

Finance Quantitative

Notes de Cours

Master M2 — 2025–2026

Yaë Ulrich Gaba

*« En matière de finance, il ne s'agit pas
de prédire l'avenir mais de s'y préparer. »*

— Pierre-Simon Laplace

Table des matières

Préface	1
1 Marchés Financiers — Modélisation	7
1.1 Introduction aux marchés financiers	7
1.1.1 Classes d’actifs	7
1.1.2 Produits dérivés fondamentaux	8
1.2 Modélisation en temps discret	8
1.2.1 Modèle à une période	8
1.2.2 Probabilité risque-neutre	9
1.3 Modèle binomial multi-périodes (CRR)	9
1.3.1 Calibration des paramètres CRR	10
1.4 Théorèmes fondamentaux de l’évaluation d’actifs	10
1.5 Réplication et couverture	10
1.6 Implémentation Python	10
1.7 Parité call-put	11
1.8 Hypothèses de marché	12
1.9 Faits stylisés des rendements financiers	12
1.10 Exercices	13
2 Modèle de Black–Scholes	15
2.1 Introduction	15
2.2 Mouvement brownien géométrique	15
2.3 Hypothèses du modèle de Black–Scholes	16
2.4 Équation de Black–Scholes	16
2.5 Formules de Black–Scholes	16
2.6 Les Grecques	17
2.7 Volatilité implicite	18
2.8 Implémentation Python	18
2.9 Extensions du modèle	19
2.9.1 Dividendes continus	19
2.9.2 Options sur futures (formule de Black)	19
2.10 Limites du modèle	20
2.11 Exercices	20
3 Pricing des Options	21
3.1 Introduction	21
3.2 Arbres binomiaux	21
3.2.1 Algorithme CRR	21
3.2.2 Arbre trinomial	22

3.3	Méthodes de Monte Carlo	22
3.3.1	Principe général	22
3.3.2	Simulation du MBG	22
3.3.3	Réduction de variance	23
3.4	Différences finies	24
3.4.1	Transformation et discrétisation	24
3.5	Comparaison des méthodes	26
3.6	Méthode de Longstaff–Schwartz (options américaines)	26
3.7	Exercices	27
4	Couverture Delta et Gestion des Risques	29
4.1	Introduction	29
4.2	Delta-hedging	29
4.2.1	Principe	29
4.2.2	Erreur de couverture en temps discret	30
4.3	Gamma-hedging	31
4.4	Vega-hedging	31
4.5	Décomposition du P&L	32
4.6	Coûts de transaction et couverture optimale	32
4.7	Couverture en marché incomplet	33
4.8	Exercices	33
5	Calcul Stochastique en Finance	35
5.1	Mouvement brownien	35
5.2	Intégrale d'Itô	36
5.2.1	Construction	36
5.2.2	Règles de calcul	37
5.3	Formule d'Itô	37
5.4	Théorème de Girsanov	38
5.5	Théorème de représentation des martingales	38
5.6	Équations différentielles stochastiques	38
5.7	Exponentielle stochastique	39
5.8	Implémentation : schémas de discrétisation	39
5.9	Exercices	40
6	Modèles à Volatilité Stochastique	41
6.1	Motivation : le smile de volatilité	41
6.2	Volatilité locale (Dupire)	41
6.3	Modèle de Heston	42
6.3.1	Pricing par fonction caractéristique	42
6.4	Modèle SABR	43
6.5	Calibration	44
6.6	Volatilité locale vs stochastique	45
6.7	Modèles à volatilité locale stochastique (SLV)	45
6.8	Exercices	46

7	Modèles de Taux d'Intérêt	47
7.1	Structure par terme des taux	47
7.2	Modèles de taux court	48
7.2.1	Modèle de Vasicek (1977)	48
7.2.2	Modèle de Cox–Ingersoll–Ross (CIR, 1985)	48
7.2.3	Modèle de Hull–White (1990)	48
7.3	Cadre Heath–Jarrow–Morton (HJM)	49
7.4	Pricing de produits de taux	50
7.4.1	Caps et floors	50
7.4.2	Swaptions	50
7.5	Modèles de marché (LIBOR Market Model)	51
7.6	Exercices	51
8	Optimisation de Portefeuille	53
8.1	Théorie de Markowitz	53
8.1.1	Cadre moyenne-variance	53
8.1.2	Problème d'optimisation	53
8.2	Modèle CAPM	55
8.3	Modèle de Black–Litterman	56
8.4	Contraintes pratiques	56
8.5	Optimisation robuste	57
8.6	Estimation des paramètres	57
8.7	Exercices	57
9	Mesures de Risque	59
9.1	Introduction	59
9.2	Value at Risk (VaR)	59
9.3	Expected Shortfall (CVaR)	60
9.4	Mesures de risque cohérentes	60
9.5	Tests de résistance (Stress Testing)	61
9.6	Copules et modélisation de la dépendance	61
9.7	Exercices	62
10	Produits Dérivés Exotiques	65
10.1	Introduction	65
10.2	Options à barrière	65
10.3	Options asiatiques	66
10.4	Options lookback	66
10.5	Options binaires (digitales)	67
10.6	Valorisation par Monte Carlo	67
10.7	Méthodes d'arbre pour les exotiques	68
10.8	Exercices	68
11	Apprentissage Machine en Finance	71
11.1	Introduction	71
11.2	Trading algorithmique	71
11.3	Réseaux de neurones pour le pricing d'options	72
11.4	Apprentissage par renforcement pour la gestion de portefeuille	72
11.5	NLP et analyse de sentiment	73

11.6 Risques et réglementation	73
11.7 Exercices	74

Préface

Objectifs du cours

La **finance quantitative** se situe au carrefour des mathématiques, de la statistique et de l'économie financière. Ce cours de niveau Master a pour ambition de fournir aux étudiants une formation rigoureuse couvrant les fondements théoriques et les méthodes numériques indispensables à la compréhension et à la pratique de la finance de marché moderne.

Les marchés financiers contemporains sont caractérisés par une complexité croissante : produits dérivés exotiques, trading algorithmique à haute fréquence, gestion quantitative de portefeuilles, et plus récemment l'irruption de l'apprentissage automatique dans la modélisation financière. Ce cours vise à donner aux étudiants les outils mathématiques et informatiques nécessaires pour évoluer dans cet environnement.

L'approche adoptée est résolument **mathématique et computationnelle** : chaque concept théorique est accompagné d'une implémentation Python permettant de vérifier les résultats analytiques et de développer l'intuition numérique.

Prérequis

Ce cours suppose que l'étudiant possède des connaissances solides dans les domaines suivants :

- **Analyse mathématique** : calcul différentiel et intégral, séries, espaces de fonctions, équations différentielles ordinaires.
- **Algèbre linéaire** : espaces vectoriels, matrices, valeurs propres, décompositions matricielles (Cholesky, SVD).
- **Probabilités** : espaces de probabilité, variables aléatoires, lois classiques, espérance conditionnelle, théorèmes limites (loi des grands nombres, TCL).
- **Statistique** : estimation paramétrique, tests d'hypothèses, régression linéaire, maximum de vraisemblance.
- **Programmation** : maîtrise de Python (ou d'un langage scientifique équivalent), notions de programmation orientée objet.

Organisation du cours

Le cours est organisé en onze chapitres, chacun conçu pour être traité en une à deux semaines de cours magistral accompagné de travaux dirigés et de séances de travaux pratiques sur ordinateur.

Chapitre 1 — Marchés financiers et modélisation. Introduction aux marchés, actifs financiers, hypothèses de modélisation, absence d'opportunité d'arbitrage (AOA), modèles en temps discret (Cox–Ross–Rubinstein).

Chapitre 2 — Modèle de Black–Scholes. Dérivation du modèle, équation aux dérivées partielles de Black–Scholes, formules fermées pour les options européennes, calcul et interprétation des Grecques.

Chapitre 3 — Méthodes de pricing. Arbres binomiaux et trinomiaux, méthodes de Monte Carlo, différences finies, techniques de réduction de variance.

Chapitre 4 — Couverture et gestion des risques. Delta-hedging, couverture dynamique, gamma-hedging, coût de la couverture, limites pratiques du modèle de Black–Scholes.

Chapitre 5 — Calcul stochastique en finance. Mouvement brownien, intégrale d'Itô, formule d'Itô, théorème de Girsanov, représentation des martingales.

Chapitre 6 — Volatilité stochastique. Smile et surface de volatilité, modèle de Heston, modèle SABR, calibration, volatilité locale vs stochastique.

Chapitre 7 — Modèles de taux d'intérêt. Structure par terme, modèles de Vasicek, CIR, Hull–White, cadre HJM, pricing d'obligations et de caps/floors.

Chapitre 8 — Optimisation de portefeuille. Théorie de Markowitz, frontière efficiente, modèle CAPM, modèle de Black–Litterman, optimisation robuste.

Chapitre 9 — Mesures de risque. Value-at-Risk (VaR), Expected Shortfall (CVaR), propriétés de cohérence, méthodes de calcul, backtesting, stress testing.

Chapitre 10 — Dérivés exotiques. Options barrières, options asiatiques, lookback, options sur panier, pricing analytique et numérique.

Chapitre 11 — Apprentissage automatique en finance. Réseaux de neurones pour le pricing, apprentissage par renforcement pour le trading, détection d'anomalies, risques de sur-apprentissage.

Méthodologie pédagogique

Chaque chapitre suit une structure cohérente :

1. **Théorie** : présentation rigoureuse des concepts mathématiques avec démonstrations complètes ou esquissées.
2. **Interprétation financière** : discussion de la signification économique des résultats et de leurs limites.

3. **Implémentation** : code Python illustrant les méthodes numériques avec les bibliothèques `numpy`, `scipy`, `matplotlib`.
4. **Exercices** : problèmes théoriques et numériques de difficulté progressive pour approfondir la compréhension.

Conventions de notation

Tout au long de ce cours, nous utiliserons les notations suivantes :

Notation	Signification
$(\Omega, \mathcal{F}, \mathbb{P})$	Espace de probabilité
$\mathbb{E}[X]$	Espérance de la variable aléatoire X
$\mathbb{E}^{\mathbb{Q}}[\cdot]$	Espérance sous la mesure risque-neutre
$\text{Var}(X)$	Variance de X
$\text{Cov}(X, Y)$	Covariance de X et Y
$\mathbb{R}, \mathbb{R}^+$	Nombres réels, réels positifs
$\mathbb{N}$	Nombres naturels
W_t ou B_t	Mouvement brownien standard
$(\mathcal{F}_t)_{t \geq 0}$	Filtration naturelle
$\mathbb{Q}$	Mesure de probabilité risque-neutre
r	Taux d'intérêt sans risque (continu)
σ	Volatilité de l'actif sous-jacent
S_t	Prix de l'actif sous-jacent à l'instant t
$C(S, t), P(S, t)$	Prix du call et du put européens
K	Prix d'exercice (strike)
T	Maturité de l'option
$\Delta, \Gamma, \Theta, \mathcal{V}, \rho$	Lettres grecques (sensibilités)

Logiciels et outils

Les implémentations numériques de ce cours utilisent l'écosystème Python scientifique :

- `numpy` : calcul matriciel, algèbre linéaire, génération de nombres aléatoires.
- `scipy` : optimisation (`scipy.optimize`), intégration, fonctions spéciales, loi normale (`scipy.stats.norm`).
- `matplotlib` : visualisation graphique, tracé de surfaces de volatilité, frontières efficientes.
- `pandas` : manipulation et analyse de séries temporelles financières.
- `scikit-learn` : algorithmes d'apprentissage automatique classiques (chapitre 11).
- `tensorflow` / `pytorch` : réseaux de neurones profonds (chapitre 11).

Thèmes transversaux

Plusieurs thèmes traversent l'ensemble du cours et méritent d'être signalés dès cette préface :

- **Absence d'arbitrage** : principe fondamental qui sous-tend toute la théorie du pricing en finance quantitative.
- **Complétude des marchés** : la question de savoir si tout actif contingent peut être répliqué par une stratégie auto-finançante.
- **Risque de modèle** : les conséquences de l'utilisation de modèles mathématiques simplifiés pour des réalités financières complexes.
- **Calibration vs estimation** : la distinction entre calibrer un modèle sur les prix de marché et estimer ses paramètres à partir de données historiques.

Références bibliographiques

Ce cours s'appuie sur les ouvrages de référence suivants :

1. SHREVE, S.E. — *Stochastic Calculus for Finance I & II*, Springer, 2004.
2. HULL, J.C. — *Options, Futures, and Other Derivatives*, Pearson, 2017 (10^e édition).
3. BJÖRK, T. — *Arbitrage Theory in Continuous Time*, Oxford University Press, 2009 (3^e édition).
4. GLASSERMAN, P. — *Monte Carlo Methods in Financial Engineering*, Springer, 2003.
5. GATHERAL, J. — *The Volatility Surface : A Practitioner's Guide*, Wiley, 2006.
6. BRIGO, D. & MERCURIO, F. — *Interest Rate Models — Theory and Practice*, Springer, 2006.
7. MCNEIL, A.J., FREY, R. & EMBRECHTS, P. — *Quantitative Risk Management : Concepts, Techniques and Tools*, Princeton University Press, 2015.
8. LÓPEZ DE PRADO, M. — *Advances in Financial Machine Learning*, Wiley, 2018.
9. LAMBERTON, D. & LAPEYRE, B. — *Introduction au calcul stochastique appliqué à la finance*, Ellipses, 2012.
10. DANA, R.-A. & JEANBLANC, M. — *Marchés financiers en temps continu*, Economica, 2003.

Remerciements

Ce cours a bénéficié des contributions de nombreux collègues enseignants-chercheurs en mathématiques financières. Nous remercions particulièrement les étudiants des promotions précédentes dont les questions et remarques pertinentes ont permis d'améliorer considérablement la présentation de ce matériel pédagogique.

L'auteur
Mars 2026

Chapitre 1

Marchés Financiers — Modélisation

En 1900, Louis Bachelier soutient sa thèse à la Sorbonne sous la direction d'Henri Poincaré. Son sujet : la « théorie de la spéculation », où il propose de modéliser les cours boursiers par un processus stochastique. Poincaré, pourtant visionnaire, juge le travail « pas sans intérêt » mais lui accorde une mention honorable plutôt que très honorable. Il faudra soixante ans, l'arrivée de Samuelson, Black, Scholes et Merton, pour que les idées de Bachelier deviennent la pierre angulaire de la finance moderne. Aujourd'hui, les marchés financiers brassent des milliers de milliards par jour, et les modèles mathématiques qui les décrivent sont au cœur de l'économie mondiale.

1.1 Introduction aux marchés financiers

Les marchés financiers constituent le lieu d'échange d'actifs dont la valeur est incertaine. Comprendre leur fonctionnement et les modéliser mathématiquement est le point de départ de toute la finance quantitative.

Définition 1.1 (Marché financier). Un **marché financier** est un système organisé permettant l'échange d'actifs financiers entre agents économiques. On distingue :

- Le **marché primaire** : émission de nouveaux titres.
- Le **marché secondaire** : échange de titres déjà émis.

1.1.1 Classes d'actifs

Définition 1.2 (Actif financier). Un **actif financier** est un contrat donnant droit à des flux de trésorerie futurs. Les principales classes sont :

1. **Actions** (equities) : parts de propriété d'une entreprise.
2. **Obligations** (bonds) : titres de créance à revenu fixe.
3. **Produits dérivés** (derivatives) : contrats dont la valeur dépend d'un actif sous-jacent.
4. **Devises** (currencies) : marché des changes (forex).
5. **Matières premières** (commodities) : or, pétrole, blé.

1.1.2 Produits dérivés fondamentaux

Définition 1.3 (Option européenne). Une **option d'achat (call) européenne** de prix d'exercice K et de maturité T sur un sous-jacent S donne à son détenteur le droit (mais non l'obligation) d'acheter S au prix K à la date T . Son payoff est :

$$(S_T - K)^+ = \max(S_T - K, 0).$$

De même, une **option de vente (put) européenne** a pour payoff :

$$(K - S_T)^+ = \max(K - S_T, 0).$$

Définition 1.4 (Contrat forward). Un **contrat forward** (ou contrat à terme) est un accord d'acheter ou de vendre un actif à une date future T à un prix F fixé aujourd'hui. Son payoff pour l'acheteur est $S_T - F$.

1.2 Modélisation en temps discret

1.2.1 Modèle à une période

Considérons un marché à deux dates $t = 0$ et $t = 1$, avec :

- Un actif sans risque de prix $B_0 = 1$, $B_1 = 1 + r$.
- Un actif risqué de prix S_0 à $t = 0$, pouvant prendre les valeurs $S_1^u = uS_0$ (hausse) ou $S_1^d = dS_0$ (baisse) avec $0 < d < 1 + r < u$.

Définition 1.5 (Stratégie de trading). Une **stratégie de trading** est un couple $(\phi^0, \phi^1) \in \mathbb{R}^2$ où ϕ^0 représente la quantité d'actif sans risque et ϕ^1 la quantité d'actif risqué détenues. La valeur du portefeuille est :

$$V_t = \phi^0 B_t + \phi^1 S_t, \quad t \in \{0, 1\}.$$

Définition 1.6 (Opportunité d'arbitrage). Une **opportunité d'arbitrage** est une stratégie (ϕ^0, ϕ^1) telle que :

$$V_0 \leq 0, \quad V_1 \geq 0 \text{ p.s.}, \quad \mathbb{P}(V_1 > 0) > 0.$$

Autrement dit, un gain possible sans risque de perte et sans mise de fonds initiale.

Théorème 1.7 (Absence d'arbitrage — modèle binomial à une période). *Le modèle binomial à une période est sans arbitrage si et seulement si $d < 1 + r < u$.*

Démonstration. Supposons qu'il existe un arbitrage (ϕ^0, ϕ^1) avec $V_0 = 0$. Alors :

$$\phi^0 + \phi^1 S_0 = 0 \implies \phi^0 = -\phi^1 S_0.$$

Les valeurs terminales sont :

$$\begin{aligned} V_1^u &= \phi^1(uS_0 - (1+r)S_0) = \phi^1 S_0(u - 1 - r), \\ V_1^d &= \phi^1(dS_0 - (1+r)S_0) = \phi^1 S_0(d - 1 - r). \end{aligned}$$

Pour que $V_1^u \geq 0$ et $V_1^d \geq 0$ simultanément avec au moins une inégalité stricte, il faut que $u - 1 - r$ et $d - 1 - r$ soient de même signe non nul, ce qui contredit $d < 1 + r < u$. $\square$

1.2.2 Probabilité risque-neutre

Théorème 1.8 (Mesure risque-neutre). *Sous la condition AOA ($d < 1 + r < u$), il existe une unique mesure de probabilité $\mathbb{Q}$ sur $\{\omega_u, \omega_d\}$ telle que le prix actualisé de l'actif risqué soit une $\mathbb{Q}$ -martingale :*

$$S_0 = \frac{1}{1+r} \mathbb{E}^{\mathbb{Q}}[S_1].$$

La probabilité risque-neutre est donnée par :

$$q = \frac{(1+r) - d}{u - d}, \quad 1 - q = \frac{u - (1+r)}{u - d}.$$

Prix risque-neutre d'un actif contingent

Le prix à $t = 0$ d'un actif contingent de payoff $H = h(S_1)$ est :

$$V_0 = \frac{1}{1+r} \mathbb{E}^{\mathbb{Q}}[H] = \frac{1}{1+r} [q \cdot h(uS_0) + (1-q) \cdot h(dS_0)].$$

Exemple 1.9. Soit $S_0 = 100$, $u = 1.2$, $d = 0.9$, $r = 0.05$, $K = 100$.

La probabilité risque-neutre est :

$$q = \frac{1.05 - 0.9}{1.2 - 0.9} = \frac{0.15}{0.30} = 0.5.$$

Le prix du call est :

$$C_0 = \frac{1}{1.05} [0.5 \times (120 - 100)^+ + 0.5 \times (90 - 100)^+] = \frac{0.5 \times 20}{1.05} \approx 9.52.$$

1.3 Modèle binomial multi-périodes (CRR)

Définition 1.10 (Modèle de Cox–Ross–Rubinstein). Le **modèle CRR** à N périodes est défini par :

- $S_{n+1} = S_n \cdot Y_{n+1}$ où $Y_{n+1} \in \{u, d\}$ sont i.i.d. sous $\mathbb{Q}$.
- Le prix actualisé $\tilde{S}_n = S_n / (1+r)^n$ est une $\mathbb{Q}$ -martingale.
- Après n périodes : $S_n = S_0 u^j d^{n-j}$ pour j hausses parmi n .

Théorème 1.11 (Formule de pricing CRR). *Le prix d'un call européen dans le modèle CRR est :*

$$C_0 = \frac{1}{(1+r)^N} \sum_{j=0}^N \binom{N}{j} q^j (1-q)^{N-j} (S_0 u^j d^{N-j} - K)^+.$$

Convergence vers Black–Scholes

Lorsque $N \rightarrow \infty$ avec le paramétrage $u = e^{\sigma\sqrt{\Delta t}}$, $d = e^{-\sigma\sqrt{\Delta t}}$, $\Delta t = T/N$, la formule CRR converge vers la formule de Black–Scholes (chapitre 2). C'est une conséquence du théorème central limite.

1.3.1 Calibration des paramètres CRR

Pour que le modèle discret approche le modèle continu avec volatilité σ et taux r , on pose :

$$u = e^{\sigma\sqrt{\Delta t}}, \quad d = e^{-\sigma\sqrt{\Delta t}} = 1/u, \quad (1.1)$$

$$q = \frac{e^{r\Delta t} - d}{u - d}. \quad (1.2)$$

1.4 Théorèmes fondamentaux de l'évaluation d'actifs

Théorème 1.12 (Premier théorème fondamental (FTAP)). *Un marché est sans arbitrage si et seulement s'il existe une mesure de probabilité $\mathbb{Q}$ équivalente à $\mathbb{P}$ sous laquelle les prix actualisés des actifs sont des martingales.*

Théorème 1.13 (Second théorème fondamental). *Un marché sans arbitrage est **complet** (tout actif contingent est répliquable) si et seulement si la mesure martingale $\mathbb{Q}$ est unique.*

Remarque 1.14. Dans le modèle binomial, le marché est complet car il y a deux états de la nature et deux actifs. En revanche, dans un modèle trinomial (trois états), le marché est incomplet et il existe une infinité de mesures martingales.

1.5 Réplication et couverture

Proposition 1.15 (Portefeuille de réplication — une période). Dans le modèle binomial à une période, l'actif contingent $H = h(S_1)$ est répliqué par (ϕ^0, ϕ^1) avec :

$$\phi^1 = \frac{h(uS_0) - h(dS_0)}{(u - d)S_0}, \quad \phi^0 = \frac{1}{1 + r} [h(uS_0) - \phi^1 u S_0].$$

La quantité ϕ^1 est appelée le **delta** de l'option.

1.6 Implémentation Python

Modèle binomial CRR en Python

```
import numpy as np

def crr_european(S0, K, T, r, sigma, N, option='call'):
    """Prix d'une option européenne par le modele CRR."""
    dt = T / N
    u = np.exp(sigma * np.sqrt(dt))
    d = 1 / u
    q = (np.exp(r * dt) - d) / (u - d)
    discount = np.exp(-r * dt)

    # Prix terminaux
    j = np.arange(N + 1)
    ST = S0 * u**j * d**(N - j)
```



```

if option == 'call':
    payoff = np.maximum(ST - K, 0)
else:
    payoff = np.maximum(K - ST, 0)

# Backward induction
for n in range(N - 1, -1, -1):
    payoff = discount * (q * payoff[1:] + (1 - q) * payoff[:-1])

return payoff[0]

# Exemple
S0, K, T, r, sigma = 100, 100, 1.0, 0.05, 0.20
for N in [10, 50, 100, 500, 1000]:
    price = crr_european(S0, K, T, r, sigma, N)
    print(f"N={N:4d}: C = {price:.4f}")

```

Parité call-put — vérification

```

def verify_put_call_parity(S0, K, T, r, sigma, N):
    """Verifie la parite call-put: C - P = S0 - K*exp(-rT)."""
    C = crr_european(S0, K, T, r, sigma, N, 'call')
    P = crr_european(S0, K, T, r, sigma, N, 'put')
    parity = S0 - K * np.exp(-r * T)
    print(f"C - P = {C - P:.6f}")
    print(f"S0 - K*exp(-rT) = {parity:.6f}")
    print(f"Erreur = {abs(C - P - parity):.2e}")

verify_put_call_parity(100, 100, 1.0, 0.05, 0.20, 500)

```

1.7 Parité call-put

Théorème 1.16 (Parité call-put). *Pour des options européennes de même strike K et maturité T :*

$$C - P = S_0 - Ke^{-rT}.$$

Démonstration. Considérons le portefeuille $\Pi = C - P - S_0 + Ke^{-rT}$. À maturité :

$$\Pi_T = (S_T - K)^+ - (K - S_T)^+ - S_T + K = S_T - K - S_T + K = 0.$$

Par AOA, $\Pi_0 = 0$, d'où le résultat. □

1.8 Hypothèses de marché

Hypothèses simplificatrices

Les modèles classiques reposent sur des hypothèses fortes :

1. Marché sans friction (pas de coûts de transaction).
2. Trading continu possible.
3. Vente à découvert autorisée sans restriction.
4. Taux d'emprunt égal au taux de prêt.
5. Pas de risque de contrepartie.

La relaxation de ces hypothèses conduit à des modèles plus réalistes mais significativement plus complexes.

1.9 Faits stylisés des rendements financiers

Les rendements logarithmiques $r_t = \ln(S_t/S_{t-1})$ présentent des propriétés empiriques bien documentées :

1. **Queues épaisses** : la distribution des rendements a des queues plus lourdes que la loi normale (kurtosis excédentaire).
2. **Asymétrie** : légère asymétrie négative.
3. **Agrégation gaussienne** : les rendements sur longue période se rapprochent de la loi normale.
4. **Absence d'autocorrélation** : les rendements sont quasi non corrélés.
5. **Clustering de volatilité** : les périodes de forte volatilité sont regroupées.
6. **Effet de levier** : corrélation négative entre rendements et volatilité.

Analyse des faits stylisés

```
import numpy as np
from scipy import stats

def stylized_facts(returns):
    """Analyse des faits stylisés des rendements."""
    print(f"Moyenne:      {np.mean(returns):.6f}")
    print(f"Ecart-type: {np.std(returns):.6f}")
    print(f"Skewness:      {stats.skew(returns):.4f}")
    print(f"Kurtosis:       {stats.kurtosis(returns):.4f}")

    # Test de normalité (Jarque-Bera)
    jb_stat, jb_pval = stats.jarque_bera(returns)
```

```

print(f"Jarque-Bera: stat={jb_stat:.2f}, p={jb_pval:.4e}")

# Autocorrelation des rendements et des carres
n = len(returns)
r_centered = returns - np.mean(returns)
acf1 = np.correlate(r_centered[:-1], r_centered[1:]) / (n *
    ↳ np.var(returns))
r2 = r_centered**2
acf1_sq = np.correlate(r2[:-1], r2[1:]) / (n * np.var(r2))
print(f"ACF(1) rendements: {acf1[0]:.4f}")
print(f"ACF(1) carres:      {acf1_sq[0]:.4f}")

# Simulation GBM pour comparaison
np.random.seed(42)
S0, mu, sigma, dt, N = 100, 0.08, 0.20, 1/252, 2520
Z = np.random.standard_normal(N)
log_returns = (mu - 0.5*sigma**2)*dt + sigma*np.sqrt(dt)*Z
stylized_facts(log_returns)

```

1.10 Exercices

Exercice 1.1 (Modèle binomial à une période). Soit $S_0 = 50$, $u = 1.3$, $d = 0.8$, $r = 0.04$.

1. Vérifier la condition AOA.
2. Calculer la probabilité risque-neutre q .
3. Calculer le prix d'un call de strike $K = 55$.
4. Déterminer le portefeuille de réplication (ϕ^0, ϕ^1) .
5. Vérifier que le prix du put satisfait la parité call-put.

Exercice 1.2 (Convergence CRR). 1. Implémenter le modèle CRR et tracer le prix du call en fonction de N pour $N = 1, 2, \dots, 200$.

2. Observer la convergence et les oscillations pair/impair.
3. Comparer avec la formule de Black–Scholes (chapitre 2).

Exercice 1.3 (Marché incomplet). Considérons un modèle trinomial : $S_1 \in \{uS_0, mS_0, dS_0\}$ avec $d < m < u$.

1. Montrer que le marché est incomplet.
2. Caractériser l'ensemble des mesures martingales.
3. En déduire les bornes de non-arbitrage pour le prix d'un call.

Exercice 1.4 (Options américaines dans le modèle CRR). 1. Modifier l'algorithme CRR pour pricer un put américain en intégrant la possibilité d'exercice anticipé.

2. Comparer les prix du put américain et européen. Vérifier que le put américain est toujours au moins aussi cher.
3. Déterminer la frontière d'exercice optimal.

Chapitre 2

Modèle de Black–Scholes

En 1973, Fischer Black et Myron Scholes publient un article qui va transformer la finance : ils dérivent une formule explicite pour le prix d’une option européenne. Robert Merton, indépendamment, étend et solidifie la théorie. Le modèle repose sur une hypothèse audacieuse : le cours de l’action suit un mouvement brownien géométrique, et dans ce cadre, il est possible de construire un portefeuille de couverture qui élimine *tout* le risque. Le prix de l’option est alors le coût de cette couverture. Scholes et Merton recevront le prix Nobel d’économie en 1997 (Black étant décédé en 1995).

2.1 Introduction

Le modèle de Black–Scholes–Merton (1973) constitue la pierre angulaire de la finance quantitative moderne. Il fournit un cadre mathématique rigoureux pour l’évaluation et la couverture des options européennes, basé sur l’hypothèse que le prix de l’actif sous-jacent suit un mouvement brownien géométrique.

2.2 Mouvement brownien géométrique

Définition 2.1 (Mouvement brownien géométrique (MBG)). Un processus $(S_t)_{t \geq 0}$ suit un **mouvement brownien géométrique** s’il satisfait l’équation différentielle stochastique :

$$dS_t = \mu S_t dt + \sigma S_t dW_t,$$

où $\mu \in \mathbb{R}$ est le rendement espéré (drift), $\sigma > 0$ la volatilité, et $(W_t)_{t \geq 0}$ un mouvement brownien standard.

Théorème 2.2 (Solution du MBG). *La solution de l’EDS ci-dessus est :*

$$S_t = S_0 \exp \left[\left(\mu - \frac{\sigma^2}{2} \right) t + \sigma W_t \right].$$

En particulier, $\ln(S_t/S_0) \sim \mathcal{N}((\mu - \sigma^2/2)t, \sigma^2 t)$.

Démonstration. Posons $X_t = \ln S_t$. Par la formule d’Itô (cf. chapitre 5) :

$$dX_t = \frac{1}{S_t} dS_t - \frac{1}{2} \frac{1}{S_t^2} (dS_t)^2 = \left(\mu - \frac{\sigma^2}{2} \right) dt + \sigma dW_t.$$

En intégrant de 0 à t : $X_t = X_0 + (\mu - \sigma^2/2)t + \sigma W_t$, d’où le résultat. □

2.3 Hypothèses du modèle de Black–Scholes

Hypothèses fondamentales

1. Le sous-jacent suit un MBG de volatilité σ constante.
2. Le taux sans risque r est constant.
3. Pas de dividendes (généralisation possible).
4. Marché sans frictions (pas de coûts de transaction).
5. Trading continu possible.
6. Vente à découvert autorisée.

2.4 Équation de Black–Scholes

Théorème 2.3 (ÉDP de Black–Scholes). *Sous les hypothèses du modèle, le prix $V(S, t)$ d'un dérivé européen satisfait l'équation aux dérivées partielles :*

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

avec la condition terminale $V(S, T) = h(S)$ où h est le payoff.

Esquisse de preuve par réplication. Considérons un portefeuille $\Pi = V - \Delta S$ où $\Delta = \partial V / \partial S$. Par Itô :

$$\begin{aligned} d\Pi &= dV - \Delta dS \\ &= \left(\frac{\partial V}{\partial t} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} \right) dt. \end{aligned}$$

Ce portefeuille est sans risque (pas de terme en dW_t), donc par AOA : $d\Pi = r\Pi dt = r(V - \Delta S) dt$. En égalisant, on obtient l'EDP. $\square$

2.5 Formules de Black–Scholes

Théorème 2.4 (Formules fermées de Black–Scholes). *Le prix d'un call européen est :*

$$C(S, t) = S \Phi(d_1) - K e^{-r(T-t)} \Phi(d_2),$$

et le prix d'un put européen :

$$P(S, t) = K e^{-r(T-t)} \Phi(-d_2) - S \Phi(-d_1),$$

où Φ désigne la fonction de répartition de la loi normale standard et :

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

Formules de Black-Scholes — Récapitulatif

$$\begin{aligned}
C &= S\Phi(d_1) - Ke^{-r\tau}\Phi(d_2), \\
P &= Ke^{-r\tau}\Phi(-d_2) - S\Phi(-d_1), \\
d_1 &= \frac{\ln(S/K) + (r + \sigma^2/2)\tau}{\sigma\sqrt{\tau}}, \quad d_2 = d_1 - \sigma\sqrt{\tau}, \quad \tau = T - t.
\end{aligned}$$

Preuve par pricing risque-neutre. Sous la mesure risque-neutre $\mathbb{Q}$:

$$C = e^{-r\tau}\mathbb{E}^{\mathbb{Q}}[(S_T - K)^+].$$

Or $S_T = S \exp[(r - \sigma^2/2)\tau + \sigma\sqrt{\tau}Z]$ avec $Z \sim \mathcal{N}(0, 1)$ sous $\mathbb{Q}$. Le payoff est positif lorsque $Z > -d_2$. On calcule :

$$\begin{aligned}
C &= e^{-r\tau} \int_{-d_2}^{+\infty} \left(Se^{(r-\sigma^2/2)\tau + \sigma\sqrt{\tau}z} - K \right) \frac{e^{-z^2/2}}{\sqrt{2\pi}} dz \\
&= S\Phi(d_1) - Ke^{-r\tau}\Phi(d_2).
\end{aligned}$$

Le passage de d_2 à d_1 dans le premier terme résulte du complètement du carré dans l'exponentielle. $\square$

2.6 Les Grecques

Les **Grecques** (Greeks) mesurent la sensibilité du prix de l'option aux différents paramètres du modèle.

Définition 2.5 (Grecques principales). Pour un call européen :

$$\begin{aligned}
\Delta &= \frac{\partial C}{\partial S} = \Phi(d_1), \\
\Gamma &= \frac{\partial^2 C}{\partial S^2} = \frac{\phi(d_1)}{S\sigma\sqrt{\tau}}, \\
\Theta &= \frac{\partial C}{\partial t} = -\frac{S\sigma\phi(d_1)}{2\sqrt{\tau}} - rKe^{-r\tau}\Phi(d_2), \\
\mathcal{V} &= \frac{\partial C}{\partial \sigma} = S\sqrt{\tau}\phi(d_1), \\
\rho &= \frac{\partial C}{\partial r} = K\tau e^{-r\tau}\Phi(d_2),
\end{aligned}$$

où $\phi(x) = \frac{1}{\sqrt{2\pi}}e^{-x^2/2}$ est la densité de la loi normale standard.

Interprétation des Grecques

- Δ : nombre d'unités du sous-jacent pour couvrir l'option.
- Γ : convexité, mesure la stabilité du delta.
- Θ : décroissance temporelle (*time decay*).

- $\mathcal{V}$ (vega) : sensibilité à la volatilité.
- ρ : sensibilité au taux d'intérêt.

Proposition 2.6 (Relation fondamentale des Grecques). Le delta, gamma et theta d'un dérivé satisfont l'EDP de Black-Scholes sous la forme :

$$\Theta + rS\Delta + \frac{1}{2}\sigma^2 S^2 \Gamma = rV.$$

2.7 Volatilité implicite

Définition 2.7 (Volatilité implicite). La **volatilité implicite** σ_{impl} est la valeur de σ qui, insérée dans la formule de Black-Scholes, reproduit le prix de marché observé C_{mkt} :

$$C_{\text{BS}}(S, K, T, r, \sigma_{\text{impl}}) = C_{\text{mkt}}.$$

Remarque 2.8. L'existence et l'unicité de σ_{impl} découlent du fait que $\mathcal{V} = \partial C / \partial \sigma > 0$ (le prix du call est strictement croissant en la volatilité).

2.8 Implémentation Python

Formules de Black-Scholes et Grecques

```
import numpy as np
from scipy.stats import norm

def bs_price(S, K, T, r, sigma, option='call'):
    """Prix Black-Scholes d'une option européenne."""
    d1 = (np.log(S/K) + (r + 0.5*sigma**2)*T) / (sigma*np.sqrt(T))
    d2 = d1 - sigma*np.sqrt(T)
    if option == 'call':
        return S*norm.cdf(d1) - K*np.exp(-r*T)*norm.cdf(d2)
    else:
        return K*np.exp(-r*T)*norm.cdf(-d2) - S*norm.cdf(-d1)

def bs_greeks(S, K, T, r, sigma):
    """Calcul des Grecques pour un call européen."""
    d1 = (np.log(S/K) + (r + 0.5*sigma**2)*T) / (sigma*np.sqrt(T))
    d2 = d1 - sigma*np.sqrt(T)
    delta = norm.cdf(d1)
    gamma = norm.pdf(d1) / (S * sigma * np.sqrt(T))
    theta = (-S*sigma*norm.pdf(d1)/(2*np.sqrt(T))
             - r*K*np.exp(-r*T)*norm.cdf(d2))
    vega = S * np.sqrt(T) * norm.pdf(d1)
    rho = K * T * np.exp(-r*T) * norm.cdf(d2)
    return {'delta': delta, 'gamma': gamma, 'theta': theta,
            'vega': vega, 'rho': rho}
```

Exemple


```

S, K, T, r, sigma = 100, 100, 1.0, 0.05, 0.20
print(f"Call = {bs_price(S,K,T,r,sigma,'call'):.4f}")
print(f"Put = {bs_price(S,K,T,r,sigma,'put'):.4f}")
greeks = bs_greeks(S, K, T, r, sigma)
for name, val in greeks.items():
    print(f"{name:>6s} = {val:.6f}")

```

Calcul de la volatilité implicite (Newton-Raphson)

```

def implied_vol(market_price, S, K, T, r, option='call',
                tol=1e-8, max_iter=100):
    """Volatilité implicite par Newton-Raphson."""
    sigma = 0.20 # initial guess
    for _ in range(max_iter):
        price = bs_price(S, K, T, r, sigma, option)
        d1 = (np.log(S/K) + (r + 0.5*sigma**2)*T) / (sigma*np.sqrt(T))
        vega = S * np.sqrt(T) * norm.pdf(d1)
        if vega < 1e-12:
            break
        sigma -= (price - market_price) / vega
        if abs(price - market_price) < tol:
            return sigma
    return sigma

# Exemple: retrouver sigma a partir d'un prix
C_mkt = bs_price(100, 100, 1.0, 0.05, 0.25, 'call')
iv = implied_vol(C_mkt, 100, 100, 1.0, 0.05)
print(f"Vol implicite retrouvée: {iv:.6f}")

```

2.9 Extensions du modèle

2.9.1 Dividendes continus

Pour un sous-jacent versant un taux de dividende continu q , la formule de Black-Scholes se modifie :

$$C = Se^{-q\tau}\Phi(d_1) - Ke^{-r\tau}\Phi(d_2),$$

$$\text{avec } d_1 = \frac{\ln(S/K) + (r - q + \sigma^2/2)\tau}{\sigma\sqrt{\tau}}.$$

2.9.2 Options sur futures (formule de Black)

Théorème 2.9 (Formule de Black (1976)). *Pour une option européenne sur un contrat futures de prix F :*

$$C = e^{-r\tau}[F\Phi(d_1) - K\Phi(d_2)],$$

$$\text{avec } d_1 = \frac{\ln(F/K) + \sigma^2\tau/2}{\sigma\sqrt{\tau}}, \quad d_2 = d_1 - \sigma\sqrt{\tau}.$$

2.10 Limites du modèle

Limites connues du modèle de Black-Scholes

1. La volatilité n'est pas constante en pratique (smile de volatilité, cf. chapitre 6).
2. Les rendements ne sont pas gaussiens (queues épaisses).
3. Le trading continu est une idéalisation.
4. Les coûts de transaction ne sont pas nuls.
5. Le modèle sous-estime le risque de crash.

2.11 Exercices

Exercice 2.1 (Vérification de la formule de Black-Scholes). 1. Vérifier par substitution directe que la formule du call satisfait l'EDP de Black-Scholes.

2. Vérifier la condition terminale $C(S, T) = (S - K)^+$.
3. Démontrer la parité call-put à partir des formules.

Exercice 2.2 (Comportements limites). Étudier les limites suivantes :

1. $C(S, t)$ lorsque $\sigma \rightarrow 0^+$.
2. $C(S, t)$ lorsque $\sigma \rightarrow +\infty$.
3. $C(S, t)$ lorsque $T - t \rightarrow 0^+$ avec $S > K$, $S = K$, $S < K$.
4. Δ lorsque $T - t \rightarrow 0^+$.

Exercice 2.3 (Grecques numériques). 1. Calculer le delta et le gamma par différences finies et comparer aux formules analytiques.

2. Tracer Δ , Γ , Θ , $\mathcal{V}$ en fonction de S pour différentes valeurs de τ .
3. Vérifier numériquement la relation $\Theta + rS\Delta + \frac{1}{2}\sigma^2 S^2 \Gamma = rC$.

Exercice 2.4 (Surface de volatilité implicite). 1. Générer des prix d'options à partir d'un modèle de Heston (chapitre 6) pour différents strikes et maturités.

2. Calculer la volatilité implicite Black-Scholes pour chaque prix.
3. Tracer la surface de volatilité implicite $(K, T) \mapsto \sigma_{\text{impl}}(K, T)$.

Chapitre 3

Pricing des Options

3.1 Introduction

La formule de Black-Scholes est élégante, mais elle ne s'applique qu'aux options européennes sur un sous-jacent suivant un mouvement brownien géométrique. Que faire pour une option américaine (exercice anticipé), une option asiatique (dépendant de la moyenne du cours), ou un produit exotique à barrière ? Les formules fermées n'existent plus, et il faut recourir au calcul numérique. Trois grandes familles de méthodes se partagent le terrain : les arbres binomiaux, hérités du modèle de Cox-Ross-Rubinstein (1979), qui discrétisent le temps et construisent l'option par récurrence arrière ; les méthodes de Monte Carlo, qui simulent des milliers de trajectoires du sous-jacent et moyennent les payoffs actualisés ; et les différences finies, qui résolvent numériquement l'EDP de Black-Scholes sur une grille.

Chacune a ses forces et ses faiblesses, et le choix dépend du produit, de la dimension et de la précision requise.

3.2 Arbres binomiaux

3.2.1 Algorithme CRR

Rappelons le modèle de Cox-Ross-Rubinstein (chapitre 1). L'algorithme procède en deux phases :

1. **Phase forward** : construction de l'arbre des prix $S_{n,j} = S_0 u^j d^{n-j}$ pour $n = 0, \dots, N$ et $j = 0, \dots, n$.
2. **Phase backward** : calcul récursif des valeurs de l'option de $n = N$ à $n = 0$.

Récurrence backward

Pour une option européenne :

$$V_{n,j} = e^{-r\Delta t} [q \cdot V_{n+1,j+1} + (1-q) \cdot V_{n+1,j}],$$

avec condition terminale $V_{N,j} = h(S_{N,j})$.

Pour une option américaine, on ajoute la possibilité d'exercice anticipé :

$$V_{n,j} = \max(h(S_{n,j}), e^{-r\Delta t} [qV_{n+1,j+1} + (1-q)V_{n+1,j}]).$$

3.2.2 Arbre trinomial

Définition 3.1 (Arbre trinomial). L'arbre trinomial utilise trois mouvements possibles :

$$S_{n+1} \in \{uS_n, S_n, dS_n\}$$

avec des probabilités (p_u, p_m, p_d) choisies pour respecter les deux premiers moments :

$$p_u = \frac{1}{2} \left(\frac{\sigma^2 \Delta t + \nu^2 \Delta t^2}{(\Delta x)^2} + \frac{\nu \Delta t}{\Delta x} \right), \quad (3.1)$$

$$p_d = \frac{1}{2} \left(\frac{\sigma^2 \Delta t + \nu^2 \Delta t^2}{(\Delta x)^2} - \frac{\nu \Delta t}{\Delta x} \right), \quad (3.2)$$

$$p_m = 1 - p_u - p_d, \quad (3.3)$$

où $\nu = r - \sigma^2/2$ et $\Delta x = \sigma\sqrt{3\Delta t}$.

Remarque 3.2. L'arbre trinomial converge plus rapidement que le binomial et présente moins d'oscillations dans le prix en fonction de N .

3.3 Méthodes de Monte Carlo

3.3.1 Principe général

Théorème 3.3 (Pricing par Monte Carlo). *Le prix d'un dérivé européen de payoff $h(S_T)$ est :*

$$V_0 = e^{-rT} \mathbb{E}^{\mathbb{Q}}[h(S_T)] \approx \frac{e^{-rT}}{M} \sum_{m=1}^M h(S_T^{(m)}),$$

où les $S_T^{(m)}$ sont des réalisations indépendantes de S_T sous $\mathbb{Q}$.

La convergence est en $O(1/\sqrt{M})$ par le théorème central limite, indépendamment de la dimension du problème. C'est l'avantage majeur de Monte Carlo pour les problèmes en grande dimension.

Proposition 3.4 (Intervalle de confiance). L'estimateur Monte Carlo $\hat{V}_M$ satisfait :

$$\mathbb{P}\left(\left|\hat{V}_M - V_0\right| \leq z_{\alpha/2} \frac{\hat{\sigma}_{\text{MC}}}{\sqrt{M}}\right) \approx 1 - \alpha,$$

où $\hat{\sigma}_{\text{MC}}$ est l'écart-type empirique des payoffs actualisés et $z_{\alpha/2}$ le quantile de la loi normale.

3.3.2 Simulation du MBG

Sous la mesure risque-neutre :

$$S_T = S_0 \exp\left[\left(r - \frac{\sigma^2}{2}\right) T + \sigma\sqrt{T} Z\right], \quad Z \sim \mathcal{N}(0, 1).$$

Pour une simulation sur une grille $0 = t_0 < t_1 < \dots < t_N = T$ (nécessaire pour les options path-dependent) :

$$S_{t_{i+1}} = S_{t_i} \exp\left[\left(r - \frac{\sigma^2}{2}\right) \Delta t + \sigma\sqrt{\Delta t} Z_i\right].$$

Monte Carlo pour un call européen

```

import numpy as np

def mc_european(S0, K, T, r, sigma, M=100000, option='call'):
    """Prix Monte Carlo d'une option européenne."""
    Z = np.random.standard_normal(M)
    ST = S0 * np.exp((r - 0.5*sigma**2)*T + sigma*np.sqrt(T)*Z)

    if option == 'call':
        payoffs = np.maximum(ST - K, 0)
    else:
        payoffs = np.maximum(K - ST, 0)

    disc_payoffs = np.exp(-r*T) * payoffs
    price = np.mean(disc_payoffs)
    std_err = np.std(disc_payoffs) / np.sqrt(M)
    ci_95 = (price - 1.96*std_err, price + 1.96*std_err)
    return price, std_err, ci_95

np.random.seed(42)
price, se, ci = mc_european(100, 100, 1.0, 0.05, 0.20)
print(f"Prix MC: {price:.4f} +/- {se:.4f}")
print(f"IC 95%: [{ci[0]:.4f}, {ci[1]:.4f}]")

```

3.3.3 Réduction de variance

Définition 3.5 (Variables antithétiques). Pour chaque Z_i , on calcule aussi le payoff avec $-Z_i$, puis on moyenne :

$$\hat{V} = \frac{e^{-rT}}{M} \sum_{m=1}^M \frac{h(S_T^{(Z_m)}) + h(S_T^{(-Z_m)})}{2}.$$

Définition 3.6 (Variable de contrôle). On utilise un estimateur Y de moyenne connue $\mathbb{E}[Y]$ corrélé au payoff $X = h(S_T)$:

$$\hat{V}_{CV} = \frac{1}{M} \sum_{m=1}^M [X^{(m)} - \beta(Y^{(m)} - \mathbb{E}[Y])],$$

où $\beta = \text{Cov}(X, Y) / \text{Var}(Y)$ est le coefficient optimal. Un choix classique est $Y = S_T$ avec $\mathbb{E}^Q[S_T] = S_0 e^{rT}$.

Monte Carlo avec réduction de variance

```

def mc_antithetic(S0, K, T, r, sigma, M=100000):
    """Monte Carlo avec variables antithétiques."""
    Z = np.random.standard_normal(M)
    ST_plus = S0 * np.exp((r - 0.5*sigma**2)*T + sigma*np.sqrt(T)*Z)
    ST_minus = S0 * np.exp((r - 0.5*sigma**2)*T - sigma*np.sqrt(T)*Z)

```

```

payoffs = 0.5*(np.maximum(ST_plus - K, 0)
               + np.maximum(ST_minus - K, 0))
disc = np.exp(-r*T) * payoffs
return np.mean(disc), np.std(disc)/np.sqrt(M)

def mc_control_variate(S0, K, T, r, sigma, M=100000):
    """Monte Carlo avec variable de controle (S_T)."""
    Z = np.random.standard_normal(M)
    ST = S0 * np.exp((r - 0.5*sigma**2)*T + sigma*np.sqrt(T)*Z)
    X = np.exp(-r*T) * np.maximum(ST - K, 0)
    Y = np.exp(-r*T) * ST
    EY = S0 # E^Q[exp(-rT)*S_T] = S_0

    beta = np.cov(X, Y)[0, 1] / np.var(Y)
    X_cv = X - beta * (Y - EY)
    return np.mean(X_cv), np.std(X_cv)/np.sqrt(M)

np.random.seed(42)
p1, s1 = mc_antithetic(100, 100, 1.0, 0.05, 0.20)
p2, s2 = mc_control_variate(100, 100, 1.0, 0.05, 0.20)
print(f"Antithetique: {p1:.4f} (se={s1:.4f})")
print(f"Controle:      {p2:.4f} (se={s2:.4f})")

```

3.4 Différences finies

L'ÉDP de Black–Scholes peut être résolue numériquement par la méthode des différences finies.

3.4.1 Transformation et discrétisation

On pose $x = \ln(S/K)$, $\tau = T - t$. L'EDP se transforme en :

$$\frac{\partial u}{\partial \tau} = \frac{\sigma^2}{2} \frac{\partial^2 u}{\partial x^2} + \left(r - \frac{\sigma^2}{2}\right) \frac{\partial u}{\partial x} - ru.$$

Définition 3.7 (Schéma explicite). On discrétise sur une grille (x_i, τ_j) :

$$\frac{u_i^{j+1} - u_i^j}{\Delta \tau} = \frac{\sigma^2}{2} \frac{u_{i+1}^j - 2u_i^j + u_{i-1}^j}{(\Delta x)^2} + \nu \frac{u_{i+1}^j - u_{i-1}^j}{2\Delta x} - ru_i^j.$$

Ce schéma est **explicite** (la valeur au pas suivant se calcule directement) mais soumis à une condition de stabilité CFL.

Définition 3.8 (Schéma implicite (Crank–Nicolson)). Le schéma de Crank–Nicolson moyenne les pas explicite et implicite :

$$\frac{u_i^{j+1} - u_i^j}{\Delta \tau} = \frac{1}{2} [\mathcal{L}u_i^j + \mathcal{L}u_i^{j+1}],$$

où $\mathcal{L}$ est l'opérateur différentiel spatial. Ce schéma est inconditionnellement stable et d'ordre 2 en temps et en espace.

Différences finies — Crank–Nicolson

```

import numpy as np
from scipy.linalg import solve_banded

def fd_crank_nicolson(S0, K, T, r, sigma, Nx=200, Nt=500):
    """Prix d'un call européen par Crank-Nicolson."""
    x_max = 5 * sigma * np.sqrt(T)
    dx = 2 * x_max / Nx
    dt = T / Nt
    x = np.linspace(-x_max, x_max, Nx + 1)

    nu = r - 0.5 * sigma**2
    alpha = 0.5 * dt * sigma**2 / dx**2
    beta_val = 0.5 * dt * nu / (2 * dx)

    # Condition terminale (call)
    u = np.maximum(K * (np.exp(x) - 1), 0)

    # Coefficients tridiagonaux
    a = -0.5 * (alpha - beta_val) # sous-diag
    b = 1 + alpha + 0.5 * r * dt # diag
    c = -0.5 * (alpha + beta_val) # sur-diag

    a_e = 0.5 * (alpha - beta_val)
    b_e = 1 - alpha - 0.5 * r * dt
    c_e = 0.5 * (alpha + beta_val)

    # Matrice bande pour solve_banded
    AB = np.zeros((3, Nx + 1))
    AB[0, 1:] = c # sur-diag
    AB[1, :] = b # diag
    AB[2, :-1] = a # sous-diag

    for j in range(Nt):
        rhs = a_e * np.roll(u, 1) + b_e * u + c_e * np.roll(u, -1)
        rhs[0] = 0 # condition aux limites
        rhs[-1] = K * (np.exp(x[-1]) - np.exp(-r * (j+1)*dt))
        AB[1, 0] = 1; AB[0, 1] = 0
        AB[1, -1] = 1; AB[2, -2] = 0
        u = solve_banded((1, 1), AB, rhs)
        AB[1, 0] = b; AB[0, 1] = c
        AB[1, -1] = b; AB[2, -2] = a

    # Interpolation pour S = S0
    x0 = np.log(S0 / K)
    price = np.interp(x0, x, u)
    return price

price_fd = fd_crank_nicolson(100, 100, 1.0, 0.05, 0.20)
print(f"Prix Crank-Nicolson: {price_fd:.4f}")

```

3.5 Comparaison des méthodes

Critère	Arbres	Monte Carlo	Diff. finies
Dimension	faible	élevée	faible
Américaines	oui	difficile	oui
Path-dependent	non	oui	difficile
Convergence	$O(1/N)$	$O(1/\sqrt{M})$	$O(h^2)$
Grecques	diff. finies	bump-and-reprice	directes

3.6 Méthode de Longstaff–Schwartz (options américaines)

Théorème 3.9 (Algorithme LSM). *Pour une option américaine, l'algorithme de Longstaff–Schwartz (2001) estime la valeur de continuation par régression sur des polynômes des prix simulés. À chaque pas de temps t_n en backward :*

1. Calculer le payoff d'exercice immédiat $h(S_{t_n}^{(m)})$.
2. Régresser les flux futurs actualisés sur des fonctions de base $\psi_k(S_{t_n}^{(m)})$ (e.g., polynômes de Laguerre).
3. Exercer si $h(S_{t_n}^{(m)}) > \hat{C}(S_{t_n}^{(m)})$ (continuation estimée).

Longstaff–Schwartz pour put américain

```
def lsm_american_put(S0, K, T, r, sigma, M=50000, N=100):
    """Put américain par Longstaff-Schwartz."""
    dt = T / N
    discount = np.exp(-r * dt)

    # Simulation des trajectoires
    Z = np.random.standard_normal((N, M))
    S = np.zeros((N + 1, M))
    S[0] = S0
    for i in range(N):
        S[i+1] = S[i] * np.exp((r - 0.5*sigma**2)*dt
                               + sigma*np.sqrt(dt)*Z[i])

    # Payoff terminal
    cashflow = np.maximum(K - S[N], 0)
    stopping = np.full(M, N)

    # Backward induction
    for n in range(N - 1, 0, -1):
        exercise = np.maximum(K - S[n], 0)
        itm = exercise > 0

        if np.sum(itm) > 0:
            X = S[n, itm]
```



```

Y = cashflow[itm] * discount**(stopping[itm] - n)
# Regression polynomiale (degre 3)
A = np.column_stack([np.ones_like(X), X, X**2, X**3])
coeffs = np.linalg.lstsq(A, Y, rcond=None)[0]
continuation = A @ coeffs

exercise_now = exercise[itm] > continuation
idx = np.where(itm)[0][exercise_now]
cashflow[idx] = exercise[idx]
stopping[idx] = n

price = np.mean(cashflow * discount**stopping)
return price

np.random.seed(42)
p_am = lsm_american_put(100, 100, 1.0, 0.05, 0.20)
print(f"Put americain (LSM): {p_am:.4f}")

```

3.7 Exercices

- Exercice 3.1** (Convergence des méthodes). 1. Comparer les prix CRR, Monte Carlo et Crank–Nicolson pour un call européen avec les paramètres $S_0 = 100$, $K = 100$, $T = 1$, $r = 0.05$, $\sigma = 0.20$.
2. Tracer l’erreur par rapport à Black–Scholes en fonction du paramètre de discrétisation (N ou M).
3. Mesurer le temps de calcul pour chaque méthode.

- Exercice 3.2** (Réduction de variance). 1. Implémenter les variables antithétiques et la variable de contrôle.
2. Comparer les erreurs standard pour $M = 10^4, 10^5, 10^6$.
3. Calculer le facteur de réduction de variance pour chaque technique.

- Exercice 3.3** (Options américaines). 1. Comparer le prix d’un put américain par l’arbre CRR, par Crank–Nicolson avec exercice anticipé, et par LSM.
2. Déterminer la prime d’exercice anticipé $P_{\text{am}} - P_{\text{eu}}$ en fonction de r et σ .

- Exercice 3.4** (Grecques par Monte Carlo). 1. Calculer Δ et Γ par *bump-and-reprice* : $\Delta \approx (V(S+h) - V(S-h))/(2h)$.
2. Calculer Δ par la méthode des *likelihood ratios* : $\Delta = e^{-rT} \mathbb{E} \left[h(S_T) \frac{Z}{\sigma S_0 \sqrt{T}} \right]$.
3. Comparer les écarts-types des deux estimateurs.

Chapitre 4

Couverture Delta et Gestion des Risques

Le modèle de Black-Scholes ne se contente pas de donner un prix : il dit comment *se couvrir*. L'idée centrale est que le risque d'une option peut être éliminé en détenant en permanence la bonne quantité du sous-jacent — le *delta*. En ajustant cette position à chaque instant, le portefeuille devient insensible aux fluctuations du marché. Mais la couverture parfaite est un idéal : en pratique, on ne peut réajuster qu'à intervalles discrets, les coûts de transaction s'accumulent, et la volatilité n'est jamais constante. Comprendre ces limites est aussi important que comprendre la théorie.

4.1 Introduction

La couverture (*hedging*) est l'activité consistant à réduire ou éliminer le risque lié à une position sur produits dérivés. Ce chapitre développe la théorie et la pratique du delta-hedging, du gamma-hedging, et analyse les coûts et limites de la couverture dynamique.

4.2 Delta-hedging

4.2.1 Principe

Définition 4.1 (Delta-hedging). Le **delta-hedging** consiste à maintenir un portefeuille $\Pi_t = V_t - \Delta_t S_t$ localement insensible aux variations de S , où $\Delta_t = \frac{\partial V}{\partial S}(S_t, t)$.

Théorème 4.2 (Portefeuille auto-finançant delta-couvert). *Dans le cadre de Black-Scholes avec trading continu, le portefeuille delta-couvert est sans risque et réplique parfaitement le dérivé. Sa valeur satisfait :*

$$d\Pi_t = r\Pi_t dt.$$

Couverture en temps discret

En pratique, le rebalancement est discret (toutes les Δt unités de temps). L'erreur de couverture provient du fait que Δ change entre deux rebalancements. Plus la fréquence est élevée, meilleure est la couverture, mais plus les coûts de transaction augmentent.

4.2.2 Erreur de couverture en temps discret

Proposition 4.3 (Erreur de couverture). Pour un call européen couvert en delta avec rebalancement aux dates $t_0, t_1, \dots, t_N$, l'erreur de couverture est :

$$\epsilon = \sum_{i=0}^{N-1} \frac{1}{2} \Gamma(S_{t_i}, t_i) S_{t_i}^2 [(\Delta S_{t_i})^2 - \sigma^2 S_{t_i}^2 \Delta t] \cdot e^{r(T-t_{i+1})},$$

où $\Delta S_{t_i} = S_{t_{i+1}} - S_{t_i}$.

L'erreur de couverture a une espérance nulle sous $\mathbb{Q}$ mais une variance non nulle, proportionnelle à Δt .

Simulation du delta-hedging discret

```
import numpy as np
from scipy.stats import norm

def bs_delta(S, K, T, r, sigma):
    """Delta d'un call europeen."""
    if T < 1e-10:
        return 1.0 if S > K else 0.0
    d1 = (np.log(S/K) + (r + 0.5*sigma**2)*T) / (sigma*np.sqrt(T))
    return norm.cdf(d1)

def bs_price(S, K, T, r, sigma):
    """Prix Black-Scholes d'un call."""
    if T < 1e-10:
        return max(S - K, 0)
    d1 = (np.log(S/K) + (r + 0.5*sigma**2)*T) / (sigma*np.sqrt(T))
    d2 = d1 - sigma*np.sqrt(T)
    return S*norm.cdf(d1) - K*np.exp(-r*T)*norm.cdf(d2)

def simulate_delta_hedge(S0, K, T, r, sigma, N_rebal, M=10000):
    """Simule le delta-hedging et calcule l'erreur de couverture."""
    dt = T / N_rebal
    errors = np.zeros(M)

    for m in range(M):
        S = S0
        # Portefeuille initial: vente du call, achat de delta*S
        V0 = bs_price(S, K, T, r, sigma)
        delta = bs_delta(S, K, T, r, sigma)
        cash = V0 - delta * S # position cash

        for i in range(N_rebal):
            tau = T - (i+1)*dt
            Z = np.random.standard_normal()
            S_new = S * np.exp((r - 0.5*sigma**2)*dt
                               + sigma*np.sqrt(dt)*Z)
            # Cash croît au taux sans risque
            cash *= np.exp(r * dt)
```

```

# Nouveau delta
if tau > 1e-10:
    delta_new = bs_delta(S_new, K, tau, r, sigma)
else:
    delta_new = 1.0 if S_new > K else 0.0
# Rebalancement
cash -= (delta_new - delta) * S_new
delta = delta_new
S = S_new

# Valeur finale du portefeuille
portfolio = cash + delta * S
payoff = max(S - K, 0)
errors[m] = portfolio - payoff

return errors

np.random.seed(42)
for N in [10, 50, 252, 1000]:
    errs = simulate_delta_hedge(100, 100, 1.0, 0.05, 0.20, N)
    print(f"N={N:4d}: mean={np.mean(errs):.4f}, "
          f"std={np.std(errs):.4f}")

```

4.3 Gamma-hedging

Définition 4.4 (Gamma-hedging). Le **gamma-hedging** consiste à rendre le portefeuille insensible non seulement aux petites variations de S (delta-neutre) mais aussi aux variations plus importantes (gamma-neutre). On utilise un deuxième instrument (typiquement une autre option) pour annuler le gamma.

Proposition 4.5 (Construction d'un portefeuille gamma-neutre). Soit V le dérivé à couvrir et O un instrument de couverture avec gamma Γ_O . La position dans O nécessaire pour annuler le gamma est :

$$w_O = -\frac{\Gamma_V}{\Gamma_O}.$$

Après cette couverture, on ajuste la position dans le sous-jacent pour restaurer la delta-neutralité :

$$\Delta_{\text{total}} = \Delta_V + w_O \Delta_O.$$

Exemple 4.6. Un trader est short un call ($\Delta = 0.55$, $\Gamma = 0.03$). Il dispose d'un call de maturité différente ($\Delta_O = 0.40$, $\Gamma_O = 0.05$).

Position gamma-neutre : $w_O = -(-0.03)/0.05 = 0.60$.

Nouveau delta : $-0.55 + 0.60 \times 0.40 = -0.31$.

Il faut acheter 0.31 unités du sous-jacent pour être delta-neutre.

4.4 Vega-hedging

Définition 4.7 (Vega-hedging). Le **vega-hedging** vise à neutraliser la sensibilité du portefeuille à la volatilité. Comme le vega du sous-jacent est nul, il faut utiliser des

options.

Remarque 4.8. En pratique, le vega-hedging est plus important que le gamma-hedging car les mouvements de volatilité implicite peuvent engendrer des profits et pertes (P&L) considérables.

4.5 Décomposition du P&L

Théorème 4.9 (Décomposition du P&L d'un portefeuille delta-couvert). *Le P&L instantané d'un portefeuille delta-couvert s'écrit :*

$$dP\&L = \frac{1}{2}\Gamma S^2[(dS/S)^2 - \sigma^2 dt] + \mathcal{V} d\sigma_{impl} + \dots$$

*Le premier terme est le **P&L gamma** (profit si la volatilité réalisée dépasse la volatilité implicite), le second le **P&L vega**.*

Approximation Taylor du P&L

Pour un portefeuille général, l'approximation Taylor donne :

$$\Delta P\&L \approx \Delta \cdot \delta S + \frac{1}{2}\Gamma(\delta S)^2 + \Theta \delta t + \mathcal{V} \delta \sigma + \rho \delta r.$$

4.6 Coûts de transaction et couverture optimale

Théorème 4.10 (Fréquence de rebalancement optimale (Leland)). *En présence de coûts de transaction proportionnels $k|\delta S|$, Leland (1985) montre que la couverture optimale utilise une volatilité modifiée :*

$$\hat{\sigma}^2 = \sigma^2 \left(1 + \sqrt{\frac{2}{\pi}} \frac{k}{\sigma \sqrt{\Delta t}} \right).$$

Proposition 4.11 (Modèle de Whalley–Wilmott). Une approche alternative consiste à ne rebalancer que lorsque le delta s'écarte suffisamment de sa cible. La bande de non-rebalancement optimale est :

$$\Delta_{\text{cible}} \pm \left(\frac{3k e^{-r\tau} \Gamma^2 S^2}{2\lambda} \right)^{1/3},$$

où λ représente l'aversion au risque du trader.

Couverture avec coûts de transaction

```
def hedge_with_costs(S0, K, T, r, sigma, N_rebal, k=0.001, M=5000):
    """Delta-hedging avec couts de transaction proportionnels."""
    dt = T / N_rebal
    total_costs = np.zeros(M)
    hedge_errors = np.zeros(M)
```

```

for m in range(M):
    S = S0
    V0 = bs_price(S, K, T, r, sigma)
    delta = bs_delta(S, K, T, r, sigma)
    cash = V0 - delta * S
    cost = k * abs(delta) * S # cout initial

    for i in range(N_rebal):
        tau = T - (i+1)*dt
        Z = np.random.standard_normal()
        S_new = S * np.exp((r-0.5*sigma**2)*dt
                           + sigma*np.sqrt(dt)*Z)
        cash *= np.exp(r*dt)

        delta_new = bs_delta(S_new, K, max(tau, 1e-10), r, sigma)
        trade = delta_new - delta
        cost += k * abs(trade) * S_new
        cash -= trade * S_new
        delta = delta_new
        S = S_new

    portfolio = cash + delta * S
    payoff = max(S - K, 0)
    hedge_errors[m] = portfolio - payoff
    total_costs[m] = cost

print(f"Cout moyen: {np.mean(total_costs):.4f}")
print(f"Erreur std: {np.std(hedge_errors):.4f}")
return hedge_errors, total_costs

np.random.seed(42)
hedge_with_costs(100, 100, 1.0, 0.05, 0.20, 252, k=0.001)

```

4.7 Couverture en marché incomplet

Remarque 4.12. Lorsque le marché est incomplet (volatilité stochastique, sauts), la couverture parfaite est impossible. On recourt alors à :

- La **couverture en variance minimale** : minimiser $\text{Var}(\Pi_T - H)$ sur les stratégies admissibles.
- La **couverture par super-réplication** : trouver le coût minimal d'un portefeuille dominant le payoff.
- La **couverture par quantile** : contrôler la probabilité de perte.

4.8 Exercices

Exercice 4.1 (Simulation du delta-hedging). 1. Simuler la couverture delta d'un call européen pour $N = 10, 50, 252, 1000$ rebalancements.

2. Tracer l'histogramme de l'erreur de couverture pour chaque N .
3. Vérifier que la variance de l'erreur décroît en $O(1/N)$.

Exercice 4.2 (Gamma-hedging). 1. Construire un portefeuille delta-gamma neutre à partir d'un call ATM court, d'un call OTM long et du sous-jacent.

2. Simuler le P&L sur une semaine et comparer avec le portefeuille uniquement delta-couvert.

Exercice 4.3 (Analyse du P&L). 1. Décomposer le P&L journalier d'un portefeuille delta-couvert en composantes delta, gamma, theta et vega.

2. Montrer que pour un portefeuille delta-neutre, $P\&L \approx \frac{1}{2}\Gamma S^2(\sigma_r^2 - \sigma_i^2) dt$ où σ_r est la volatilité réalisée et σ_i la volatilité implicite.

Exercice 4.4 (Coûts de transaction). 1. Implémenter la couverture de Leland avec volatilité modifiée.

2. Comparer le coût total et l'erreur de couverture avec la méthode standard pour différentes valeurs de k .
3. Implémenter la bande de Whalley–Wilmott et analyser le compromis coût/risque.

Chapitre 5

Calcul Stochastique en Finance

Le mouvement brownien est la brique fondamentale de la finance quantitative — mais c'est un objet mathématiquement monstrueux : ses trajectoires sont continues mais *nulle part différentiables*. On ne peut pas dériver au sens classique, et l'intégrale ordinaire ne fonctionne plus. C'est Kiyosi Itô qui, dans les années 1940 au Japon, a construit le calcul différentiel adapté à cette situation : l'*intégrale d'Itô* et la célèbre *formule d'Itô*, qui est au calcul stochastique ce que la règle de la chaîne est au calcul classique — avec un terme correctif supplémentaire, le terme en $\frac{1}{2}\sigma^2 f''$, qui change tout. Ce chapitre construit ces outils et montre comment ils s'appliquent à la modélisation financière.

5.1 Mouvement brownien

Définition 5.1 (Mouvement brownien standard). Un **mouvement brownien standard** (ou processus de Wiener) $(W_t)_{t \geq 0}$ est un processus stochastique continu tel que :

1. $W_0 = 0$ p.s.
2. Les trajectoires $t \mapsto W_t(\omega)$ sont continues p.s.
3. Les accroissements sont indépendants : pour $0 \leq s < t \leq u < v$, $W_t - W_s$ et $W_v - W_u$ sont indépendants.
4. Les accroissements sont gaussiens : $W_t - W_s \sim \mathcal{N}(0, t - s)$.

Proposition 5.2 (Propriétés du mouvement brownien). 1. $\mathbb{E}[W_t] = 0$ et $\text{Var}(W_t) = t$ pour tout $t \geq 0$.

2. $\text{Cov}(W_s, W_t) = \min(s, t)$.
3. W_t est une martingale par rapport à sa filtration naturelle.
4. $W_t^2 - t$ est une martingale.
5. Les trajectoires sont p.s. nulle part différentiables.
6. La variation quadratique satisfait $\langle W \rangle_t = t$.

Simulation de trajectoires browniennes

```

import numpy as np
import matplotlib.pyplot as plt

def brownian_paths(T, N, M):
    """Simule M trajectoires browniennes sur [0, T] avec N pas."""
    dt = T / N
    dW = np.sqrt(dt) * np.random.standard_normal((M, N))
    W = np.zeros((M, N + 1))
    W[:, 1:] = np.cumsum(dW, axis=1)
    t = np.linspace(0, T, N + 1)
    return t, W

np.random.seed(42)
t, W = brownian_paths(1.0, 1000, 5)
for i in range(5):
    plt.plot(t, W[i])
plt.xlabel('t'); plt.ylabel('$W_t$')
plt.title('Trajectoires browniennes')
plt.grid(True)
plt.show()

```

5.2 Intégrale d'Itô

5.2.1 Construction

L'intégrale stochastique étend l'intégrale classique aux intégrandes aléatoires par rapport au mouvement brownien.

Définition 5.3 (Intégrale d'Itô). Pour un processus adapté $(H_t)_{t \geq 0}$ vérifiant $\mathbb{E} \left[\int_0^T H_t^2 dt \right] < \infty$, l'intégrale d'Itô est définie comme :

$$I_T(H) = \int_0^T H_t dW_t = \lim_{n \rightarrow \infty} \sum_{i=0}^{n-1} H_{t_i} (W_{t_{i+1}} - W_{t_i}).$$

La limite est prise en moyenne quadratique (L^2).

Théorème 5.4 (Propriétés de l'intégrale d'Itô). Soit $I_t = \int_0^t H_s dW_s$. Alors :

1. I_t est une martingale : $\mathbb{E}[I_t | \mathcal{F}_s] = I_s$ pour $s \leq t$.
2. **Isométrie d'Itô** : $\mathbb{E}[I_T^2] = \mathbb{E} \left[\int_0^T H_t^2 dt \right]$.
3. $\mathbb{E}[I_T] = 0$.
4. La variation quadratique est $\langle I \rangle_t = \int_0^t H_s^2 ds$.

Itô vs Stratonovich

L'intégrale d'Itô évalue l'intégrande au bord **gauche** de chaque intervalle : H_{t_i} et non $H_{t_{i+1}}$ ou $\frac{1}{2}(H_{t_i} + H_{t_{i+1}})$. Le choix de Stratonovich (évaluation au milieu) donne un résultat différent et satisfait les règles de calcul classiques, mais l'intégrale d'Itô est privilégiée en finance car elle préserve la propriété de martingale (et donc le pricing risque-neutre).

5.2.2 Règles de calcul

Les règles de calcul diffèrent du calcul classique. On retient :

$$(dW_t)^2 = dt, \quad (dt)^2 = 0, \quad dW_t \cdot dt = 0.$$

5.3 Formule d'Itô

Théorème 5.5 (Formule d'Itô — cas unidimensionnel). *Soit (X_t) un processus d'Itô satisfaisant $dX_t = \mu_t dt + \sigma_t dW_t$ et $f \in C^{1,2}(\mathbb{R}^+ \times \mathbb{R})$. Alors :*

$$df(t, X_t) = \frac{\partial f}{\partial t} dt + \frac{\partial f}{\partial x} dX_t + \frac{1}{2} \frac{\partial^2 f}{\partial x^2} \sigma_t^2 dt.$$

Sous forme développée :

$$df(t, X_t) = \left(\frac{\partial f}{\partial t} + \mu_t \frac{\partial f}{\partial x} + \frac{1}{2} \sigma_t^2 \frac{\partial^2 f}{\partial x^2} \right) dt + \sigma_t \frac{\partial f}{\partial x} dW_t.$$

Formule d'Itô — forme compacte

$$df = \left(\partial_t f + \mu \partial_x f + \frac{1}{2} \sigma^2 \partial_{xx} f \right) dt + \sigma \partial_x f dW_t.$$

Le terme supplémentaire $\frac{1}{2} \sigma^2 \partial_{xx} f dt$ n'a pas d'analogue en calcul déterministe ; il provient de la variation quadratique non nulle du brownien.

Exemple 5.6. Appliquons la formule d'Itô à $f(x) = x^2$ avec $X_t = W_t$:

$$d(W_t^2) = 2W_t dW_t + \frac{1}{2} \cdot 2 dt = 2W_t dW_t + dt.$$

En intégrant : $W_T^2 = 2 \int_0^T W_t dW_t + T$, d'où :

$$\int_0^T W_t dW_t = \frac{W_T^2 - T}{2}.$$

On note la différence avec le calcul classique qui donnerait $W_T^2/2$.

Théorème 5.7 (Formule d'Itô multidimensionnelle). *Soient $dX_t^i = \mu_t^i dt + \sum_j \sigma_t^{ij} dW_t^j$ et $f \in C^{1,2}$. Alors :*

$$df = \partial_t f dt + \sum_i \partial_{x_i} f dX_t^i + \frac{1}{2} \sum_{i,j} \partial_{x_i x_j} f d\langle X^i, X^j \rangle_t.$$

5.4 Théorème de Girsanov

Théorème 5.8 (Girsanov). Soit (W_t) un $\mathbb{P}$ -brownien et (θ_t) un processus adapté vérifiant la condition de Novikov :

$$\mathbb{E}^{\mathbb{P}} \left[\exp \left(\frac{1}{2} \int_0^T \theta_t^2 dt \right) \right] < \infty.$$

Définissons la densité de Radon–Nikodym :

$$\left. \frac{d\mathbb{Q}}{d\mathbb{P}} \right|_{\mathcal{F}_T} = \mathcal{E}_T(\theta) = \exp \left(- \int_0^T \theta_t dW_t - \frac{1}{2} \int_0^T \theta_t^2 dt \right).$$

Alors sous $\mathbb{Q}$, le processus $\tilde{W}_t = W_t + \int_0^t \theta_s ds$ est un mouvement brownien standard.

Application à la finance

Le théorème de Girsanov permet de passer de la mesure historique $\mathbb{P}$ (sous laquelle $dS_t/S_t = \mu dt + \sigma dW_t$) à la mesure risque-neutre $\mathbb{Q}$ (sous laquelle $dS_t/S_t = r dt + \sigma d\tilde{W}_t$) en posant $\theta = (\mu - r)/\sigma$. La quantité θ est le **prix du risque de marché** (*market price of risk*).

5.5 Théorème de représentation des martingales

Théorème 5.9 (Représentation des martingales). Toute martingale de carré intégrable $(M_t)_{0 \leq t \leq T}$ adaptée à la filtration du brownien peut s'écrire :

$$M_t = M_0 + \int_0^t H_s dW_s,$$

pour un unique processus prévisible (H_t) vérifiant $\mathbb{E} \left[\int_0^T H_t^2 dt \right] < \infty$.

Remarque 5.10. Ce théorème garantit que dans le modèle de Black–Scholes (marché complet), tout actif contingent de carré intégrable est répliquable. Le processus H_t fournit la stratégie de couverture.

5.6 Équations différentielles stochastiques

Définition 5.11 (ÉDS générale). Une **équation différentielle stochastique** (EDS) est de la forme :

$$dX_t = b(t, X_t) dt + \sigma(t, X_t) dW_t, \quad X_0 = x_0,$$

où b est le coefficient de drift et σ le coefficient de diffusion.

Théorème 5.12 (Existence et unicité). Si b et σ sont lipschitziennes en x uniformément en t et à croissance linéaire, alors l'EDS admet une unique solution forte $(X_t)_{0 \leq t \leq T}$.

Théorème 5.13 (Lien EDP–EDS (Feynman–Kac)). Soit $u(t, x)$ solution de :

$$\partial_t u + b(t, x) \partial_x u + \frac{1}{2} \sigma^2(t, x) \partial_{xx} u - r u = 0, \quad u(T, x) = g(x).$$

Alors :

$$u(t, x) = \mathbb{E} \left[e^{-r(T-t)} g(X_T) \mid X_t = x \right],$$

où $dX_s = b(s, X_s) ds + \sigma(s, X_s) dW_s$.

Remarque 5.14. Le théorème de Feynman–Kac établit le lien fondamental entre le pricing par EDP (Black–Scholes) et le pricing par espérance risque-neutre (Monte Carlo).

5.7 Exponentielle stochastique

Définition 5.15 (Exponentielle stochastique). L'**exponentielle stochastique** (ou de Doléans-Dade) d'un processus d'Itô X_t est :

$$\mathcal{E}_t(X) = \exp\left(X_t - X_0 - \frac{1}{2}\langle X \rangle_t\right).$$

Elle satisfait $d\mathcal{E}_t = \mathcal{E}_t dX_t$, $\mathcal{E}_0 = 1$.

Proposition 5.16. Le MBG $S_t = S_0 \exp[(\mu - \sigma^2/2)t + \sigma W_t]$ est l'exponentielle stochastique de $\mu t + \sigma W_t$:

$$S_t = S_0 \mathcal{E}_t(\mu \cdot + \sigma W).$$

5.8 Implémentation : schémas de discrétisation

Schémas d'Euler et de Milstein

```
import numpy as np

def euler_maruyama(x0, b, sigma, T, N, M):
    """Schema d'Euler-Maruyama pour dX = b(X)dt + sigma(X)dW."""
    dt = T / N
    X = np.full(M, x0, dtype=float)
    for i in range(N):
        dW = np.sqrt(dt) * np.random.standard_normal(M)
        X = X + b(X)*dt + sigma(X)*dW
    return X

def milstein(x0, b, sigma, sigma_prime, T, N, M):
    """Schema de Milstein (ordre 1.0 fort)."""
    dt = T / N
    X = np.full(M, x0, dtype=float)
    for i in range(N):
        dW = np.sqrt(dt) * np.random.standard_normal(M)
        X = (X + b(X)*dt + sigma(X)*dW
              + 0.5*sigma(X)*sigma_prime(X)*(dW**2 - dt))
    return X

# Application au MBG: dS = r*S*dt + sig*S*dW
r_val, sig = 0.05, 0.20
S0, T, N, M = 100, 1.0, 100, 100000

np.random.seed(42)
ST_euler = euler_maruyama(S0, lambda S: r_val*S, lambda S: sig*S,
                           T, N, M)
ST_milstein = milstein(S0, lambda S: r_val*S, lambda S: sig*S,
```

```

lambda S: sig, T, N, M)

# Solution exacte
Z = np.random.standard_normal(M)
ST_exact = S0 * np.exp((r_val - 0.5*sig**2)*T + sig*np.sqrt(T)*Z)

print(f"Euler:      E[S_T] = {np.mean(ST_euler):.2f}")
print(f"Milstein:   E[S_T] = {np.mean(ST_milstein):.2f}")
print(f"Exact:       E[S_T] = {S0*np.exp(r_val*T):.2f}")

```

5.9 Exercices

Exercice 5.1 (Formule d'Itô). 1. Calculer $d(e^{W_t})$ et $d(e^{\alpha W_t - \alpha^2 t/2})$.

2. Montrer que $\mathcal{E}_t = e^{\sigma W_t - \sigma^2 t/2}$ est une martingale.

3. Appliquer la formule d'Itô à $f(t, x) = e^{-rt}x$ pour retrouver que $e^{-rt}S_t$ est une $\mathbb{Q}$ -martingale.

Exercice 5.2 (Girsanov). 1. Vérifier explicitement le changement de mesure pour le MBG : écrire S_t sous $\mathbb{P}$ puis sous $\mathbb{Q}$.

2. Calculer le prix du risque de marché $\theta = (\mu - r)/\sigma$ pour $\mu = 0.10$, $r = 0.03$, $\sigma = 0.25$.

3. Montrer que la densité de Radon–Nikodym $\mathcal{E}_T(\theta)$ est une $\mathbb{P}$ -martingale.

Exercice 5.3 (Convergence des schémas). 1. Comparer l'erreur forte $\mathbb{E}[|S_T^N - S_T|]$ des schémas d'Euler et de Milstein en fonction de N .

2. Vérifier que l'ordre de convergence forte est $1/2$ pour Euler et 1 pour Milstein.

3. Implémenter un schéma d'ordre faible 2 (Platen) et comparer.

Exercice 5.4 (Feynman–Kac). 1. Vérifier le théorème de Feynman–Kac pour le call européen en comparant la solution de l'EDP de Black–Scholes avec l'espérance Monte Carlo.

2. Appliquer Feynman–Kac à l'équation de la chaleur $\partial_t u = \frac{1}{2}\partial_{xx}u$ et interpréter.

Chapitre 6

Modèles à Volatilité Stochastique

Le modèle de Black-Scholes suppose que la volatilité est constante. Mais les marchés racontent une autre histoire : lorsqu'on calcule la volatilité implicite à partir des prix d'options observés, elle varie avec le strike et la maturité, dessinant le célèbre « smile de volatilité ». Ce phénomène, devenu flagrant après le krach de 1987, a motivé une nouvelle génération de modèles où la volatilité elle-même est un processus stochastique. Le modèle de Heston (1993), le modèle SABR (Hagan et al., 2002), et les modèles à volatilité locale de Dupire (1994) tentent chacun de capturer cette dynamique.

6.1 Motivation : le smile de volatilité

Le modèle de Black-Scholes suppose une volatilité constante σ . Or, en pratique, la volatilité implicite $\sigma_{\text{impl}}(K, T)$ dépend du strike K et de la maturité T , formant une **surface de volatilité** présentant un *smile* ou *skew* caractéristique.

Définition 6.1 (Smile de volatilité). Le **smile de volatilité** est la courbe $K \mapsto \sigma_{\text{impl}}(K, T)$ pour une maturité fixée. Sur les marchés actions, on observe généralement un *skew* négatif : la volatilité implicite est plus élevée pour les puts OTM (strikes bas) que pour les calls OTM (strikes hauts).

Remarque 6.2. Le smile est incompatible avec le modèle de Black-Scholes et motive l'introduction de modèles à volatilité stochastique ou à sauts.

6.2 Volatilité locale (Dupire)

Théorème 6.3 (Équation de Dupire (1994)). Si les prix d'options européennes $C(K, T)$ sont connus pour tout (K, T) , la **volatilité locale** $\sigma_{\text{loc}}(K, T)$ est définie par :

$$\sigma_{\text{loc}}^2(K, T) = \frac{\partial_T C + rK \partial_K C}{\frac{1}{2} K^2 \partial_{KK} C}.$$

Le sous-jacent satisfait alors : $dS_t = rS_t dt + \sigma_{\text{loc}}(S_t, t)S_t dW_t^{\mathbb{Q}}$.

Limites de la volatilité locale

Le modèle de Dupire reproduit exactement la surface de volatilité aujourd'hui, mais ses prédictions de dynamique future du smile sont souvent irréalistes (le smile s'apla-

tit trop vite). Les modèles à volatilité stochastique offrent une meilleure dynamique.

6.3 Modèle de Heston

Définition 6.4 (Modèle de Heston (1993)). Le **modèle de Heston** décrit le sous-jacent et sa variance par le système :

$$dS_t = rS_t dt + \sqrt{v_t} S_t dW_t^1, \quad (6.1)$$

$$dv_t = \kappa(\bar{v} - v_t) dt + \xi \sqrt{v_t} dW_t^2, \quad (6.2)$$

avec $d\langle W^1, W^2 \rangle_t = \rho dt$ et les paramètres :

- $v_0 > 0$: variance initiale,
- $\kappa > 0$: vitesse de retour à la moyenne,
- $\bar{v} > 0$: variance de long terme,
- $\xi > 0$: volatilité de la variance (*vol of vol*),
- $\rho \in [-1, 1]$: corrélation ($\rho < 0$ pour le skew).

Théorème 6.5 (Condition de Feller). *La variance v_t reste strictement positive si et seulement si la **condition de Feller** est satisfaite :*

$$2\kappa\bar{v} \geq \xi^2.$$

6.3.1 Pricing par fonction caractéristique

Théorème 6.6 (Formule semi-analytique de Heston). *Le prix d'un call européen dans le modèle de Heston est :*

$$C = S_0 P_1 - K e^{-rT} P_2,$$

où les probabilités P_j sont obtenues par inversion de Fourier :

$$P_j = \frac{1}{2} + \frac{1}{\pi} \int_0^\infty \operatorname{Re} \left[\frac{e^{-iu \ln K} \varphi_j(u)}{iu} \right] du,$$

et $\varphi_j(u)$ est la fonction caractéristique du log-prix sous les mesures associées.

Pricing Heston par transformée de Fourier

```
import numpy as np
from scipy.integrate import quad

def heston_char_func(u, T, r, v0, kappa, vbar, xi, rho):
    """Fonction caractéristique du log-prix sous Heston."""
    d = np.sqrt((rho*xi*1j*u - kappa)**2 + xi**2*(1j*u + u**2))
    g = (kappa - rho*xi*1j*u - d) / (kappa - rho*xi*1j*u + d)

    C = r*1j*u*T + (kappa*vbar/xi**2) * (
```



```

        (kappa - rho*xi*1j*u - d)*T
        - 2*np.log((1 - g*np.exp(-d*T))/(1 - g))
    )
    D = ((kappa - rho*xi*1j*u - d)/xi**2) * (
        (1 - np.exp(-d*T))/(1 - g*np.exp(-d*T))
    )
    return np.exp(C + D*v0)

def heston_call(S0, K, T, r, v0, kappa, vbar, xi, rho):
    """Prix d'un call europeen dans le modele de Heston."""
    ln_K = np.log(K)
    ln_S = np.log(S0)

    def integrand(u):
        phi = heston_char_func(u - 0.5j, T, r, v0,
                                kappa, vbar, xi, rho)
        return np.real(np.exp(-1j*u*ln_K) * phi / (u**2 + 0.25))

    integral, _ = quad(integrand, 0, 200, limit=200)
    return S0 - np.sqrt(S0*K)*np.exp(-r*T)/np.pi * integral

# Parametres Heston
S0, K, T, r = 100, 100, 1.0, 0.05
v0, kappa, vbar, xi, rho = 0.04, 2.0, 0.04, 0.3, -0.7

price = heston_call(S0, K, T, r, v0, kappa, vbar, xi, rho)
print(f"Prix Heston: {price:.4f}")

```

6.4 Modèle SABR

Définition 6.7 (Modèle SABR). Le modèle **SABR** (Stochastic Alpha Beta Rho, Hagan et al., 2002) est défini par :

$$dF_t = \alpha_t F_t^\beta dW_t^1, \quad (6.3)$$

$$d\alpha_t = \nu \alpha_t dW_t^2, \quad (6.4)$$

avec $d\langle W^1, W^2 \rangle_t = \rho dt$, $F_0 = f$ (taux forward), $\alpha_0 = \alpha$, et $\beta \in [0, 1]$.

Théorème 6.8 (Approximation de Hagan pour la volatilité implicite). *La volatilité implicite Black dans le modèle SABR est approximée par :*

$$\sigma_B(K, f) \approx \frac{\alpha}{(fK)^{(1-\beta)/2} \left[1 + \frac{(1-\beta)^2}{24} \ln^2 \frac{f}{K} + \dots \right]} \cdot \frac{z}{x(z)} \cdot \left[1 + \left(\frac{(1-\beta)^2}{24} \frac{\alpha^2}{(fK)^{1-\beta}} + \frac{1}{4} \frac{\rho\beta\nu\alpha}{(fK)^{(1-\beta)/2}} + \frac{2-3\rho}{24} \right) \right]$$

$$\text{où } z = \frac{\nu}{\alpha} (fK)^{(1-\beta)/2} \ln \frac{f}{K} \text{ et } x(z) = \ln \frac{\sqrt{1-2\rho z + z^2} + z - \rho}{1-\rho}.$$

Volatilité implicite SABR

```

import numpy as np

def sabr_vol(K, f, T, alpha, beta, rho, nu):
    """Approximation de Hagan pour la vol implicite SABR."""
    if abs(f - K) < 1e-10:
        # ATM
        fk = f**(1 - beta)
        vol = (alpha / fk) * (
            1 + ((1-beta)**2/24 * alpha**2/fk**2
                + 0.25*rho*beta*nu*alpha/fk
                + (2-3*rho**2)/24 * nu**2) * T
        )
        return vol

    fk_mid = (f * K)**((1 - beta)/2)
    ln_fk = np.log(f / K)
    z = nu / alpha * fk_mid * ln_fk
    x_z = np.log((np.sqrt(1 - 2*rho*z + z**2) + z - rho)
                 / (1 - rho))

    prefix = alpha / (fk_mid * (1 + (1-beta)**2/24 * ln_fk**2
                                   + (1-beta)**4/1920 * ln_fk**4))
    correction = 1 + ((1-beta)**2/24 * alpha**2 / (f*K)**(1-beta)
                      + 0.25*rho*beta*nu*alpha / fk_mid
                      + (2-3*rho**2)/24 * nu**2) * T

    return prefix * z / x_z * correction

# Exemple: smile SABR
f, T = 0.03, 5.0
alpha, beta, rho, nu = 0.02, 0.5, -0.3, 0.4
strikes = np.linspace(0.01, 0.06, 50)
vols = [sabr_vol(K, f, T, alpha, beta, rho, nu) for K in strikes]

```

6.5 Calibration

Définition 6.9 (Problème de calibration). La **calibration** consiste à déterminer les paramètres du modèle θ^* minimisant l'écart avec les prix (ou volatilités implicites) de marché :

$$\theta^* = \arg \min_{\theta} \sum_{i=1}^N w_i (\sigma_{\text{mod}}(K_i, T_i; \theta) - \sigma_{\text{mkt}}(K_i, T_i))^2.$$

Calibration du modèle de Heston

```

from scipy.optimize import minimize
from scipy.stats import norm

```

```
def calibrate_heston(market_data, S0, r):
    """Calibration de Heston sur des vol implicites de marche.

    market_data: list of (K, T, sigma_mkt)
    """
    def objective(params):
        v0, kappa, vbar, xi, rho = params
        if v0<0 or kappa<0 or vbar<0 or xi<0 or abs(rho)>1:
            return 1e10
        total = 0
        for K, T, sigma_mkt in market_data:
            try:
                C_mod = heston_call(S0, K, T, r, v0, kappa,
                                    vbar, xi, rho)
                # Vol implicite du prix modele
                sigma_mod = implied_vol_from_price(
                    C_mod, S0, K, T, r)
                total += (sigma_mod - sigma_mkt)**2
            except:
                total += 1e6
        return total

    x0 = [0.04, 1.5, 0.04, 0.3, -0.5]
    bounds = [(0.001,1), (0.01,10), (0.001,1),
              (0.01,2), (-0.99,0.99)]
    result = minimize(objective, x0, method='L-BFGS-B',
                      bounds=bounds)
    return result.x
```

6.6 Volatilité locale vs stochastique

Propriété	Locale (Dupire)	Stochastique (Heston)
Calibration exacte	Oui	Approchée
Dynamique du smile	Irréaliste	Réaliste
Nombre de facteurs	1	2
Marché	Complet	Incomplet
Grecques	Simple	Complexes

6.7 Modèles à volatilité locale stochastique (SLV)

Définition 6.10 (Modèle SLV). Les modèles **SLV** combinent volatilité locale et stochastique :

$$dS_t = rS_t dt + L(S_t, t)\sqrt{v_t} S_t dW_t^1,$$

où $L(S, t)$ est la fonction de *leverage* calibrée pour reproduire exactement la surface de volatilité, tandis que la dynamique de v_t contrôle le comportement futur du smile.

6.8 Exercices

Exercice 6.1 (Surface de volatilité). 1. Générer les prix d'options dans le modèle de Heston pour $K/S_0 \in \{0.8, 0.9, 1.0, 1.1, 1.2\}$ et $T \in \{0.25, 0.5, 1, 2\}$.

2. Calculer la volatilité implicite Black–Scholes pour chaque prix.

3. Tracer le smile pour chaque maturité et commenter l'impact de ρ .

Exercice 6.2 (Condition de Feller). 1. Pour $\kappa = 2$, $\bar{v} = 0.04$, déterminer la valeur maximale de ξ satisfaisant Feller.

2. Simuler des trajectoires de v_t avec et sans Feller et observer le comportement près de zéro.

Exercice 6.3 (Calibration SABR). 1. Calibrer le modèle SABR sur des données de swaptions (volatilités implicites pour différents strikes).

2. Analyser la stabilité des paramètres (α, ρ, ν) en fonction de la maturité.

3. Comparer avec la calibration Heston sur les mêmes données.

Exercice 6.4 (Monte Carlo pour Heston). 1. Implémenter le schéma d'Euler tronqué pour le modèle de Heston : $v_{t+\Delta t} = \max(v_t + \kappa(\bar{v} - v_t)\Delta t + \xi\sqrt{v_t^+}\Delta W^2, 0)$.

2. Comparer avec la méthode de la fonction caractéristique.

3. Étudier l'impact du paramètre ρ sur le skew du smile.

Chapitre 7

Modèles de Taux d'Intérêt

Jusqu'ici, nous avons traité le taux d'intérêt comme une constante. Mais quiconque a observé les marchés obligataires sait que les taux sont tout sauf constants : ils forment une *courbe* — la structure par terme — qui évolue au fil du temps de manière stochastique. Modéliser cette évolution est l'un des défis les plus riches de la finance quantitative. Oldrich Vasicek, en 1977, propose le premier modèle stochastique de taux court avec retour à la moyenne, s'inspirant du processus d'Ornstein–Uhlenbeck. John Cox, Jonathan Ingersoll et Stephen Ross (1985) corrigent le défaut de Vasicek — la possibilité de taux négatifs — en introduisant une volatilité proportionnelle à $\sqrt{r}$. Puis David Heath, Robert Jarrow et Andrew Morton (1992) renversent la perspective : au lieu de modéliser le taux court, ils modélisent directement toute la courbe forward. Chaque approche répond à des besoins différents, et les comprendre, c'est comprendre comment les marchés évaluent le temps et l'incertitude.

7.1 Structure par terme des taux

Définition 7.1 (Zéro-coupon et courbe des taux). Une **obligation zéro-coupon** de maturité T est un contrat payant 1 euro à la date T . Son prix à la date t est noté $P(t, T)$. Le **taux zéro-coupon continu** est :

$$R(t, T) = -\frac{\ln P(t, T)}{T - t}.$$

La courbe $T \mapsto R(t, T)$ est la **structure par terme** des taux à la date t .

Définition 7.2 (Taux forward instantané). Le **taux forward instantané** est :

$$f(t, T) = -\frac{\partial \ln P(t, T)}{\partial T},$$

de sorte que $P(t, T) = \exp\left(-\int_t^T f(t, u) du\right)$.

Définition 7.3 (Taux court). Le **taux court** (ou taux instantané) est : $r_t = f(t, t) = \lim_{T \rightarrow t^+} R(t, T)$.

7.2 Modèles de taux court

7.2.1 Modèle de Vasicek (1977)

Définition 7.4 (Modèle de Vasicek). Le taux court suit un processus d'Ornstein–Uhlenbeck :

$$dr_t = a(b - r_t) dt + \sigma dW_t,$$

avec $a > 0$ (vitesse de retour à la moyenne), b (taux de long terme), $\sigma > 0$ (volatilité).

Théorème 7.5 (Prix du zéro-coupon — Vasicek). *Le prix du zéro-coupon est de la forme affine :*

$$P(t, T) = A(t, T) \exp(-B(t, T) r_t),$$

où, avec $\tau = T - t$:

$$B(\tau) = \frac{1 - e^{-a\tau}}{a}, \quad (7.1)$$

$$A(\tau) = \exp \left[\left(b - \frac{\sigma^2}{2a^2} \right) (B(\tau) - \tau) - \frac{\sigma^2}{4a} B(\tau)^2 \right]. \quad (7.2)$$

Remarque 7.6. Le modèle de Vasicek permet des taux négatifs, ce qui était considéré comme un défaut jusqu'aux années 2010, mais s'est révélé réaliste dans l'environnement de taux négatifs européens.

7.2.2 Modèle de Cox–Ingersoll–Ross (CIR, 1985)

Définition 7.7 (Modèle CIR).

$$dr_t = a(b - r_t) dt + \sigma \sqrt{r_t} dW_t.$$

Le terme $\sqrt{r_t}$ dans la diffusion garantit $r_t \geq 0$ sous la condition de Feller $2ab \geq \sigma^2$.

Théorème 7.8 (Prix du zéro-coupon — CIR). *Le prix est affine en r_t : $P(t, T) = A(\tau) \exp(-B(\tau) r_t)$ avec :*

$$B(\tau) = \frac{2(e^{\gamma\tau} - 1)}{(\gamma + a)(e^{\gamma\tau} - 1) + 2\gamma}, \quad (7.3)$$

$$A(\tau) = \left[\frac{2\gamma e^{(a+\gamma)\tau/2}}{(\gamma + a)(e^{\gamma\tau} - 1) + 2\gamma} \right]^{2ab/\sigma^2}, \quad (7.4)$$

où $\gamma = \sqrt{a^2 + 2\sigma^2}$.

7.2.3 Modèle de Hull–White (1990)

Définition 7.9 (Modèle de Hull–White). Extension de Vasicek avec paramètres dépendant du temps :

$$dr_t = [\theta(t) - a r_t] dt + \sigma dW_t,$$

où $\theta(t)$ est choisie pour calibrer exactement la courbe des taux initiale :

$$\theta(t) = \frac{\partial f^M(0, t)}{\partial t} + a f^M(0, t) + \frac{\sigma^2}{2a} (1 - e^{-2at}),$$

avec $f^M(0, t)$ le taux forward de marché.

Modèles de Vasicek et CIR

```
import numpy as np

def vasicek_bond(r, t, T, a, b, sigma):
    """Prix zero-coupon Vasicek."""
    tau = T - t
    B = (1 - np.exp(-a * tau)) / a
    A = np.exp((b - sigma**2/(2*a**2)) * (B - tau)
               - sigma**2/(4*a) * B**2)
    return A * np.exp(-B * r)

def cir_bond(r, t, T, a, b, sigma):
    """Prix zero-coupon CIR."""
    tau = T - t
    gamma = np.sqrt(a**2 + 2*sigma**2)
    eg = np.exp(gamma * tau)
    denom = (gamma + a)*(eg - 1) + 2*gamma
    B = 2*(eg - 1) / denom
    A = (2*gamma * np.exp((a + gamma)*tau/2) / denom)**(2*a*b/sigma**2)
    return A * np.exp(-B * r)

# Courbe zero-coupon
r0 = 0.03
maturities = np.linspace(0.25, 30, 100)
P_vas = [vasicek_bond(r0, 0, T, 0.5, 0.04, 0.01) for T in maturities]
P_cir = [cir_bond(r0, 0, T, 0.5, 0.04, 0.06) for T in maturities]
R_vas = -np.log(P_vas) / maturities
R_cir = -np.log(P_cir) / maturities

print(f"Vasicek R(10y): {R_vas[39]:.4f}")
print(f"CIR      R(10y): {R_cir[39]:.4f}")
```

7.3 Cadre Heath–Jarrow–Morton (HJM)

Théorème 7.10 (Cadre HJM (1992)). *Le cadre HJM modélise directement les taux forward instantanés :*

$$df(t, T) = \alpha(t, T) dt + \sigma(t, T) dW_t.$$

*La condition d'absence d'arbitrage impose la **condition de drift HJM** :*

$$\alpha(t, T) = \sigma(t, T) \int_t^T \sigma(t, u) du.$$

Remarque 7.11. Le cadre HJM est très général mais conduit généralement à des modèles non markoviens (sauf pour des choix spécifiques de σ). Les modèles de Vasicek et Hull–White sont des cas particuliers de HJM.

7.4 Pricing de produits de taux

7.4.1 Caps et floors

Définition 7.12 (Caplet). Un **caplet** sur la période $[T_i, T_{i+1}]$ avec taux d'exercice K et nominal N paie :

$$N \delta_i (L(T_i, T_i, T_{i+1}) - K)^+,$$

où $L(T_i, T_i, T_{i+1})$ est le taux LIBOR et $\delta_i = T_{i+1} - T_i$.

Théorème 7.13 (Formule de Black pour les caplets). *Dans le modèle de Black, le prix du caplet est :*

$$\text{Caplet} = N \delta_i P(0, T_{i+1}) [L_0 \Phi(d_1) - K \Phi(d_2)],$$

avec $d_1 = \frac{\ln(L_0/K) + \sigma_i^2 T_i / 2}{\sigma_i \sqrt{T_i}}$, $L_0 = L(0, T_i, T_{i+1})$.

7.4.2 Swaptions

Définition 7.14 (Swaption). Une **swaption** donne le droit d'entrer dans un swap de taux à une date future. Une swaption payeuse de strike K et maturité T_0 sur un swap de dates $T_0, \dots, T_n$ a pour payoff :

$$\left(\sum_{i=0}^{n-1} \delta_i P(T_0, T_{i+1}) \right) (S_{T_0} - K)^+,$$

où S_{T_0} est le taux swap au pair.

Simulation de trajectoires de taux (Vasicek)

```
def vasicek_paths(r0, a, b, sigma, T, N, M):
    """Simule M trajectoires du taux court Vasicek."""
    dt = T / N
    r = np.zeros((M, N + 1))
    r[:, 0] = r0
    for i in range(N):
        dW = np.sqrt(dt) * np.random.standard_normal(M)
        r[:, i+1] = r[:, i] + a*(b - r[:, i])*dt + sigma*dW
    return r

def cir_paths(r0, a, b, sigma, T, N, M):
    """Simule M trajectoires du taux court CIR (Euler tronque)."""
    dt = T / N
    r = np.zeros((M, N + 1))
    r[:, 0] = r0
    for i in range(N):
        dW = np.sqrt(dt) * np.random.standard_normal(M)
        r_pos = np.maximum(r[:, i], 0)
        r[:, i+1] = r[:, i] + a*(b - r_pos)*dt + sigma*np.sqrt(r_pos)*dW
        r[:, i+1] = np.maximum(r[:, i+1], 0)
    return r

np.random.seed(42)
```



```
r_vas = vasicek_paths(0.03, 0.5, 0.04, 0.01, 10, 2520, 5)
r_cir = cir_paths(0.03, 0.5, 0.04, 0.06, 10, 2520, 5)
```

7.5 Modèles de marché (LIBOR Market Model)

Définition 7.15 (Modèle BGM/LMM). Le **LIBOR Market Model** (Brace–Gatarek–Musiela) modélise directement les taux forward LIBOR $L_i(t) = L(t, T_i, T_{i+1})$ sous la mesure forward T_{i+1} :

$$dL_i(t) = \sigma_i(t)L_i(t) dW_t^{T_{i+1}}.$$

Sous d'autres mesures, un drift apparaît dépendant des corrélations entre taux forward.

7.6 Exercices

Exercice 7.1 (Courbe des taux). 1. Calculer et tracer la courbe des taux $T \mapsto R(0, T)$ pour Vasicek et CIR avec $r_0 = 0.03$, $a = 0.5$, $b = 0.04$, $\sigma_V = 0.01$, $\sigma_C = 0.06$.

2. Comparer les formes de courbe (croissante, décroissante, bossée).

3. Étudier l'impact des paramètres a , b , σ sur la forme.

Exercice 7.2 (Pricing Monte Carlo d'un zéro-coupon). 1. Pricer une obligation zéro-coupon 5 ans par Monte Carlo dans le modèle de Vasicek : $P(0, T) = \mathbb{E}^{\mathbb{Q}} \left[e^{-\int_0^T r_t dt} \right]$.

2. Comparer avec la formule analytique.

3. Répéter pour le modèle CIR.

Exercice 7.3 (Caplet pricing). 1. Pricer un caplet à 1 an avec la formule de Black.

2. Pricer le même caplet par Monte Carlo dans le modèle de Hull–White.

3. Comparer les résultats.

Exercice 7.4 (Calibration Hull–White). 1. Calibrer le modèle Hull–White sur une courbe de taux donnée en déterminant $\theta(t)$.

2. Vérifier que les prix zéro-coupon du modèle coïncident avec les prix de marché.

Chapitre 8

Optimisation de Portefeuille

En 1952, Harry Markowitz, alors étudiant en doctorat à l'université de Chicago, publie un article de quatorze pages qui va révolutionner la finance : « Portfolio Selection ». Son idée est simple mais profonde : un investisseur ne devrait pas choisir ses actifs individuellement, mais optimiser son *portefeuille* dans son ensemble, en tenant compte des corrélations entre les rendements. La diversification — “ne pas mettre tous ses œufs dans le même panier” — reçoit ainsi une fondation mathématique rigoureuse, exprimée comme un problème d'optimisation quadratique sous contraintes. Markowitz recevra le prix Nobel d'économie en 1990 pour ce travail. Ce chapitre développe la théorie moyenne-variance, la frontière efficiente, le modèle CAPM de Sharpe, et les extensions modernes incluant le risque CVaR et l'optimisation robuste.

8.1 Théorie de Markowitz

8.1.1 Cadre moyenne-variance

Définition 8.1 (Portefeuille). Considérons n actifs de rendements aléatoires $R = (R_1, \dots, R_n)^\top$ avec vecteur de rendements espérés $\mu = \mathbb{E}[R] \in \mathbb{R}^n$ et matrice de covariance $\Sigma = \text{Cov}(R) \in \mathbb{R}^{n \times n}$ (symétrique définie positive).

Un **portefeuille** est défini par un vecteur de poids $w = (w_1, \dots, w_n)^\top$ avec $\mathbf{1}^\top w = 1$. Le rendement du portefeuille est :

$$R_p = w^\top R, \quad \mathbb{E}[R_p] = w^\top \mu, \quad \text{Var}(R_p) = w^\top \Sigma w.$$

8.1.2 Problème d'optimisation

Définition 8.2 (Problème de Markowitz). Le **problème de Markowitz** (1952) cherche le portefeuille de variance minimale pour un rendement cible μ_p :

$$\min_w \frac{1}{2} w^\top \Sigma w \quad \text{s.c.} \quad w^\top \mu = \mu_p, \quad \mathbf{1}^\top w = 1.$$

Théorème 8.3 (Solution analytique). *La solution du problème de Markowitz est :*

$$w^* = \Sigma^{-1} [\lambda_1 \mu + \lambda_2 \mathbf{1}],$$

où les multiplicateurs de Lagrange λ_1, λ_2 sont déterminés par les contraintes. En posant :

$$A = \mathbf{1}^\top \Sigma^{-1} \mu, \quad B = \mu^\top \Sigma^{-1} \mu, \quad C = \mathbf{1}^\top \Sigma^{-1} \mathbf{1}, \quad D = BC - A^2, \quad (8.1)$$

on obtient :

$$\lambda_1 = \frac{C\mu_p - A}{D}, \quad \lambda_2 = \frac{B - A\mu_p}{D}.$$

Variance minimale du portefeuille optimal

$$\sigma_p^2 = \frac{C\mu_p^2 - 2A\mu_p + B}{D}.$$

C'est une parabole en μ_p dans le plan (σ_p^2, μ_p) , ou une hyperbole dans le plan (σ_p, μ_p) .

Définition 8.4 (Frontière efficiente). La **frontière efficiente** est la partie supérieure de l'hyperbole : l'ensemble des portefeuilles optimaux pour lesquels il est impossible d'augmenter le rendement sans augmenter le risque.

Définition 8.5 (Portefeuille de variance minimale globale (GMV)). Le portefeuille **GMV** minimise la variance sans contrainte de rendement :

$$w_{\text{GMV}} = \frac{\Sigma^{-1}\mathbf{1}}{\mathbf{1}^\top \Sigma^{-1}\mathbf{1}}.$$

Son rendement espéré est $\mu_{\text{GMV}} = A/C$ et sa variance $\sigma_{\text{GMV}}^2 = 1/C$.

Frontière efficiente de Markowitz

```
import numpy as np
from scipy.optimize import minimize

def efficient_frontier(mu, Sigma, n_points=100):
    """Calcul de la frontiere efficiente."""
    n = len(mu)
    # Portefeuille GMV
    Sigma_inv = np.linalg.inv(Sigma)
    ones = np.ones(n)
    w_gmv = Sigma_inv @ ones / (ones @ Sigma_inv @ ones)
    mu_gmv = w_gmv @ mu

    # Plage de rendements cibles
    mu_range = np.linspace(mu_gmv - 0.02, max(mu) + 0.05, n_points)
    sigmas = []
    weights = []

    for mu_target in mu_range:
        constraints = [
            {'type': 'eq', 'fun': lambda w: np.sum(w) - 1},
            {'type': 'eq', 'fun': lambda w, m=mu_target: w @ mu - m}
        ]
        result = minimize(
            lambda w: 0.5 * w @ Sigma @ w,
            np.ones(n)/n, method='SLSQP',
            constraints=constraints
        )
        sigmas.append(np.sqrt(result.fun * 2))
```

```

weights.append(result.x)

return mu_range, np.array(sigmas), np.array(weights)

# Exemple avec 3 actifs
mu = np.array([0.08, 0.12, 0.15])
Sigma = np.array([[0.04, 0.006, 0.002],
                  [0.006, 0.09, 0.009],
                  [0.002, 0.009, 0.16]])

mu_range, sigmas, weights = efficient_frontier(mu, Sigma)
print(f"GMV: sigma={sigmas.min():.4f}")

```

8.2 Modèle CAPM

Théorème 8.6 (Capital Asset Pricing Model (Sharpe, 1964)). *À l'équilibre, le rendement espéré de tout actif i satisfait :*

$$\mathbb{E}[R_i] - r = \beta_i(\mathbb{E}[R_M] - r),$$

où $\beta_i = \text{Cov}(R_i, R_M) / \text{Var}(R_M)$ est le **bêta** de l'actif par rapport au portefeuille de marché M , et r le taux sans risque.

Définition 8.7 (Ratio de Sharpe). Le **ratio de Sharpe** d'un portefeuille p est :

$$SR_p = \frac{\mathbb{E}[R_p] - r}{\sigma_p}.$$

Le portefeuille tangent (intersection de la CML avec la frontière efficiente) maximise le ratio de Sharpe.

Portefeuille tangent (max Sharpe)

```

def tangent_portfolio(mu, Sigma, rf):
    """Portefeuille tangent maximisant le ratio de Sharpe."""
    n = len(mu)
    Sigma_inv = np.linalg.inv(Sigma)
    excess = mu - rf
    w_tan = Sigma_inv @ excess / (np.ones(n) @ Sigma_inv @ excess)
    mu_tan = w_tan @ mu
    sigma_tan = np.sqrt(w_tan @ Sigma @ w_tan)
    sharpe = (mu_tan - rf) / sigma_tan
    return w_tan, mu_tan, sigma_tan, sharpe

rf = 0.03
w_tan, mu_tan, sig_tan, sr = tangent_portfolio(mu, Sigma, rf)
print(f"Portefeuille tangent: w = {w_tan}")
print(f"Rendement: {mu_tan:.4f}, Risque: {sig_tan:.4f}")
print(f"Ratio de Sharpe: {sr:.4f}")

```

8.3 Modèle de Black–Litterman

Définition 8.8 (Modèle de Black–Litterman (1992)). Le modèle combine les rendements d'équilibre (CAPM) $\Pi = \delta \Sigma w_M$ avec les **vues** de l'investisseur $P\mu = Q + \epsilon$, $\epsilon \sim \mathcal{N}(0, \Omega)$, pour obtenir des rendements postérieurs :

$$\hat{\mu} = [(\tau \Sigma)^{-1} + P^\top \Omega^{-1} P]^{-1} [(\tau \Sigma)^{-1} \Pi + P^\top \Omega^{-1} Q].$$

Le portefeuille optimal utilise alors $\hat{\mu}$ dans l'optimisation de Markowitz.

Avantage de Black–Litterman

Black–Litterman résout deux problèmes de Markowitz classique : (1) les poids extrêmes dus à la sensibilité aux rendements estimés, et (2) l'absence de lien avec l'équilibre de marché. Les vues sont exprimées de manière intuitive et le modèle génère des portefeuilles plus stables.

8.4 Contraintes pratiques

Remarque 8.9. En pratique, on ajoute souvent des contraintes supplémentaires :

- Interdiction de la vente à découvert : $w_i \geq 0$.
- Limites de concentration : $w_i \leq w_{\max}$.
- Contraintes sectorielles ou de liquidité.
- Coûts de transaction (optimisation dynamique).

Optimisation avec contraintes

```
def efficient_frontier_constrained(mu, Sigma, rf, n_points=50):
    """Frontiere efficiente avec w >= 0 (pas de vente a decouvert)."""
    n = len(mu)
    bounds = [(0, 1) for _ in range(n)]
    mu_range = np.linspace(min(mu), max(mu), n_points)
    results = []

    for mu_target in mu_range:
        constraints = [
            {'type': 'eq', 'fun': lambda w: np.sum(w) - 1},
            {'type': 'eq', 'fun': lambda w, m=mu_target: w @ mu - m}
        ]
        res = minimize(lambda w: w @ Sigma @ w, np.ones(n)/n,
                       method='SLSQP', bounds=bounds,
                       constraints=constraints)
        if res.success:
            results.append((mu_target, np.sqrt(res.fun), res.x))

    return results
```

8.5 Optimisation robuste

Définition 8.10 (Optimisation robuste). L'**optimisation robuste** tient compte de l'incertitude sur les paramètres estimés. On résout :

$$\min_w \max_{\mu \in \mathcal{U}} \{-w^\top \mu + \lambda w^\top \Sigma w\},$$

où $\mathcal{U}$ est un ensemble d'incertitude autour de $\hat{\mu}$. Pour $\mathcal{U} = \{\mu : \|\mu - \hat{\mu}\|_{\Sigma^{-1}} \leq \kappa\}$, le problème se réduit à :

$$\min_w \left\{ -w^\top \hat{\mu} + \kappa \sqrt{w^\top \Sigma w} + \lambda w^\top \Sigma w \right\}.$$

8.6 Estimation des paramètres

Erreurs d'estimation

Les rendements espérés μ sont notoirement difficiles à estimer. Avec T observations, l'erreur standard de $\hat{\mu}_i$ est $\sigma_i/\sqrt{T}$, souvent du même ordre que le signal. La matrice Σ est mieux estimée mais peut être singulière ou mal conditionnée lorsque $n \approx T$.

Proposition 8.11 (Estimateur de Ledoit–Wolf). L'estimateur **shrinkage** de Ledoit–Wolf (2004) régularise la matrice de covariance :

$$\hat{\Sigma}_{\text{LW}} = (1 - \alpha)\hat{\Sigma} + \alpha \nu I,$$

où $\alpha \in [0, 1]$ et $\nu = \text{tr}(\hat{\Sigma})/n$ sont déterminés analytiquement pour minimiser la perte quadratique.

8.7 Exercices

Exercice 8.1 (Frontière efficiente). 1. Calculer et tracer la frontière efficiente pour 5 actifs.

2. Identifier le portefeuille GMV et le portefeuille tangent.

3. Tracer la Capital Market Line.

4. Comparer les frontières avec et sans contrainte de positivité.

Exercice 8.2 (CAPM et bêta). 1. Estimer les bêtas de 5 actions par régression linéaire sur un indice de marché.

2. Vérifier la relation du CAPM en traçant le rendement moyen historique en fonction du bêta.

3. Calculer l'alpha de Jensen pour chaque action.

Exercice 8.3 (Black–Litterman). 1. Implémenter le modèle de Black–Litterman.

2. Partir des rendements d'équilibre et ajouter la vue "l'actif 1 surperformera l'actif 2 de 2%".

3. Comparer les poids avant et après l'intégration de la vue.

Exercice 8.4 (Estimation robuste). 1. Simuler 250 jours de rendements pour 10 actifs et estimer $\hat{\mu}$ et $\hat{\Sigma}$.

2. Comparer les portefeuilles optimaux avec $\hat{\Sigma}$ et $\hat{\Sigma}_{LW}$.

3. Effectuer un backtest et mesurer la performance out-of-sample.

Chapitre 9

Mesures de Risque

9.1 Introduction

Après les crises financières des années 1990 — débâcle d'Orange County (1994), faillite de Barings (1995), crise LTCM (1998) — la question de la mesure du risque est devenue centrale. J.P. Morgan popularise la *Value at Risk* (VaR) en 1994 avec son système Risk-Metrics : un chiffre unique résumant la perte maximale probable sur un horizon donné avec un certain niveau de confiance. Mais Philippe Artzner et ses collègues montrent en 1999 que la VaR n'est pas *cohérente* : elle ne respecte pas la sous-additivité, ce qui signifie que diversifier un portefeuille peut, paradoxalement, augmenter la VaR mesurée. L'*Expected Shortfall* (CVaR) corrige ce défaut. Ce chapitre parcourt cette histoire, des mesures empiriques aux axiomes théoriques, en passant par les copules et les tests de résistance.

Intuition

Une mesure de risque répond à la question : « Combien puis-je perdre au maximum avec une probabilité donnée sur un horizon donné ? » La VaR répond par un quantile, mais ne dit rien sur l'ampleur des pertes au-delà. L'Expected Shortfall comble cette lacune en moyennant les pertes dans la queue de la distribution.

9.2 Value at Risk (VaR)

Définition 9.1 (Value at Risk). Soit L la variable aléatoire représentant la perte d'un portefeuille sur un horizon Δt . La **Value at Risk** au niveau de confiance $\alpha \in (0, 1)$ est

$$\text{VaR}_\alpha(L) = \inf\{l \in \mathbb{R} : \mathbb{P}(L > l) \leq 1 - \alpha\} = F_L^{-1}(\alpha),$$

où F_L^{-1} désigne le quantile généralisé de L .

Exemple 9.2 (VaR gaussienne). Si $L \sim \mathcal{N}(\mu, \sigma^2)$, alors $\text{VaR}_\alpha = \mu + \sigma \Phi^{-1}(\alpha)$, où Φ est la fonction de répartition de la loi normale standard. Pour $\alpha = 0,99$: $\text{VaR}_{99\%} = \mu + 2,326 \sigma$.

Proposition 9.3 (Méthodes de calcul de la VaR). 1. **Historique** : VaR_α = quantile empirique des rendements passés.

2. **Paramétrique (variance-covariance)** : suppose une distribution (souvent gaussienne) et utilise μ, σ estimés.

3. **Monte Carlo** : simulation de scénarios et calcul du quantile sur les pertes simulées.

Limites de la VaR

La VaR ne satisfait pas la propriété de sous-additivité en général : on peut avoir $\text{VaR}_\alpha(L_1 + L_2) > \text{VaR}_\alpha(L_1) + \text{VaR}_\alpha(L_2)$. Cela signifie que la diversification peut *sembler* augmenter le risque, ce qui est un défaut conceptuel majeur.

9.3 Expected Shortfall (CVaR)

Définition 9.4 (Expected Shortfall). L'**Expected Shortfall** (ou CVaR, Conditional Value at Risk) au niveau α est

$$\text{ES}_\alpha(L) = \frac{1}{1 - \alpha} \int_\alpha^1 \text{VaR}_u(L) \, du.$$

Si F_L est continue, $\text{ES}_\alpha(L) = \mathbb{E}[L \mid L \geq \text{VaR}_\alpha(L)]$.

Exemple 9.5 (ES gaussien). Si $L \sim \mathcal{N}(\mu, \sigma^2)$:

$$\text{ES}_\alpha = \mu + \sigma \frac{\phi(\Phi^{-1}(\alpha))}{1 - \alpha},$$

où ϕ est la densité de la loi normale standard. Pour $\alpha = 0,99$: $\text{ES}_{99\%} \approx \mu + 2,665 \sigma$.

Théorème 9.6 (Représentation duale de l'ES).

$$\text{ES}_\alpha(L) = \sup_{Q \in \mathcal{Q}_\alpha} \mathbb{E}_Q[L],$$

où $\mathcal{Q}_\alpha = \{Q \ll \mathbb{P} : dQ/d\mathbb{P} \leq 1/(1 - \alpha)\}$. Cette représentation montre que l'ES est le pire cas d'espérance sur un ensemble d'ambiguïté.

9.4 Mesures de risque cohérentes

Définition 9.7 (Mesure de risque cohérente). Une fonctionnelle $\rho : \mathcal{X} \rightarrow \mathbb{R}$ est une **mesure de risque cohérente** (Artzner, Delbaen, Eber, Heath, 1999) si elle satisfait :

1. **Monotonie** : $L_1 \leq L_2$ p.s. $\implies \rho(L_1) \leq \rho(L_2)$.
2. **Invariance par translation** : $\rho(L + c) = \rho(L) + c$ pour tout $c \in \mathbb{R}$.
3. **Homogénéité positive** : $\rho(\lambda L) = \lambda \rho(L)$ pour tout $\lambda > 0$.
4. **Sous-additivité** : $\rho(L_1 + L_2) \leq \rho(L_1) + \rho(L_2)$.

Théorème 9.8 (ADEH, 1999). 1. L'Expected Shortfall est une mesure de risque **cohérente**.

2. La VaR n'est **pas** cohérente (elle viole la sous-additivité).
3. Toute mesure de risque cohérente admet une représentation $\rho(L) = \sup_{Q \in \mathcal{Q}} \mathbb{E}_Q[L]$ pour un ensemble convexe fermé $\mathcal{Q}$ de mesures de probabilité.

Comparaison VaR vs ES

Propriété	VaR	ES
Sous-additivité	Non	Oui
Cohérence	Non	Oui
Sensibilité à la queue	Non	Oui
Réglementation Bâle III+	Non	Oui
Facilité de backtesting	Oui	Difficile

9.5 Tests de résistance (Stress Testing)

Définition 9.9 (Stress test). Un **test de résistance** consiste à évaluer les pertes d'un portefeuille sous des scénarios extrêmes prédéfinis ou historiques :

$$L_{\text{stress}} = \sum_{i=1}^n \Delta_i \cdot S_i^{\text{stress}},$$

où Δ_i sont les sensibilités et S_i^{stress} les chocs appliqués aux facteurs de risque.

Proposition 9.10 (Types de stress tests). 1. **Historiques** : reproduction de crises passées (2008, 2020).

2. **Hypothétiques** : scénarios construits (choc de taux de 300 bp, effondrement d'un secteur).

3. **Reverse stress tests** : recherche du scénario qui mène à une perte critique donnée.

9.6 Copules et modélisation de la dépendance

Définition 9.11 (Copule). Une **copule** $C : [0, 1]^d \rightarrow [0, 1]$ est une fonction de répartition dont les marginales sont uniformes sur $[0, 1]$. Elle capture la structure de dépendance indépendamment des lois marginales.

Théorème 9.12 (Sklar, 1959). Pour toute loi jointe F de marginales $F_1, \dots, F_d$, il existe une copule C telle que

$$F(x_1, \dots, x_d) = C(F_1(x_1), \dots, F_d(x_d)).$$

Si les F_i sont continues, C est unique.

Exemple 9.13 (Copule gaussienne). La **copule gaussienne** de paramètre Σ (matrice de corrélation) est

$$C_{\Sigma}^{\text{Ga}}(u_1, \dots, u_d) = \Phi_{\Sigma}(\Phi^{-1}(u_1), \dots, \Phi^{-1}(u_d)),$$

où Φ_{Σ} est la fonction de répartition de la loi normale multidimensionnelle.

Définition 9.14 (Copule de Clayton). La **copule de Clayton** (bivariée) est

$$C_{\theta}(u, v) = (u^{-\theta} + v^{-\theta} - 1)^{-1/\theta}, \quad \theta > 0.$$

Elle modélise une forte dépendance dans la queue inférieure (risque de co-crash).

Remarque 9.15. La copule gaussienne sous-estime la dépendance de queue. C'est l'une des causes de la crise des subprimes de 2008, où les modèles de CDO utilisaient la copule gaussienne de Li (2000) sans tenir compte de la dépendance extrême.

Définition 9.16 (Dépendance de queue). Le **coefficient de dépendance de queue inférieure** est

$$\lambda_L = \lim_{u \rightarrow 0^+} \frac{C(u, u)}{u}.$$

Pour la copule gaussienne, $\lambda_L = 0$ (sauf si $\rho = 1$). Pour la copule de Clayton, $\lambda_L = 2^{-1/\theta} > 0$.

Calcul de la VaR et de l'ES par simulation Monte Carlo

```
import numpy as np
from scipy.stats import norm

np.random.seed(42)
n_simulations = 100_000
mu, sigma = 0.0, 0.02 # rendement journalier
alpha = 0.99

# Simulation des rendements
returns = np.random.normal(mu, sigma, n_simulations)
losses = -returns # perte = -rendement

# VaR historique
var_99 = np.percentile(losses, 100 * alpha)

# Expected Shortfall
es_99 = losses[losses >= var_99].mean()

print(f"VaR 99%: {var_99:.4f}")
print(f"ES 99%: {es_99:.4f}")
print(f"VaR analytique: {-mu + sigma * norm.ppf(alpha):.4f}")
```

9.7 Exercices

Exercice 9.1 (VaR d'un portefeuille). Un portefeuille de valeur 10 M€ a des rendements journaliers $\mathcal{N}(0,05\%, (1,5\%)^2)$. Calculer la VaR à 99% à 1 jour et à 10 jours (règle de la racine carrée du temps).

Exercice 9.2 (Violation de la sous-additivité). Construire un exemple explicite de deux variables aléatoires L_1, L_2 telles que $\text{VaR}_{0,95}(L_1 + L_2) > \text{VaR}_{0,95}(L_1) + \text{VaR}_{0,95}(L_2)$. *Indication* : utiliser des pertes de Bernoulli concentrées.

Exercice 9.3 (Copule de Clayton). Simuler 10 000 paires (U, V) suivant une copule de Clayton avec $\theta = 2$. Estimer le coefficient de dépendance de queue inférieure et comparer avec la valeur théorique $\lambda_L = 2^{-1/2}$.

Exercice 9.4 (Mesure de risque convexe). Montrer que toute mesure de risque cohérente est convexe. Donner un exemple de mesure convexe qui n'est pas cohérente (homogénéité positive violée).

Exercice 9.5 (Stress test). Concevoir un stress test historique basé sur la crise de 2008 pour un portefeuille contenant 60% d'actions et 40% d'obligations. Calculer la perte sous ce scénario et comparer avec la VaR à 99%.

Exercice 9.6 (Backtesting). Expliquer le test de Kupiec pour le backtesting de la VaR. Si sur 250 jours de trading, on observe 5 dépassements de la VaR à 99%, la VaR est-elle correctement calibrée au seuil de 5% ?

Chapitre 10

Produits Dérivés Exotiques

10.1 Introduction

Les options vanilles — calls et puts européens — sont les briques élémentaires de la finance des dérivés. Mais les besoins des entreprises et des investisseurs sont infiniment plus variés : une compagnie aérienne veut se couvrir contre la *moyenne* du prix du kérosène sur un trimestre (option asiatique) ; un exportateur veut une protection qui s'active uniquement si le taux de change franchit un seuil (option barrière) ; un investisseur veut le meilleur rendement parmi plusieurs actifs (option sur panier). Ces *options exotiques*, nées dans les salles de marché des années 1980–1990, posent des défis mathématiques considérables : la plupart n'admettent pas de formule fermée à la Black–Scholes et nécessitent des méthodes numériques (Monte Carlo, différences finies, arbres).

Intuition

Les options vanilles sont des briques élémentaires. Les options exotiques combinent ces briques ou ajoutent des conditions (barrières, moyennes, extréma) pour créer des profils de gain sur mesure. Leur valorisation requiert souvent des méthodes numériques avancées car les formules fermées sont rares.

10.2 Options à barrière

Définition 10.1 (Option à barrière). Une **option à barrière** est une option dont l'existence ou l'activation dépend du franchissement d'un niveau B par le sous-jacent S_t durant la vie de l'option. On distingue :

- **Knock-out** (désactivante) : l'option disparaît si S_t atteint B .
- **Knock-in** (activante) : l'option n'existe que si S_t atteint B .

Combiné avec la direction (up/down), cela donne 8 types fondamentaux.

Théorème 10.2 (Relation in-out). *Pour toute barrière B , le prix d'un call (ou put) vanille vérifie :*

$$V_{\text{vanille}} = V_{\text{knock-in}} + V_{\text{knock-out}}.$$

Théorème 10.3 (Formule de Merton pour un down-and-out call). *Dans le modèle de Black-Scholes, pour un call down-and-out avec barrière $B < S_0$ et $B < K$:*

$$C_{\text{do}} = C_{\text{BS}} - \left(\frac{B}{S_0}\right)^{2\lambda} C_{\text{BS}}\left(\frac{B^2}{S_0}, K, r, \sigma, T\right),$$

où $\lambda = \frac{r-\sigma^2/2}{\sigma^2} + \frac{1}{2}$ et C_{BS} est le prix de Black-Scholes.

Risque de couverture des barrières

Le delta d'une option barrière diverge lorsque le sous-jacent approche de la barrière, rendant la couverture dynamique extrêmement coûteuse. En pratique, on utilise des barrières discrètes (observées quotidiennement) avec une correction de Broadie, Glasserman et Kou.

10.3 Options asiatiques

Définition 10.4 (Option asiatique). Une **option asiatique** a un payoff dépendant de la moyenne du sous-jacent :

- **Average price** : payoff = $(\bar{S} - K)^+$ (call) où $\bar{S} = \frac{1}{T} \int_0^T S_t dt$ (moyenne continue) ou $\bar{S} = \frac{1}{n} \sum_{i=1}^n S_{t_i}$ (moyenne discrète).
- **Average strike** : payoff = $(S_T - \bar{S})^+$.

Proposition 10.5 (Propriétés de l'asiatique). 1. La moyenne réduit la volatilité effective : $\text{Var}(\bar{S}) < \text{Var}(S_T)$, donc les options asiatiques sont moins chères que les vanilles correspondantes.

2. La distribution de $\bar{S}$ n'est pas lognormale, même si S_t suit un mouvement brownien géométrique.
3. Il n'existe pas de formule fermée exacte dans Black-Scholes pour la moyenne arithmétique (contrairement à la moyenne géométrique).

Théorème 10.6 (Prix de l'asiatique géométrique). *Pour la moyenne géométrique continue $\bar{S}_G = \exp\left(\frac{1}{T} \int_0^T \ln S_t dt\right)$, le prix du call est donné par la formule de Black-Scholes modifiée avec*

$$\hat{\sigma} = \frac{\sigma}{\sqrt{3}}, \quad \hat{r} = \frac{1}{2} \left(r - \frac{\sigma^2}{6} \right).$$

10.4 Options lookback

Définition 10.7 (Option lookback). Une **option lookback** dépend de l'extrémum du sous-jacent :

- **Floating strike lookback call** : payoff = $S_T - \min_{0 \leq t \leq T} S_t$.
- **Fixed strike lookback call** : payoff = $(\max_{0 \leq t \leq T} S_t - K)^+$.

Théorème 10.8 (Goldman, Sosin, Gatto (1979)). *Dans Black-Scholes, le prix du floating strike lookback call est*

$$V = S_0 \Phi(d_1) - S_0 e^{-rT} \frac{\sigma^2}{2r} \Phi(-d_1) - S_{\min} e^{-rT} \left[\Phi(d_2) - \frac{\sigma^2}{2r} \left(\frac{S_0}{S_{\min}} \right)^{-2r/\sigma^2} \Phi(d_3) \right],$$

où d_1, d_2, d_3 sont des fonctions explicites de $S_0, S_{\min}, r, \sigma, T$.

10.5 Options binaires (digitales)

Définition 10.9 (Option binaire). Une **option binaire** (ou digitale) a un payoff discontinu :

- **Cash-or-nothing call** : payoff = $Q \cdot \mathbb{1}_{S_T > K}$ (paye Q si $S_T > K$).
- **Asset-or-nothing call** : payoff = $S_T \cdot \mathbb{1}_{S_T > K}$.

Proposition 10.10 (Prix dans Black-Scholes).

$$\begin{aligned} V_{\text{cash}} &= Q e^{-rT} \Phi(d_2), \\ V_{\text{asset}} &= S_0 \Phi(d_1), \end{aligned}$$

où d_1, d_2 sont les quantités standard de Black-Scholes. Un call vanille se décompose : $C = V_{\text{asset}} - K e^{-rT} \Phi(d_2)$.

Remarque 10.11. Le delta d'une option binaire diverge près du strike à l'échéance, ce qui rend la couverture très difficile. En pratique, on les réplique par un *call spread* serré.

10.6 Valorisation par Monte Carlo

Définition 10.12 (Estimateur Monte Carlo). Le prix d'une option exotique de payoff $h(S_{t_1}, \dots, S_{t_n})$ est

$$V_0 = e^{-rT} \mathbb{E}^{\mathbb{Q}}[h(S_{t_1}, \dots, S_{t_n})] \approx \frac{e^{-rT}}{M} \sum_{j=1}^M h(S_{t_1}^{(j)}, \dots, S_{t_n}^{(j)}),$$

où les trajectoires $(S^{(j)})$ sont simulées sous la mesure risque-neutre.

Proposition 10.13 (Réduction de variance). Les techniques principales sont :

1. **Variables antithétiques** : pour chaque trajectoire W_t , utiliser aussi $-W_t$.
2. **Variable de contrôle** : utiliser le prix connu de la géométrie pour corriger l'estimateur de l'arithmétique.
3. **Importance sampling** : modifier la mesure pour échantillonner les événements rares.

Simulation de trajectoires sous $\mathbb{Q}$

Dans le modèle de Black-Scholes :

$$S_{t_{i+1}} = S_{t_i} \exp \left[\left(r - \frac{\sigma^2}{2} \right) \Delta t + \sigma \sqrt{\Delta t} Z_i \right], \quad Z_i \sim \mathcal{N}(0, 1).$$

10.7 Méthodes d'arbre pour les exotiques

Définition 10.14 (Arbre trinomial). Un **arbre trinomial** discrétise le sous-jacent avec trois mouvements à chaque pas : $S \rightarrow Su$, $S \rightarrow Sm$, $S \rightarrow Sd$. Les probabilités risque-neutres (p_u, p_m, p_d) sont calibrées pour reproduire les deux premiers moments de $\ln S$.

Proposition 10.15 (Adaptation aux barrières). Pour une option barrière, on ajuste l'arbre pour qu'un nœud coïncide exactement avec la barrière B , éliminant les erreurs de spécification dues à la discrétisation.

Prix d'une option asiatique par Monte Carlo

```
import numpy as np

S0, K, r, sigma, T = 100, 100, 0.05, 0.2, 1.0
n_steps, n_paths = 252, 100_000
dt = T / n_steps

np.random.seed(42)
Z = np.random.randn(n_paths, n_steps)
S = np.zeros((n_paths, n_steps + 1))
S[:, 0] = S0

for i in range(n_steps):
    S[:, i+1] = S[:, i] * np.exp((r - 0.5*sigma**2)*dt
                                + sigma*np.sqrt(dt)*Z[:, i])

S_mean = S[:, 1:].mean(axis=1)
payoff = np.maximum(S_mean - K, 0)
price = np.exp(-r*T) * payoff.mean()
stderr = np.exp(-r*T) * payoff.std() / np.sqrt(n_paths)

print(f"Prix asiatique (call): {price:.4f} +/- {1.96*stderr:.4f}")
```

10.8 Exercices

Exercice 10.1 (Relation in-out). Démontrer la relation in-out pour les options à barrière en utilisant l'absence d'opportunité d'arbitrage.

Exercice 10.2 (Asiatique vs vanille). Montrer analytiquement que le prix d'un call asiatique (average price) est inférieur au prix du call vanille correspondant.

Exercice 10.3 (Lookback pricing). Implémenter un pricer Monte Carlo pour une option lookback floating strike et comparer avec la formule fermée de Goldman-Sosin-Gatto.

Exercice 10.4 (Réplication d'une digitale). Montrer qu'un call binaire cash-or-nothing de payoff $Q\mathbb{1}_{S_T > K}$ peut être approché par un call spread : $(C(K) - C(K + \epsilon))/\epsilon$ multiplié par Q , et calculer la limite $\epsilon \rightarrow 0$.

Exercice 10.5 (Variables antithétiques). Montrer que l'estimateur Monte Carlo avec variables antithétiques a une variance inférieure à l'estimateur standard si et seulement si les payoffs sont négativement corrélés sous l'inversion $W \rightarrow -W$.

Exercice 10.6 (Barrière discrète). Pour un down-and-out call avec barrière discrète observée quotidiennement, appliquer la correction de Broadie-Glasserman-Kou : $B_{\text{eff}} = B e^{-\beta\sigma\sqrt{\Delta t}}$ avec $\beta \approx 0,5826$ et comparer avec la barrière continue.

Chapitre 11

Apprentissage Machine en Finance

11.1 Introduction

L'apprentissage machine (ML) transforme la finance quantitative en permettant de découvrir des structures non linéaires dans les données de marché. Ce chapitre présente les principales applications : trading algorithmique, pricing d'options par réseaux de neurones, gestion de portefeuille par apprentissage par renforcement, analyse de sentiment par NLP, ainsi que les risques et la réglementation.

Intuition

Les modèles classiques (Black-Scholes, GARCH) imposent une structure paramétrique. Le ML permet de « laisser parler les données » en apprenant des relations complexes directement, au prix d'un risque de surapprentissage accru et d'une moindre interprétabilité.

11.2 Trading algorithmique

Définition 11.1 (Stratégie de trading algorithmique). Une **stratégie de trading algorithmique** est une règle systématique $\delta_t = g(x_t; \theta)$ qui détermine la position δ_t à l'instant t en fonction de features observables x_t et de paramètres θ optimisés sur données historiques.

Proposition 11.2 (Features classiques). Les prédicteurs typiques incluent :

1. **Indicateurs techniques** : moyennes mobiles, RSI, bandes de Bollinger, MACD.
2. **Microstructure** : déséquilibre du carnet d'ordres (order flow imbalance), spread bid-ask, volume.
3. **Features fondamentaux** : ratios financiers, earnings surprises.
4. **Features alternatifs** : données satellites, trafic web, sentiment de réseaux sociaux.

Surapprentissage et data snooping

Le backtesting massif sur données historiques produit des stratégies qui performant parfaitement in-sample mais échouent out-of-sample. Il faut : validation croisée tem-

porelle (walk-forward), correction de Bonferroni pour les tests multiples, et réserve de données fraîches pour le test final.

Théorème 11.3 (Borne de généralisation). *Pour une classe d'hypothèses $\mathcal{H}$ de dimension de Vapnik-Chervonenkis d_{VC} , avec probabilité au moins $1 - \delta$ sur un échantillon de taille n :*

$$R(h) \leq \hat{R}(h) + \sqrt{\frac{d_{VC}(\ln(2n/d_{VC}) + 1) + \ln(4/\delta)}{n}},$$

où $R(h)$ est le risque réel et $\hat{R}(h)$ le risque empirique.

11.3 Réseaux de neurones pour le pricing d'options

Définition 11.4 (Réseau de neurones feedforward). Un **réseau de neurones feedforward** à L couches est une fonction

$$f_{\theta}(x) = W_L \sigma(W_{L-1} \sigma(\cdots \sigma(W_1 x + b_1) \cdots) + b_{L-1}) + b_L,$$

où σ est une fonction d'activation non linéaire (ReLU, tanh, etc.), W_{ℓ} les matrices de poids et b_{ℓ} les biais.

Théorème 11.5 (Approximation universelle). *Pour toute fonction continue $g : \mathbb{R}^d \rightarrow \mathbb{R}$ sur un compact K et tout $\epsilon > 0$, il existe un réseau feedforward à une couche cachée avec activation sigmoïde tel que $\sup_{x \in K} |f_{\theta}(x) - g(x)| < \epsilon$.*

Proposition 11.6 (Pricing par réseau de neurones). L'approche consiste à :

1. Générer N jeux de paramètres $(S_0^{(i)}, K^{(i)}, T^{(i)}, \sigma^{(i)}, r^{(i)})$ et calculer les prix $V^{(i)}$ par Black-Scholes ou Monte Carlo.
2. Entraîner un réseau f_{θ} minimisant $\frac{1}{N} \sum_{i=1}^N (f_{\theta}(x^{(i)}) - V^{(i)})^2$.
3. En inférence, f_{θ} donne le prix et les grecs ($\Delta = \partial f_{\theta} / \partial S$) quasi instantanément.

Remarque 11.7. Les réseaux de neurones accélèrent le pricing d'un facteur 10^3 à 10^5 par rapport à Monte Carlo, ce qui est crucial pour le calcul de la XVA (CVA, DVA, FVA) sur des portefeuilles contenant des milliers de transactions.

11.4 Apprentissage par renforcement pour la gestion de portefeuille

Définition 11.8 (Processus de décision markovien (MDP)). Un **MDP** est un tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$ où $\mathcal{S}$ est l'espace d'états, $\mathcal{A}$ l'espace d'actions, $P(s'|s, a)$ la dynamique de transition, $R(s, a)$ la récompense, et $\gamma \in [0, 1)$ le facteur d'actualisation.

Définition 11.9 (Politique et fonction de valeur). Une **politique** $\pi : \mathcal{S} \rightarrow \mathcal{A}$ spécifie l'action à chaque état. La **fonction de valeur** est $V^{\pi}(s) = \mathbb{E}_{\pi} [\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \mid s_0 = s]$. La politique optimale satisfait $V^*(s) = \max_{\pi} V^{\pi}(s)$.

Théorème 11.10 (Équation de Bellman). *La fonction de valeur optimale satisfait*

$$V^*(s) = \max_{a \in \mathcal{A}} \left[R(s, a) + \gamma \sum_{s'} P(s'|s, a) V^*(s') \right].$$

Proposition 11.11 (Application à la gestion de portefeuille). • **État** : $s_t = (w_t, S_t, \text{features}_t)$ (richesse, prix, indicateurs).

- **Action** : $a_t = (\delta_1, \dots, \delta_d)$ (allocations dans d actifs).
- **Récompense** : $R_t = \ln(w_{t+1}/w_t)$ (rendement logarithmique) ou ratio de Sharpe différentiel.
- **Algorithmes** : DDPG, PPO, SAC pour actions continues.

11.5 NLP et analyse de sentiment

Définition 11.12 (Analyse de sentiment financière). L'**analyse de sentiment** en finance consiste à extraire un signal haussier/baissier à partir de sources textuelles : communiqués de presse, rapports d'analystes, tweets, transcriptions d'appels de résultats.

Proposition 11.13 (Pipeline NLP pour la finance). 1. **Collecte** : flux Reuters/Bloomberg, API Twitter/Reddit.

2. **Prétraitement** : tokenisation, suppression des stop words, lemmatisation.
3. **Modèle** : FinBERT (BERT fine-tuné sur du texte financier) ou GPT spécialisé.
4. **Agrégation** : score de sentiment moyen pondéré par source et par récence.
5. **Signal de trading** : combinaison avec des features quantitatifs.

Métriques de performance des stratégies ML

$$\begin{aligned} \text{Ratio de Sharpe} &= \frac{\mathbb{E}[R_p] - r_f}{\sqrt{\text{Var}(R_p)}} \\ \text{Ratio de Sortino} &= \frac{\mathbb{E}[R_p] - r_f}{\sqrt{\mathbb{E}[\min(R_p - r_f, 0)^2]}} \\ \text{Maximum Drawdown} &= \max_t \frac{\max_{s \leq t} W_s - W_t}{\max_{s \leq t} W_s} \end{aligned}$$

11.6 Risques et réglementation

Définition 11.14 (Risque de modèle). Le **risque de modèle** en ML inclut :

- **Surapprentissage** : le modèle mémorise le bruit.
- **Distribution shift** : les données futures diffèrent des données d'entraînement (changements de régime).

- **Boîte noire** : impossibilité d'expliquer les décisions (problème d'interprétabilité).
- **Attaques adversariales** : manipulation délibérée des inputs pour tromper le modèle.

Remarque 11.15. Le règlement européen AI Act (2024) classe les systèmes de scoring de crédit et de trading algorithmique comme à « haut risque », imposant des exigences de transparence, de documentation et de supervision humaine.

Pricing d'options par réseau de neurones

```
import numpy as np
import torch
import torch.nn as nn
from scipy.stats import norm

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

# Generate training data
n_train = 50_000
S = np.random.uniform(50, 150, n_train)
K = np.random.uniform(50, 150, n_train)
T = np.random.uniform(0.1, 2.0, n_train)
sigma = np.random.uniform(0.1, 0.5, n_train)
r_rate = np.full(n_train, 0.05)
prices = bs_call(S, K, T, r_rate, sigma)

X = torch.tensor(np.column_stack([S, K, T, sigma]), dtype=torch.float32)
y = torch.tensor(prices, dtype=torch.float32).unsqueeze(1)

model = nn.Sequential(
    nn.Linear(4, 64), nn.ReLU(),
    nn.Linear(64, 64), nn.ReLU(),
    nn.Linear(64, 1))
optimizer = torch.optim.Adam(model.parameters(), lr=1e-3)

for epoch in range(200):
    pred = model(X)
    loss = nn.MSELoss()(pred, y)
    optimizer.zero_grad(); loss.backward(); optimizer.step()
print(f"MSE finale: {loss.item():.6f}")
```

11.7 Exercices

Exercice 11.1 (Feature engineering). Pour une stratégie de trading sur le S&P 500, proposer 10 features pertinents. Discuter les problèmes de look-ahead bias et de multicollinéarité.

Exercice 11.2 (Réseau pour les grecs). Entraîner un réseau de neurones pour pricer des calls européens. Calculer le delta par différentiation automatique et comparer avec la formule analytique.

Exercice 11.3 (RL pour allocation). Implémenter un agent DQN simple pour l'allocation entre un actif risqué (rendement $\mathcal{N}(8\%, 20\%)$) et un actif sans risque (3%). Comparer la politique apprise avec la règle de Merton.

Exercice 11.4 (Analyse de sentiment). Collecter 1000 titres d'actualités financières et les classer en positif/négatif/neutre. Mesurer la corrélation entre le score de sentiment agrégé et les rendements du lendemain.

Exercice 11.5 (Validation temporelle). Expliquer pourquoi la validation croisée standard (k-fold) est inappropriée en finance. Implémenter une validation walk-forward avec une fenêtre glissante de 2 ans d'entraînement et 6 mois de test.

Exercice 11.6 (Interprétabilité). Appliquer les valeurs de Shapley (SHAP) à un modèle XGBoost de prédiction de rendements. Identifier les features les plus importants et discuter la stabilité des explications au cours du temps.

Bibliographie

- [1] Hull, J.C., *Options, Futures, and Other Derivatives*, 11th ed., Pearson, 2022.
- [2] Shreve, S.E., *Stochastic Calculus for Finance II*, Springer, 2004.
- [3] Brigo, D. and Mercurio, F., *Interest Rate Models*, 2nd ed., Springer, 2006.
- [4] Björk, T., *Arbitrage Theory in Continuous Time*, 4th ed., Oxford, 2020.
- [5] Lamberton, D. and Lapeyre, B., *Introduction au calcul stochastique appliqué à la finance*, 3rd ed., Ellipses, 2012.