Simulation MONTE CARLO

Encadré par: Mr.EL ASRI
Réalisé par: Imane DIBE

Contents

1 Introduction

1.1 Définition de Monte Carlo

La méthode Monte-Carlo, est une méthode algorithmique visant à calculer une valeur numérique approchée en utilisant des procédés aléatoires, c'est-à-dire des techniques probabilistes. Il est particulièrement utilisée pour calculer des intégrales en dimensions plus grandes que 1. Elles sont également couramment utilisées en physique des particules, où des simulations probabilistes permettent d'estimer la forme d'un signal ou la sensibilité d'un détecteur.

La méthode de simulation de Monte-Carlo permet aussi d'introduire une approche statistique du risque dans une décision financière. Elle consiste à isoler des variables clés du projet, telles que le chiffre d'affaires ou la marge, et à leur affecter une loi de probabilités

1.2 Monte Carlo en finance

MC est largement utilisé en ingénierie financière pour la prévision boursière. Cela a un sens intuitif, car le marché est impossible à modéliser, a une dimensionnalité incroyablement élevée et dispose de nombreuses données à échantillonner. L'importance du risque est un autre facteur important pour lequel les analystes financiers utilisent les méthodes de MC. Une application relativement simple de Monte Carlo dans ce domaine est l'optimisation de portefeuille.

2 Exemples de Monte Carlo

2.1 Simulation de loi normale

Théoriquement:

La méthode de Monte Carlo repose sur le principe suivant : au lieu de tirer des échantillons directement de la distribution normale, nous générons des échantillons à partir d'une distribution uniforme et utilisons une transformation pour les convertir en échantillons de la distribution normale. les étapes suivie:

1. Nous générons n échantillons aléatoires Ui à partir d'une distribution uniforme sur l'intervalle [0, 1]. C'est-à-dire que chaque échantillon est une valeur aléatoire comprise entre 0 et 1.

2. Nous utilisons une fonction inverse de la fonction de répartition cumulative de la distribution normale :
$$F^{-1}(u) = \mu + \sigma\sqrt{2}\mathrm{erf}^{-1}(2u - 1) \tag{1}$$
pour convertir les échantillons uniformes en échantillons de la distribution normale Xi. Cette fonction inverse est calculée à partir de la fonction de répartition cumulative de la distribution normale.

3. Les paramètres de la distribution normale, tels que la moyenne (μ) et l'écart-type (σ), sont utilisés pour ajuster les échantillons uniformes transformés afin de correspondre à la distribution normale spécifiée.

4. Une fois que les échantillons uniformes sont transformés et ajustés, ils représentent des échantillons de la distribution normale avec les paramètres spécifié.

5. Enfin, nous répétons ces étapes pour générer autant d'échantillons normalement distribués que nécessaire pour notre simulation.

<u>*Pratiquement:*</u>

```python
import numpy as np
import matplotlib.pyplot as plt

# Fonction de simulation de la loi normale avec Monte Carlo
def monte_carlo_normal(m , std , n):
    echantillons = np.random.normal(m , std , n)
    return echantillons

# Simulation
echantillons = monte_carlo_normal(0 , 1, 10000)

# Affichage de l'histogramme des échantillons
plt.hist(echantillons, bins=50, density=True, color='blue', edgecolor='black', alpha=0.5)
plt.xlabel('Valeurs')
plt.ylabel('Densité de probabilité')
plt.title('Simulation de loi normale avec Monte Carlo')
plt.grid(True)
plt.show()
```

Figure 1: Le code python de loi normale

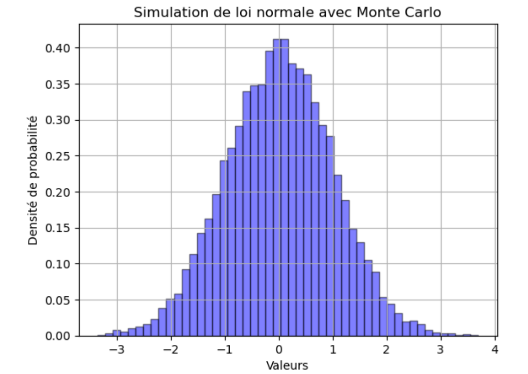

Figure 2: visualisation de loi normale simulée

2.2 Mouvement Brownien

<u>Définition 1:</u>Le mouvement brownien, également connu sous le nom de mouvement brownien standard, est un concept fondamental en mathématiques et en finance. Il a été initialement observé par le botaniste écossais Robert Brown au début du XIXe siècle lorsqu'il a observé le mouvement aléatoire des particules de pollen dans l'eau. Ce phénomène a été formalisé mathématiquement par le mathématicien français Louis Bachelier et développé plus tard par Albert Einstein. La description physique la plus élémentaire de ce phénomène est la suivante :

- entre deux chocs, la grosse particule se déplace en ligne droite avec une vitesse constante.
- la grosse particule est accélérée lorsqu'elle rencontre une molécule de fluide ou une paroi.

<u>Définition 2:</u> Le mouvement brownien est un processus stochastique continu caractérisé par deux propriétés principales :

1. Indépendance et incrément stationnaire : Les incréments du mouvement brownien sur des intervalles de temps non chevauchants sont indépendants les uns des autres, et la distribution des incréments est stationnaire.
2. Trajectoire continue mais non dérivable : Le mouvement brownien a une trajectoire continue mais non dérivable. Cela signifie que la trajectoire du mouvement brownien est continue dans le temps, mais sa dérivée n'existe pas en tout point.

<u>*Théoriquement:*</u>
Mathématiquement, le mouvement brownien est défini par les caractéristiques suivantes :

- À chaque instant t, le mouvement brownien $W(t)$ suit une distribution normale avec une moyenne de 0 et une variance de t : $W(t) \sim \mathcal{N}(0, t)$.
- Les incréments du mouvement brownien sur des intervalles de temps $[s, t]$ sont distribués normalement avec une moyenne de 0 et une variance de $t - s$.
- Le mouvement brownien a une trajectoire continue presque sûrement mais non dérivable.

Pratiquement:

```python
import numpy as np
import matplotlib.pyplot as plt

# Fonction pour simuler un mouvement brownien standard
def standard_brownien(dt, n, p):
    # p: le pas
    t = np.linspace(0, n*dt, num=n)
    W = np.random.standard_normal(size=(p, n))
    W = np.cumsum(W, axis=1) * np.sqrt(dt)  # Mouvement Brownien
    return t, W

dt=0.01
n=1000
p=1000

# Simulation du mouvement brownien standard
t, W = standard_brownien(dt , n , p)

# Affichage des trajectoires
for i in range(p):
    plt.plot(t, W[i])
plt.xlabel('Temps')
plt.ylabel('Position')
plt.title('Mouvement Brownien Standard')
plt.grid(True)
plt.show()
```

Figure 3: Le code python du mouvement brownien

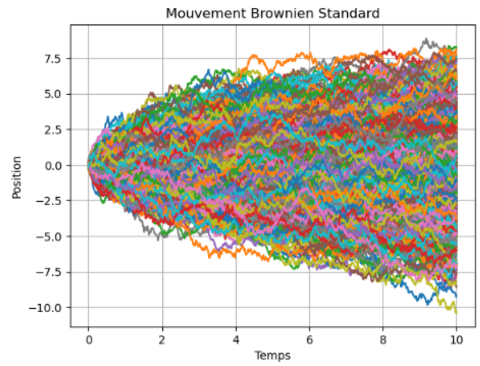

Figure 4: La visualisation du mouvement brownien

2.3 Mouvement Brownien Géometrique

Définition:Le mouvement brownien géométrique est un processus stochastique continu qui modélise l'évolution d'un actif financier, tel qu'une action, dans le temps. Contrairement au mouvement brownien standard, où les variations de prix sont normalement distribuées, le mouvement brownien géométrique assume que les rendements de l'actif sont log-normalement distribués.

Théoriquement:

Mathématiquement, le mouvement brownien géométrique est décrit par l'équation différentielle stochastique suivante : $dS = \mu S dt + \sigma S dW$

- S représente le prix de l'actif à un instant t.
- μ est le taux de rendement moyen de l'actif.
- σ est la volatilité de l'actif.
- dt est un intervalle de temps infinitésimal.
- dW est l'incrément d'un mouvement brownien standard.

- La solution de cette équation différentielle stochastique est :

$$S(t) = S(0) \exp\left((\mu - \frac{1}{2}\sigma^2)t + \sigma W(t)\right)$$

- $S(0)$ est le prix initial de l'actif à $t = 0$.
- $W(t)$ est le mouvement brownien standard.

Pratiquement:

```python
import numpy as np
import matplotlib.pyplot as plt

# Fonction pour simuler un mouvement brownien géométrique
def brownien_geometrique(mu, sigma, S0, dt, n, p):
    t = np.linspace(0, n*dt, num=n)
    W = np.random.standard_normal(size=(p , n))
    W = np.cumsum(W, axis=1) * np.sqrt(dt)  # Mouvement Brownien
    X = (mu - 0.5 * sigma**2) * t + sigma * W
    S = S0 * np.exp(X)  # Mouvement Brownien Geometrique
    return t, S

# Paramètres
S0 = 100
mu = 0.2
sigma = 0.3
dt = 0.01
n = 1000
p = 1000

# Simulation du mouvement brownien géométrique
t, S = brownien_geometrique(mu, sigma, S0, dt, n, p)

# Affichage des trajectoires
for i in range(p):
    plt.plot(t, S[i])
plt.xlabel('Temps')
plt.ylabel('Prix')
plt.title('Mouvement Brownien Géométrique')
plt.show()
```

Figure 5: Le code python du mouvement brownien géométrique

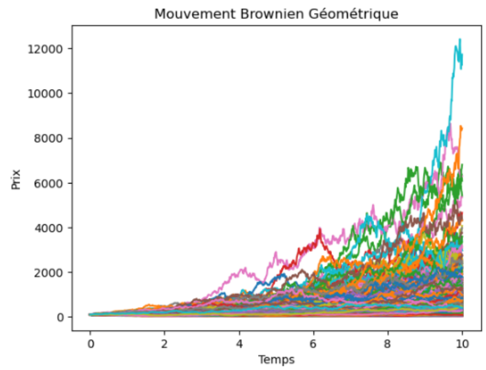

Figure 6: La visualisation du mouvement brownien géométrique

2.4 Mouvement Brownien Arithmétique

Définition:Le mouvement brownien arithmétique est un autre type de processus stochastique utilisé en finance pour modéliser l'évolution des prix des actifs financiers. Contrairement au mouvement brownien géométrique, où les variations de prix sont proportionnelles au prix lui-même, le mouvement brownien arithmétique modélise les variations de prix de manière additive.

Théoriquement:

Mathématiquement, le mouvement brownien arithmétique est décrit par l'équation différentielle stochastique suivante : $dS = \mu dt + \sigma dW$

- S représente le prix de l'actif à un instant t.
- μ est le taux de rendement moyen de l'actif.
- σ est la volatilité de l'actif.
- dt est un intervalle de temps infinitésimal.
- dW est l'incrément d'un mouvement brownien standard.

• La solution de cette équation différentielle stochastique est :

$$S(t) = S(0) + \mu t + \sigma W(t)$$

- $S(0)$ est le prix initial de l'actif à $t = 0$.
- $W(t)$ est le mouvement brownien standard.

Pratiquement:

```python
import numpy as np
import matplotlib.pyplot as plt

# Fonction pour simuler un mouvement brownien arithmétique
def brownien_arithmetique(mu, sigma, S0, dt, n, p):
    t = np.linspace(0, n*dt, num=n)
    W = np.random.standard_normal(size=(p, n))
    W = np.cumsum(W, axis=1) * np.sqrt(dt)  # Mouvement Brownien
    X = (mu - 0.5 * sigma**2) * t + sigma * W
    S = S0 + X  # Mouvement Brownien Arithmetic
    return t, S

# Paramètres
S0 = 100
mu = 0.1
sigma = 0.2
dt = 0.01
n = 1000
p = 1000

# Simulation du mouvement brownien arithmétique
t, S = brownien_arithmetique(mu, sigma, S0, dt, n, p)

# Affichage des trajectoires
for i in range(p):
    plt.plot(t, S[i])
plt.xlabel('Temps')
plt.ylabel('Prix')
plt.title('Mouvement Brownien Arithmétique')
plt.show()
```

Figure 7: Le code python du mouvement brownien arithmétique

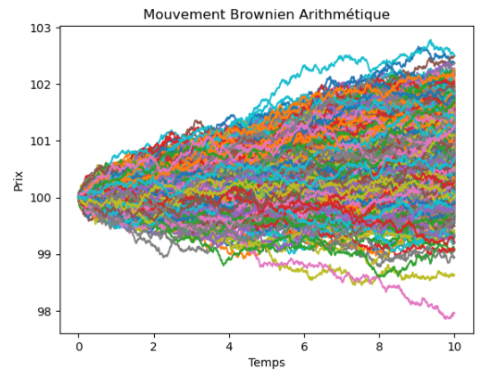

Figure 8: La visualization du mouvement brownien arithmétique

3 Modèle de Black Scholes

3.1 Fondements théoriques

3.1.1 Origines et contexte historique du modèle

Le modèle Black-scholes est une formule mathématique qui peut être utilisée pour calculer la juste valeur d'une option, qui est un contrat qui donne à l'acheteur le droit, mais non l'obligation, d'acheter ou de vendre un actif sous-jacent à un prix spécifié et temps. Le modèle a été développé par deux économistes, Fischer Black et Myron Scholes, en 1973, puis étendu par Robert Merton.

3.1.2 Hypothèses sous-jacentes du modèle

Le modèle repose sur plusieurs hypothèses, telles que :

1. L'option est européenne, c'est-à-dire qu'elle ne peut être exercée qu'à l'échéance.
2. L'actif sous-jacent suit un mouvement brownien géométrique, ce qui signifie que ses changements de prix sont aléatoires et normalement distribués, avec une dérive et une volatilité constantes.
3. Le taux d'intérêt sans risque et la volatilité de l'actif sous-jacent sont connus et constants.
4. Il n'y a pas de frais de transaction, de taxes, de dividendes ou d'opportunités d'arbitrage.
5. Le marché est efficace et liquide.

3.2 Formulation mathématique

3.2.1 Équation de Black-Scholes originale

L'équation de Black-Scholes est une équation aux dérivées partielles (EDP) qui décrit le comportement du prix d'une option au fil du temps. Elle est basée sur plusieurs hypothèses clés, notamment l'absence d'opportunités d'arbitrage, l'efficience des marchés, et le mouvement brownien géométrique.

L'équation de Black-Scholes pour le prix $V(S,t)$ d'une option européenne est donnée par :

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0$$

où :

- $\frac{\partial V}{\partial t}$ est la dérivée partielle du prix de l'option par rapport au temps t.
- $\frac{\partial^2 V}{\partial S^2}$ est la dérivée seconde du prix de l'option par rapport au prix de l'actif sous-jacent S.
- S est le prix de l'actif sous-jacent.
- t est le temps.
- σ est la volatilité de l'actif sous-jacent.
- r est le taux d'intérêt sans risque.

Cette équation exprime l'équilibre entre le flux de gains anticipés sur l'option (le terme $\frac{\partial V}{\partial t}$) et le coût d'opportunité associé à la détention de l'option (le terme $-rV$).

3.2.2 Développement et justification des équations différentielles stochastiques

La justification des équations différentielles stochastiques (EDS) utilisées dans le modèle de Black-Scholes repose principalement sur deux hypothèses : l'absence d'opportunités d'arbitrage et l'hypothèse de rendement constant. Ces hypothèses conduisent à l'utilisation du mouvement brownien géométrique dans le modèle.

Le processus stochastique qui décrit l'évolution du prix de l'actif sous-jacent S dans le modèle de Black-Scholes est décrit par l'équation différentielle stochastique suivante :

$$dS = rSdt + \sigma SdW$$

où :

- r est le taux d'intérêt sans risque.
- σ est la volatilité de l'actif sous-jacent.
- dW est l'accroissement élémentaire du mouvement brownien standard.

Cette équation décrit le mouvement du prix de l'actif sous-jacent en fonction du taux d'intérêt sans risque, de la volatilité et du mouvement brownien, qui représente le facteur de risque aléatoire.

3.2.3 Dérivation des formules pour le prix d'une option d'achat (call) et d'une option de vente (put)

En utilisant l'équation de Black-Scholes et en appliquant les conditions aux limites appropriées (par exemple, les conditions d'exercice à l'expiration), on peut dériver les formules pour le prix d'une option d'achat (call) et d'une option de vente (put). Ces formules sont données par :

Pour une option d'achat (call) :

$$C(S,t) = SN(d_1) - Ke^{-r(T-t)}N(d_2)$$

Pour une option de vente (put) :

$$P(S,t) = Ke^{-r(T-t)}N(-d_2) - SN(-d_1)$$

où :

- $C(S,t)$ est le prix de l'option d'achat (call).
- $P(S,t)$ est le prix de l'option de vente (put).
- K est le prix d'exercice de l'option.
- $N(\cdot)$ est la fonction de répartition cumulative de la loi normale standard.

- d_1 et d_2 sont définis comme suit :

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

avec T représentant la maturité de l'option.

Ces formules donnent le prix théorique des options d'achat et de vente en fonction du prix de l'actif sous-jacent, du prix d'exercice, du taux d'intérêt sans risque, de la volatilité et du temps restant jusqu'à l'expiration de l'option.

4 Interface utilisateur

Afin d'améliorer l'expérience utilisateur et de faciliter l'interaction avec notre modèle, nous avons développé une interface graphique utilisant **Streamlit**. Cette interface permet aux utilisateurs de visualiser et d'analyser les résultats des simulations en temps réel, tout en bénéficiant d'une navigation intuitive et d'une personnalisation avancée.

L'interface comprend plusieurs sections accessibles via un menu latéral interactif. Les utilisateurs peuvent explorer différentes fonctionnalités telles que la loi normale, les processus de mouvement brownien (classique, géométrique et arithmétique), ainsi que l'évaluation des options financières. De plus, un thème personnalisé a été implémenté afin d'offrir une expérience plus ergonomique, avec des couleurs harmonieuses et des icônes adaptées à chaque module.

L'interface est accessible via le lien suivant : https://project-monte-carlo-2023.streamlit.app/.

Figure 9: Interface utilisateur

5 Conclusion

Ce projet a permis de mettre en place un cadre méthodologique robuste pour l'analyse et la modélisation des processus stochastiques appliqués à l'évaluation des options financières. En combinant des approches théoriques et des implémentations numériques, nous avons pu simuler et comparer différentes méthodes d'évaluation, notamment à travers les modèles de **mouvement brownien** et de **mouvement brownien géométrique**.

L'intégration d'une **interface utilisateur** via **Streamlit** a permis de rendre ces modèles interactifs et accessibles, offrant aux utilisateurs une visualisation intuitive des simulations et des résultats obtenus. Cette interface constitue un outil efficace pour explorer les comportements des options en fonction des paramètres du marché.

En perspective, des améliorations peuvent être envisagées, notamment l'intégration d'autres modèles d'évaluation plus sophistiqués, tels que les méthodes numériques avancées (Monte-Carlo, différences finies), ainsi que l'optimisation de l'interface pour une meilleure ergonomie et une plus grande flexibilité dans l'analyse des options.

Ce travail ouvre ainsi la voie à une meilleure compréhension des dynamiques de marché et à une utilisation plus intuitive des modèles mathématiques en finance quantitative.