 = S_0$$

D'autre part, on a :

$$S_2(DU) = duS_0 = S_0$$

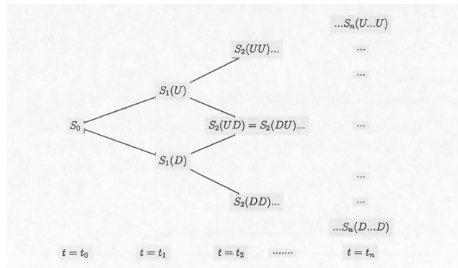

FIGURE 1 – Modèle binomiale a n périodes

Il en résulte que :

$$S_2(DU) = S_2(UD) = S_0$$

La figure suivante résume ce qui précède dans un arbre binomial à deux périodes.

L'objectif étant toujours de pouvoir évaluer C_0 qui est le prix de l'option à la date t_0 , pour cela, on applique une technique itérative (Hull, 2003). Cette technique consiste à calculer les valeurs de l'option en commençant par la date d'échéance où les valeurs de l'option sont connues, et en rétrogradant d'une manière itérative jusqu'à déterminer la valeur de l'option à la date désirée. En rappelant qu'à chaque instant le négociateur équilibre son portefeuille constitué par une action S, un actif sans risque B et une option d'achat européenne C, le prix de l'option à la date de maturité est exprimé par $C_2 = (S_2 - K)^+$, ainsi que le prix de l'action à la même date dépendent tous les deux des résultats des deux premiers lancers de la pièce de monnaie. En effet, à l'instant t_1, on dispose d'une richesse X_1, comme on a vu d'ailleurs précédemment. Suivant le résultat obtenu au premier lancer de la pièce de monnaie, cette richesse peut prendre deux valeurs. À la lumière de ces valeurs, le négociateur réajuste son portefeuille par l'achat de a_1 parts de l'actif risqué. D'une manière générale, la richesse à l'instant t_{n+1} s'écrit sous la forme (Shreve, 2004) :

$$\begin{cases} X_1(U) = \alpha_0 S_1(U) + (1+r)(C_0 - \alpha_0 S_0) \\ X_1(D) = \alpha_0 S_1(D) + (1+r)(C_0 - \alpha_0 S_0) \end{cases}$$

De la même manière que précédemment, le prix de l'option à l'instant t_2 est de la forme :

$$C_2 = (S_2 + K)^+$$

Cependant, cette fois-ci le négociateur veut que sa

$$\begin{cases} C_2(U,U) = \alpha_1(U) S_2(UU) + (1+r)(X_1(U) - \alpha_1(U) S_1(U)) \\ C_2(U,D) = \alpha_1(U) S_2(UD) + (1+r)(X_1(U) - \alpha_1(U) S_1(U)) \\ C_2(D,U) = \alpha_1(D) S_2(DU) + (1+r)(X_1(D) - \alpha_1(D) S_1(D)) \\ C_2(D,D) = \alpha_1(D) S_2(DD) + (1+r)(X_1(D) - \alpha_1(D) S_1(D)) \end{cases}$$

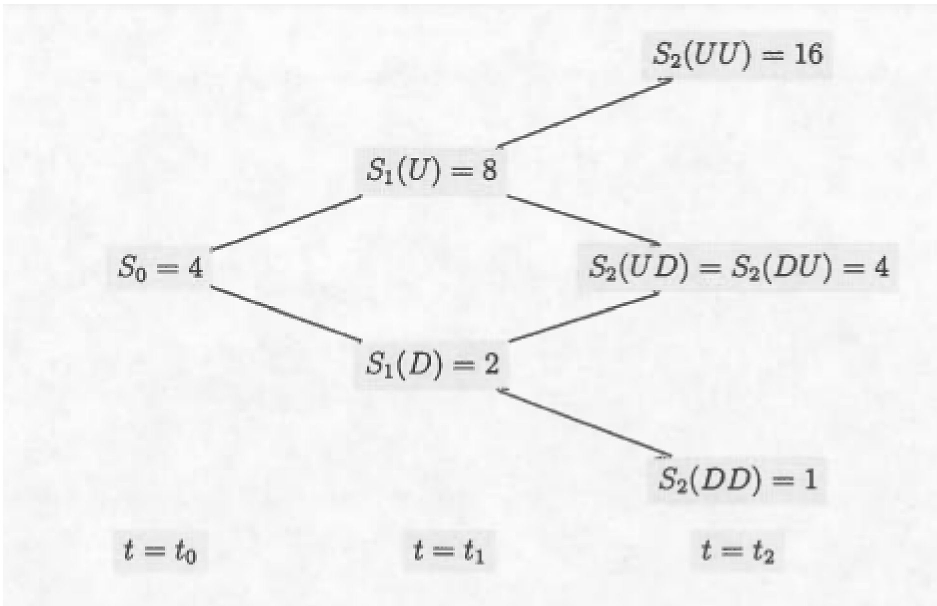

FIGURE 2 – Arbre binomial à 2 périodes représentant l'évolution du prix de l'actif risqué sur la période $[0,\, t_2]$

2 Modèle Cox, Ross et Rubinstein (CRR)

Le modèle binomial (ou modèle Cox-Ross-Rubinstein (CRR)) est une méthode numérique très simple pour calculer les prix de produits dérivés. En fait, l'arbre binomial donne une approximation de l'espérance des flux monétaires actualisés de l'option dans un univers neutre au risque. Il a été proposé pour la première fois par Cox, Ross et Rubinstein (1979).

2.1 Description du modèle

Le modèle CRR, est un modèle à temps discret dans le cours de l'actif risqué peut d'un instant à l'autre soit monter, nous allons nous restreindre au cas où le marché contient un seul actif risqué, $d = 1$: Cette simplification est adoptée par souci de clarté du texte, les résultats énoncés dans cette section peuvent être étendus au cas général $d \geq 1$ sans difficulté.

La création de l'arbre de prix s'effectue en partant de la date à laquelle on veut valoriser l'option et ce jusqu'à la date d'expiration de l'option. À chaque étape, on accepte que le sous-jacent augmente (up) ou diminue (down) en fonction d'un facteur spécifique (u ou d) et ce pour toutes les étapes. (Par définition, $u \geq 1$ et $0 \leq d \leq 1$). Par conséquent, si S est Si S est le prix actuel, alors le prix de la période suivante sera soit $S_{\text{up}} = S_u$ ou $S_{\text{down}} = S_d$.

2.2 Modélisation probabiliste du marché

La modélisation probabiliste du marché est la donnée de trois choses :

— Ω est l'ensemble des états du monde : 2 états possibles selon la valeur de l'actif risqué en $t = 1$, état "haut" ω_u ou "bas" ω_d, c'est-à-dire $\Omega = \omega_u, \omega_d$.

— P est la probabilité historique sur Ω. $P(\omega_u) = p$ et $P(\omega_d) = 1 - p$. Le prix a une probabilité réelle p de monter et $1 - p$ de descendre.

— $\mathcal{F} = F_0, F_1, F_2$ est une tribu représentant l'information globale disponible sur le marché aux instants $t = 0$, $t = 1$ et $t = 2$.

Considérons un actif valant S_0 à la période initiale et qui, à chaque période, peut être haussier (et avoir un rendement u) avec une probabilité p ou baissier (rendement d) avec une probabilité $q = (1 - p)$. Sur 3 périodes on a :

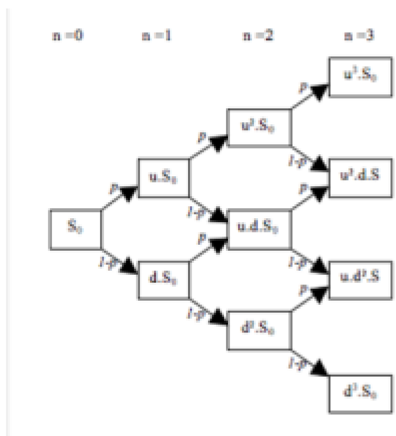

FIGURE 3 – Graphe représentant l'évolution du prix à chaque noeud

L'ensemble des états uniaux est $\Omega = uuu, uud, udu, udd, duu, dud, ddu, ddd$. Le prix de l'actif à chaque période (sauf à $t = 0$) est une variable aléatoire.

$$S_1 = \begin{cases} S_0 u, & \text{avec probabilité } p \\ S_0 d, & \text{avec probabilité } 1 - p \end{cases}$$

$$S_2 = \begin{cases} S_0 u^2, & \text{avec probabilité } P \\ S_0 ud, & \text{avec probabilité } 2p(1-p) \\ S_0 d^2, & \text{avec probabilité } (1-p)^2 \end{cases}$$

$$S_3 = \begin{cases} S_0 u^3, & \text{avec probabilité } P^3 \\ S_0 u^2 d, & \text{avec probabilité } 3p^2(1-p) \\ S_0 ud^2, & \text{avec probabilité } 3p(1-p)^2 \\ S_0 d^3, & \text{avec probabilité } (1-p)^3 \end{cases}$$

Remarque Un produit dérivé (ou actif contingent) : G est une variable aléatoire $\mathcal{F}T$ mesurable et s'écrit donc sous la forme :

$$G = \varphi({St_0}^1, S^1_{t_1}, \dots, S^1_{t_N})$$

avec φ application borélienne. Lorsque φ dépend seulement de la dernière variable, on dira que l'actif dérivé est indépendant du chemin : calls et puts européens (dits aussi "vanille"), calls et puts digitaux.

Théorème (Existence et unicité de la probabilité risque neutre) : Supposons que $t_k = k\Delta t$ et que dans le modèle binomial, il y a AOA. Alors :

1. $d < e^{r\Delta t} < u$;
2. Il existe une unique probabilité risque neutre Q. Sous cette probabilité les rendements $Y_1, \dots, Y_N$ sont indépendants et de même loi (iid) donnée par :

$$Q(Y_1 = u) = q$$
$$Q(Y_1 = d) = 1 - q$$

ou

$$q = \frac{e^{r\delta t} - d}{u - d}$$

Théorème (Évaluation des actifs indépendants du chemin) : Soit un actif dérivé G indépendant du chemin $G = \varphi(S_1(T))$ avec $\varphi : \mathbb{R} \to \mathbb{R}$ mesurable. Supposons que $t_k = k\delta t$ et que dans le modèle binomial, il y a AOA. Alors pour $k = 0, \dots, N$, le prix de G à la date t_k ne dépend que de la valeur du sous-jacent à cette date.

on note $p^G_{t_k}(X)$ ce prix quand $S^1_{t_k} = X$ et on n'a

$$p^G_{t_N}(X) = \phi(X) \quad etp^G_{t_k}(X) = e^{-r\delta t}[qp^G_{t_{k+1}}(X_u) + (1-q)p^G_{t_{k+1}}(X_d)] \quad sik = 0; \dots; N$$

Proposition (Prix d'un actif réplicable) : On suppose qu'il existe une mesure de probabilité neutre au risque Q. On suppose que l'actif contingent G admet une stratégie de réplication $\theta \in A$ à partir du capital initial v_0. On suppose de plus que θ est bornée. Si on note par $p^G_{t_k}$ le prix de G à l'instant t_k alors :

$$\forall k \in \{0, \dots, N\} : \quad p^G_{t_k} = e^{r(t_k)} \mathbb{E}^Q(G_e^{-rT} \mathcal{F}_{t_k})$$

3 Modèle de Black-Scholes

3.1 Présentation du modèle

On considère un marché formé par un actif sans risque S_0 et un actif risqué S. On suppose que le prix de l'actif sans risque vérifie :

$$S^0_t = S^0_0 exp(rt)$$

où r est une constante donnée, $r \geq 0$. $S_0^0 = 1$. Pour modéliser l'incertitude concernant le prix de l'actif risqué S, on introduit un Espace de probabilité $(\Omega, \mathcal{F}, \mathbb{P})$. On suppose que le processus de prix de S, $S = \{S_t, t \geq 0\}$, est décrit par le modèle de Black-Scholes.

$$dS_t = \mu S_t dt + \sigma S_t dB_t$$

ou $B = B_t, t \geq 0$ est un mouvement Brownien sur $(\Omega, \mathcal{F}, \mathbb{P})$. Ici, μ et σ sont deux constantes données, avec $\sigma > 0$.

— La constante μ est appelée "tendance" de S.

— La constante σ est appelée "volatilité" de S.

3.2 La formule de Black-Scholes dans le cas d'un call européen :

$$C_0 = S_0(\frac{1}{\sqrt(2 * \pi)} \int_{-\infty}^{d_1} e^{-\frac{y^2}{2}} dy - Ke^(-rT)(\frac{1}{\sqrt(2 * \pi)}) \int_{-\infty}^{d_2} e^{-\frac{y^2}{2}} dy$$

ou $d_1 = \frac{\ln\left(\frac{S_0}{K}\right)+\left(r+\frac{\sigma^2}{2}\right)T}{\sigma\sqrt{T}}$ et $d_2 = \frac{\ln\left(\frac{S_0}{K}\right)+\left(r-\frac{\sigma^2}{2}\right)T}{\sigma\sqrt{T}}$

Elle donne le prix à l'instant initial (la prime) d'un call Européen de date d'exercice T et de prix d'exercice K sur un actif sous-jacent de prix initial S_0, de volatilité σ, sachant que le taux sans risque est r. On notera que d_1 et d_2 sont des constantes qui vérifient $d_2 = d_1 - \sigma\sqrt{T}$. Cette formule permet de calculer la prime du call dès lors qu'on se donne les constantes T, K, S_0, σ et r dont elle dépend. On la réécrit souvent en utilisant la fonction de répartition de la loi normale centrée réduite :

Elle s'écrit alors plus simplement :

$$C_0 = S_0 N(d_1) - Ke^{-rT} N(d_2) \tag{1}$$

3.3 Convergence du modèle CRR vers le modèle de Black-Scholes

3.3.1 limite du prix CRR

On a vu que dans un modèle Cox, Ross et Rubinstein, la prime d'un Put (Européen), tout comme celle d'un Call, est égale à l'espérance, sous la probabilité risque neutre, du payoff actualisé :

$$P_0(n) = e^{-rT} E_\phi(S_T)$$

où ϕ désigne ce payoff qui, dans le cas du Put, vaut $\phi(S) = (K - S)^+$. Cette formule CRR permet, tout comme la formule de Black-Scholes, de calculer la prime du Put dès que l'on s'est donné les constantes T, K, S_0, σ et r. La principale différence est que le prix dépend cette fois, en plus de ces constantes, du nombre n de pas de discrétisation de l'intervalle $[0, T]$. D'où la notation adoptée à présent, $P_0(n)$. l'objectif de ce paragraphe est de montrer que, lorsque n tend vers l'infini, la limite de P0(n) existe et qu'elle est precisement ´egale au prix Black-Scholes (formule (1), c'est-'a-dire de montrer que :

$$\lim_{n\to\infty} e^{-rT} E\left((K - S_T)^+\right) = Ke^{-rT} N(-d_2) - S_0 N(-d_1) \tag{2}$$

Désignons par Z une variable aléatoire prenant les valeurs -1 et +1 avec les probabilités $1-p$ et p respectivement, où p est la probabilité risque-neutre($p = \frac{R-d}{u-d}$, où $R = e^{r\delta t}$, $u = e^{\sigma\sqrt{\delta t}}$ et $d = e^{-\sigma\sqrt{\delta t}}$) . À noter que dans ce cas, la loi de Z dépend de n puisque $\delta t = T/n$ est une fonction de n. On peut alors réécrire, dans le modèle CRR, la variable aléatoire S_T, donnant le prix en $t = T$ de l'actif sous-jacent, de la façon suivante : étant donnée une suite de v.a. $Z_1, Z_2, \ldots, Z_n$ indépendantes et ayant la même loi que Z, S_T est égal à

$$ST = S_0 e^{Z_1\sigma\sqrt{\Delta t} + Z_2\sigma\sqrt{\Delta t} + \cdots + Z_n\sigma\sqrt{\Delta t}} = S_0 e^{\sigma\sqrt{T}\tilde{Z}_n}$$

ou $\tilde{Z}_n = \{\frac{1}{\sqrt{n}} \sum_{i=1}^{n} Z_i$. Donc si l'on pose $f(z) = (K - S_0 e^{\sigma\sqrt{T}z})^+$, la limite que l'on veut calculer $\lim_{n\to\infty} e^{-rT}\mathbb{E}((K - ST)^+)$ s'écrit simplement :

$$\lim_{n\to\infty} e^{-rT}\mathbb{E}((K - ST)^+) = \lim_{n\to\infty} e^{-rT}\mathbb{E}(f(\tilde{Z}_n)). \tag{3}$$

Où f est la fonction définie précédemment qui est à la fois continue et bornée. Nous allons voir que cette dernière propriété de f est importante et on peut noter qu'elle ne serait plus vraie pour un Call on'a le resultat suivant :

Proposition Soient T, r, σ des constantes strictement positives, $\delta t = \frac{T}{n}$, et R, d et u les quantités

$$R = e^{r\delta t}, \quad d = e^{-\sigma\sqrt{\delta t}} \quad \text{et} \quad u = e^{\sigma\sqrt{\delta t}}.$$

Soit $Z_1, Z_2, \ldots, Z_n$ une suite de v.a. independantes prenant deux valeurs -1 et $+1$ avec les probabilités $1-p$ et p respectivement, où p est la fonction de n donnée par

$$p = \frac{R-d}{u-d}.$$

Alors la suite $\tilde{Z}_n = \{\frac{1}{\sqrt{n}} \sum_{i=1}^{n} Z_i$ converge en loi vers une v.a. de la forme $\frac{\sqrt{T}}{\sigma}(r - \frac{\sigma^2}{2}) + \tilde{Z}_n$, ou $\tilde{Z}_n$ suit une loi normale centrée et réduite.

Preuve : admise vous pouvez le voir parmi la bibliographie

3.3.2 Convergence vers Black-Scholes :

Pour verifier que la limite du prix CRR est bien le prix Black-Scholes, il reste a vérifier que, si Z_0 est une v.a. normale centrée réduite et si $f(z) = (K - S_0 e^{\sigma\sqrt{T}z})^+$, on a bien :

$$e^{-rT}\mathbb{E}(f(Z_0)) = S_0 N(d_1) - Ke^{-rT} N(d_2)$$

Cela decoule du calcul suivant :

$$e^{-rT}E(f(Z_0)) = e^{-rT}\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}\left(K - S_0 e^{\sigma\frac{\sqrt{T}}{\sqrt{n}}(\sqrt{n}z+\frac{\sqrt{T}}{\sigma(r-\sigma^2)}\sqrt{n})}\right)^+\left(e^{-\frac{1}{2}z^2}\right)dz$$

On peut vérifier que la quantité $S_0 e^{\sigma\sqrt{n}z e^{T(r-\sigma^2/2)}} < K$ si et seulement si $z < -d_2$ et en faisant dans le second terme le changement de variable $y = z - \sigma\sqrt{T}$ pour lequel il est facile de voir que $z \in (-\infty, -d_2]$ si et seulement si $y \in (-\infty, -d_1]$. D'où on aura la formule cherchée :

$$e^{-rT}E(f(Z_0)) = e^{(}-rt)KN(-d_2) - S_0N(-d_1)$$

3.3.3 Vitesse de convergence

Sachant que la limite du prix Cox, Ross et Rubinstein est égale au prix Black-Scholes, on peut se demander comment se comporte le premier tend vers le second lorsque n tend vers l'infini. La convergence est-elle monotone, ou non, et surtout est-elle rapide ? Si l'on représente sur un graphe du prix $C_0(n)$ comme une fonction de n, on s'aperçoit (voir la figure ci-dessous) que la convergence est très irrégulière et qu'elle ne semble pas particulièrement rapide. En réalité, on peut montrer que :

$$C_0(n) - C_{BS} = \epsilon\left(\frac{1}{n}\right)$$

c'est-à-dire que cet écart tend vers zéro comme $\frac{1}{n}$, ce qui est assez rapide (dans le cas du théorème de la limite centrale, on attend plutôt une convergence en $\sqrt{\frac{1}{n}}$).

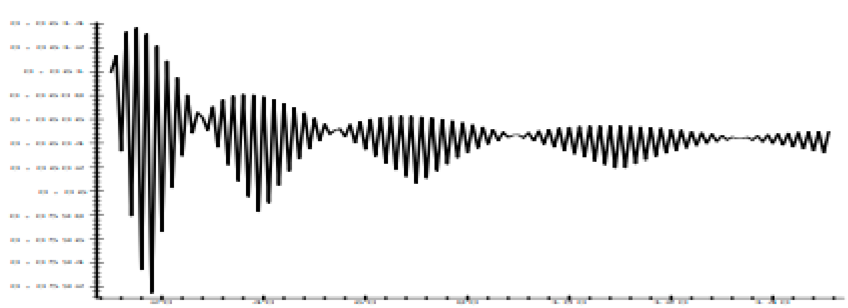

FIGURE 4 – Tracé du prix Cox, Ross et Rubinstein $C_0(n)$ d'un Call Européen en fonction de n et de sa limite, le prix Black-Scholes.

4 Application Numerique

Dans cette partie, nous allons voir comment calculer le prix d'un call européen d'une façon numérique. Nous allons utiliser le langage de programmation Python pour y succéder.

Le code que nous avons utilisé pour exécuter cette tâche est présent dans l'annexe de ce document.

Pour ce faire, nous allons d'abord montrer l'application numérique de l'algorithme de Cox-Ross-Rubistein :

Nous devons d'abord prendre en compte 5 paramètres dans notre modèle :

- $S0$: C'est le prix d'entrée de l'actif sous-jacent (par exemple : une action).
- K : Il s'agit ici du prix de l'exercice de l'option d'achat de call.
- T : C'est le temps jusqu'à l'échéance de l'option, qui est exprimé en années.
- r : C'est le taux d'intérêt sans risque, qui est exprimé en décimales.
- $sigma$: Il s'agit ici de la volatilité de l'actif sous-jacent, qui est une mesure de la fluctuation du prix de l'actif.
- N : C'est le nombre d'étapes dans l'arbre binomial utilisé lors de l'implémentation de l'algorithme de Cox-Ross-Rubistein. Il n'est pas utilisé pour celui de Black-Scholes.

Nous allons ensuite pouvoir définir le code suivant pour l'algorithme CCR (Cox-Ross-Rubistein) :

4.1 Algorithme CCR sur Python

Importons d'abord la bibliothèque `numpy`, elle nous servira à simplifier les calculs matriciels (on l'attribue à la variable `np`). Ensuite, nous allons définir les propriétés de l'arbre binomial

4.1.1 Élaboration de l'arbre binomial

D'abord, on calcule la durée d'une étape dans ce même arbre, en divisant le temps total jusqu'à l'échéance (T) par le nombre d'étapes (n) et on l'attribue à la variable `dt`. Ensuite on calcule le facteur d'augmentation des prix (attribué à la variable `u`) pour chaque étape de l'arbre. On utilise la volatilité de l'actif (sigma) et la durée d'une étape définie précédemment en `dt`. La fonction exponentielle (`np.exp`) et la racine carré (`np.sqrt`) sont utilisées par le raccourci de la bibliothèque `numpy`.

On calcule ensuite le facteur de diminution des prix (qu'on attribue à la variable `d`) pour chaque étape de l'arbre binomial en prenant l'inverse du facteur d'augmentation des prix `u`. À la fin, il nous reste plus qu'à définir la probabilité neutre au risque (en variable `p`) pour chaque étape de l'arbre binomial en utilisant le taux d'intérêt sans risque (r), la durée d'une étape (exprimée en `dt`), et les facteurs d'augmentation et de diminution des prix (`u` et `d`). Nous utilisons aussi la fonction exponentielle (`np.exp`) du raccourci de la bibliothèque `numpy` pour effectuer ce calcul.

Après avoir définie les propriétés de l'arbre, nous allons maintenant calculer le prix des options de l'échéance en utilisant cette arbre.

4.1.2 Élaboration des prix de l'actif sous-jacent

Pour cette partie du code, nous allons d'abord créer une matrice de taille $(N+1)$ x $(N+1)$ remplie de 0, qui représente l'arbre binomial des prix de l'actif. On a choisit cette taille car il y a n étapes dans ce même arbre, et que chaque étape a une position supplémentaire par rapport à l'étape précédente (d'où $N+1$). Ensuite on initialise la valeur du noeud racine de l'arbre binomial avec le prix actuel de l'actif (ici `S0`).

On créait ensuite une boucle `for` qui itère sur chaque étape de l'arbre binomial (de l'étape 1 à l'étape N). À l'intérieur de cette boucle, on calcule la valeur du noeud le plus à gauche de l'étape `i` en multipliant la valeur du noeud le plus à gauche de l'étape précédente $(i-1)$ par le facteur d'augmentation des prix (défini par `u` dans la précédente partie).

On recréait une autre boucle `for` qui itère sur chaque position j à l'intérieur de l'étape i. A l'intérieur de cette boucle, on calcule la valeur du noeud à la position j de l'étape i en multipliant la valeur du noeud à la position $(j-1)$ de l'étape précédente $(i-1)$ par le facteur de diminution des prix (défini dans la partie précédente par `d`).

Enfin, on calcule le prix de l'option d'achat européenne à l'échéance pour chaque noeud de la dernière étape de l'arbre binomial en soustrayant le prix de l'exercice (K) de la valeur de l'actif à chaque noeud. La fonction `np.maximum` est utilisée pour s'assurer que les prix des options sont au moins égaux à 0 (car les options d'achats ont un prix minimum de 0).

4.1.3 Retropropagation des prix des options

Nous allons dans cette partie rétropropager les prix des options à travers l'arbre binomial en utilisant les probabilités neutres au risque (`p`) et les facteurs d'actualisations. Pour se faire, nous allons créer une boucle `for` qui itère sur chaque étape l'arbre en ordre inverse, en commençant par l'étape $(n-1)$ et en remontant jusqu'à l'étape 0. L'argument `-1` dans la fonction `range()` indique que l'itération se fait en sens inverse. On créait ensuite une autre boucle `for` qui itère sur chaque position j à l'intérieur de l'étape i. Comme nous rétropropagons les prix des options à partir de l'étape précédente, il est nécessaire d'itérer sur toutes les positions jusqu'à celle de i.

Dans cette boucle, on va calculer le prix de l'option d'achat européenne à la position j de l'étape i en utilisant les probabilités neutres au risque (`p`) et les facteurs d'actualisation. La fonction exponentielle (marqué par `np.exp`) est utilisée pour calculer le facteur d'actualisation basé sur la taux d'intérêt sans risque (r) et la durée d'une étape (`dt`). Ensuite, le prix de l'option est actualisé en prenant la moyenne pondérée des prix des options aux positions j et $j + 1$ de l'étape suivante ($i + 1$), en utilisant les probabilités neutres au risque (`p` et leurs compléments (1-`p`)

4.1.4 Affichage du prix de l'action call européen

Notre modèle CRR a été élaborer sur Python, maintenant nous allons afficher les résultats qu'on obtient avec celui-ci. Pour ce faire, on va définir une variable `call_price` qui extrait le prix de l'option d'achat à partir du tableau `option_prices`. Comme mentionné précédemment, après avoir rétropropagé les prix des options à travers l'arbre binomial, le prix de l'option d'achat européenne aujourd'hui est stocké dans `option_prices[0]`. Le `0` ici fait référence à une valeur initiale, car il n'y a pas de valeur d'option à l'échéance pour le noeud racine de l'arbre.

On affiche ensuite le prix de l'option call européen à l'aide de la fonction `print`. Le texte "Prix du call européen :" est utilisé comme étiquette pour indiquer clairement que la valeur affichée est la prix de l'option d'achat européenne.

Voici donc comment calculer le prix d'achat d'un call européen grâce à l'algorithme CRR sur Python. Maintenant, on va voir comment calculer l'algorithme Black-Scholes (BS) sur ce même support.

4.2 Algorithme BS sur Python

Pour commencer, on va reprendre nos paramètres qu'on avait appliquée pour l'algorithme de CCR. Cependant, nous n'allons pas prendre le N car il ne prend pas en compte l'itération du nombre de pas.

D'abord, nous allons prendre les options `log`, `exp` et `sqrt` de la bibliothèque `numpy`. Elles servent à faciliter les calculs. Ensuite, on va importer la bibliothèque `scipy.stats`, elle sert à générer des fonctions de répartition cumulative (CDF) de loi normale standard.

On définit ensuite la fonction `bs_price` qui prend en paramètre S, K, r, T et $sigma$. Après avoir fait cela, on définit deux autres variables, `d1` et `d2` : `d1` est le facteur par lequel la valeur actuelle du reçu contingent dépasse le prix actuel de l'action, tandis ce que `d2` représente la probabilité ajustée au risque

que l'option soit exercée.

On définit ensuite le calcul du prix de l'option call. L'objectif de ce calcul est de définir `N(d1)` et `N(d2)`, c'est à dire les probabilités de `d1` et `d2`, d'où l'utilisation de la fonction `scipy.stats.norm.cdf`. En effet, pour rappel, la formule globale pour calculer le prix d'une option d'achat call européen de l'algorithme de BS est la suivante :

$$C = S \times \mathrm{N}(d_1) - K \times \exp(-r \cdot T) \times \mathrm{N}(d_2)$$

Tandis ce que la fonction de répartition cumulative de $N(d)$ est la suivante :

$$\mathrm{N}(d) = \frac{1}{2\pi} \int_{-\infty}^{d} e^{-\frac{t^2}{2}} \, dt$$

On retourne ensuite la valeur de l'option d'achat avec `call_price`. De ce fait, nous pouvons ensuite afficher la valeur de celle-ci avec `print` + la phrase "Prix du call européen". Voici comment on calcule donc le prix d'option d'achat call européen avec l'algorithme de BS.

4.3 Mise en place d'une simulation pour calculer le prix d'option d'achat

L'objectif ici est de montrer que l'algorithme de CRR converge vers le modèle BS. Pour ce faire, nous allons faire deux simulations différentes. En effet, nous allons changer les paramètres pour la première et la deuxième afin de voir si les algorithmes arrivent à gérer des valeurs différentes.

4.3.1 Simulation du premier modèle

Dans ce premier modèle, on va prendre ces valeurs pour les paramètres suivants :

- $S0$: 100
- K : 120
- T : 1
- r : 0.05
- $sigma$: 0.2
- N : 300 (car une valeur plus élevée rendrait le graphique illisible)

Pour mieux avoir une analyse précise sur l'objectif que l'on doit chercher, nous allons générer un graphique pour N tâtonnement sur Python. Il suffit de rajouter à notre code la bibliothèque `matplotlib.pyplot` qui prend en charge les graphiques sur Python et le code pour afficher un graphique. Pour avoir une meilleure visualisation, nous avons aussi calculer toutes les valeurs de pour chaque N, c'est à dire que si $N = 150$ par exemple, le plot affichera un graphique qui va de 0 à 150.

Plus N est grand, plus le temps de simulation sera élevé car l'algorithme doit calculer plus de valeurs. Ainsi, lorsque $N = 300$, on aura les résultat suivants pour CRR et BS :

Résultat CRR	3.2433496401240802
Résultat BS	3.2474774165608125

Les résultats numériques nous montrent que lorsque N=300, alors la valeur que donne l'algorithme CCR sera très proche de celle donnée par celui de BS.

Ce résultat peut-être complémenté par un graphique :

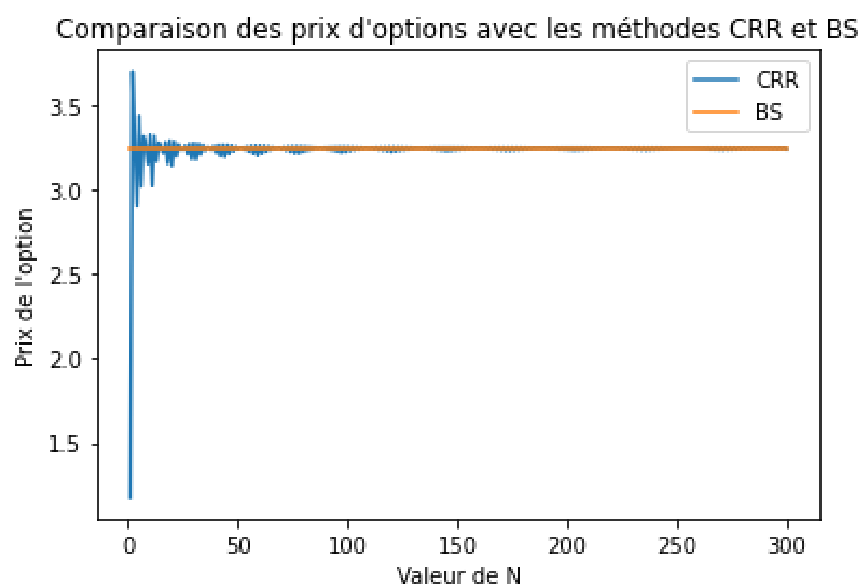

Figure 5 – Comparaison des prix d'options CRR et BS pour la simulation 1

Sur ce graphique, on peut voir que lorsque N a une petite valeur, le prix d'option pour l'algorithme CRR a une grand fluctuation comparé à celui de BS. Cependant, plus N à une grand valeur, plus cette variance à une valeur faible. Il arrive à un moment où la variance du modèle CRR se rapproche de 0 pour converger vers le modèle BS. Ce qui veut dire qu'en effet, le modèle Cox-Ros-Rubistein converge bien vers le modèle de Black-Scholes.

Pour prouver ce point, nous allons faire une seconde simulation avec d'autres valeurs pour les paramètres.

4.3.2 Simulation du second modèle

Pour cette seconde simulation, on va prendre des valeurs un plus plus complexe pour les paramètres :
— $S0$: 450
— K : 200
— T : 15
— r : 0.01
— $sigma$: 0.8
— N : 300
Nous obtenons les résultats numérique suivants :

Résultat CRR	417.2748831984571
Résultat BS	417.2962628576625

Les résultats numériques nous montrent encore qu'à $N = 300$, le résultat du modèle CCR est presque égale à celui de BS. Vient ensuite le graphique suivant :

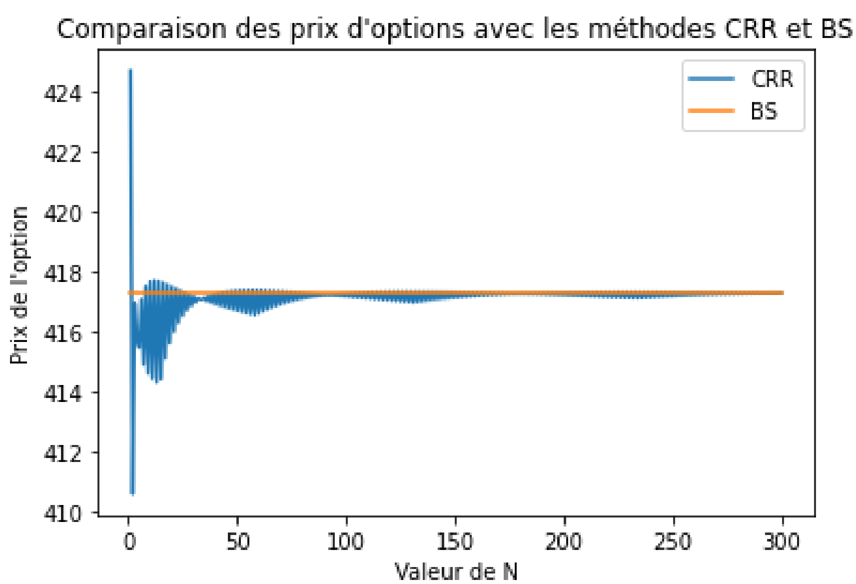

FIGURE 6 – Comparaison des prix d'options CRR et BS pour la simulation 2

Là encore, on peut voir qu'il y a une grande variance au début avec des écarts de plus d'une dizaine d'euros au niveau du prix de l'option. Si N devient de plus en plus grand, alors il y a aussi une convergence du modèle CCR vers le modèle BS.

Ainsi, pour conclure sur nos résultats, on peut voir qu'en effet le modèle Cox-Ross-Rubistein converge bien vers le modèle Black-Scholes quand $N \to +\infty$.

5 Autres applications

Dans cette dernière partie, nous allons voir si nous pouvons appliquer cette même logique à d'autres types d'options. En premier lieu, on va expliquer comment on pourrait prouver qu'un modèle CCR puisse converger vers un modèle BS avec des prix de vente put européen.

5.1 Application sur les prix de vente (put) européen

Il n'y a pas de grandes différences dans l'application numérique. En effet pour utiliser l'algorithme CRR avec les prix des options de vente, il faut changer la ligne qui calcule les prix des options en fonction des prix des actions et du prix d'exercice.

Pour l'option call, c'était `np.maximum(0, S[-1] -K)`. Pour les put, la formule est `np.maximum(0, K - S[-1])`. La différence entre les deux formules réside dans la manière dont les profits sont calculés pour les options call et put. Pour les options call, le profit est la différence entre le prix de l'actions et le prix d'exercice, tandis que pour les options put, c'est l'inverse, c'est-à-dire la différence entre le prix d'exercice et le prix de l'action.

On doit aussi faire un autre changement dans la fonction `bs_price`. En effet, la formule du prix du call européen est `S0 * scipy.stats.norm.cdf(d1) - K * exp(-r * T) * scipy.stats.norm.cdf(d2)`. Pour calculer le prix du put européen, on utilise la formule suivante : `-S0 * scipy.stats.norm.cdf(-d1) + K * exp(-r * T) * scipy.stats.norm.cdf(-d2)`. La différence entre les deux formules provient du fait que pour les options put, on utilise les valeurs négatives de `d1` et `d2` dans la fonction de répartition cumulative de la loi normale. Cette modification reflète la différence dans la calcul des profits pour les options call et put, comme expliqué précédemment.

On voit donc qu'il y a peu de différence pour calculer les call et put européens. Maintenant que nous avons vu cela, nous allons appliquer cette fois-ci le même principe sur les options composés.

5.2 Application sur les options composés

Nous allons d'abord définir qu'est ce qu'une option composée : c'est une option financière dont le sous-jacent est une autre option. Il s'agit donc d'une option sur option. Les options composées peuvent être des options d'achat (call) ou de vente (put) européens. Cependant, il est à noter que les options composées ne peuvent pas être utiliser avec le modèle BS car il n'existe pas de formule analytique directe dans ce modèle, mais nous pouvons utiliser une autre approche numérique, comme la méthode de Monte-Carlo (nous allons uniquement faire le calcule pour le modèle CRR).

D'abord, nous allons adapter la fonction `crr_price` pour qu'elle prenne deux arguments supplémentaires, `option_type` qui correspond au type d'option externe (l'option composée d'elle même) et `inner_option_type` (l'option sous-jacente) qui correspond au prix des options internes. Ces arguments peuvent prendre les valeurs "call" ou "put".

Les changements vont être affectés dans une nouvelle fonction se nommant `crr_compound_price` qui va calculer d'abord le prix des options internes en utilisant une approche similaire à celle de la fonction `crr_price`. Ensuite, on utilise les prix d'options internes comme actifs sous-jacents pour calculer les prix des options composées.

Conclusion

Pour conclure, nous avons pu voir comment les modèles CCR et BS étaient appliqués mathématiquement et numériquement. Les deux modèles sont bons, cependant ils peuvent contenir des limites. Par exemple, le modèle BS a été développé à partir d'une série d'hypothèses strictes, notamment l'absence de frais de transaction et de dividende, ainsi que la constance du taux d'intérêt et de la volatilité. Ce fait fait, ce modèle tend à être moins précis lorsque l'on s'éloigne des hypothèses strictes, et peut nécessiter des ajustements pour mieux refléter les conditions du marché.

Le modèle CCR, en revanche, peut-être plus flexible grâce à son approche de construction d'un arbre binomial qui permet de simuler les fluctuations possibles du prix de l'actif sous-jacent (notamment les variations de la volatilité et des taux d'intérêt). Néanmoins, ce modèle peut-être plus complexe à calculer, notamment lorsqu'il s'agit de considérer plusieurs facteurs de risque simultanément.

Pour la répartition du travail, Mamadou s'est occupé de faire tout ce qui touche aux calculs mathématiques et l'introduction des deux modèles. Quant à Jérémy, il s'est occupé de l'application numérique et de l'analyse des résultats pour prouver qu'il y a bien une convergence entre du modèle CCR vers le modèle BS.

Références

[1] Diener, M. (2013). Cours de calcul stochastique. Université Nice Sophia Antipolis. Récupéré sur https ://math.unice.fr/ diener/L3MASS13/CalSto13cours10.pdf

[2] Investopedia. (2022). Binomial Option Pricing. Récupéré sur https ://www.investopedia.com/terms/b/binomialoptionpricing.asp

[3] Germain, P. (2018). Introduction à Python. Cours donné à l'École des Neurosciences de Paris Île-de-France. Récupéré sur http ://chercheurs.lille.inria.fr/pgermain/neurones2018/coursIntroPython.pdf

[4] Pythoninoffice. (2022). Calculate Black Scholes option price in Python. Récupéré sur https ://pythoninoffice.com/calculate-black-scholes-option-price-in-python/

[5] Cox, J. C., Ross, S. A., Rubinstein, M. (1979). Option Pricing : A Simplified Approach. Journal of Financial Economics, 7(3), 229-263. Récupéré sur http ://argento.bu.edu/hes/cox79py538.pdf

[6] Importq. (2016). Option Pricing in Python : Cox-Ross-Rubinstein. Récupéré sur https ://importq.wordpress.com/2016/07/02/option-pricing-in-python-cox-ross-rubinstein/

[7] Black-Scholes comme limite de CRR. Récupéré sur https ://math.unice.fr/ diener/L3MASS08/chap8.pdf

[8] https ://archipel.uqam.ca/8952/1/M14463.pdf