Modélisation et Pricing de produits structurés

Documentation technique

Fanny Gauducheau, Lauryn Letaconnoux et Giovanni Manche

Cours de Produits structurés

M2 Ingénierie Économique et Financière - 272, Parcours Quantitatif

Université Paris-Dauphine — PSL

Avril 2025

Table des matières

Introduction

L'objectif de ce document est de présenter de manière détaillée les outils utilisés dans le développement d'un pricer de produits structurés, réalisé dans le cadre du cours de Produits Structurés en Python. Ce projet vise à concevoir une application générale, scalable et adaptable, capable de traiter le pricing d'un large éventail de produits financiers : obligations, options vanilles ou exotiques, ainsi que d'autres instruments structurés.

Compte tenu de la diversité des produits structurés — tant sur le plan de leur modélisation que de leur méthode de valorisation — un exposé clair des méthodes utilisées s'impose. C'est la vocation de cette documentation, qui permettra au lecteur de comprendre la structure de l'application et d'en saisir les sous-jacents économiques et financiers.

Grâce à cette lecture, l'utilisateur pourra comprendre l'architecture du code, identifier les produits déjà intégrés dans le *scope* de l'application, et saisir la logique d'ajout de nouveaux instruments. En effet, l'approche orientée objet adoptée repose sur une structure modulaire, facilitant l'implémentation ou le retrait de produits et de modèles supplémentaires.

Le document suit l'ordre de construction du code, et donc du pricing de produits structurés. Effectivement, nous commençons par poser les bases de calculs avec les différents modèles utilisés pour calibrer les taux ainsi que la volatilité. Puis, nous décrivons brièvement les différents types de produits structurés disponibles sur notre application. Ceux-ci sont organisés en catégories en fonction de caractéristiques communes. La dernière partie explique le fonctionnement de l'outil de pricing, selon 3 méthodes différentes (Monte Carlo, Black Scholes ainsi qu'une méthode "hybride", la décomposition des produits).

1 Modèles de volatilité

Dans cette partie, nous allons présenter deux modèles de volatilité que nous avons implémentés dans notre application, à savoir le modèle SSVI et le modèle de Dupire. Ces deux modèles ont chacun leurs avantages et leurs limites, et l'utilisateur a le choix du modèle.

1.1 Volatilité paramétrique - le modèle SSVI

1.1.1 Définition et formules

Le modèle SSVI (*Surface Stochastic Volatility Inspired*) est un modèle de volatilité **paramétrique**, c'est-à-dire que la volatilité est fonction d'un nombre limité de paramètres. En l'occurrence, le modèle SSVI est une extension du modèle SVI (*Stochastic Volatility Inspired*), qui paramétrise la variance totale au niveau de *moneyness* $k = \ln\left(\frac{K}{S}\right)$. Le modèle SSVI étend le modèle SVI et propose une paramétrisation pour toute la surface de volatilité, sur la base des équations suivantes :

$$w(k) = \sigma_{BS}^2 \times t = \frac{\theta_t}{2} \times \left(1 + \rho\phi(\theta_t)k + \sqrt{(\phi(\theta_t)k + \rho)^2 + (1-\rho^2)}\right)$$
$$|\rho| < 1, k = \ln(\frac{K}{S})$$
$$\theta_t = \left(\frac{1-\exp(-\kappa t)}{\kappa t}(\nu_0 - \nu_\infty) + \nu_\infty\right) t$$
$$\phi(\theta_t) = \eta\theta_t^\lambda, \eta > 0, \lambda \in]0,1]$$

où :

- $w(k)$ est la variance implicite totale (volatilité implicite de Black-Scholes au carré multipliée par le temps jusqu'à l'échéance)
- $k = \ln\left(\frac{K}{S}\right)$ est le log-moneyness
- σ_{BS} est la volatilité implicite de Black-Scholes
- t est le temps jusqu'à l'échéance
- ρ est le paramètre de corrélation avec $|\rho| < 1$
- θ_t représente la variance totale ATM (At-The-Money) pour la maturité t
- $\phi(\theta_t)$ est une fonction de paramétrisation de la forme du smile

1.1.2 Calibration et implémentation

La calibration du modèle SSVI se fait en deux étapes :

1. Estimation de κ, v_0 et v_∞ par minimisation de l'erreur au carré ATM :

$$\min \sum_{i,ATM} (Prix_{BS}(\sigma_t = \theta_t(\kappa, v_0, v_\infty)) - MktPrice)^2$$

2. Estimation de ρ, λ, η par minimisation de :

$$\sum_i (Prix_{BS}(\sigma_t = w(k, \rho, \eta, \lambda, \theta_t(\hat{\kappa}, \hat{v_0}, \hat{v_\infty}))) - MktPrice)^2$$

Dans notre code, la minimisation est assurée par la méthode `calibrate`, qui procède en trois temps :

1. On récupère les strikes ATM. En effet, en raison d'effets de seuil, il se peut que plusieurs strikes soient proches du prix du sous-jacent. La tolérance est de 0.05% (sinon, le strike le plus proche du prix du sous-jacent est le seul sélectionné).
2. Minimisation ATM
3. Minimisation globale avec $\hat{\kappa}, \hat{v_0}, \hat{v_\infty}$ fixés

Différentes technicalités de l'optimisation sous Python sont utilisées (bornes inférieures pour les paramètres, essais à partir de différents points de départ,...) sans que cela ne gêne l'optimisation.

En utilisant ce modèle sur une surface de prix de calls sur l'action AAPL pour différents strikes et maturités, à date du 17 avril 2025, avec un prix du sous-jacent de 198, nous obtenons la surface de volatilité suivante :

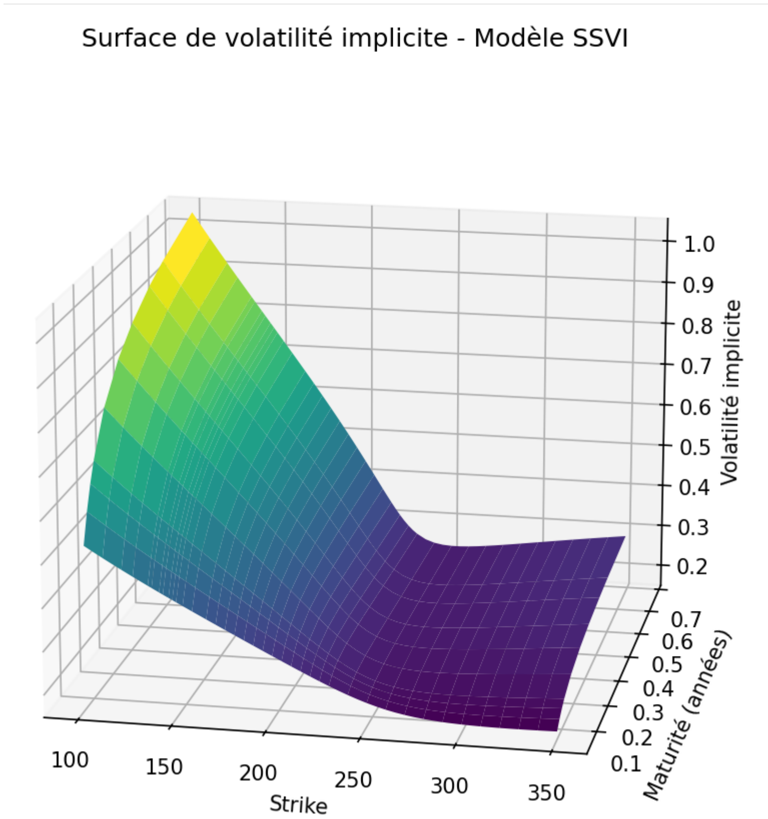

FIGURE 1 – Surface de volatilité des options d'achat sur l'action Apple, modèle SSVI

1.2 Volatilité locale - le modèle de Dupire

1.2.1 Définition et formules

Le modèle de Dupire est un modèle de volatilité **locale**, c'est-à-dire qu'elle considère la volatilité comme une fonction déterministe du prix du sous-jacent et du temps. L'idée clé du modèle de Dupire est que la volatilité est une fonction déterministe du prix spot et du temps, et cette fonction peut être déduite de la surface de volatilité implicite observée sur le marché. La formule de Dupire permet de reconstruire la fonction de volatilité locale à partir des options :

Fonction de la volatilité implicite : $\sigma_{loc}^2(K,t) = \dfrac{\sigma_{IMP}^2 + 2\sigma_{IMP}T \times \left(\frac{\partial\sigma}{\partial T} + (r-q)K \times \frac{\partial\sigma}{\partial K}\right)}{\left(1 + Kq \times \frac{\partial\sigma}{\partial K}\sqrt{T}\right)^2 + \sigma_{IMP}K^2T \times \left(\frac{\partial^2\sigma}{\partial K^2} - q\left(\frac{\partial\sigma}{\partial K}\right)^2\sqrt{T}\right)}$

Forme en fonction du prix du call : $\sigma_{loc}^2(K,t) = \sqrt{2 \times \dfrac{\frac{\partial C}{\partial T} + rK \times \frac{\partial C}{\partial K}}{K^2 \times \frac{\partial^2 C}{\partial K^2}}}$

avec :

— $\sigma_{loc}^2(K,t)$: La variance locale qui dépend du prix d'exercice K et du temps t

— σ_{IMP}^2 : La variance implicite (volatilité implicite au carré)

— σ_{IMP} : La volatilité implicite extraite des prix de marché des options

— T : Le temps jusqu'à l'échéance (maturité) de l'option

— $\frac{\partial\sigma}{\partial T}$: La dérivée partielle de la volatilité par rapport au temps

— $\frac{\partial\sigma}{\partial K}$: La dérivée partielle de la volatilité par rapport au prix d'exercice, "skew" de volatilité

— $\frac{\partial^2\sigma}{\partial K^2}$: La dérivée seconde de la volatilité par rapport au prix d'exercice, mesurant la courbure du "smile" de vo

— r : Le taux d'intérêt sans risque

— q : Le taux de dividende continu ou le coût de portage

— K : Le prix d'exercice de l'option

1.2.2 Calibration et implémentation

Pour le modèle de Dupire, il n'y a pas de "calibration" à proprement parlé, étant donnée l'absence de paramètres. Néanmoins, il fut nécessaire d'implémenter des méthodes de calculs de dérivées (nous avons utilisé la méthode des différences finies), afin de récupérer la volatilité locale en fonction de la volatilité implicite. Ainsi, nous procédons en deux temps :

1. On calcule la volatilité implicite en inversant la relation de Black-Scholes,
2. On calcule la volatilité locale en fonction de la volatilité implicite.

2 Options

Dans toute la suite, nous utiliserons les notations suivantes :

— S_0 désigne le prix actuel du sous-jacent,

— S_T désigne le prix à maturité du sous-jacent,

— K désigne le prix d'exercice,

— T désigne la maturité en années,

— r désigne le taux sans risque,

— q désigne le taux de dividende (continu),

— σ désigne la volatilité du sous-jacent,

— $N(.)$ désigne la fonction de répartition de la loi Normale centrée et réduite. $\mathcal{N}(.)$ désigne la fonction de **densité** de cette même loi.

2.1 Options vanilles

2.1.1 Pricing et payoff

Par options vanilles, nous entendons les options d'achat (*call*) et de vente (*put*), classiques, qui donnent le droit (mais non l'obligation) d'acheter (resp. vendre) le sous-jacent à maturité. Les payoffs de ces options sont :

$$\text{Payoff d'un call : } \max(S_T - K, 0)$$
$$\text{Payoff d'un put : } \max(K - S_T, 0)$$

Pour en déterminer le prix, Black & Scholes proposent des formules fermées bien connues :

$$\text{Prix d'un call : } C_0 = S_T e^{-qT} \times N(d_1) + Ke^{-rT} N(d_2)$$
$$\text{Prix d'un put : } P_0 = Ke^{-rT} N(-d_2) + S_T e^{-qT} \times N(-d_1)$$
$$d_1 = \frac{ln(\frac{S_0}{K}) + (r - q + \frac{\sigma^2}{2})T}{\sigma\sqrt{T}}$$
$$d_2 = d_1 - \sigma\sqrt{T}$$

Un autre moyen est d'utiliser la méthode de Monte Carlo, que nous décrirons ci-après.

2.1.2 Grecques

Le calcul des sensibilités des options vanilles est relativement simple à partir des formules de Black & Scholes :

— **Delta :** dérivée du prix de l'option par rapport au prix du sous-jacent :

$$\Delta_{call} = e^{-qT} N(d_1)$$
$$\Delta_{put} = e^{-qT}(N(d_1) - 1)$$

— **Gamma :** dérivée du delta par rapport au prix du sous-jacent :

$$\Gamma_{call} = \Gamma_{put} = e^{-qT}$$

— **Vega :** dérivée du prix de l'option par rapport à la volatilité implicite :

$$\nu_{call} = \nu_{put} = S_0 e^{-qT} \mathcal{N}(d_1)\sqrt{T}$$

— **Theta** : dérivée du prix de l'option par rapport au temps :

$$\Theta_{call} = -\frac{S_0 \sigma e^{-qT} \mathcal{N}(d_1)}{2\sqrt{T}} + qS_0 e^{-qT} N(d_1) - rKe^{-rT} N(d_2)$$
$$\Theta_{put} = -\frac{S_0 \sigma e^{-qT} \mathcal{N}(d_1)}{2\sqrt{T}} - qS_0 e^{-qT} N(-d_1) + rKe^{-rT} N(-d_2)$$

— **Rho :** dérivée du prix de l'option par rapport au taux sans risque :

$$\rho_{call} = KTe^{-rT} N(d_2)$$
$$\rho_{put} = -KTe^{-rT} N(-d_2)$$

2.2 Options exotiques

2.2.1 Pricing d'options exotiques par la méthode de Monte Carlo

Pour la plupart des options exotiques, il n'existe pas de formules fermées comme pour les options vanilles. La méthode de Monte Carlo permet de fournir un prix à ces options. La méthode est la suivante :

1. Faire un grand nombre de simulations de trajectoires du sous-jacent. On suppose que le sous-jacent suit un mouvement brownien géométrique classique (en supposant un univers *risk-free*) :
$$S_t = S_0 \exp\left(\left(r - q - \frac{1}{2}\sigma^2\right) t + \sigma W_t\right)$$
avec W_t un processus de Wiener standard. En utilisant les propriétés du mouvements browniens, en particulier la normalité de ses incréments, il nous est possible de discrétiser l'équation du processus à des fins de simulation numérique :
$$S_{t+\Delta t} = S_t \times \exp\left(\left(r - q - \frac{1}{2}\sigma^2\right) \Delta t + \sigma\sqrt{\Delta t} Z\right)$$
avec $Z \sim N(0, 1)$
2. Pour chaque simulation, calculer le payoff de l'option.
3. Le prix de l'option est alors donné par la moyenne actualisée des payoffs.

2.2.2 Grecques d'options exotiques par la méthodes des différences finies

Si on utilise la méthode de Monte Carlo, il est impossible de calculer des grecques par formules fermées. En revanche, nous pouvons utiliser la méthode des différences finies pour y parvenir. Concrètement, nous utilisons la formule de Taylor pour aboutir aux formules suivantes :

$$f'(a) \approx \frac{f(a+h) - f(a)}{h}$$
$$f''(a) \approx \frac{f(a+h) + f(a-h) - 2f(a)}{h^2}$$

En remplaçant a par la variable (prix du sous-jacent, volatilité, taux sans risque, temps) choquée et h la valeur de ce choc, on obtient les grecques souhaitées.

2.2.3 Options exotiques implémentées : options digitales et options barrières

Les options digitales paient un montant fixe à l'échéance si l'option est dans la monnaie, sinon elles ne paient rien :

$$\text{Payoff}_{call} = M \text{ si } S_T \geq K \text{ sinon } 0$$

Les **options barrières** sont des options dont l'existence ou l'extinction dépend du franchissement d'un certain niveau prédéfini appelé *barrière*. Elles sont particulièrement utilisées pour structurer des produits plus accessibles ou pour répondre à des besoins spécifiques de couverture. Il existe quatre types principaux d'options barrières :

— **Up-and-In** : L'option n'existe que si le prix du sous-jacent atteint ou dépasse la barrière à la hausse pendant la vie de l'option.

— **Up-and-Out** : L'option devient nulle (extinction) si le prix du sous-jacent atteint ou dépasse la barrière à la hausse.

— **Down-and-In** : L'option n'existe que si le prix du sous-jacent atteint ou franchit la barrière à la baisse pendant la vie de l'option.

— **Down-and-Out** : L'option devient nulle si le prix du sous-jacent atteint ou franchit la barrière à la baisse.

Ainsi, les options barrières permettent d'ajuster finement l'exposition au risque en fonction du scénario de marché envisagé. Elles sont généralement moins chères que leurs équivalents vanille, car elles incorporent des conditions supplémentaires qui restreignent leur activation ou leur maintien.

3 Produits structurés

Dans cette partie, nous présentons la liste des produits structurés que nous avons implémentés, ainsi que la logique derrière leur construction, en plus d'explications sur leurs fonctionnements.

3.1 Architecture et classe de base

Au vu de la quantité et diversité de produits structurés, nous avons tenter d'en reproduire les plus importants. Pour cela, nous avons mis en place un dossier "Produits", dans lequel nous avons différents fichiers, dont l'un "ProductBase" contient la classe mère "Product", ainsi que "Decomposabe Product", qui hérite de cette dernière.
La classe abstraite Product a été mise en place pour harmoniser tous les produits et faire ressortir leur caractéristique commune, utilisée au moment du pricing : leur payoff.
Pour le payoff, nous avons mis en place un système qui puisse traiter des produits plus complexes, tout en étant également compatible avec des produits vanille. Donc, nous avons mis en place un système de simulation de chemins. L'argument paths attend une matrice numpy

de dimension `(n_simulations x n_steps)` représentant les trajectoires simulées du sous-jacent. Cette approche permet de traiter uniformément :

- Les produits vanille dont le payoff ne dépend que de la valeur finale
- Les produits path-dependent sensibles à l'évolution complète du sous-jacent (barrières, lookback, etc.)
- Les produits à exercice anticipé comme les autocalls

Le type de retour dual `Union[np.ndarray, Tuple[np.ndarray, np.ndarray]]` répond à deux scénarios :

- Pour les produits standards avec un unique paiement à maturité (comme les options européennes), la méthode retourne directement un tableau de payoffs
- Pour les produits complexes avec plusieurs flux à différentes dates (autocalls, produits à coupons, etc.), la méthode retourne un tuple contenant les payoffs et leurs dates de paiement respectives

Intégration de la dimension temporelle Le paramètre optionnel `time_grid` permet aux implémentations concrètes d'accéder à la discrétisation temporelle utilisée lors de la simulation. Cette information est cruciale pour :

- Identifier les indices correspondant aux dates d'observation
- Calculer correctement les facteurs d'actualisation
- Déterminer précisément les moments d'exercice potentiels

De plus, il est souvent possible de "décomposer" un produit structuré en fonction des différents produits qui le composent. D'où la création d'une classe qui permet de décomposer chaque produit. Cela permet notamment de valoriser le produit par la somme de ses parties, mais aussi fournit un cadre d'analyse unifié pour des structures parfois complexes.
Exemple : Une note à capital protégé est composée 1) Une obligation zéro-coupon assurant la protection du capital et 2) Une option sur le sous-jacent offrant le potentiel de rendement supplémentaire.
La suite de cette partie est structurée en suivant les différents types de produit, répartis dans les catégories suivantes : produits avec garantie en capital, produits de participation, produits de rendement et produits les plus complexes.

3.2 Produits à capital garanti

Les produits avec une garantie en capital offrent à l'investisseur l'assurance de récupérer un pourcentage minimum de son investissement initial à l'échéance, quelle que soit la performance du sous-jacent. Cette protection du capital est systématiquement assurée par une composante obligataire, généralement une obligation zéro-coupon.

Structure fondamentale Tous nos produits à capital garanti partagent un principe commun de structuration :

— Une partie du nominal est allouée à l'achat d'une obligation zéro-coupon pour assurer le remboursement minimum garanti

— Le capital restant est utilisé pour acquérir des instruments dérivés offrant une exposition au potentiel de hausse du sous-jacent

Cette architecture permet d'optimiser l'allocation entre sécurité et performance potentielle.

Types de produits à capital garanti

Capital Protected Note (CPN) La forme la plus "simple" de produit à capital garanti, composée de :

— Une obligation zéro-coupon garantissant le capital à l'échéance

— Une option d'achat (call) sur le sous-jacent, offrant une participation à la hausse

Le payoff à l'échéance est typiquement : $Nominal \times \max(Garantie, 1 + Participation \times Performance)$, où la garantie est généralement exprimée en pourcentage du nominal.

Capital Protected Note avec Barrière Variante de la CPN classique intégrant un mécanisme de barrière qui modifie le payoff conditionnel :

— Si le sous-jacent n'a jamais franchi la barrière à la baisse : protection complète et participation à la hausse

— Si la barrière a été touchée : protection minimale garantie avec performance potentiellement réduite

Cette structure permet généralement d'obtenir un taux de participation plus élevé au prix d'un risque conditionnel.

Autocall à capital garanti Produit combinant une protection du capital avec un mécanisme de remboursement anticipé automatique :

— À chaque date d'observation, si le sous-jacent est au-dessus du niveau d'autocall, le produit est remboursé avec un coupon prédéterminé

— En l'absence de remboursement anticipé, le capital est garanti à l'échéance

— Une composante obligataire assure la garantie, tandis que des options digitales financent les coupons conditionnels

TwinWin à capital garanti Structure permettant de bénéficier à la fois des hausses et des baisses modérées du sous-jacent, tout en protégeant le capital :

— Une obligation zéro-coupon pour la garantie

— Une option call standard pour capter la performance positive

— Une option put avec plafond pour transformer une baisse limitée en gain

Le TwinWin offre ainsi un profil de performance en “V” avec une protection du capital à l’échéance.

Produit à capital garanti avec coupons Combine la sécurité du capital avec des flux intermédiaires :

— Le capital est garanti à l’échéance par une obligation classique ou zéro-coupon

— Des options sont utilisées pour générer des coupons conditionnels ou garantis à des dates prédéterminées

— Les coupons peuvent être fixes, variables ou indexés à la performance du sous-jacent

Structure fondamentale Tous nos produits à capital garanti partagent un principe commun de structuration :

— Une partie du nominal est allouée à l’achat d’une obligation zéro-coupon pour assurer le remboursement minimum garanti

— Le capital restant est utilisé pour acquérir des instruments dérivés offrant une exposition au potentiel de hausse du sous-jacent

3.3 Produits d’amélioration du rendement

Les produits d’amélioration du rendement (Yield Enhancement) représentent une catégorie de produits structurés conçus pour offrir un rendement supérieur au marché, en contrepartie d’une prise de risque sur le capital investi. Contrairement aux produits à capital garanti, ces structures ne comportent pas de protection du nominal à l’échéance.

Caractéristiques fondamentales Ces produits partagent un principe économique commun :

— Génération d’un rendement supérieur aux taux sans risque

— Acceptation d’un risque de perte en capital à l’échéance

— Structure généralement équivalente à la vente d’options (stratégie optionnelle à prime)

Notre implémentation de ces produits repose sur la classe abstraite `YieldEnhancementProduct`, qui établit un cadre commun tout en permettant la spécialisation par type de produit.

Types de produits d’amélioration du rendement

Reverse Convertible Produit structuré offrant un coupon fixe élevé en échange d’une exposition au risque de baisse du sous-jacent :

— Coupon fixe garanti versé à l’échéance

— Remboursement du nominal si le sous-jacent termine au-dessus du niveau de strike

— Remboursement en fonction de la performance (avec perte potentielle) si le sous-jacent termine en-dessous du strike

En termes de décomposition, un Reverse Convertible équivaut à :

— Une obligation zéro-coupon pour le remboursement du nominal

— Une obligation pour le versement du coupon garanti

— Une position courte sur une option put (vente de protection)

Le payoff à l'échéance est : $Nominal + Coupon$ si le sous-jacent est au-dessus du strike, sinon $Nominal \times \frac{S_T}{S_0} + Coupon$, où S_T est le niveau final du sous-jacent et S_0 son niveau initial.

Discount Certificate Produit permettant d'acquérir une exposition au sous-jacent avec une décote (discount) en échange d'un plafonnement des gains potentiels :

— Prix d'achat inférieur au prix direct du sous-jacent (décote/discount)

— Participation à la hausse du sous-jacent jusqu'à un niveau plafond (cap)

— Exposition directe à la baisse du sous-jacent, sans protection

La décomposition d'un Discount Certificate révèle :

— Une exposition directe au sous-jacent (équivalent à une détention)

— Une position courte sur une option call avec strike au niveau du cap (vente du potentiel de hausse)

Le payoff à l'échéance suit trois cas : $Nominal \times Cap$ si le sous-jacent est au-dessus du cap, ou $Nominal \times \frac{S_T}{S_0}$ si le sous-jacent est en-dessous du cap, reflétant précisément la performance du sous-jacent avec la limitation à la hausse.

3.4 Produits de participation

Les produits de participation offrent à l'investisseur une exposition directe à la performance d'un actif sous-jacent, généralement avec des caractéristiques additionnelles qui les distinguent d'un investissement direct. Notre architecture modélise cette catégorie via la classe abstraite `ParticipationProduct`, qui intègre le concept fondamental du taux de participation.

Caractéristiques communes Les produits de participation partagent plusieurs attributs fondamentaux :

— Exposition directe à la performance de l'actif sous-jacent

— Taux de participation modulable pour ajuster la sensibilité au mouvement du sous-jacent

— Absence de revenus garantis (contrairement aux produits à coupon)

— Diverses structures possibles d'optimisation du profil rendement/risque

Types de produits de participation

Certificat Tracker Le produit le plus simple de cette famille, offrant une réplication quasi-parfaite de la performance du sous-jacent :

- Participation directe à la hausse comme à la baisse du sous-jacent
- Taux de participation permettant d'ajuster la sensibilité (effet de levier possible)
- Structure pouvant intégrer des frais de gestion qui réduisent progressivement la performance

Le payoff standard d'un Tracker Certificate est : $Nominal \times (1 + ParticipationRate \times Performance) \times (1 - ManagementFee)^T$, où T représente la durée de détention.
Cette structure simple équivaut économiquement à une détention directe du sous-jacent, éventuellement ajustée par le taux de participation et diminuée des frais de gestion cumulés. Le taux de participation est un multiplicateur qui s'applique à la performance du sous-jacent pour calculer le rendement final de l'investissement : il définit à quel point nous sommes exposés au sous-jacent.

Certificat Bonus Produit offrant une protection conditionnelle et un niveau de bonus garanti si une barrière prédéfinie n'est pas franchie :

- Protection conditionnelle tant que le sous-jacent reste au-dessus de la barrière
- Rendement minimum garanti (bonus) si la barrière n'est jamais touchée
- Exposition directe à la performance du sous-jacent si celle-ci dépasse le niveau de bonus

Le payoff d'un Bonus Certificate suit deux scénarios :

- Si la barrière n'est jamais touchée : $Nominal \times \max(1 + ParticipationRate \times Performance, BonusLevel)$
- Si la barrière est touchée : $Nominal \times (1 + ParticipationRate \times Performance)$

En termes de décomposition, ce produit combine une exposition directe au sous-jacent avec des options exotiques à barrière.

Certificat Outperformance Structure sophistiquée offrant une participation asymétrique à la performance du sous-jacent :

- Participation accrue à la hausse du sous-jacent au-delà d'un certain niveau (strike)
- Exposition potentiellement réduite à la baisse du sous-jacent
- Point pivot (strike) définissant le changement de régime de participation

Le payoff d'un Outperformance Certificate présente deux régimes :

- Si Performance ¿ StrikeReturn : $Nominal \times (1 + StrikeReturn + (Performance - StrikeReturn) \times UpsideParticipationRate)$
- Si Performance StrikeReturn : $Nominal \times (1 + Performance \times DownsideParticipationRate)$

Ce produit peut être décomposé en une exposition directe au sous-jacent combinée à des options call supplémentaires pour amplifier la participation à la hausse.

3.5 Produits complexes

Les produits structurés complexes représentent une classe d'instruments financiers dont le payoff dépend de conditions spécifiques liées au comportement d'un ou plusieurs sous-jacents. Nous avons choisi de mettre en place le code pour trois types de produits : l'Athéna, le Phoenix et la Range Accrual Note.

Athena Le produit Athéna est l'un des produits structurés les plus courants sur le marché. Son fonctionnement se caractérise par :

- Remboursement anticipé automatique aux dates d'observation si le sous-jacent est au-dessus d'un seuil prédéfini
- Versement d'un coupon conditionnel (potentiellement avec effet mémoire)
- Barrière de protection du capital à maturité
- Exposition à la baisse si le sous-jacent franchit la barrière à maturité

Le calcul du payoff d'un produit Athéna suit la logique suivante :

- Pour chaque date d'observation (sauf la dernière) : Si le niveau du sous-jacent est supérieur ou égal à la barrière d'autocall correspondante, le produit est remboursé par anticipation avec le nominal plus les coupons applicables.Si l'effet mémoire est activé, tous les coupons non versés précédemment sont également payés.
- À maturité (dernière date d'observation), pour les chemins non remboursés par anticipation : Si le niveau du sous-jacent est supérieur ou égal à la barrière de protection du capital, l'investisseur reçoit le nominal plus les coupons applicables. Si le niveau est inférieur à la barrière de protection, l'investisseur est exposé à la performance négative du sous-jacent (calculée comme Niveau final/Niveau initial).

Décomposition de l'Athéna

- Une obligation zéro-coupon pour le remboursement du nominal
- Des options digitales pour les coupons conditionnels
- Des options digitales supplémentaires pour l'effet mémoire (si activé)
- Des options digitales pour le mécanisme d'autocall
- Une option barrière down-and-in put pour modéliser l'exposition à la baisse si la barrière de protection est franchie à maturité

Phoenix Le produit Phoenix est une variante de l'Athéna qui se caractérise par :

- Remboursement anticipé automatique aux dates d'observation si le sous-jacent est au-dessus d'un seuil défini
- Coupons conditionnels payés périodiquement si le sous-jacent reste au-dessus d'une barrière de coupon (généralement plus basse que la barrière d'autocall)
- Effet mémoire pour les coupons non versés
- Barrière de protection du capital à maturité
- Exposition à la baisse si le sous-jacent franchit la barrière à maturité

La principale différence avec l'Athéna réside dans la séparation des mécanismes de paiement des coupons et de remboursement anticipé. Dans un Phoenix, les coupons peuvent être versés même si la condition d'autocall n'est pas remplie, ce qui peut offrir un rendement même en période de baisse modérée du marché. On remarque cette différence au niveau du payoff.

Range Accrual Note La Range Accrual Note est un produit structuré dont le paiement des coupons dépend du nombre de jours où le sous-jacent se trouve dans une plage spécifique. Elle a donc un paiement de taux d'intérêt prédéfini multiplié par le nombre de jours où le sous-jacent se trouve dans une plage spécifiée. Généralement, une garantie de capital à maturité est inclue. Enfin, pour vérifier le niveau du sous-jacent par rapport au niveau du benchmark, il y a plusieurs dates d'observation.

Décomposition La Range Accrual Note peut être décomposée en :

- Une obligation zéro-coupon pour le remboursement du capital protégé
- Une série d'options digitales pour chaque date d'observation : des calls pour la barrière inférieure et des puts pour la barrière supérieure.

Ces options digitales combinées ne paient que si le sous-jacent se trouve dans la plage définie, ce qui modélise parfaitement le comportement de la Range Accrual Note.

4 L'outil de pricing

La classe PricingEngine est un moteur de pricing pour produits financiers qui implémente trois méthodologies principales : Black-Scholes, Monte Carlo et décomposition en composantes élémentaires. Cette classe permet d'évaluer le prix et les greeks des produits financiers.

4.1 Architecture

La classe est structurée autour de trois approches de pricing complémentaires :

1. **Black-Scholes** : Méthode analytique pour les options vanilles

2. **Monte Carlo** : Méthode de simulation pour les produits plus complexes
3. **Décomposition** : Méthode hybride pour les produits structurés décomposables

Pour chaque méthode, deux fonctions principales sont implémentées :

— Une fonction de calcul du prix (`price_xxx`)

— Une fonction de calcul des greeks (`calculate_greeks_xxx`)

Ces fonctions spécialisées sont ensuite accessibles via deux méthodes génériques (`price()` et `calculate_greeks()`) qui redirigent vers la méthode appropriée selon le type de produit et la méthode demandée.

4.2 Paramètres d'Initialisation

Le moteur requiert plusieurs paramètres lors de l'initialisation :

— `spot_price` : Prix actuel du sous-jacent

— `domestic_rate` : Modèle de taux pour le taux domestique (cas où nous avons des taux étrangers)

— `volatility` : Modèle de volatilité

— `dividend` (optionnel) : Taux de dividende continu

— `foreign_rate` (optionnel) : Taux étranger pour options de change

— `num_paths` : Nombre de trajectoires pour Monte Carlo

— `num_steps` : Nombre de pas temporels pour la simulation

— `seed` (optionnel) : Graine pour la reproductibilité des simulations

Donc, afin d'obtenir les prix, nous avons besoin dans un premier lieu de générer la volatilité ainsi que les taux en fonction des modèles décrits plus haut dans cette documentation.

4.3 Méthodes principales

4.3.1 Black-Scholes

Implémente la formule analytique de Black-Scholes pour les options européennes standards. Calcule les paramètres d_1 et d_2, puis détermine le prix en fonction du type d'option (call/put). Les greeks sont calculés analytiquement.

4.3.2 Monte Carlo

Génère des trajectoires de prix du sous-jacent selon un modèle log-normal, puis évalue le payoff du produit sur ces trajectoires. Les greeks sont calculés par différences finies ("bumping").
Processus de simulation :

1. Génération d'une grille temporelle

2. Simulation des trajectoires du sous-jacent
3. Calcul des payoffs sur chaque trajectoire
4. Actualisation des payoffs

4.3.3 Décomposition

Décompose les produits structurés en composantes élémentaires (obligations et options), puis évalue séparément chaque composante avec la méthode appropriée. L'agrégation des résultats donne le prix total et les greeks du produit.

4.4 Points d'attention

— La méthode Black-Scholes n'est applicable qu'aux instances de la classe `Option`

— La méthode par décomposition n'est applicable qu'aux instances de la classe `DecomposableProduct`

— La méthode Monte Carlo est applicable à tous les produits mais est plus coûteuse en calcul

Cette architecture modulaire permet d'étendre facilement le moteur pour supporter de nouveaux produits financiers ou de nouvelles méthodes de pricing.