

Documentation Fonctionnelle - Projet Pricing de Produits Structurés en Python

CHARBONNIER - DUFOUR - HELLOT - LE MASNE DE CHERMONT

Dauphine 272 - Avril 2025

L'architecture du moteur de pricing a été pensée pour être modulaire et évolutive. Elle repose sur une séparation entre la configuration des paramètres de valorisation, l'instanciation du marché, la définition des modèles de diffusion, le choix du moteur de pricing, et enfin la structuration des résultats.

Le point d'entrée du système est la classe **PricingLauncher**. Elle prend en paramètre un objet **PricingSettings**, qui centralise tous les réglages nécessaires pour exécuter un pricing : nom du sous-jacent, type de courbe de taux et interpolation, type de surface de volatilité, fréquence d'observation, modèle de diffusion (par exemple Black-Scholes ou Heston), moteur de pricing utilisé (Monte Carlo générique, spécifique à des produits callables, etc.), nombre de chemins simulés, pas de temps, ainsi que des options telles que le calcul des grecques ou des coupons callables.

Le **PricingLauncher** initialise à partir de ces paramètres un objet **Market** qui regroupe les données de marché nécessaires à la valorisation. Il contient la courbe de taux, interpolée selon la méthode définie, les caractéristiques du sous-jacent, ainsi qu'une surface de volatilité implicite calibrée à partir des données d'options. Ces éléments sont accessibles, permettant aux moteurs de pricing de récupérer facilement les taux, les volatilités et les facteurs d'actualisation au moment du calcul. puis sélectionne dynamiquement l'engine adapté via l'énumération. Le launcher sélectionne dynamiquement l'engine adapté via l'enum **PricingEngineType**. Ce mécanisme permet de déléguer le calcul du prix et des sensibilités à des moteurs spécialisés. Chaque moteur implémente une méthode **get_results()** commune, qui retourne un objet structuré de type **PricingResults**.

Cette structure **PricingResults** contient toutes les informations utiles : le prix de l'instrument, son écart-type (avec les bornes de confiance associées), les grecques éventuellement calculées (Delta, Vega, Gamma, etc.), ainsi que la valeur de tout coupon callable le cas échéant. Une méthode statique permet d'agréger plusieurs résultats entre eux, utile notamment pour des calculs répartis ou des valorisations multi-sources.

Dans le cas d'un moteur Monte Carlo, l'engine utilise un processus stochastique (réutilisable dans les moteurs **Callable** et **American**) pour simuler des chemins d'évolution du sous-jacent. Ce processus est instancié dynamiquement selon le modèle sélectionné (par exemple Black-Scholes ou Heston) via la méthode **get_stochastic_process()**. Une fois construit, le processus est simulé à l'aide d'un schéma d'Euler, qui applique à chaque pas de temps la dynamique du modèle pour générer les trajectoires de prix.

Les processus sont organisés en deux grandes familles : **OneFactorStochasticProcess** pour les modèles à un facteur (comme Black-Scholes), où l'unique source de risque est l'évolution du prix du sous-jacent ; et **TwoFactorStochasticProcess** pour les modèles à deux facteurs (comme Heston), qui intègrent à la fois le prix du sous-jacent et la dynamique de sa variance. Chaque type de processus définit sa propre structure de dérive, de volatilité et de bruit aléatoire, avec une gestion des mouvements browniens (corrélés ou non). Cette abstraction permet au moteur de pricing de rester indépendant de la complexité du modèle sous-jacent.

Chaque moteur de pricing contient une méthode **get_price()** dans laquelle on simule des trajectoires à l'aide du processus stochastique et du schéma d'Euler décrits précédemment. Une fois les chemins générés, on appelle simplement la méthode **payoff()** du produit, qui prend un chemin en entrée et retourne le flux associé. Le prix de l'instrument correspond alors à la moyenne des payoffs obtenus, actualisée avec le discount factor approprié.

Le calcul des grecques repose également sur cette méthode `get_price()`, en utilisant une approche par différences finies. Pour le delta, le gamma ou le vega, on modifie directement les paramètres du processus stochastique de référence (par exemple le prix initial ou la volatilité), puis on recalcule le prix avec ces nouvelles valeurs. Pour le rho, la logique reste la même, mais le bump est appliqué directement aux fichiers d'entrée de la courbe de taux. On effectue un bump flat, c'est-à-dire une hausse ou une baisse uniforme de l'ensemble des points de la courbe, afin d'éviter un recalcul par bucket, plus coûteux en temps de calcul. Cela implique de recalibrer la courbe interpolée avant de relancer la simulation. Enfin, le theta est déduit à partir de l'équation aux dérivées partielles (EDP) du modèle : en combinant le prix obtenu et les sensibilités calculées, on peut reconstituer le terme de dérivée temporelle propre à chaque dynamique.

1 Description de l'interface

L'interface développée dans le cadre de ce projet de *pricing de produits structurés* repose sur une architecture logicielle en **trois couches**, garantissant une séparation claire entre la présentation, le traitement métier et la communication des données.

1.1 Frontend – Interface utilisateur

La couche **frontend** correspond à l'interface visible par l'utilisateur. Elle est développée en *HTML*, *CSS* et *JavaScript*, et permet une interaction intuitive à travers une interface web ergonomique. L'interface comprend plusieurs pages spécifiques dédiées au *pricing* de différents types de produits structurés. Chaque page propose des formulaires dynamiques permettant la saisie des paramètres de marché, des caractéristiques du produit, ainsi que le lancement des calculs.

1.2 Backend / API – Traitement des données

La couche centrale est constituée de l'**API**, développée en *Python* à l'aide du framework *Django*. Cette API joue un rôle fondamental dans la gestion et le traitement des données. Elle fait l'intermédiaire entre l'interface web (frontend) et la logique de calcul du pricing. Elle assure les responsabilités suivantes :

- Réception et traitement des requêtes issues du frontend (via des requêtes HTTP) ;
- Appel des fonctions de pricing et des modèles mathématiques internes ;
- Formatage et envoi des résultats vers le frontend, généralement en format *JSON*.

1.3 Lancement de l'interface

Le projet repose sur le framework *Django*, ce qui permet un démarrage simple et rapide de l'interface web. Pour exécuter l'application en local, il suffit d'exécuter la commande suivante dans un terminal à la racine du projet :

```
python manage.py runserver
```

Le serveur de développement se lance alors, et l'interface est accessible à l'adresse suivante :

`http://127.0.0.1:8000/`

L'utilisateur peut alors naviguer sur l'interface, saisir les paramètres nécessaires et lancer les calculs de pricing de manière interactive.

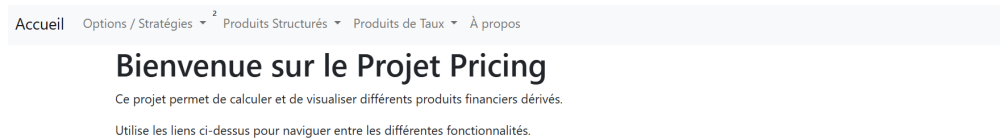


Figure 1: Page d'accueil de l'interface

1.4 Page de pricing des options

La page dédiée au **pricing des options** permet de sélectionner et de configurer différents types d'options, regroupés en plusieurs catégories :

- **Options vanilles** : Call et Put européens classiques ;
- **Options path-dépendantes** :
 - Options américaines ;
 - Options bermudéennes ;
 - Options asiatiques ;
 - Options lookback ;
 - Options à strike flottant.
- **Options binaires** ;
- **Options à barrière** : up-and-in, up-and-out, down-and-in, down-and-out.

En fonction du type d'option sélectionné, des **champs de saisie dynamiques** sont générés pour permettre à l'utilisateur de renseigner les paramètres pertinents (prix du sous-jacent, volatilité, taux, maturité, niveau de barrière, etc.).

L'utilisateur peut également choisir le type de volatilité avec laquelle pricer le produit :

- **SVI** ;
- **SSVI** ;
- **Volatilité Locale** ;

Pricer d'Options

Choisissez un type d'option et remplissez les champs nécessaires pour calculer le prix.

Type d'Option

| | | | |
|---------|----------------|---------|----------|
| Vanille | Path-Dependent | Binaire | Barrière |
|---------|----------------|---------|----------|

Sous-Type d'Option

Call

Ticker

SPX

Prix du Ticker

5768

Strike

5768

Maturité (Date)

23/04/2026

Nombre de Paths

10000

Type de Volatilité

SVI

Calculer le Prix

Figure 2: Interface des options

Une fois les paramètres saisis, l'utilisateur peut cliquer sur le bouton **Calculer le prix**. L'application effectue alors les traitements nécessaires côté backend et renvoie à l'utilisateur :

- le **prix de l'option** ;
- les **grecs** associés (Delta, Gamma, Vega, Theta, Rho) ;
- un **graphique du payoff** représentant le profil de gain/perte à maturité.

Résultats

Prix Calculé :

Price: 429.11 USD

Grecs de l'Option

| Grecs | Valeur |
|-------|-----------|
| Delta | 0.5844 |
| Gamma | -0.0000 |
| Vega | 2083.5946 |
| Theta | -74.2289 |
| Rho | 3407.7586 |

Graphique du Payoff

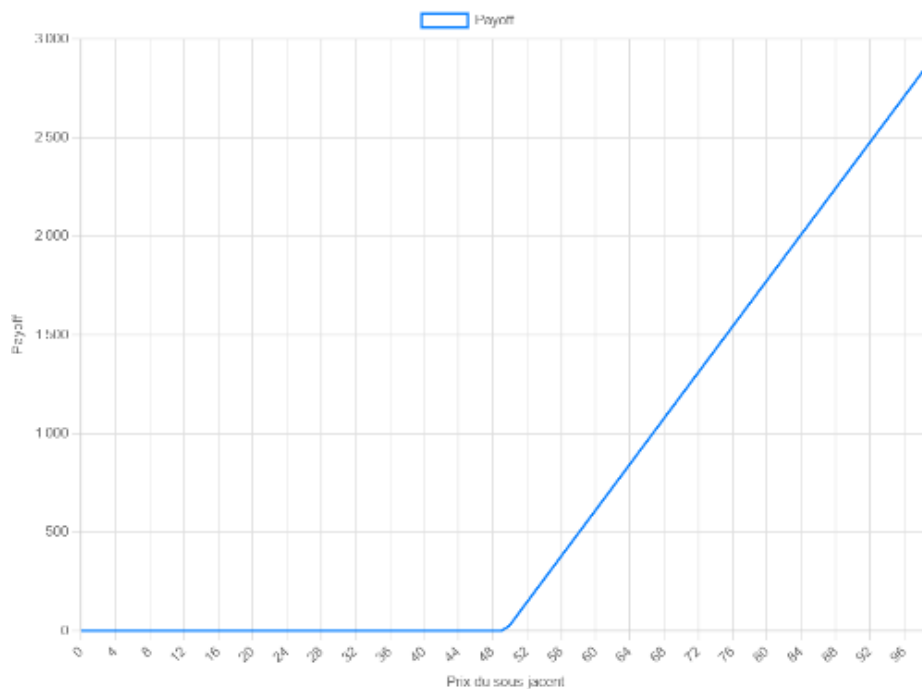


Figure 3: Exemple de pricing d'une option

1.5 Pricing de stratégies optionnelles

Une section dédiée permet également le pricing de **stratégies optionnelles complexes**, telles que :

- Call Spread, Put Spread ;
- Straddle, Strangle , Strip, Strap;
- Butterfly, Condor ;

L'utilisateur peut sélectionner plusieurs options, fixer leurs strikes, et visualiser le **payoff global** de la stratégie, ainsi que le prix total et les sensibilités agrégées.

Cette approche modulaire et visuelle facilite grandement l'analyse des positions optionnelles, que ce soit pour une couverture ou une prise de position directionnelle.

Pricer de Stratégies Optionnelles

Choisissez une stratégie d'options et remplissez les champs nécessaires pour calculer le prix.

Choisissez une Stratégie

Straddle

Strangle

Bull Spread

Bear Spread

Butterfly Spread

Condor Spread

Strip

Strap

Calendar Spread

Ticker

SPX

Prix du Ticker

5768

Strike 1

5500

Strike 2

5800

Maturité (Date)

23/04/2026

<

Type de Volatilité

SVI

Nombre de Paths MC

10000

Calculer le Prix

Figure 4: Interface des stratégies optionnelles

Résultats

Prix Calculé :

Price: 652.13 USD

Grecs de l'Option

| Grecs | Valeur |
|-------|-----------|
| Delta | 0.2601 |
| Gamma | -0.0000 |
| Vega | 4046.5492 |
| Theta | -21.4072 |
| Rho | 1516.3870 |

Graphique du Payoff

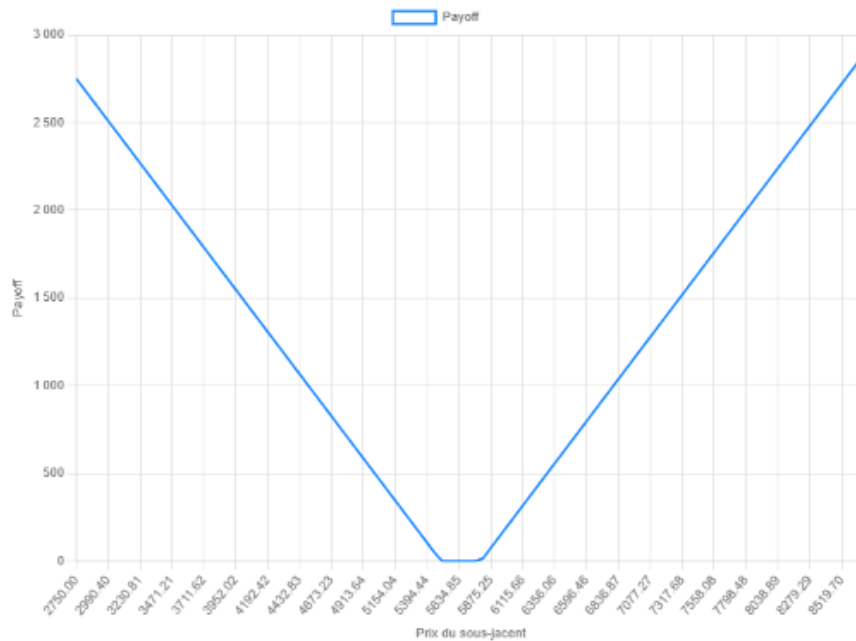


Figure 5: Exemple de résultats des stratégies optionnelles

2 Pricing des produits structurés : Autocalls

L'interface de pricing permet de calculer le prix de plusieurs types de **produits Autocall**, chacun ayant ses propres mécanismes d'observation et de versement de coupons. Les types de produits disponibles sont les suivants :

2.1 Autocall Phoenix

L' **Autocall Phoenix** est un produit structuré où l'investisseur peut recevoir des **coupons** à chaque date d'observation, sous réserve que le sous-jacent soit supérieur à une **barrière de coupon** définie. Si le sous-jacent est supérieur à la barrière à la date d'observation, un coupon est versé et le produit peut être rappelé à cette date, ce qui met fin à l'investissement avant la maturité du produit. En cas d'observation infructueuse (le sous-jacent est en-dessous de la barrière), l'investisseur attend la prochaine observation.

2.2 Autocall Eagle

L' **Autocall Eagle** est similaire à l'Autocall Phoenix, mais avec des différences importantes dans la gestion des coupons et des bonus. Dans la version **Eagle+**, les coupons manqués peuvent être rattrapés à la **maturité** du produit, sous réserve que le sous-jacent ait dépassé une certaine barrière à cette date. Ce produit peut également offrir un bonus supplémentaire si la condition est remplie à la dernière date d'observation.

2.3 Autocall avec mémoire

Les **Autocalls avec mémoire** permettent également de récupérer les **coupons manqués**, mais selon des règles spécifiques à chaque produit. Par exemple :

- Un **Phoenix+** permet de récupérer les coupons manqués dès que le sous-jacent est au-dessus de la barrière à une date d'observation ;
- Un **Eagle+** permet de récupérer les coupons uniquement à la **maturité** si le sous-jacent a dépassé la barrière.

Ce mécanisme permet à l'investisseur de maximiser ses gains, même si certaines conditions n'ont pas été remplies lors d'observations précédentes.

2.4 Autocall Security

L' **Autocall Security** est une version plus défensive d'un produit structuré, dans laquelle la protection du capital est renforcée. Contrairement aux versions classiques, les pertes ne sont comptabilisées qu'à partir du franchissement de la **barrière en capital**. Cette version est particulièrement adaptée aux investisseurs cherchant à minimiser leur exposition aux risques de marché tout en restant exposés à un produit structuré.

Exemple d'Autocall Security Dans un scénario défavorable où le sous-jacent clôture en dessous de la barrière en capital, les pertes seront limitées et ne commenceront à se matérialiser qu'à partir de ce seuil. Le mécanisme utilisé est celui d'un put leveragé contrement au put down and in classique des autocall.

2.5 Interface de Pricing

L'interface permet à l'utilisateur de choisir parmi ces différents types d'Autocalls et de renseigner les paramètres nécessaires pour obtenir un prix. Pour chaque type d'Autocall, l'utilisateur peut effectuer deux actions :

- **Pricer le produit** : Calculer le prix du produit en fonction d'un coupon donné, en renseignant les paramètres de l'Autocall tels que la **barrière de coupon**, la **fréquence des observations**, la **maturité**, et le **coupon**.
- **Solver le coupon** : Calculer le **coupon** nécessaire pour obtenir un prix de 100, c'est-à-dire déterminer le coupon qui permet d'équilibrer le prix du produit à une valeur prédéfinie (ici 100).

Les paramètres disponibles pour chaque produit sont les suivants :

- **Type de produit** : l'utilisateur peut choisir entre un Phoenix et Eagle, avec mémoire ou Security ;
- **Barrière de capital** : la barrière à franchir pour que le capital soit protégé à maturité ;
- **Fréquence des observations** : la fréquence à laquelle les conditions sont vérifiées ;
- **Maturité** : la durée du produit ;
- **Coupon** : le montant du paiement périodique (utilisé dans le mode **Pricer**, mais calculé dans le mode **Solver**).

Pricer d'Autocalls

Mode de calcul

☒ Solver de Coupon ☐ Pricing d'Autocall

Choisissez le produit autocallable

Phoenix
Eagle

Ticker

AAPL

Prix du Ticker

110

Barrière de Capital (%)

60

Barrière de Rappel (%)

100

Barrière de Coupon (%)

80

Maturité (Date)

23/04/2028

Fréquences d'observations

Annuelles

Type de Volatilité

SVI

Nombre de Paths MC

10000

Calculer le Coupon

Figure 6: Solver de coupon

Pricer d'Autocalls

Mode de calcul

☐ Solver de Coupon ☒ Pricing d'Autocall

Choisissez le produit autocallable

Phoenix
Eagle

Ticker

AAPL

Prix du Ticker

110

Barrière de Capital (%)

60

Barrière de Rappel (%)

100

Barrière de Coupon (%)

80

Coupon (%)

5

Maturité (Date)

23/04/2028

Fréquences d'observations

Annuelles

Type de Volatilité

SVI

Nombre de Paths MC

10000

Calculer le Prix

Figure 7: Pricing d'autocall

Une fois les informations saisies, l'utilisateur peut obtenir :

- Le **prix du produit**, calculé en fonction des paramètres définis (en mode **Pricer**) ;
- Le **coupon nécessaire** pour obtenir un prix de 100 (en mode **Solver**) ;
- Les **grecs**, qui permettent d'évaluer la sensibilité du prix du produit aux variations des paramètres de marché.

Résultats
Coupon : 5.25 %
Greecs de l'Option

| Greecs | Valeur |
|--------|----------|
| Delta | 0.0000 |
| Gamma | 0.0000 |
| Vega | -89.3408 |
| Theta | 2.5235 |
| Rho | 7.2106 |

Figure 8: Exemple de solver de coupon

Résultats
Prix : 106.08 €
Greecs de l'Option

| Greecs | Valeur |
|--------|----------|
| Delta | 0.0000 |
| Gamma | 0.0000 |
| Vega | -96.6975 |
| Theta | 2.6769 |
| Rho | 7.2106 |

Figure 9: Exemple de résultat de pricing

3 Pricing des produits structurés de Participation, Airbag et Twinwin

L'interface de pricing permet également de calculer le prix de plusieurs produits structurés de participation, notamment les **Airbag** et **Twinwin**. Ces produits sont conçus pour offrir une exposition au sous-jacent avec des profils de participation spécifiques, tout en présentant différents niveaux de protection du capital.

3.1 Airbag

Les **Airbag** sont des produits de participation à la hausse, permettant à l'investisseur de bénéficier d'un gain lié à l'évolution positive du sous-jacent. L'investisseur participe à la hausse du sous-jacent de manière plafonnée, c'est-à-dire qu'il reçoit 100% de la hausse du sous-jacent entre le niveau d'émission et une barrière haute. Cependant, ce produit comporte un risque en capital, car la protection du capital est limitée à une certaine barrière basse.

Le mécanisme de l'Airbag peut être résumé comme suit :

- Si le cours du sous-jacent termine entre le niveau d'émission et la barrière de capital, le capital investi est garanti et remboursé à 100%.
- Si le cours termine en dessous de la barrière de capital, l'investisseur subit une perte en capital proportionnelle à la baisse du sous-jacent.

Le **payoff de l'Airbag** est défini par la formule :

$$\text{payoff}_{\text{airbag}} = \text{nominal} \times \frac{S_T}{S_0}$$

où S_T est le niveau final du sous-jacent et S_0 est le niveau d'émission.

Une fois les paramètres saisis, l'interface permet de visualiser un **graphique du gain du produit** en fonction du niveau du sous-jacent, ainsi que les niveaux de barrière associés à ce produit.

3.2 Twinwin

Les produits **Twinwin** sont des produits de participation absolue, conçus pour anticiper une forte variation du sous-jacent, qu'elle soit à la hausse ou à la baisse. L'investisseur bénéficie d'une participation positive même dans le cas où le sous-jacent termine en dessous du niveau d'émission.

Le produit Twinwin présente plusieurs caractéristiques clés :

- À la hausse, la participation est bornée.
- À la baisse, le capital est protégé jusqu'à la barrière de capital. Cependant, en dessous de cette barrière, l'investisseur subit une perte proportionnelle à la baisse du sous-jacent.

Le **payoff du Twinwin** est défini par la formule :

$$\text{payoff}_{\text{twinwin}} = \text{nominal} \times \left\lceil \frac{S_T}{S_0} \right\rceil$$

où S_T est le niveau final du sous-jacent et S_0 est le niveau d'émission.

Le produit Twinwin peut être vu comme une stratégie optionnelle type **straddle**, car il profite des variations du sous-jacent dans les deux directions (haussière et baissière). Le produit Twinwin peut être proposé sous différentes formes, telles que :

- **Versión capital garanti** : L'investisseur bénéficie d'une protection complète du capital, avec participation sur les hausses et une perte limitée au capital garanti.
- **Versión avec coupon** : Un coupon est versé si le sous-jacent sort de l'intervalle de participation (par exemple, si le sous-jacent clôture en dehors de la fourchette de participation définie).

Le **payoff avec coupon** du Twinwin est :

$$\text{payoff}_{\text{twinwin avec coupon}} = \text{nominal} \times \frac{S_T}{S_0}$$

3.3 Interface de Pricing

Dans l'interface, l'utilisateur peut choisir parmi ces produits de participation (Airbag et Twinwin) et renseigner les paramètres nécessaires pour obtenir un prix. Les options disponibles pour chaque produit incluent :

- **Type de produit** : l'utilisateur peut choisir entre Airbag ou Twinwin ;
- **Barrière de capital** : la barrière à partir de laquelle le capital est protégé ;
- **Maturité** : la durée du produit ;
- **Rebate** : le potentiel rebate du produit en cas de franchissement de la barrière haute;
- **Leveraeg** : le levier de participation aux variations du sous jacent. Le levier est exprimé en

Pricer de Produits structurés de participation

Choisissez un produit

Airbag

Twin Win

Ticker Prix du Ticker

Barrière Basse (%) Barrière Haute (%)

Leverage (%) Rebate (%)

Maturité (Date)

Type de Volatilité

Nombre de Paths MC

Calculer le Prix

Figure 10: Interface de produits de participation

Une fois les informations saisies, l'utilisateur peut obtenir :

- Le **prix du produit**, calculé en fonction des paramètres définis ;
- Le **graphique du payoff**, qui montre l'évolution du remboursement du produit en fonction des variations du sous-jacent ;
- Les **grecs**, qui permettent d'évaluer la sensibilité du prix du produit aux variations des paramètres de marché.

Résultats

Prix Calculé :

Price: 91.85 USD

Grecs de l'Option

| Grecs | Valeur |
|-------|-----------|
| Delta | 0.0000 |
| Gamma | 0.0000 |
| Vega | -166.1098 |
| Theta | 2.3176 |
| Rho | 46.2410 |

Graphique du Payoff

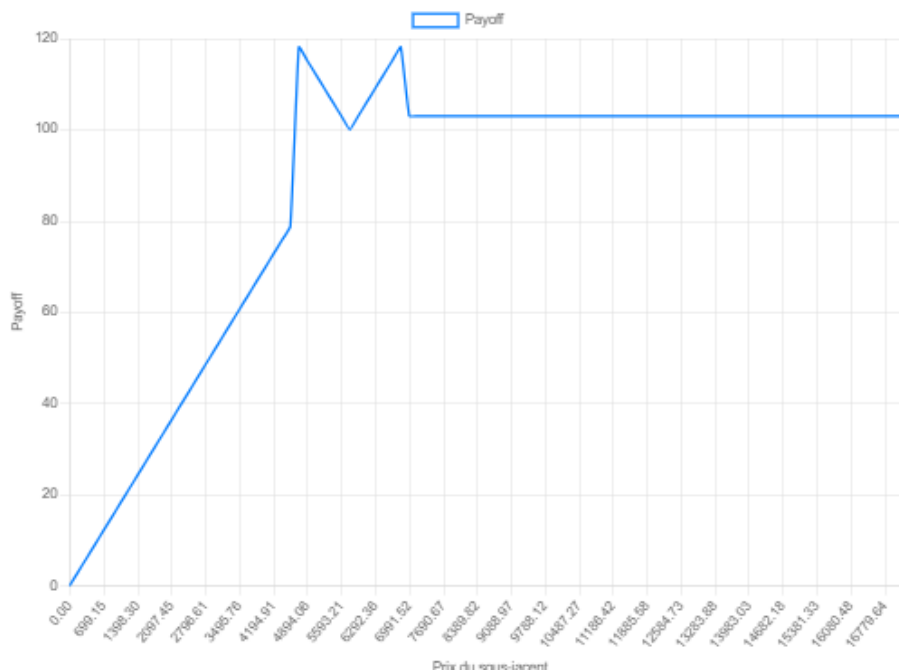


Figure 11: Résultats de pricing de twin win

4 Pricing des Obligations

L'interface propose également une section dédiée au **pricing des obligations**, permettant à l'utilisateur de calculer :

- soit le **prix de l'obligation** à partir d'un *yield to maturity* (YTM) donné ;
- soit le **YTM** correspondant à un *prix* d'obligation observé sur le marché.

4.1 Fonctionnalités disponibles

L'utilisateur peut configurer les paramètres suivants :

- **Type de calcul** : choix entre *Calcul du prix* ou *Calcul du YTM* ;
- **Valeur nominale** de l'obligation ;
- **Coupon** : taux d'intérêt nominal de l'obligation ;
- **Fréquence de coupon** : annuelle, semestrielle, trimestrielle, etc. ;
- **Date de règlement** (ou de valorisation) ;
- **Date d'échéance** de l'obligation ;
- **Convention calendaire** : Act/Act, 30/360, Actual/360, Actual/365, etc. ;

4.2 Interface utilisateur

L'interface permet une saisie simple et dynamique des différents paramètres via des menus déroulants et des champs de date interactifs. Une fois les paramètres saisis, l'utilisateur peut lancer le calcul et obtenir instantanément le résultat.

Pricer d'obligations
Mode de calcul

☒ Solver de Coupon ☐ Pricing d'obligations

Notional (€) : 100 Prix (€) : 90

Emission (Date) : 23/04/2025 Maturité (Date) : 23/04/2028 Achat (Date) : 23/04/2025

Taux de coupon (%) : 5

Fréquences de coupons : Annuelles Convention Calendaire : Actual/360

Calculer le YTM

Résultats
YTM : 8.9377 %

Figure 12: Exemple de résultat de pricing de YTM obligataire

Pricer d'obligations

Mode de calcul

☐ Solver de Coupon ☒ Pricing d'obligations

| | | |
|---|---|---|
| Notionnel (€) | YTM (%) | |
| <input type="text" value="100"/> | <input type="text" value="7"/> | |
| Emission (Date) | Maturité (Date) | Achat (Date) |
| <input type="text" value="23/04/2025"/> | <input type="text" value="23/04/2028"/> | <input type="text" value="23/04/2025"/> |
| Taux de coupon (%) | | |
| <input type="text" value="5"/> | | |
| Fréquences de coupons | Convention Calendaire | |
| <input type="text" value="Annuelles"/> | <input type="text" value="Actual/360"/> | |

Calculer le Prix

Résultats
Prix : 94.73 €

Figure 13: Exemple de résultat de pricing obligataire

5 Pricing de Swaps

L'application inclut également une interface dédiée au **pricing de swaps de taux d'intérêt**, permettant deux types d'opérations :

- **Détermination du taux fixe** (*fixed rate*) tel que la valeur du swap soit initialement nulle (i.e. prix = 0) à la date d'initiation ;
- **Valorisation du swap** en fonction d'un *taux fixe donné* (par exemple, un taux issu d'un contrat existant sur le marché).

L'utilisateur peut configurer les paramètres suivants : les dates d'échéance, la fréquence des flux, les conventions calendaires.

Cette interface permet d'effectuer des simulations rapides et d'obtenir, selon les cas, soit le taux fixe équivalent, soit la valeur actuelle nette du swap.

Pricer de Swaps

Mode de calcul

☒ Calculer le taux fixe pour un prix nul ☐ Pricing de Swap

| | | |
|---|---|---|
| Notionnel (€) | Prix (€) | |
| <input type="text" value="100"/> | <input type="text" value="0"/> | |
| Emission (Date) | Maturité (Date) | Achat (Date) |
| <input type="text" value="24/04/2025"/> | <input type="text" value="24/04/2028"/> | <input type="text" value="24/04/2026"/> |
| Fréquences de coupons | Convention Calendaire | |
| <input type="text" value="Annuelles"/> | <input type="text" value="Actual/360"/> | |

Calculer le taux fixe

Résultats
Taux Fixe : 0.67 %

Figure 14: Interface de pricing de swap