# Concepts Mathématiques en Finance Quantitative

Alexis Fabre

July 20, 2025

## 1. Mouvement Brownien

**Formule :**

$$B(t + \Delta t) = B(t) + \epsilon \cdot \sqrt{\Delta t}$$

**Définitions :**

- $B(t)$ : valeur du processus à l'instant $t$
- $\epsilon$ : bruit aléatoire standard normal (i.e., $\epsilon \sim \mathcal{N}(0, 1)$)
- $\Delta t$ : petit incrément de temps

**Utilité :** Ce processus sert de fondation pour modéliser les mouvements aléatoires dans le temps, comme ceux d'un actif financier.

**Code Python :**

```
B = np.zeros(T)
for t in range(1, T):
    B[t] = B[t-1] + np.random.normal(0, np.sqrt(dt))
```

## 2. Mouvement Brownien Géométrique (GBM)

**Formule :**

$$S_{t+1} = S_t \cdot \exp\left[(\mu - \frac{\sigma^2}{2}) \cdot dt + \sigma \cdot \sqrt{dt} \cdot Z\right]$$

**Définitions :**

- $S_t$ : prix de l'actif à l'instant $t$
- $\mu$ : rendement moyen
- $\sigma$ : volatilité
- $dt$ : incrément de temps
- $Z$ : variable aléatoire gaussienne standard ($\sim \mathcal{N}(0, 1)$)

**Utilité :** Utilisé pour simuler les trajectoires de prix d'un actif financier.

**Code Python :**

```
S[t] = S[t-1] * np.exp((mu - 0.5 * sigma**2) * dt + sigma * np.sqrt(dt) * np.random.normal())
```

## 3. Simulation de Monte Carlo

**Concept :** On répète la simulation d'un processus aléatoire (comme un GBM) de nombreuses fois pour approximer la distribution future d'un actif.

**Code Python :**

```
for i in range(num_simulations):
    for t in range(1, num_days):
        Z = np.random.standard_normal()
        simulations[t, i] = simulations[t-1, i] * np.exp((mu - 0.5 * sigma**2) *
        dt + sigma * np.sqrt(dt) * Z)
```

## 4. Processus d'Ornstein-Uhlenbeck

**Formule :**

$$dX_t = \theta(\mu - X_t)dt + \sigma dW_t$$

**Définitions :**

- $X_t$ : valeur du processus à l'instant $t$
- $\mu$ : valeur moyenne autour de laquelle le processus revient
- $\theta$ : vitesse de réversion
- $\sigma$ : volatilité
- $dW_t$ : incrément d'un mouvement brownien

**Utilité :** Modélise des variables qui reviennent vers une moyenne (ex: taux d'intérêt).
**Code Python :**

```
X[t] = X[t-1] + theta * (mu - X[t-1]) * dt + sigma * np.sqrt(dt) * np.random.normal()
```

## 5. Vraisemblance (Likelihood)

**Formule :**

$$L(\theta) = \prod_{i=1}^{n} P(x_i \mid \theta) \quad \text{ou en log:} \quad \log L(\theta) = \sum_{i=1}^{n} \log P(x_i \mid \theta)$$

**Définitions :**

- $x_i$ : observations
- $\theta$ : paramètre du modèle
- $P(x_i \mid \theta)$ : probabilité de $x_i$ donnée $\theta$

**Utilité :** Permet de déterminer le paramètre le plus probable ayant généré les données observées.
**Code Python :**

```
def log_likelihood(theta, data):
    mu, sigma = theta
    return np.sum(norm.logpdf(data, mu, sigma))
```