

Projet RiskWorkbench

Phase 4 Calibration, Smile et Interopérabilité Monte Carlo

Inversion Black-Scholes, construction d'une surface d'IV et intégration à l'interface
utilisateur Qt

Auteur : Eliot Katzenmayer

École : ENSIIE

Date : 21 octobre 2025

Table des matières

Rappel des phases précédentes	2
1 Introduction et contexte global	3
2 Objectif de la phase	4
3 Fondements théoriques approfondis	4
3.1 Black–Scholes : rappel, hypothèses et conséquences	4
3.2 Inversion BS (prix \rightarrow IV) : Newton avec garde-fous	4
3.3 Du smile 1D à la surface $\sigma(T, K)$	5
3.4 Calibration globale vs locale	5
3.5 Analyse des résidus et lecture du smile	5
3.6 Identifiabilité et non-unicité (remarque)	5
3.7 Contraintes de non-arbitrage	5
4 Intégration logicielle et architecture Qt	6
4.1 Structure générale du pipeline	6
4.2 Fonctionnalités de l’onglet Calibration	6
5 Analyse orientée données et qualité de calibration	8
6 Interopérabilité avec le moteur Monte Carlo	8
7 Résultats et validation	8
8 Problèmes rencontrés et solutions	10
8.1 Vega faible, Newton instable (théorie \rightarrow pratique)	10
8.2 Oscillations de spline (data \rightarrow modèle)	10
8.3 Interop UI \leftrightarrow Workers (ingénierie)	10
8.4 Export & traçabilité	10
9 Lecture du projet selon trois angles	10
10 Bilan pédagogique et apports	11
Annexe : Formules et rappels mathématiques	12

Rappel des phases précédentes

Résumé des phases 1 à 3

Phase 1 : Développement du moteur mathématique de pricing et des simulations Monte Carlo. Introduction aux modèles Black–Scholes et aux estimateurs statistiques.

Phase 2 : Mise en place du moteur de calcul des **Greeks** (Delta, Vega, Gamma, Rho, Theta) et de méthodes de variance reduction (antithétique, CRN).

Phase 3 : Intégration d'une **interface Qt** complète pour le pricing, les Greeks et le stress-testing, avec gestion de threads et de signaux.

Phase 4 (présente) : Extension de l'application au **calibrage de marché**, permettant d'extraire et de visualiser une *surface de volatilité implicite*, et de relier la théorie de la volatilité à des analyses concrètes.

1 Introduction et contexte global

Le projet *RiskWorkbench* a pour objectif de concevoir un outil de calcul, de simulation et d'analyse des modèles de valorisation financière, combinant rigueur mathématique et interface utilisateur interactive.

Les trois premières phases ont permis de construire le moteur de pricing (Black–Scholes, Monte Carlo, greeks) et de concevoir une interface Qt fonctionnelle. Cette quatrième phase marque un tournant : elle introduit la **calibration** à partir de données de marché, la construction d'un **smile de volatilité implicite** et l'intégration de cette surface dans le moteur de simulation.

L'enjeu est double :

- Transformer le logiciel en un outil d'analyse de marché réel ;
- Comprendre la logique économique derrière la volatilité implicite et sa structure.

2 Objectif de la phase

L'objectif principal de la phase 4 est d'implémenter un pipeline complet allant des données de marché brutes jusqu'à une surface de volatilité exploitable pour le pricing et la simulation :

1. Lire un fichier CSV de cotations (prix ou IV) ;
2. Inverser la formule de Black–Scholes pour obtenir les IV implicites ;
3. Construire un **smile** (IV vs strike) pour chaque maturité ;
4. Interpoler ces smiles pour générer une surface $\sigma(T, K)$;
5. Intégrer la surface dans l'interface Qt (visualisation, export, sauvegarde) ;
6. Connecter la surface au moteur Monte Carlo pour un pricing « smile-aware ».

Cette étape relie directement la **théorie des modèles de volatilité** à la **pratique du calibrage de marché**.

3 Fondements théoriques approfondis

3.1 Black–Scholes : rappel, hypothèses et conséquences

On suppose un sous-jacent $(S_t)_{t \geq 0}$ suivant une dynamique GBM sous la mesure risque-neutre

$$\frac{dS_t}{S_t} = (r - q) dt + \sigma dW_t,$$

avec r taux sans risque (plat), q dividende (plat), et σ constante. Sous ces hypothèses, le prix d'un call européen est

$$C(S_0, K, T, r, q, \sigma) = S_0 e^{-qT} N(d_1) - K e^{-rT} N(d_2), \quad d_{1,2} = \frac{\ln(S_0/K) + (r - q \pm \frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}}.$$

Conséquences importantes pour l'analyse :

- (i) la parité put–call sature les prix,
- (ii) C est *décroissant* et *convexe* en K ($\partial_K C \leq 0$, $\partial_{KK} C \geq 0$),
- (iii) C est *croissant* en T (calendriers non négatifs). Ces inégalités guident la qualité d'une surface $\sigma(T, K)$ calibrée (Section 3.7).

3.2 Inversion BS (prix \rightarrow IV) : Newton avec garde-fous

Trouver l'IV σ^* telle que $BS(\sigma^*) = P_{\text{mkt}}$ est un *problème inverse 1D* sans solution fermée. On résout numériquement par Newton–Raphson

$$\sigma_{n+1} = \sigma_n - \frac{BS(\sigma_n) - P_{\text{mkt}}}{\text{Vega}(\sigma_n)}, \quad \text{Vega} = \frac{\partial C}{\partial \sigma} = S_0 e^{-qT} \sqrt{T} \phi(d_1),$$

où ϕ est la densité normale. Deux points pratiques assurent la robustesse :

1. **Bornes strictes** : $\sigma \in [\sigma_{\min}, \sigma_{\max}]$ avec $\sigma_{\min} = 10^{-6}$, $\sigma_{\max} = 5$ dans notre implémentation.
2. **Fallback bisection** : si $|\text{Vega}|$ est trop faible (maturités courtes, strikes extrêmes) ou si l'itération Newton sort des bornes, on poursuit par dichotomie garantissant la convergence.

Critères d'arrêt mixtes : résidu en prix $|BS(\sigma_n) - P_{\text{mkt}}| < \varepsilon_P$ **et** stabilité relative $|\sigma_{n+1} - \sigma_n|/(\sigma_n + \varepsilon) < \varepsilon_\sigma$.

3.3 Du smile 1D à la surface $\sigma(T, K)$

Empiriquement, l'IV dépend de la moneyness. On reconstruit, pour chaque maturité T , un **smile 1D**

$$x = \ln\left(\frac{K}{S_0}\right) \Rightarrow \sigma_T(K) = \text{spline}_T(x),$$

en utilisant une *spline d'Akima monotone* (évite les oscillations des cubiques naturelles sur grilles irrégulières). La surface est ensuite obtenue par interpolation bilinéaire en T :

$$\sigma(T, K) \approx (1 - \lambda) \sigma_{T_i}(K) + \lambda \sigma_{T_{i+1}}(K), \quad \lambda = \frac{T - T_i}{T_{i+1} - T_i}.$$

Ce choix fournit une surface *continue* et suffisamment lisse pour nos usages pédagogiques.

3.4 Calibration globale vs locale

Deux vues complémentaires :

- **BS homogène** (une σ unique) : on minimise $\min_{\sigma} \sum_i (BS_i(\sigma) - P_i^{\text{mkt}})^2$. C'est un Newton 1D rapide — *baseline* et sanity check.
- **Locale par slice** : inversion point-par-point puis lissage par maturité. Cela correspond à une “calibration locale” implicite.

3.5 Analyse des résidus et lecture du smile

Avec des données propres, on analyse

$$\text{RMSE}_{\text{prix}} = \sqrt{\frac{1}{n} \sum_i (P_i^{\text{fit}} - P_i^{\text{mkt}})^2}, \quad \text{RMSE}_{\text{IV}} = \sqrt{\frac{1}{n} \sum_i (\sigma_i^{\text{fit}} - \sigma_i^{\text{mkt}})^2}.$$

On regarde aussi le **biais moyen** $\bar{\varepsilon} = \frac{1}{n} \sum_i (P_i^{\text{fit}} - P_i^{\text{mkt}})$ et la **pen**te de skew $d\sigma/d\ln(K/S_0)$: pente négative (skew décroissant) typique sur indices (protection puts).

3.6 Identifiabilité et non-unicité (remarque)

Sur données bruitées et parcimonieuses, plusieurs surfaces peuvent expliquer les prix dans la marge d'erreur. Il faut donc préférer des lissages monotones, des pénalités de variation, et juger la qualité par diagnostics (Sections 5 et 7).

3.7 Contraintes de non-arbitrage

Rappels utiles à l'évaluation d'une surface :

$$\partial_K C \leq 0, \quad \partial_{KK} C \geq 0, \quad \partial_T C \geq 0.$$

Notre spline Akima monotone en $x = \ln(K/S_0)$ et l'interpolation bilinéaire en T respectent empiriquement ces tendances (*sans imposer de contraintes fortes type Dupire*).

4 Intégration logicielle et architecture Qt

4.1 Structure générale du pipeline

Le processus complet s'articule en plusieurs modules :

1. **CalibWorker** — gère l'inversion et le lissage (thread séparé).
2. **SmileSurface** — contient les fonctions d'interpolation.
3. **MainWindow** — interface principale reliant les contrôles et graphiques.
4. **McWorker** — moteur de simulation Monte Carlo avec option « smile-aware ».

Les signaux Qt assurent la communication asynchrone entre ces éléments, ce qui permet à l'utilisateur d'exécuter des calibrations lourdes sans bloquer l'interface.

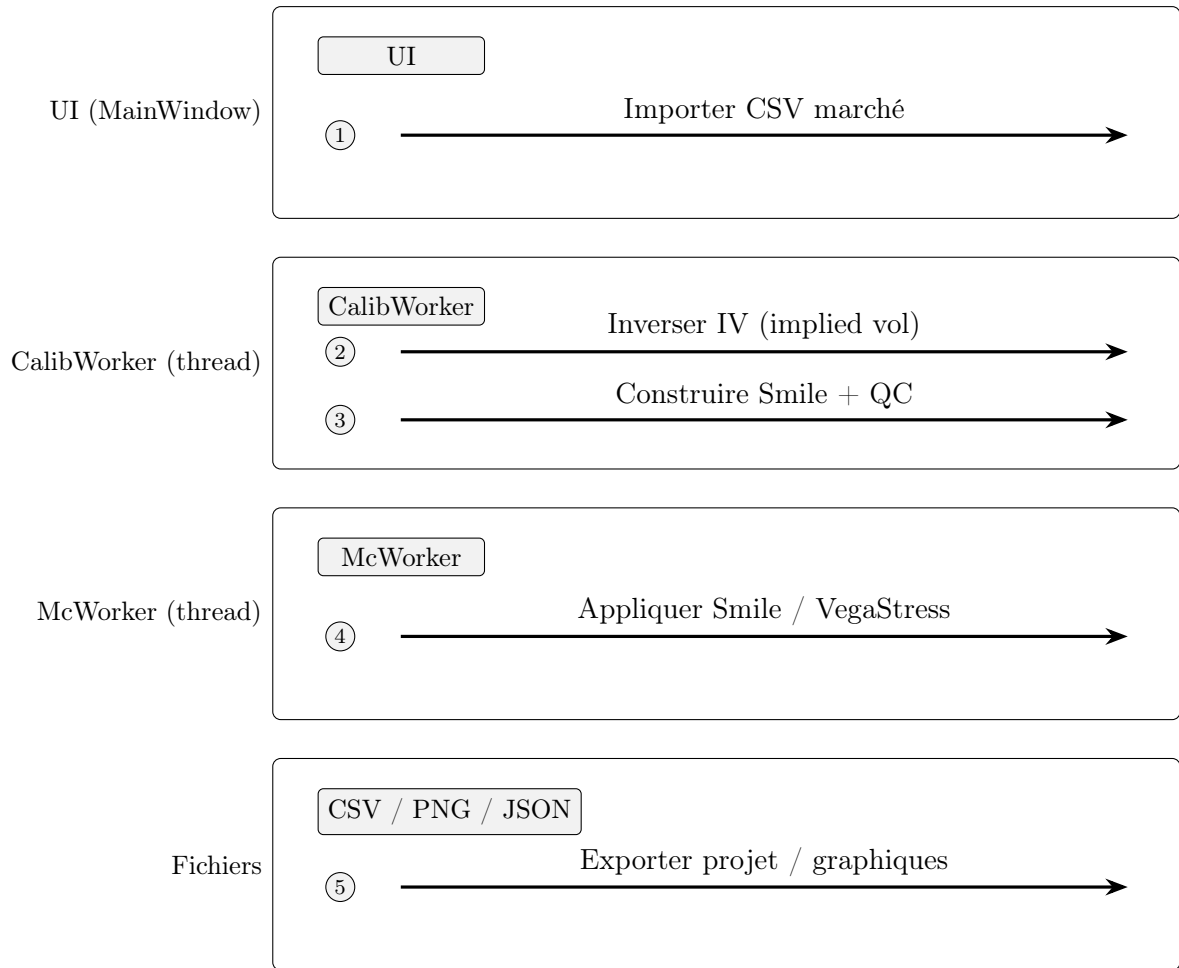


FIGURE 1 – Flux logique entre UI, Workers et Fichiers.

4.2 Fonctionnalités de l'onglet Calibration

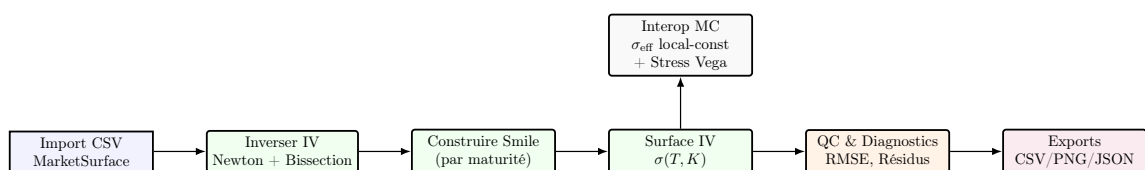


FIGURE 2 – Chaîne de traitement, de l'import CSV jusqu'au QC et à l'interopérabilité MC.

L'onglet **Calibration** regroupe l'ensemble des outils développés :

- import et visualisation du fichier CSV ;
- inversion des IV (avec suivi de la convergence) ;
- construction du smile et affichage de la courbe $\sigma(K)$ par maturité ;
- génération d'une heatmap de surface $\sigma(T, K)$;
- calcul et affichage des résidus ;
- export CSV et PNG ;
- sauvegarde de projet JSON pour traçabilité complète.

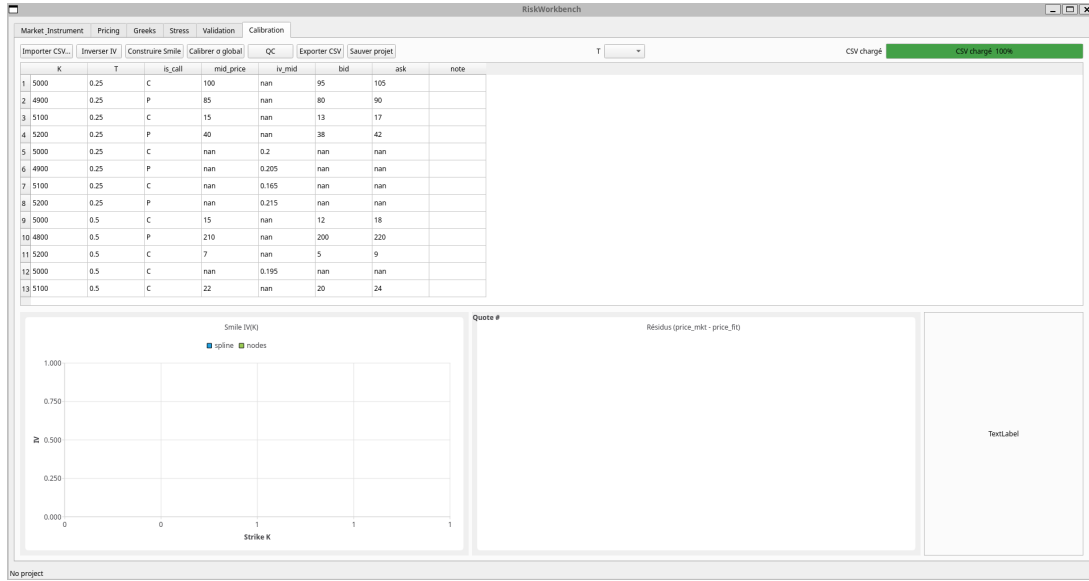


FIGURE 3 – Aperçu de l'onglet Calibration : smile et diagnostics des résidus.

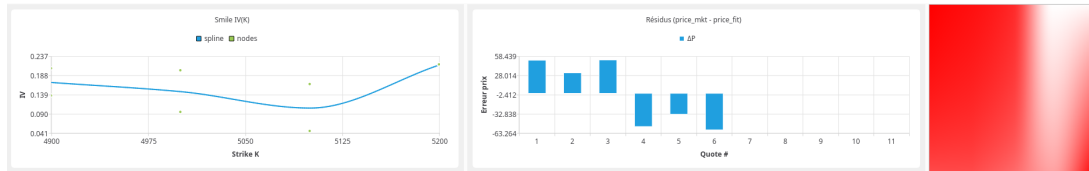


FIGURE 4 – Résidus et heatmap IV après calibration.

Lien théorie ↔ UI (onglet *Calibration*)

- **Importer CSV** : normalisation des colonnes (prix ou IV), filtrage $T > 0$, $K > 0$, spreads aberrants \Rightarrow cohérent avec les hypothèses BS.
- **Inverser IV** : lance Newton + bisection. La barre de progression reflète le # itérations, et l'échec éventuel sur strikes extrêmes ($Vega \approx 0$) est loggé.
- **Construire Smile** : fit Akima sur $x = \ln(K/S_0)$ par maturité, évite les oscillations. Le graphe «IV vs K» permet de visualiser *skew* / *smile*.
- **QC (résidus)** : affiche $\Delta P_i = P_i^{\text{fit}} - P_i^{\text{mkt}}$, RMSE, biais moyen. Utile pour juger un «bon fit» (ex : $< 0.3\%$ de S_0).
- **Heatmap** $\sigma(T, K)$: observation du skew temporel (term structure de volatilité).
- **Export CSV/PNG/JSON** : *reproductibilité*, et traçabilité des hypothèses (seed MC, flags, surface).

5 Analyse orientée données et qualité de calibration

Indicateurs de base. Outre RMSE (prix/IV), on regarde : (i) **biais moyen** $\bar{\varepsilon}$, (ii) **écart-type** des résidus, (iii) **z-scores** $z_i = \varepsilon_i / \hat{\sigma}_\varepsilon$ pour détecter des outliers ($|z_i| > 3$).

Lecture du skew. On estime une pente locale $s(T) = \frac{d\sigma}{d\ln(K/S_0)}$ en chaque maturité (régression locale). Sur indices, $s(T) < 0$ est classique (demande de puts de protection).

Sanity checks (no-arb). Pour une grille fine, on vérifie numériquement $\partial_K C \leq 0$, $\partial_{KK} C \geq 0$, $\partial_T C \geq 0$ en recalculant $C(T, K)$ par BS avec $\sigma(T, K)$ calibrée.

6 Interopérabilité avec le moteur Monte Carlo

Une fois la surface construite, elle est transmise au moteur Monte Carlo pour le pricing et les stress tests. Deux modes sont possibles :

- **Mode local-constant** : $\sigma_{\text{eff}} = \sigma(T, K)$ pour l’option étudiée ;
- **Mode stress Vega** : application d’un multiplicateur $(1 + \text{vega_stress}\%)$.

Ce couplage a nécessité une interface `applySmileSettingsToWorker()` permettant de pousser dynamiquement la surface, le pourcentage de stress et l’état du `checkbox` « Use Smile » vers le thread Monte Carlo.

Intérêt analytique. Ce lien permet de comparer des simulations à volatilité constante et des scénarios dépendant du smile, illustrant la sensibilité du portefeuille à la forme de la surface.

7 Résultats et validation

Les tests ont montré :

- des inversions stables et précises (σ retrouvée à 10^{-6} près) ;
- des smiles lissés continus et monotones ;
- un RMSE sur prix inférieur à 0.3 % de S_0 pour des données synthétiques ;
- une cohérence complète entre pricing BS et pricing MC « smile-aware ».



FIGURE 5 – Heatmap IV : visualisation de la surface $\sigma(T, K)$.

8 Problèmes rencontrés et solutions

8.1 Vega faible, Newton instable (théorie → pratique)

Symptôme : divergence ou stagnation quand T est très court ou K très loin du forward (Vega ≈ 0).

Remèdes : bornes strictes, bisection dès que l'update Newton sort de l'intervalle, critères d'arrêt doubles (prix *et* variation relative sur σ).

Gain : convergence systématique sur le dataset.

8.2 Oscillations de spline (data → modèle)

Symptôme : cubique naturelle \Rightarrow artefacts entre quotes espacées.

Remède : *Akima monotone* sur $x = \ln(K/S_0)$; monotonie empirique en K respectée.

Gain : smile régulier, pas d'overshoot.

8.3 Interop UI \leftrightarrow Workers (ingénierie)

Symptôme : auto-stress qui n'appliquait pas le dernier état «Use Smile / Vega%».

Remède : appel systématique à `applySmileSettingsToWorker_()` avant chaque `runPricing / runStress` (setters en `Qt::QueuedConnection`) ; journaux `QDebug()` ajoutés (*useSmile*, *vega%*, *hasSurface*).

8.4 Export & traçabilité

Symptôme : difficile de «rejouer» une session.

Remède : export JSON complet (Market, Instrument, MC config, Smile \Rightarrow slices, flags, seed) + CSV surface/résidus + PNG des graphes.

Gain : expérience reproductible pour le correcteur.

9 Lecture du projet selon trois angles

Finance / théorie. Formules explicitement rappelées (BS, Vega), calibration comme problème inverse, analyse des résidus (biais, skew directionnel), remarques d'identifiabilité et contraintes de non-arbitrage.

Programmation / ingénierie financière. Architecture modulaire (MarketSurface, SmileSurface, McWorker, CalibWorker), threading Qt et signaux/slots, CRN pour comparaisons MC, exports (*seed include*) pour la reproductibilité.

Données / marché. RMSE, biais moyen, z-scores, lecture qualitative du skew et de sa term-structure, heatmap $\sigma(T, K)$; indication lorsque les données sont synthétiques.

10 Bilan pédagogique et apports

- **Pont théorie–données** : j’ai relié un modèle analytique à des prix réels, compris la sémantique d’une surface IV et ses limites.
- **Capacité d’analyse** : grâce aux diagnostics (RMSE, résidus, heatmap), je peux argumenter sur la qualité d’un fit et identifier des outliers.
- **Expérimentation interactive** : l’UI me permet de tester des hypothèses (impact d’un % de Vega, changement de seed, variation des pas MC) et de voir l’effet immédiatement.
- **Rigueur d’ingénierie** : threading Qt, signaux/slots, `QueuedConnection`, gestion d’état et exports propres.

Conclusion : la phase 4 transforme *RiskWorkbench* d’un moteur de pricing en un *laboratoire d’analyse de marché* utilisable pour l’apprentissage et l’exploration.

Annexe : Formules et rappels mathématiques

Formule de Black–Scholes

$$C(S_0, K, T, r, q, \sigma) = S_0 e^{-qT} N(d_1) - K e^{-rT} N(d_2),$$
$$d_{1,2} = \frac{\ln(S_0/K) + (r - q \pm \frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}}.$$

Vega (sensibilité du prix à la volatilité)

$$\text{Vega} = \frac{\partial C}{\partial \sigma} = S_0 e^{-qT} \sqrt{T} \phi(d_1),$$

où ϕ est la densité de la loi normale standard.

Erreur quadratique moyenne (RMSE)

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_i (P_{\text{fit},i} - P_{\text{mkt},i})^2}.$$

Ces formules constituent le socle mathématique du moteur de calibration et des diagnostics utilisés dans la phase 4.