# Master 272 - Ingénierie Économique et Financière

DOCUMENTATION TECHNIQUE : PRODUITS STRUCTURÉS

**Projet : Pricer de produits dérivés & structurés en Python OOP**

Ryhan Chebrek, Alexandre Remiat, Jules Mourgues-Haroche, Yann Merakeb, Walid Boudounit

Sous la direction de Laurent Davoust

Année Académique 2024-2025

# 1. Vue d'ensemble de l'architecture

Le projet est conçu pour gérer la valorisation de dérivés financiers (ici sur single stock), incluant les produits structurés, les options et les produits de taux en utilisant diverses méthodologies de pricing. L'architecture tend à être modulaire, scalable et interconnectée par l'approche OOP avec une séparation claire des responsabilités entre les différents composants :

## Arborescence du projet

```
project_root/
 config/
    config.ini
 data/
    management/
       data_retriever.py
       data_utils.py
    options/       # *.xlsx (SPX, AMZN, MSFT, TSLA, ESTX50, STXX500)
    prices/        # loader.py + historical *.csv
    rates/
        curve/     # ESTER, SOFR, EURIBOR swap quotes
        spot/      # historical short rates
 market/
    day_count_convention.py
    market.py
    market_factory.py
 rate/
    abstract_taux.py
    interpolation/
    nelson_siegel.py
    svensson.py
    vasicek.py
    zc_curve.py
 option/
    abstract_option.py
    option.py
 pricers/
    pricing_model.py
    bs_pricer.py
    mc_pricer.py
    tree_pricer.py
    structured_pricer.py
    regression.py
 investment_strategies/
    abstract_strategy.py
    vanilla_strategies.py
    structured_strategies.py
 risk_metrics/
```

```
   greeks.py
   rate_product_sensitivity.py
   options_backtest.py
   sensitivity_analyzer.py
stochastic_process/
   abstract_stochastic.py
   brownian_motion.py
   gbm_process.py
   ou_process.py
volatility/
    implied_volatility.py
    models/
       heston.py
       svi.py
    calibration/
    vol_surface.py
```

- **Fichier Data** : Fichier comprenant notamment des modules data_retriever, data_utils permettant d'accéder directement à la donnée présente dans les fichiers de données csv, xlsx, d'assurer un parsing, un nettoyage et un format homogène.
- **Fichier Market** : Représente globalement le marché financier précisé par l'utilisateur à une date donnée et un actif avec des caractéristiques concernant les prix spot, les courbes de d'intérêt (swap EURIBOR/SOFR/ESTR comme historique), la volatilité (historique via une fenêtre choisie ou implciite si disponible) ou encore les niveaux de dividendes jugés fictifs ou éventuellement laissés à l'utilisateur.
- **Fichier Rate** : Représente l'univers de taux avec notamment les différents produits (Obligataire à taux fixe ou flottant, Zéro Coupon, FRA, Forward, IRS...) et notamment la construction de courbes de taux interpolés via les modèles. De tels modules ont visé à respecter au maximum le concept d'héritage (pricing de bond comme portefeuile de ZC...). Existence de différents modules liés aux modèles paramétriques et à taux court pour ajuster ou simuler la structure par terme. Parmis ces modèles, on peut retrouver :
  - **Modèle Nelson–Siegel** : Ajustement paramétrique parcimonieux de la courbe des taux spot.
  - **Modèle Nelson–Siegel–Svensson** : Extension avec un terme supplémentaire flexible.
  - **Pricing Vasicek des Obligations** : Prix analytiques de ZCB sous un taux court à retour vers la moyenne. La calibration d'un tel modèle étant peu évidente, il n'a pas pu être possible de lier le process OU au processus de diffusion de prix de sous-jacent.
  - **ZC Curve** : Module de classes permettant d'obtenir des courbes boostrapées à partir de taux de swap pour obtenir les courbes interpolées de discount, de forward ou zero rates selon le modèle choisi.
- **Fichiers Investment Strategies & Option** : Ces fichiers incluent des modules définissant différents types d'options (par ex. européennes, américaines, barrières)

et leurs propriétés comme leur payoff, leurs niveaux de barrières (Digitales, PDI, CUO...), des stratégies sur options vanilles (type Bear/BullCallSpread, Butterfly, Condor...) avec des legs représentant les options sous jacentes et leur nombre dans la stratégie. Egalement des stratégies structurées avec une approche hybride entre composition obligation & optionnelle ainsi qu'autocallable. Parmis ces modules on peut y retrouver :

- **Structured Strategies** : Agrège plusieurs options et ZC pour la valorisation de différents produits structurés par composition mais également pricing pour Autocallable. Approche orientée objet par héritage et abstraction de `Strategy`.
- **Vanilla Strategies** : Module arborant différentes stratégies optionnelles populaires évoquées précédemment encapsulant les legs (via `get_legs()` et les poids comme un portefeuille d'options pour faciliter le pricing vectoriel.
- **Liste des stratégies Vanilla disponibles** : Bear Call Spread, Bull Call Spread, Put–Call Spread, Butterfly, Straddle, Strap, Strip, Strangle, Condor.
- **Produits Exotiques/Structurés disponibles** : Autocallables (générique, sweet), Reverse Convertibles, Twin Win, Bonus Certificate, Capped Participation, Discount Certificate, Barrier Reverse Convertibles.

- **Fichier Pricers** : Implémente des moteurs de pricing utilisant des simulations Monte Carlo, des arbres trinomiaux ainsi que Longstaff pour le pricing d'options américaines (en bonus). Une approche orientée objet et vectorielle pour faciliser l'agrégation a principalement été adoptée ici avec un abstract `pricing_model` encapsulant des paramètres communs comme `n_steps`, `n_paths`, `seed`, `dt`, `calculate_T()`. A noter que le modèle trinomial n'étant pas totalement adapté à des approches paths dependant, nous ne préconisons pas d'employer ce dernier pour des options barrières. Un effort supplémentaire a été réalisé pour tenir compte de versement de dividendes fictivement versés à l'ex-div date ou bien au format continu à un taux `q`. On peut ainsi retrouver les différents moteurs :
  - **Moteur Monte Carlo** : Simule conjointement les trajectoires du sous-jacent et des taux d'intérêt, supporte la réduction de variance et la méthode Longstaff–Schwartz.
  - **Modèle Trinomial** : Implémente des arbres recombinants (binomial/trinomial) pour le sous-jacent, extensibles pour inclure les taux courts.
  - **Modèle Black & Scholes** : Implémente les formules fermées du modèle de Black & Scholes pour le pricing d'options européennes en prenant en compte les dividendes et également les principales grecques.

- **Modèles de Volatilité** : Implémente la calibration de la volatilité implicite, par exemple le modèle SVI pour les surfaces de volatilité.
  - **Modèle SVI** : Calibre les smiles de volatilité implicite et interpole selon les maturités.
  - **Volatilité Implicite** : Extrait la volatilité implicite $\sigma_{\mathrm{IV}}$ telle que le prix modèle corresponde au prix observé sur le marché par méthode de dichotomie et d'optimisation, la méthode de Newton Raphson a été mise de côté étant donné l'approximation des grecques assez sensibles au paramétrage et donnant des valeurs proches de zéro souvent.

- **Module Mesures de Risque** : Modules permettant le calcul de sensibilités de prix de produit par rapport aux variables d'influence : produits optionnels avec le delta (spot), gamma (delta), theta (temps), rho (shift de la courbe des taux), vega (volatilité) mais également pour les produits de taux avec les DV01, la convexité, la duration. Le calcule des Greeks a été fait par différences finies (Delta, Gamma, Vega, Theta, Rho) via de petites perturbations $\varepsilon$ pour des approches Monte Carlo & Trinomial contre formules fermées pour Black & Scholes. Enfin une approche traditionnelle de risque par le calcul de Value at Risk pour un portefeuille d'options de manière vectorielle agrégée. On peut ainsi retrouver les modules :
  - **SensitivityAnalyzer** : Génère des profils de sensibilité de la valeur du portefeuille en fonction du prix du sous-jacent, de la volatilité et du temps à maturité. Ces deux outils enveloppent le pricer choisi (MC, Longstaff, trinomial) et mettent en cache les prix intermédiaires pour plus d'efficacité.
  - **Greeks** : Instancie un GreekCalculator pour calculer les grecques d'un portefeuille selon le modèle choisi par recréation de modèle avec les nouveaux paramètres "shiftés".
  - **Options_backtest** : Calcul des VaR (Théorique, Monte Carlo, Cornish Fisher) au seuil alpha de manière vectorisée et en utilisant les grecques selon un modèle encapsulé.

# 2. Bref rappel des méthodologies utilisées

## Modèle de Vasicek à taux court

On modélise le taux sans risque instantané $r_t$ par l'EDS à retour vers la moyenne :

$$dr_t = \kappa(\theta - r_t)\, dt + \eta\, dW_t,$$

où :

- $\kappa > 0$ est la *vitesse de réversion*: rapidité avec laquelle $r_t$ revient vers $\theta$.
- $\theta$ est la *moyenne long terme* du taux court.
- $\eta$ est la *volatilité du taux court.*
- $W_t$ est un mouvement brownien standard.

Cette spécification en temps continu garantit que $r_t$ fluctue autour de $\theta$ sous l'effet de chocs aléatoires.

### Discrétisation Euler–Maruyama

Pour simuler numériquement sur des pas uniformes $\Delta t$ :

$$r_{t+\Delta t} = r_t + \kappa(\theta - r_t)\, \Delta t + \eta\, \sqrt{\Delta t}\, Z_t, \quad Z_t \sim \mathcal{N}(0, 1).$$

Le terme de drift $\kappa(\theta - r_t)\Delta t$ ramène $r_t$ vers $\theta$, tandis que le terme de diffusion $\eta\sqrt{\Delta t}\, Z_t$ ajoute de l'aléa.

### Calibration

On ajuste $\kappa, \theta, \eta$ aux taux historiques $\{r_{t_i}\}$ en régressant :

$$r_{t_{i+1}} = \beta\, r_{t_i} + \alpha + \varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}\Big(0, \tfrac{\eta^2}{2\kappa}(1 - e^{-2\kappa\Delta t})\Big),$$

avec $\beta = e^{-\kappa\Delta t}$ et $\alpha = \theta(1-\beta)$. À partir des estimés OLS $\hat{\beta}, \hat{\alpha}$ et de la variance résiduelle $\hat{\sigma}_\varepsilon^2$, on déduit :

$$\hat{\kappa} = -\frac{\ln \hat{\beta}}{\Delta t}, \quad \hat{\theta} = \frac{\hat{\alpha}}{1 - \hat{\beta}}, \quad \hat{\eta} = \sqrt{\frac{2\,\hat{\kappa}\,\hat{\sigma}_\varepsilon^2}{1 - \hat{\beta}^2}}.$$

### Pricing d'une obligation zéro-coupon

Sous le modèle de Vasicek risque-neutre, le prix à $t$ d'une ZCB à l'échéance $T$ est :

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

avec

$$B(t,T) = \frac{1 - e^{-\kappa(T-t)}}{\kappa}, \quad A(t,T) = \Big(\theta - \tfrac{\eta^2}{2\kappa^2}\Big)\big(B(t,T) - (T-t)\big) - \frac{\eta^2}{4\kappa}\, B(t,T)^2.$$

## Ajustement de la courbe de taux

### Modèle Nelson–Siegel

$$y(t) = \beta_0 + \beta_1 \frac{1 - e^{-t/\lambda}}{t/\lambda} + \beta_2\Big(\frac{1 - e^{-t/\lambda}}{t/\lambda} - e^{-t/\lambda}\Big),$$

où :

- $\beta_0$ est le *niveau long terme.*
- $\beta_1$ contrôle la *pente* (effet court terme).
- $\beta_2$ contrôle la *courbure* (bosse moyen terme).
- $\lambda$ est le *facteur de décroissance.*

### Modèle Nelson–Siegel–Svensson

$$y(t) = \beta_0 + \beta_1 \frac{1 - e^{-t/\lambda_1}}{t/\lambda_1} + \beta_2\Big(\frac{1 - e^{-t/\lambda_1}}{t/\lambda_1} - e^{-t/\lambda_1}\Big) + \beta_3\Big(\frac{1 - e^{-t/\lambda_2}}{t/\lambda_2} - e^{-t/\lambda_2}\Big),$$

avec : $\beta_3$ courbure supplémentaire, et $\lambda_1, \lambda_2$ deux facteurs de décroissance.

Avec la donnée de marché des taux sans risque et taux à 3 mois européens (ESTR & EURIBOR 3M), nous avons calibrés les différents modèles par bootstrapping et obtention des taux zéros, en comparaison avec une simple interpolation cubique et/ou linéaire :

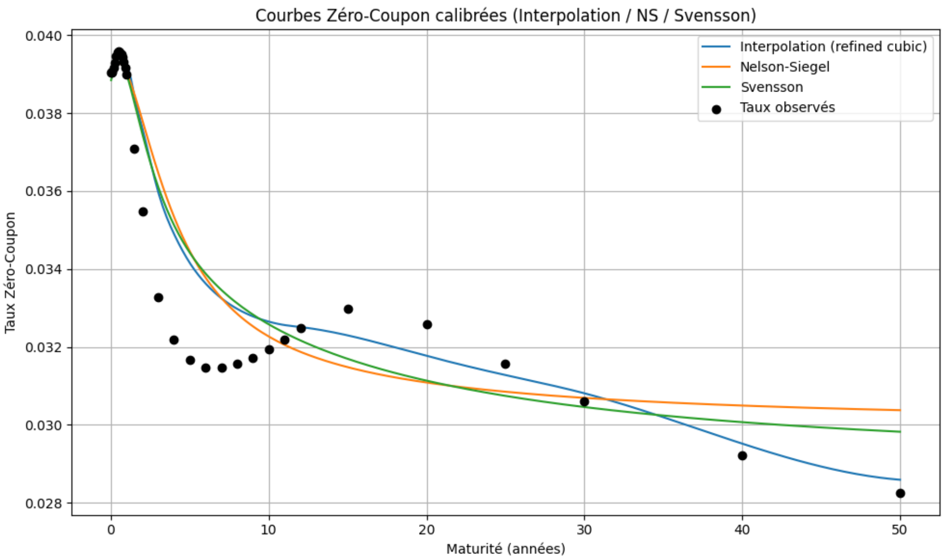


Figure 1: Courbe des taux zéros sans risque ESTR par calibration des différents modèles et interpolation.

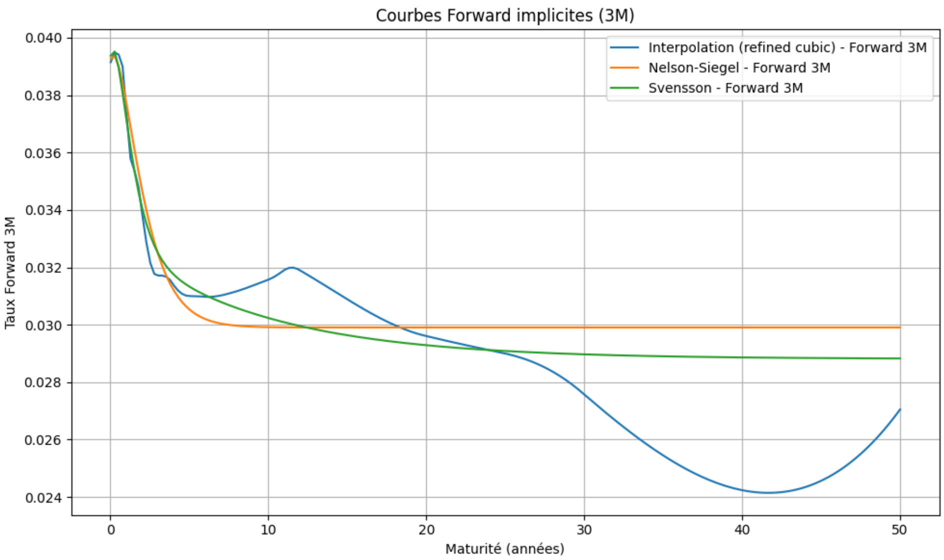


Figure 2: Courbe des taux zéros EURIBOR 3M par calibration des différents modèles et interpolation.

## Simulation Monte Carlo

Pour valoriser sous des taux stochastiques et un sous-jacent $S_t$ :

$$r_{t+\Delta t} = r_t + \kappa(\theta - r_t)\,\Delta t + \eta\sqrt{\Delta t}\,Z_t^r,$$

$$S_{t+\Delta t} = S_t \exp\Big((r_t - \tfrac{\sigma^2}{2})\Delta t + \sigma\sqrt{\Delta t}\,Z_t^S\Big),$$

avec $\mathrm{Corr}(Z_t^r, Z_t^S) = \rho$. Chaque payoff est actualisé par $\exp(-\sum_k r_{t_k}\Delta t)$ puis moyenné sur $N$ trajectoires.

## Arbre trinomial

### Probabilités de transition

$$p_u = \tfrac{1}{2}\Big(\tfrac{\sigma^2\Delta t+(\mu\Delta t)^2}{\sigma^2\Delta t} + \tfrac{\mu\Delta t}{\sigma\sqrt{\Delta t}}\Big),\ p_d = \tfrac{1}{2}\Big(\tfrac{\sigma^2\Delta t+(\mu\Delta t)^2}{\sigma^2\Delta t} - \tfrac{\mu\Delta t}{\sigma\sqrt{\Delta t}}\Big),\ p_m = 1 - p_u - p_d.$$

### Niveaux d'actif

$$S_{i,j} = S_0\, u^j\, d^{\,i-j}, \quad u = e^{\sigma\sqrt{\Delta t}}, \quad d = e^{-\sigma\sqrt{\Delta t}}.$$

### Rétro-induction

$$V_{i,j} = e^{-r\Delta t}\big(p_u V_{i+1,j+1} + p_m V_{i+1,j} + p_d V_{i+1,j-1}\big),$$

et pour les options américaines :

$$V_{i,j} = \max\{\text{exercice immédiat, valeur de continuation}\}.$$

## Méthode Longstaff–Schwartz

1. Simuler de nombreuses trajectoires de $S_t$.
2. À chaque étape (en remontant), régresser les payoffs in-the-money sur des fonctions de base de $S_t$.
3. Estimer la valeur de continuation.
4. Exercer si la valeur intrinsèque dépasse la continuation, sinon continuer.
5. Actualiser et moyenniser les flux pour obtenir le prix.

## Calibration du modèle SVI

Variance implicite totale $w(k)$ en fonction de la log-moneyness $k = \ln(K/S_0)$ :

$$w(k) = a + b\Big(\rho\,(k-m) + \sqrt{(k-m)^2+\sigma^2}\Big),$$

où $a, b, \rho, m, \sigma$ sont calibrés puis interpolés selon la maturité.

### Volatilité implicite

On résout

$$C_{\text{model}}(S_0, K, T, r, q, \sigma_{\text{IV}}) = C_{\text{mkt}}$$

par dichotomie ou Newton–Raphson, avec Vega calculée par différence centrale :

$$V(\sigma) \approx \frac{C_{\text{model}}(\sigma+\varepsilon) - C_{\text{model}}(\sigma-\varepsilon)}{2\varepsilon}.$$

Afin d'avoir une meilleure représentation et dimensionalité des volatilités implicites sur le marché des options, nous représenton la surface sur le SP500 de façon multidimensionnelle :

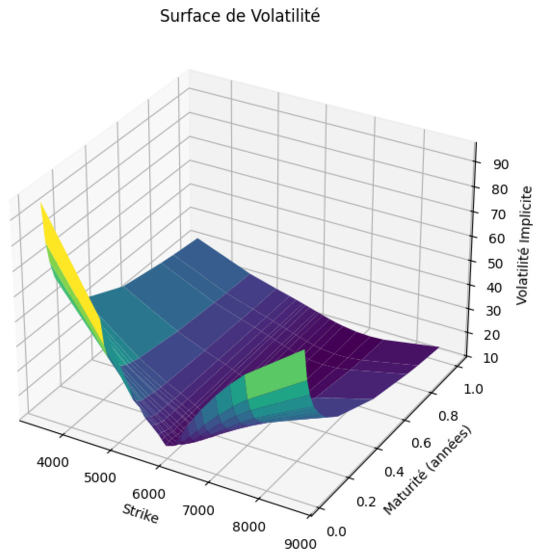


Figure 3: Surface de volatilité pour différentes options sur SP500 pour différentes maturités et strikes.

Les calibrations du modèle SVI sur ces mêmes données et pour différentes maturités sont représentés dynamiquement (ouvrir le fichier via un reader tel qu'Acrobat) :

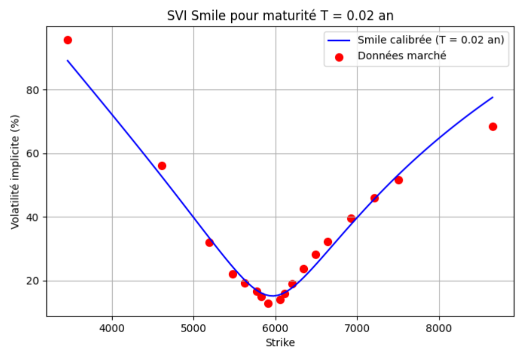


Figure 4: Animation des surfaces de volatilité 2D (à différents strike pour différentes maturité données)

# 3. Processus stochastiques pour le pricing

- **Mouvement Brownien Géométrique (GBM)** :

$$dS_t = \mu S_t \, dt + \sigma S_t \, dW_t, \quad S_t = S_0 \exp\Big((\mu - \tfrac{\sigma^2}{2})t + \sigma W_t\Big).$$

- **Mesure risque-neutre** :

$$dS_t = r S_t \, dt + \sigma S_t \, dW_t, \quad S_t = S_0 \exp\Big((r - \tfrac{\sigma^2}{2})t + \sigma W_t\Big).$$

- **Propriétés de $W_t$** : incréments indépendants, $W_t \sim \mathcal{N}(0, t)$.
- **Processus Ornstein–Uhlenbeck** :

$$dX_t = \theta(\mu - X_t)\, dt + \sigma\, dW_t, \quad X_t = X_0 e^{-\theta t} + \mu(1 - e^{-\theta t}) + \sigma \int_0^t e^{-\theta(t-s)} dW_s.$$

  Le modèle de Black & Scholes étant réduit à une contrainte de constance sur les taux d'intérêts pour la diffusion, représentant une hypothèse forte et peu réaliste, nous avons ici joint au processus GBM un OU Process issus d'une calibration (arbitraire) du modèle de Vasicek :

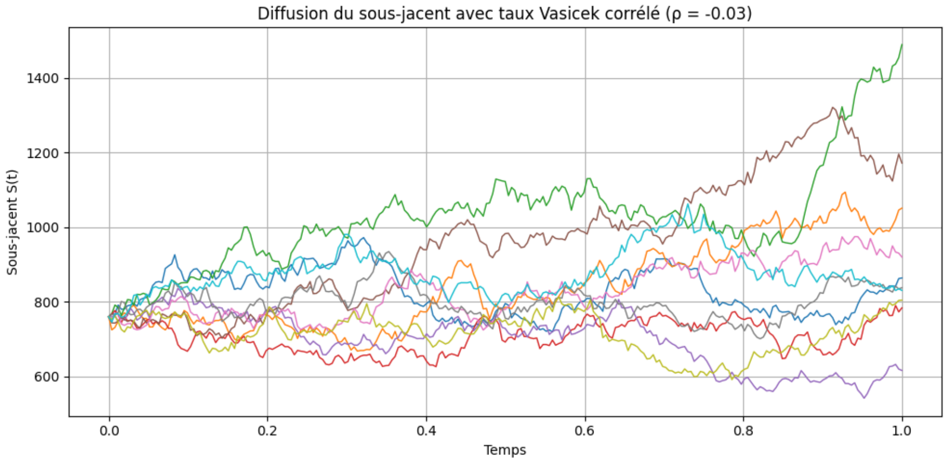


Figure 5: Diffusions GBM avec corrélation aux simulations par processus OU des taux d'intérêts.

# 4. Produits structurés

Ces produits financiers sont des stratégies d'investissement hybrides (combinaison de produit fixed income  dérivé optionnel souvent) et sur-mesures au profil de risque de l'investisseur. Notre méthodologie de pricing respecte une logique de *"all residual invested"* pour toute position ayant une partie long pour la performance (d'où le fait que certains produits peuvent pricer 100% du notionnel) contrairement aux positions short.

## Produit Autocallable

Flux de trésorerie path-dépendants aux dates d'observation $t_1, \ldots, t_N$, définis par trois barrières :

- Barrière coupon $B_c$ (fraction de $S_0$)
- Barrière autocall $B_a$
- Barrière protection $B_p$

Soit $C_i$ le coupon annuel payable à $t_i$. On note $S_{t_i}$ le spot à cette date.

$$\begin{aligned}
&\text{À la première date de call } t_i : \\
&\quad \text{si } S_{t_i} \geq B_a S_0, \ \text{payer } N\Big(1 + \sum_{j=1}^{i} C_j\Big), \ \text{fin du produit.} \\
&\text{Sinon, à maturité } t_N : \\
&\quad \begin{cases} S_{t_N} \geq B_p S_0 : & N\big(1 + \sum_{j=1}^{N} C_j\big), \\ S_{t_N} < B_p S_0 : & N \frac{S_{t_N}}{S_0}. \end{cases}
\end{aligned}$$

Les coupons $C_i$ ne sont versés que si le produit n'a pas été autocallé auparavant et si $S_{t_i} \geq B_c S_0$.

## Sweet Autocall

Cas particulier de l'autocall générique, avec dates et coupons définis par l'utilisateur, même structure de payoff.

## Reverse Convertible

Combine :

- Long ZCB de valeur nominale $N$ à l'échéance $T$.
- Short Put européen de strike $K$ arrivant à $T$.

$$\Pi_{RC}(S_T) = N - \max(K - S_T, 0).$$

## Twin Win

Combine :

- Long ZCB nominal $N$.
- Long Call Up-and-Out ($K$, barrière $B_u$).
- Long Put Down-and-Out ($K$, barrière $B_d$).

$$\Pi_{TW}(S_T) = N + 1_{\{\sup_{t \leq T} S_t < B_u S_0\}} \max(S_T - K, 0) + 1_{\{\inf_{t \leq T} S_t > B_d S_0\}} \max(K - S_T, 0).$$

## Bonus Certificate

Combine :

- Long ZCB ($N$).
- Long Call ($K$).
- Short Put Down-and-Out ($K$, barrière $B$).

$$\Pi_{BC}(S_T) = N + \max(S_T - K, 0) - 1_{\{\min_{t \leq T} S_t > BS_0\}} \max(K - S_T, 0).$$

Si la barrière est touchée, le put disparaît et le payoff devient $N + \max(S_T - K, 0)$.

### Capped Participation Certificate

Combine :

- Long ZCB ($N$).
- Long Call ($K$).
- Long Digital Call ($K$, payoff $L = \text{cap} - K$).

$$\Pi_{CPC}(S_T) = N + \max(S_T - K, 0) + L\, 1_{\{S_T \geq K\}}.$$

### Discount Certificate

Combine :

- Long ZCB ($N$).
- Short Put ($K$).

$$\Pi_{DC}(S_T) = N - \max(K - S_T, 0).$$

### Reverse Convertible à barrière

Combine :

- Long ZCB ($N$).
- Short Put ($K$).
- Long Put Down-and-Out ($K$, barrière $B$).

$$\Pi_{RCB}(S_T) = N - \max(K - S_T, 0) + 1_{\{\min_{t\leq T} S_t > BS_0\}} \max(K - S_T, 0).$$

## 5. Stratégies d'options Vanilla

### Bear Call Spread

**Opinion de marché:** Baissier modéré ou neutre. **Composition:** Vente d'un call strike $K_1$, achat d'un call strike $K_2 > K_1$.

$$\Pi_{\text{BCS}}(S_T) = -(S_T - K_1)^+ + (S_T - K_2)^+.$$

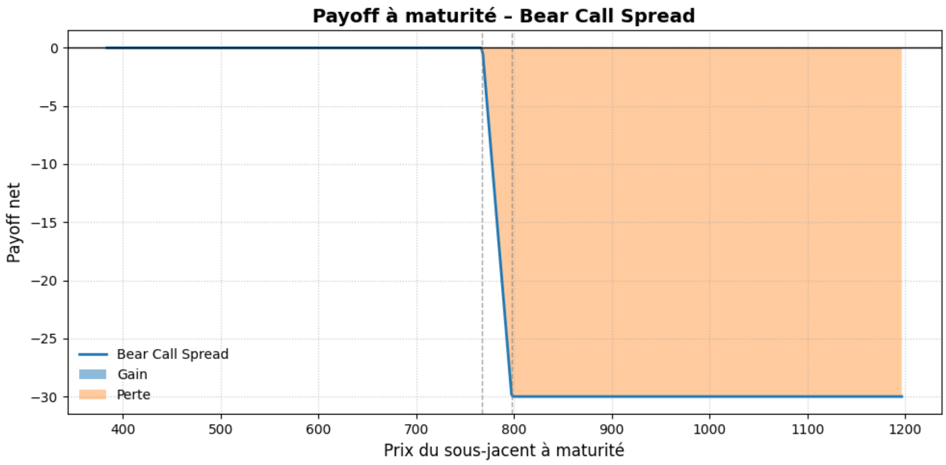


Figure 6: Payoff du Bear Call Spread *(paramètres du cas d'exemple sec. 7)*

## Bull Call Spread

**Opinion de marché:** Haussier modéré. **Composition:** Achat d'un call $K_1$, vente d'un call $K_2 > K_1$.

$$\Pi_{\text{BuCS}}(S_T) = (S_T - K_1)^+ - (S_T - K_2)^+.$$

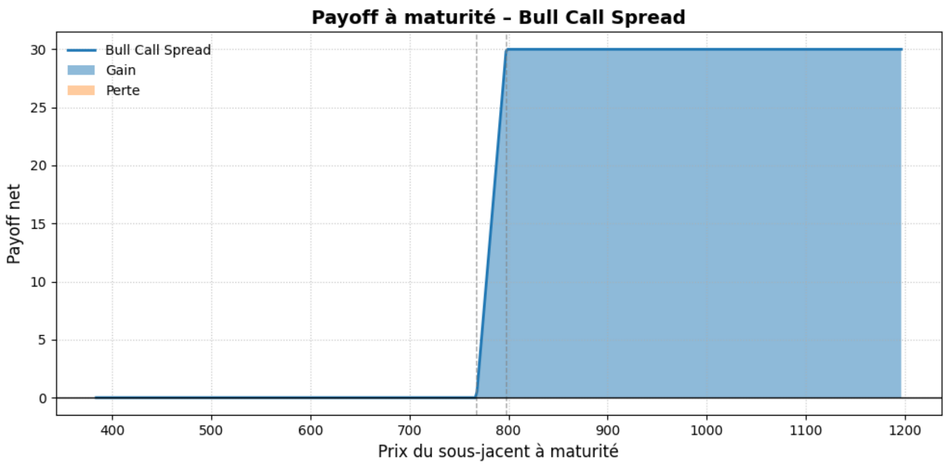


Figure 7: Payoff du Bull Call Spread *(paramètres du cas d'exemple sec. 7)*

## Put–Call Spread

**Opinion de marché:** Neutre à baissier, joue la différence de prime. **Composition:** Achat d'un put et vente d'un call au même strike $K$.

$$\Pi_{\text{PCS}}(S_T) = (K - S_T)^+ - (S_T - K)^+.$$

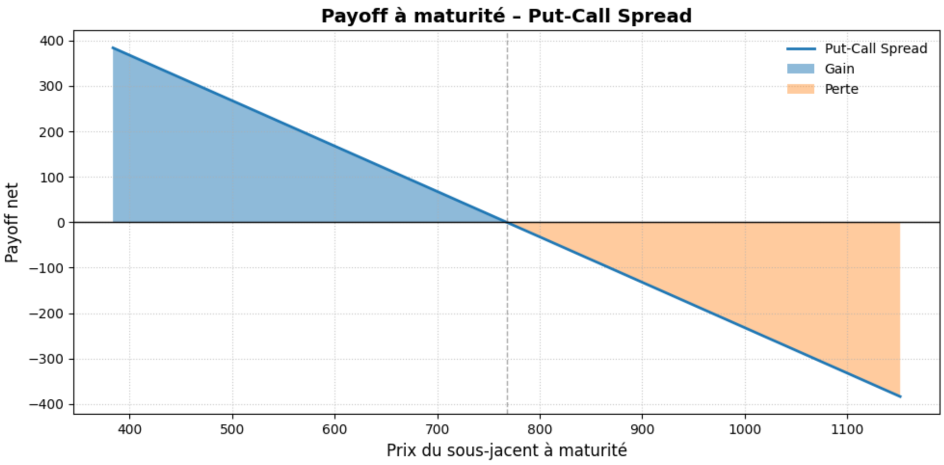


Figure 8: Payoff du Put-Call Spread *(paramètres du cas d'exemple sec. 7)*

## Butterfly (Approche calls)

**Opinion de marché:** Neutre, faible volatilité attendue. **Composition:** +1 call $K_1$, –2 calls $K_2$, +1 call $K_3$ avec $K_1 < K_2 < K_3$.

$$\Pi_{\mathrm{BF}}(S_T) = (S_T - K_1)^+ - 2(S_T - K_2)^+ + (S_T - K_3)^+.$$

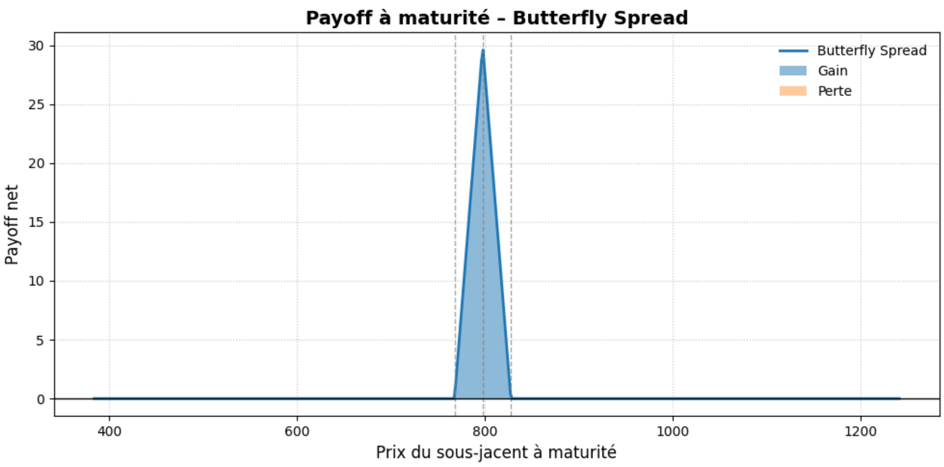


Figure 9: Payoff du Butterfly (Calls) *(paramètres du cas d'exemple sec. 7)*

## Straddle

**Opinion de marché:** Attente d'un grand mouvement. **Composition:** Achat d'un call et d'un put au même strike $K$.

$$\Pi_{\mathrm{Straddle}}(S_T) = |S_T - K|.$$

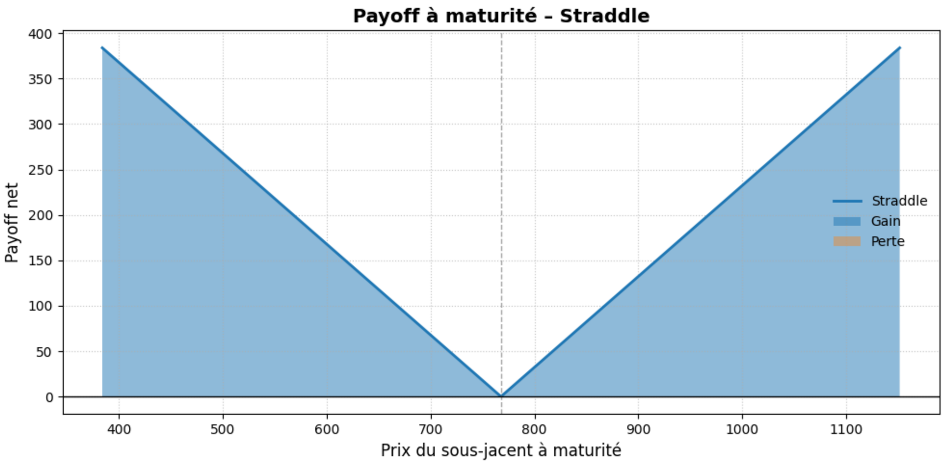


Figure 10: Payoff du Straddle *(paramètres du cas d'exemple sec. 7)*

## Strap

**Opinion de marché:** Très haussier. **Composition:** Achat de 2 calls et 1 put au strike $K$.

$$\Pi_{\text{Strap}}(S_T) = 2(S_T - K)^+ + (K - S_T)^+.$$

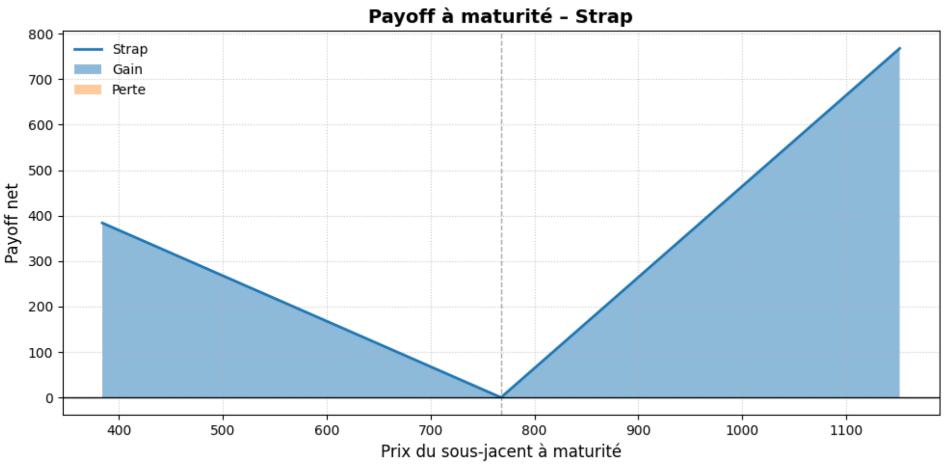


Figure 11: Payoff du Strap *(paramètres du cas d'exemple sec. 7)*

## Strip

**Opinion de marché:** Très baissier. **Composition:** Achat de 1 call et 2 puts au strike $K$.

$$\Pi_{\text{Strip}}(S_T) = (S_T - K)^+ + 2(K - S_T)^+.$$

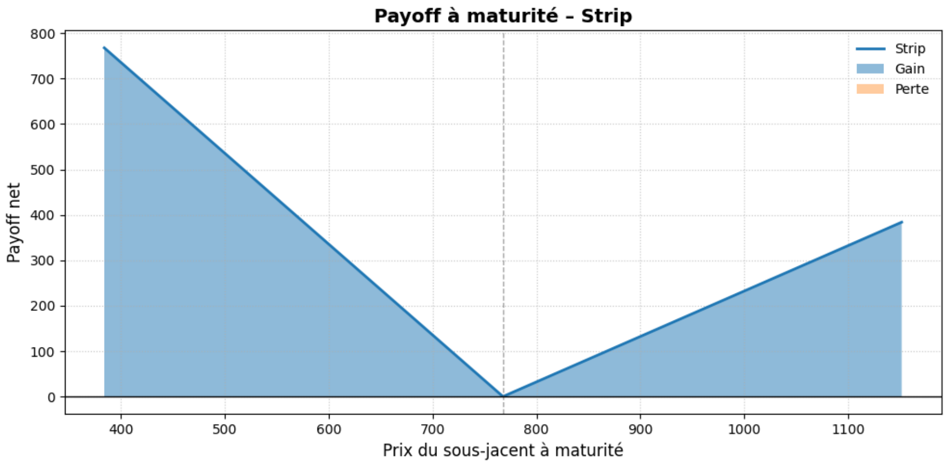


Figure 12: Payoff du Strip *(paramètres du cas d'exemple sec. 7)*

## Strangle

**Opinion de marché:** Grand mouvement sans direction. **Composition:** Achat d'un put $K_L$ et d'un call $K_H > K_L$.

$$\Pi_{\text{Strangle}}(S_T) = (K_L - S_T)^+ + (S_T - K_H)^+.$$

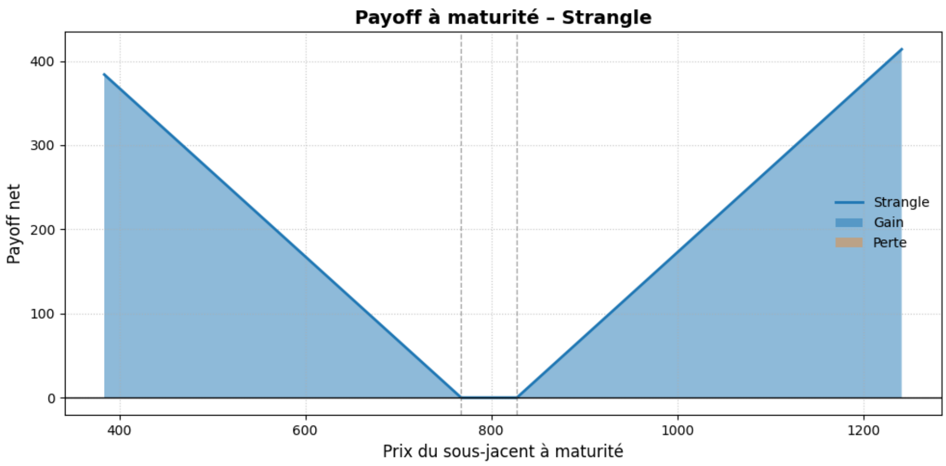


Figure 13: Payoff du Strangle *(paramètres du cas d'exemple sec. 7)*

## Condor (Approche Calls)

**Opinion de marché:** Neutre, plateau de profit. **Composition:** +1 call $K_1$, –1 call $K_2$, –1 call $K_3$, +1 call $K_4$ avec $K_1 < K_2 < K_3 < K_4$.

$$\Pi_{\text{Condor}}(S_T) = (S_T - K_1)^+ - (S_T - K_2)^+ - (S_T - K_3)^+ + (S_T - K_4)^+.$$

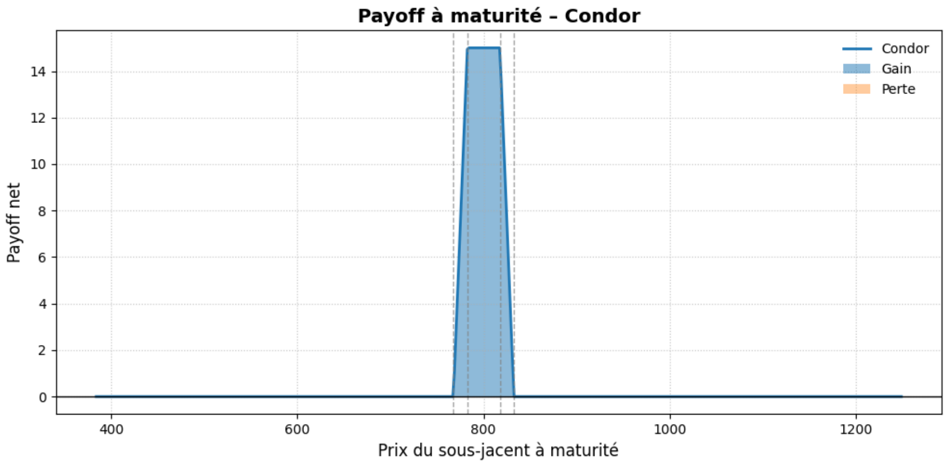


Figure 14: Payoff du Condor *(paramètres du cas d'exemple sec. 7)*

# 6 Produits de Taux

## Obligation Zéro-Coupon (ZCB)

**Principe :** Instrument à revenu fixe sans paiement intermédiaire. **Flux :** Paiement unique à maturité d'un montant nominal $N$.

$$\Pi_{\text{ZCB}}(t) = N \cdot P(t,T)$$

où $P(t,T)$ est le facteur d'actualisation de $t$ à $T$.

## Obligation à Taux Fixe (Fixed Rate Bond)

**Principe :** Série de paiements réguliers de coupons $C = \text{coupon\_rate} \cdot N$, puis remboursement du nominal. **Flux :**

$$\Pi_{\text{FRB}}(t) = \sum_{i=1}^{n} C \cdot P(t,T_i) + N \cdot P(t,T_n)$$

avec $T_i$ les dates de paiement et $P(t,T_i)$ les facteurs d'actualisation.

## Obligation à Taux Flottant (Floating Rate Bond)

**Principe :** Le coupon est indexé sur une courbe de taux forward $f(t_i, t_{i+1})$, avec une marge (spread) éventuelle.

$$C_i = N \cdot (f(t_i, t_{i+1}) + \text{margin}) \cdot \delta_i$$

**Flux :** Somme des coupons flottants et remboursement final du nominal.

### Forward Rate

**Principe :** Taux d'intérêt implicite entre deux dates futures $T_1 < T_2$, déduit de la courbe des ZCB :

$$f(T_1, T_2) = \frac{1}{\delta}\left(\frac{P(0,T_1)}{P(0,T_2)} - 1\right), \quad \delta = T_2 - T_1$$

### Forward Rate Agreement (FRA)

**Principe :** Contrat à règlement unique qui spécule sur un taux futur $L(T_1, T_2)$ par rapport à un strike $K$.

$$\mathrm{MtM}_{\mathrm{FRA}} = N \cdot \delta \cdot \frac{L(T_1,T_2) - K}{1 + \delta \cdot L(T_1,T_2)}$$

### Interest Rate Swap (IRS)

**Principe :** Échange entre un flux à taux fixe et un flux à taux flottant. **MtM :**

$$\mathrm{MtM}_{\mathrm{swap}} = N \cdot \left(\sum_{i=1}^{n} \left(f(t_{i-1}, t_i) - K\right) \cdot \delta_i \cdot P(0, t_i)\right)$$

où $f(t_{i-1}, t_i)$ est le taux forward, $K$ est le taux fixe du swap.

## 7. Mesures de risque

Après avoir décrit les payoffs, on passe aux *risk metrics*, mesures de sensibilité de la valeur $V$ aux paramètres de marché, calculées par différences finies ($\varepsilon \approx 10^{-2}$) ou formules analytiques pour les produits de taux.

### Greeks par différences finies

$$\begin{aligned}
\Delta &\approx \frac{V\big(S_0(1+\varepsilon)\big) - V\big(S_0(1-\varepsilon)\big)}{2\,\varepsilon\,S_0}, \\
\Gamma &\approx \frac{V\big(S_0(1+\varepsilon)\big) - 2\,V(S_0) + V\big(S_0(1-\varepsilon)\big)}{\varepsilon^2\,S_0^2}, \\
\mathrm{Vega} &\approx \frac{V(\sigma+\varepsilon) - V(\sigma-\varepsilon)}{2\,\varepsilon}, \\
\Theta &\approx V\big(t + 1 \text{ jour}\big) - V(t), \\
\rho &\approx \frac{V(r+\varepsilon) - V(r-\varepsilon)}{2\,\varepsilon}.
\end{aligned}$$

### Profils de sensibilité

Pour compléter les Greeks, on trace des profils de *sensibilité globale* :

- $V(S_0)$ en fonction du spot $S$.
- $V(\sigma)$ en fonction de la volatilité.
- $V(T)$ en fonction du temps à maturité.
- $V(r)$ en fonction d'un shift parallèle des taux (VaR de taux).

### Sensibilités produits de taux

Pour un produit à cash-flows $CF_i$ payés aux temps $t_i$ :

$$\mathrm{PV} = \sum_i CF_i \, DF(t_i), \quad DF(t) = e^{-y(t)\,t}.$$

**DV01**

$$\mathrm{DV01} = -\frac{\partial \mathrm{PV}}{\partial y} \times 10^{-4} = -\sum_i t_i \, CF_i \, DF(t_i) \times 10^{-4}.$$

**Macaulay Duration**

$$D_M = \frac{\sum_i t_i \, CF_i \, DF(t_i)}{\sum_i CF_i \, DF(t_i)}.$$

**Convexité**

$$\text{Convexity} = \sum_i t_i^2 \, CF_i \, DF(t_i) \times 10^{-4}.$$

### Profils de sensibilité de taux

En complément, on trace $V$ d'un produit de taux en fonction d'un shift parallèle $h$ de la courbe :

$$DF_h(t) = DF(t)\, e^{-h\,t},$$

pour $h \in [-50\text{bp}, +50\text{bp}]$.

## 8. Cas de test : Paramètres, actifs et résultats

### Paramètres de marché et sous-jacent

### Actifs et stratégies testés

- **Options vanilles** : Call, Put, DigitalCall (payoff €10), DigitalPut (payoff €10).
- **Options barrières** : UpAndOutCall, UpAndInCall, DownAndOutPut, DownAndInPut.

| | |
|---|---|
| **Date de pricing** | 2023-04-25 |
| **Maturité** | 2028-04-25 |
| **Sous-jacent** | LVMH ($S_0 = \texttt{853.002}$) |
| **Strike** | $K = 0.9\,S_0$ |
| **Barrières** | down $= 0.8\,S_0$, up $= 1.2\,S_0$ |
| **Convention jours** | Actual/365 |
| **Courbe de taux** | Svensson (guess = $[0.02, -0.01, 0.01, 0.005, 1.5, 3.5]$) |
| **Générateurs de price** | MC (10 000 paths, 300 steps), Trinomial (300 steps), Black–Scholes |
| **Exercise** | Européen |

Table 1: Paramètres clés de l'étude et configuration des pricers.

- **Produits structurés** : SweetAutocall, TwinWin, ReverseConvertible, BonusCertificate, CappedParticipationCertificate, DiscountCertificate, ReverseConvertibleBarrier.
- **Produits de taux** : ZeroCouponBond, FixedRateBond (6 % semestriel), FloatingRateBond, ForwardRate (1→2 ans), ForwardRateAgreement, InterestRateSwap (payer fixe 5 ans).
- **Stratégies vanilles** : BearCallSpread, BullCallSpread, ButterflySpread, Straddle, Strap, Strip, Strangle, Condor, PutCallSpread.

## Résultats numériques

| **Modèle** | $\mathrm{VaR}_{\mathrm{Th}}$ | $\mathrm{VaR}_{\mathrm{MC}}$ | $\mathrm{VaR}_{\mathrm{CF}}$ |
|---|---|---|---|
| Black–Scholes | 92.53 | 91.87 | 89.36 |
| Trinomial tree | 92.57 | 117.60 | 89.38 |
| Monte Carlo | 93.31 | 91.50 | 89.97 |

Table 2: VaR (théorique, MC, Cornish–Fisher) pour un straddle via trois modèles.

| **Produit** | Prix / Notional [%] |
|---|---|
| Sweet Autocall | 87.52 |
| Twin Win | 100.00 |
| Reverse Convertible | 75.58 |
| Bonus Certificate | 99.98 |
| Capped Participation Certificate | 100.00 |
| Discount Certificate | 75.58 |
| Reverse Convertible Barrier | 90.04 |

Table 3: Prix estimés des produits structurés (% du nominal).

| **Produit** | Prix | DV01 | Duration | Convexité |
|---|---|---|---|---|
| Zero-Coupon Bond | 855.34 (€) | -0.4281 | 5.0055 | 2.1430 |
| Fixed Rate Bond 6 % | 113.03 (%) | -0.5016 | 4.4375 | 2.3937 |
| Floating Rate Bond | 101.86 (%) | -0.4709 | 4.6233 | 2.2881 |
| Swap payer fixe (MtM / TF) | -6 358.56 (€) / 3.15% | 2.8204 | -0.0139 | -12.5998 |

Table 4: Prix et sensibilités (DV01, Macaulay duration, convexité) des produits de taux.

| **Stratégie** | Prix | CI 95 % |
|---|---|---|
| Bear Call Spread | -13.84 | [–13.85, –13.82] |
| Bull Call Spread | +13.84 | [13.82, 13.85] |
| Butterfly Spread | +0.65 | [0.64, 0.66] |
| Straddle | 400.52 | [393.33, 407.70] |
| Strap | 701.47 | [688.50, 714.44] |
| Strip | 500.08 | [491.49, 508.67] |
| Strangle | 373.49 | [366.29, 380.70] |
| Condor | +0.55 | [0.54, 0.55] |
| Put–Call Spread | -201.39 | [–205.77, –197.01] |

Table 5: Prix et intervalles de confiance (MC, $\alpha = 5\%$) pour les stratégies vanilles.

# 9. Lancement et gestion de l'application Streamlit

## 9.1 Prérequis

Avant de démarrer l'application Web, assurez-vous d'avoir :

- Python **3.11+** installé et configuré sur votre poste.
- Le gestionnaire de paquets `uv` (optionnel) ou, à défaut, `pip`.

- Le dépôt du projet cloné en local, avec le fichier `streamlit`$_{o}ption_{p}ricer.py\grave{a}laracine.$

## 9.2 Installation des dépendances

Vous pouvez choisir l'une des deux méthodes suivantes :

**Avec** `uv`

```
pip install uv    % Installer le gestionnaire uv
uv sync          % Synchroniser et installer toutes les dépendances
```

**Avec** `pip`

```
pip install .     % Installation directe depuis le dépôt courant
```

## 9.3 Démarrage de l'application

Positionnez-vous dans le répertoire racine du projet (contenant `streamlit_option_pricer.py`), puis exécutez :

```
# Si vous utilisez uv :
uv run streamlit run streamlit_option_pricer.py

# Sinon, directement avec Streamlit :
streamlit run streamlit_option_pricer.py
```

Le serveur Streamlit se lancera alors et affichera une URL locale (par défaut `http://localhost:8501`) à ouvrir dans votre navigateur.

## 9.4 Navigation et utilisation

- **Barre latérale** : configurer le marché (ticker, date de valorisation, source de volatilité, fenêtre historique), la méthode de construction des courbes et les paramètres Monte Carlo / Tree.
- **Onglets** : choisir la catégorie de produits (Structured, Option, Strategy, Rate).
- **Widgets dynamiques** : renseigner les paramètres spécifiques au produit (strike, maturité, coupons, barrières, etc.).
- **Méthode de pricing** : sélectionner entre MC, Tree ou BS (pour les options et stratégies).

- **Bouton Pricer** : lancer le calcul et afficher le prix, les Greeks et/ou les sensibilités (DV01, Duration, Convexité). Pour les stratégies, le graphique de payoff est généré automatiquement.

## 9.5 Organisation du code Streamlit

Le fichier `streamlit_option_pricer.py` orchestre :

- L'import des modules `market`, `pricers`, `rate`, `option`, `risk_metrics` et `investment_strategies`.
- La construction des widgets Streamlit à partir des dictionnaires `COMMON_PARAMS` et `SPECIFIC_PARAMS` définis dans `app.py`.
- La logique de dispatch vers le bon moteur de pricing (Monte Carlo, Tree, BSM ou StructuredPricer).
- La conversion et la normalisation des paramètres (dates, conventions, quotes en pourcentage vs valeurs absolues).