Modèle de Cox-Ross-Rubinstein

Compte rendu du TP - MAP552

Alexis VO

Université Paris-Saclay

École polytechnique

June 3, 2025

Contents

Toutes les implémentations sont en Python, avec les librairies NumPy, Scipy et Matplotlib.

1 Modèle binomial à n périodes

On considère un actif avec prix initial S_0, maturité T, drift μ, volatilité σ, et n périodes.
Le pas de temps est : $h_n = \frac{T}{n}$
Les coefficients d'évolution sont définis par :

$$u_n = e^{\mu h_n + \sigma\sqrt{h_n}}$$

$$d_n = e^{\mu h_n - \sigma\sqrt{h_n}}$$

Les prix possibles à la date j sont donnés par :

$$\boxed{S_j^{(i)} = S_0 \cdot u^{j-i} d^i, \quad \text{pour } i = 0, 1, \ldots, j \text{ et } S_0 > 0}$$

Implémentation :

```
def Sn(T, n, mu, sigma, j):
    h = T / n
    u = np.exp(mu * h + sigma * np.sqrt(h))
    d = np.exp(mu * h - sigma * np.sqrt(h))
    return np.array([S0 * u**(j - i) * d**i for i in range(j + 1)])
```

2 Option européenne de type call

Le gain à la maturité pour une option call de strike K est :

$$\boxed{\text{Payoff}(i) = \max(S_n^{(i)} - K, 0)}$$

Implémentation :

```
def Payoff(T, n, mu, sigma, K):
    prices = Sn(T, n, mu, sigma, n)
    return np.maximum(prices - K, 0)
```

3 Évaluation backward : prix de l'option

On utilise la *probabilité risque-neutre* : $q = \frac{e^{rh} - d}{u - d}$
La formule backward est :

$$\boxed{V_j^{(i)} = e^{-rh}\left(qV_{j+1}^{(i)} + (1-q)V_{j+1}^{(i+1)}\right)}$$

Pour avoir l'intuition de cette formule, on peut la voir comme une moyenne pondérée des valeurs futures de l'option, actualisée au temps présent. On estime la valeur actuelle $V_j^{(i)}$ comme la valeur espérée sous mesure risque-neutre des deux états possibles au pas suivant :

- montée : $V_{j+1}^{(i)}$ avec probabilité q
- descente : $V_{j+1}^{(i+1)}$ avec probabilité $1 - q$

Puis on actualise à aujourd'hui avec le facteur e^{-rh}.
Remarque : la probabilité risque neutre q est calculée pour s'assurer que le modèle respecte l'absence d'arbitrage.

Implémentation :

```
def Calln(T, n, r, mu, sigma, K):
    h = T / n
    u = np.exp(mu * h + sigma * np.sqrt(h))
    d = np.exp(mu * h - sigma * np.sqrt(h))
    q = (np.exp(r * h) - d) / (u - d)
    V = Payoff(T, n, mu, sigma, K)
    for j in range(n - 1, -1, -1):
        V = np.exp(-r * h) * (q * V[:-1] + (1 - q) * V[1:])
    return V[0]
```

4 Delta-hedging

La couverture à la date j est :

$$\boxed{\Delta_j^{(i)} = \frac{V_{j+1}^{(i)} - V_{j+1}^{(i+1)}}{S_{j+1}^{(i)} - S_{j+1}^{(i+1)}}}$$

où l'on a :

- $V_{j+1}^{(i)}$: valeur de l'option à la date $j+1$ dans l'état i
- $V_{j+1}^{(i+1)}$: valeur de l'option à la date $j+1$ dans l'état $i+1$
- $S_{j+1}^{(i)}$: prix de l'actif sous-jacent à la date $j+1$ dans l'état i
- $S_{j+1}^{(i+1)}$: prix de l'actif sous-jacent à la date $j+1$ dans l'état $i+1$

Cette formule nous donne la quantité d'actif à détenir pour couvrir l'option à la date j. Elle est calculée telle que

$$\Delta = \frac{\text{variation de l'option}}{\text{variation de l'actif}}$$

Implémentation :

```
def Deltan(T, n, r, mu, sigma, K, j):
    ...
    delta = (V_up - V_down) / (Sj1[:-1] - Sj1[1:])
    return delta
```

5 Comparaison avec Black-Scholes

Après avoir étudié la valorisation d'une option européenne dans le cadre discret du modèle binomial de Cox-Ross-Rubinstein, on va désormais s'intéresser à sa version continue : la formule de Black-Scholes. Cette dernière est la limite du modèle CRR quand le nombre de pas tend vers l'infini. Elle calcule directement le prix de l'option en fonction des paramètres du marché, sans avoir besoin de construire un arbre binomial comme dans le modèle CRR.

Formule de Black-Scholes :

$$\boxed{C = S_0\Phi(d_1) - Ke^{-rT}\Phi(d_2)}$$

avec :

$$d_1 = \frac{\ln(S_0/K) + (r + \frac{\sigma^2}{2})T}{\sigma\sqrt{T}}, \quad d_2 = d_1 - \sigma\sqrt{T}$$

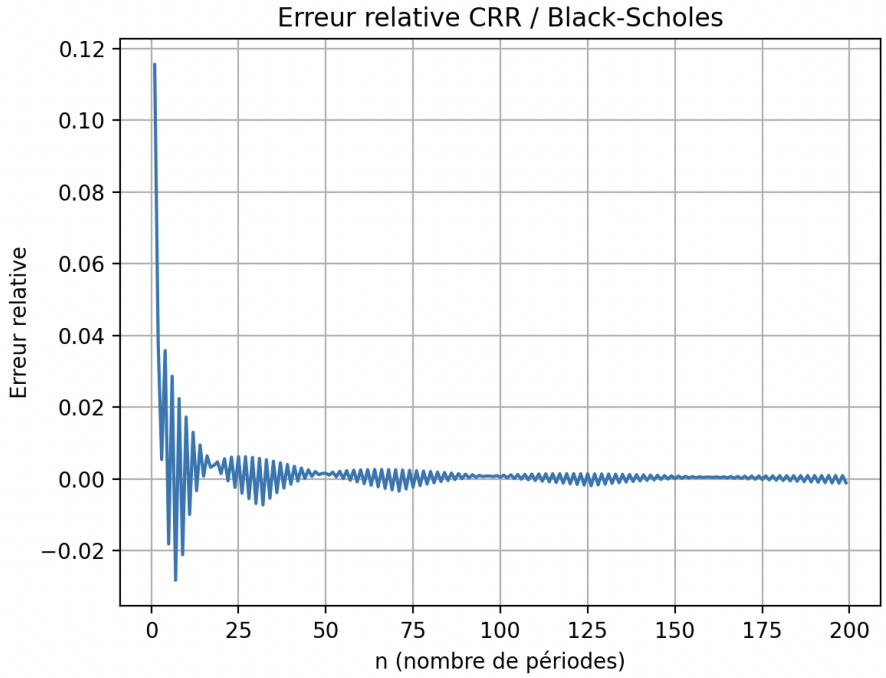
Implémentation :

```
def Call(T, r, sigma, K):
    d1 = (np.log(S0 / K) + (r + sigma**2 / 2) * T) / (sigma * np.sqrt(T))
    d2 = d1 - sigma * np.sqrt(T)
    return S0 * norm.cdf(d1) - K * np.exp(-r * T) * norm.cdf(d2)
```

6 Écart relatif et convergence

$$\boxed{\text{erreur}(n) = \frac{\text{Calln}(n)}{\text{Call}} - 1}$$

Graphe de l'erreur relative en fonction du nombre de périodes n



Interprétation : on observe que l'erreur relative diminue lorsque le nombre de périodes n augmente, ce qui confirme la convergence du modèle binomial vers la formule de Black-Scholes. Le graphe nous montre par ailleurs une décroissance rapide pour n faible, puis un palier asymptotique vers 0 pour des valeurs plus élevées de n. La convergence est donc plus lente. On peut conclure que l'arbre binomial reste fiable pour des valeurs de n relativement faibles.