



# ÉVALUATION ET GESTION DE RISQUES

**Site Web du projet :**

<https://github.com/GearlessJohn/deep-pricing>

8 mars 2023

---

Xuan Wang, Mateus Siqueira Thimóteo



# TABLE DES MATIÈRES

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Méthodologie</b>	<b>2</b>
2.1	Présentation des Modèles . . . . .	2
2.1.1	Modèle de Black-Scholes . . . . .	2
2.1.2	Modèle de volatilité stochastique de Heston . . . . .	3
2.1.3	Nos Modèles . . . . .	3
2.2	Méthodes de Fourier . . . . .	4
2.2.1	Transformée de Fourier . . . . .	4
2.2.2	Fonction caractéristique . . . . .	5
2.3	Justification théorique . . . . .	9
2.3.1	Calcul de la covariance du processus $X_t$ . . . . .	9
<b>3</b>	<b>Simulation numérique</b>	<b>11</b>
3.1	Méthodologie . . . . .	11
3.1.1	Méthode de Monte-Carlo . . . . .	11
3.1.2	Calibration . . . . .	13
3.1.3	Données et différentes versions du modèle . . . . .	13
3.2	Résultats . . . . .	15
3.2.1	Tests intermédiaires . . . . .	15
3.2.2	Calibration à échéance unique . . . . .	17
3.2.3	Calibration à multi-écheances . . . . .	20
<b>4</b>	<b>Conclusion</b>	<b>23</b>
<b>5</b>	<b>Appendix</b>	<b>24</b>
5.1	Information du logiciel et de l'ordinateur utilisés . . . . .	24

# 1

## INTRODUCTION

---

La modélisation de la volatilité dans les marchés financiers est d'extrême importance. Elle est évidemment nécessaire pour la valorisation et le *trading* d'options, mais impacte aussi les prix des sous-jacents, grâce à des réplifications de portefeuilles et au principe de non-arbitrage - par *couverture delta*, par exemple. En particulier, la volatilité doit être (très) prise en compte même par les acteurs qui négocient à peine les sous-jacents. Elle fournit beaucoup de précisions sur la distribution de probabilité des rendements d'un actif.

Le modèle de Black-Scholes (1973), qui suppose une volatilité constante par rapport au temps et au prix *strike* d'une option, n'est pas assez flexible pour s'ajuster aux données du marché d'options. Afin d'avoir un modèle plus fidèle à celles-ci, il faut affaiblir les hypothèses sur la volatilité. L'on peut permettre qu'elle évolue en fonction du prix et du temps - ou, plus généralement, de toute la trajectoire - et cela de manière déterministe *ou* stochastique.

Plusieurs modèles paramétriques de volatilité peuvent être mis en place à partir de ces généralisations. Les problèmes de *calibration* auxquels l'on s'intéresse consistent à optimiser les paramètres de ces modèles afin de maximiser l'ajustement aux données du marché d'options. Cet ajustement est, bien entendu, mesuré par une fonction de perte à choisir. L'algorithme d'optimisation est également à définir, mais il ne fait pas d'objet d'étude de ce rapport.

Ici, l'on s'intéresse à définir deux modèles spécifiques de volatilité stochastique, dits *fractionnaire* et *Exponentiel*, et à vérifier leur capacité d'ajustement aux données de marché. Dans cet article, nous commençons par les calculs et simulations pertinents du modèle. Après avoir été en mesure de générer des échantillons  $S_t$  sur la base des paramètres donnés, nous commençons par une calibration pour une seule échéance. Puis nous réalisons des calibrations globales pour les données composées de différentes échéances ensemble. On veut aussi toujours tenir compte du fait que les calculs nécessaires puissent être faits en temps raisonnable.

## 2

# MÉTHODOLOGIE

## 2.1 PRÉSENTATION DES MODÈLES

### 2.1.1 • MODÈLE DE BLACK-SCHOLES

Black-Scholes (1973) ont dérivé des solutions en forme fermée pour le des prix des options d'achat et de vente européennes dans le modèle de mouvement brownien géométrique où la dynamique du prix de l'action  $S_t$  est donnée par :

$$dS_t = S_t \sigma dB_t, \quad S_0 > 0 \quad (1)$$

avec  $B$  un mouvement Brownien standard et  $\sigma$  le paramètre constant de volatilité instantanée.

**Volatilité implicite** Dans le cadre du modèle Black-Scholes, pour chaque prix d'option observé dans un marché, l'on peut calculer l'unique valeur de volatilité qui donne à l'option la valorisation observée (dès qu'elle ne viole pas des contraintes de non-arbitrage). Cette correspondance entre prix et volatilité est très importante pour optimiser la calibration ; elle peut être adressée de deux manières principales :

- Méthode de Newton : permet une convergence rapide
- Méthode de bisection : converge dans une plage plus grande de *strikes*

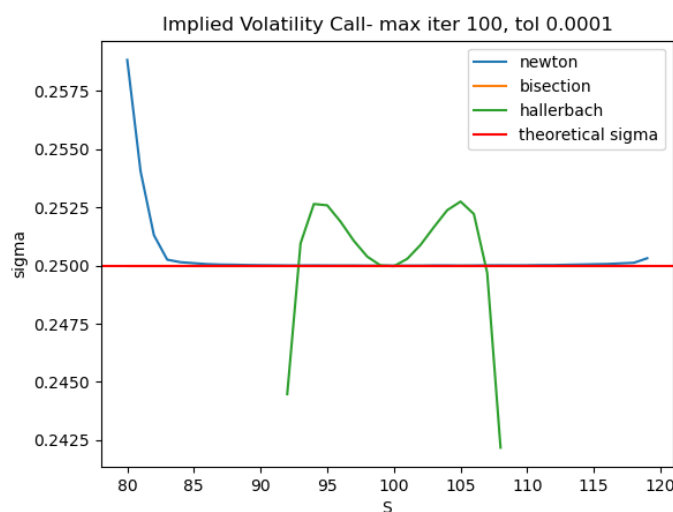


FIGURE 1 – Champ d'application des différentes méthodes de calcul de la volatilité implicite

Une troisième approche, basée sur l'approximation de la fonction inverse de la valorisation Black-Scholes (Hallerbach), n'est efficace que pour les options sur la monnaie, n'ayant donc pas d'utilité pratique pour ce travail.

### 2.1.2 • MODÈLE DE VOLATILITÉ STOCHASTIQUE DE HESTON

Le modèle de volatilité stochastique de Heston (1993) est spécifié comme suit :

$$dS_t = \mu_t S_t + S_t \sqrt{v_t} dB_t, \quad S_0 > 0 \quad (2)$$

$$dv_t = \kappa(\theta - v_t)dt + \omega \sqrt{v_t} dW_t^1 \quad (3)$$

Pour tenir compte de l'effet de levier, les processus stochastiques de Wiener  $B_t$ ,  $W_t^1$  doivent être corrélés  $dB_t \cdot dW_t^1 = \rho dt$ . L'actif sous-jacent  $S$  a une variance stochastique  $v$ , qui est modélisée comme un processus de racine carrée à retour à la moyenne. Le paramètre  $\kappa$  est le taux de retour à la moyenne de la variance,  $\theta$  est le niveau de long terme de la variance et  $\omega$  est la volatilité de la variance. Enfin, la dérive  $\mu_t$  est utilisée pour s'adapter à la courbe à terme du sous-jacent. Notons que la partie déterministe du processus (3) est asymptotiquement stable si  $\kappa > 0$ . Clairement, ce point d'équilibre est  $v_t = \theta$ .

Au-delà de cela, nous avons les hypothèses suivantes concernant le modèle :

$$\kappa > 0, \omega > 0, |\rho| < 1 \quad (4)$$

Le modèle proposé par Heston étend le modèle de Black-Scholes et l'inclut comme un cas particulier. Le cadre de Heston prend en compte la distribution non-lognormale des rendements des actifs, l'effet de levier, l'importante propriété de retour à la moyenne de la volatilité et il reste analytiquement tractable.

### 2.1.3 • NOS MODÈLES

Maintenant, on se concentre à décrire l'ensemble des modèles à vol stochastique en analyse, qui sont des modèles à *noyau* :

$$dS_t = S_t \sigma_t dB_t \quad (5)$$

$$\sigma_t = \xi(t) \frac{P(X_t)}{\sqrt{E[P(X_t)^2]}} \quad (6)$$

$$X_t = \int_0^t K(t, s) dW_s^1 \quad (7)$$

$$B_t = \rho W_t^1 + \sqrt{1 - \rho^2} W_t^2 \quad (8)$$

$$P(x) = \sum_n \alpha_n x^n \quad (9)$$

Ici,  $W^1$  et  $W^2$  sont des Browniens standards *indépendants*.

L'on définit une dynamique de prix log-normale à court-terme, de façon analogue à pour le cas de vol déterministe, à l'exception d'utiliser  $B_t$  à la place de  $W_t$ . En revanche,  $\sigma_t$  est un processus **stochastique**, décrit en fonction de  $x$ . De plus, on observe que  $B_t$  et  $\sigma(t, x)$  sont corrélés, puisque  $\sigma$  dépend de  $x$ .

Ici,  $K : [0, T]^2 \rightarrow \mathbb{R}$  est un noyau dans  $L^2$ , i.e., tel que  $\int_0^T \int_0^T K(t, s)^2 ds dt < +\infty$ . On considère l'utilisation de deux types de noyaux :

- Exponentiel :  $K(t, s) = \eta e^{-\lambda(t-s)} \mathbb{1}_{\{s \leq t\}}$
- Fractionnaire :  $K(t, s) = c(t-s)^{H-\frac{1}{2}} \mathbb{1}_{\{s \leq t\}}$

**Espace de paramètres** En fonction du noyau choisi, nous avons bien  $\eta$  et  $\lambda$  ou  $c$  et  $H$  à être optimisés. De plus, il nous faut :

- Définir le format spécifique de la série formelle (9) (typiquement, l'on s'intéresse aux cas linéaire et exponentiel), ainsi qu'optimiser ses paramètres. Un paramètre redondant sera ainsi créé (avec  $\lambda$  ou  $c$ ).
- Calibrer  $\rho$ . L'on se restreint (au moins) à l'intervalle  $[-1, 0]$ .

**Explication des paramètres** Les observations suivantes expliquent la pertinence pratique du modèle :

- $\rho$  représente (indirectement) la corrélation entre la variation de la volatilité et celle du prix. En pratique, on s'attend à une corrélation particulièrement négative ; les moments de crise dans un marché sont typiquement les plus volatiles/risqués.
- $x$  est un facteur de la volatilité. Si  $p$  affine,  $x$  est (réductible à) la volatilité.
- Dans le modèle exponentiel,  $\eta$  est la volatilité du facteur  $x$ , donc une mesure de à quel point la volatilité est non-déterministe.
- Dans le noyau fractionnaire,  $\lambda$  représente un retour à la moyenne (de la volatilité).

## 2.2 MÉTHODES DE FOURIER

Concentrons-nous maintenant aux méthodes d'inversion de Fourier pour calculer le prix d'un actif. Cela nous permettra d'établir une procédure de calibration plus rapide pour les modèles de volatilité stochastique en analyse. La modélisation présentée ici est issue de référence [4].

### 2.2.1 • TRANSFORMÉE DE FOURIER

Tout d'abord, nous établirons les définitions et propriétés principales de la transformée de Fourier, outil de base pour cette méthode de pricing.

**Définition** Soit  $L^1(\mathbb{R})$  l'ensemble des fonctions  $f : \mathbb{R} \rightarrow \mathbb{R}$  absolument intégrables, i.e., telles que

$$\int_{\mathbb{R}} |f(x)| dx < \infty \quad (10)$$

Pour toute fonction  $f$  dans l'espace  $L^1$ , on définit sa transformée de Fourier  $\hat{f} : \mathbb{R} \rightarrow \mathbf{C}$  par

$$\hat{f}(z) = \int_{\mathbb{R}} e^{izx} f(x) dx \quad (11)$$

pour tout  $z \in \mathbb{R}$ .

L'intérêt du sujet pour cette méthode se justifie par le fait que la fonction caractéristique, calculable explicitement pour le modèle considéré, est une transformée de Fourier.

**Propriétés** Certaines propriétés remarquables de la transformée de Fourier  $\hat{f}$  sont :

- Continuité :  $\hat{f}$  est continue, avec  $|\hat{f}(z)| \rightarrow 0$  lorsque  $|z| \rightarrow \infty$  ;
- Dérivation de la transformée :  $f^{(n)}(z) = \int_{\mathbb{R}} f(x) e^{izx} (ix)^n dx$ , dès que  $g(x) := x^n f(x)$  soit dans  $C^n(\mathbb{R})$  ;
- Dérivation : pour tout  $k \in \mathbf{N}$ , si  $f \in C^n(\mathbb{R})$  est telle que  $f^{(k)} \in L^1(\mathbb{R})$ , alors  $g(z) := z^k \hat{f}(z)$  est bornée. De plus :

$$g(z) = i^k \int_{\mathbb{R}} f^{(k)}(x) e^{ixz} dx \quad (12)$$

- Anti-symmétrie : puisque  $f$  prend des valeurs réelles,  $f(x) = \frac{1}{2\pi} \int_{\mathbb{R}} e^{-izx} \hat{f}(z) dz$  presque partout dans  $\mathbb{R}$  ;
- Invariance de la parité : si  $f$  est paire (resp. impaire),  $\hat{f}$  le sera aussi.

**Formule d'inversion** La transformée de Fourier a l'importante propriété de que, si l'on l'applique deux fois de suite à une fonction dans  $L^1(\mathbb{R})$ , on obtient, à une constante près, la fonction originale.

Si  $f \in L^1(\mathbb{R})$  est telle que  $\hat{f} \in L^1(\mathbb{R})$ , alors :

$$f(x) = \frac{1}{2\pi} \int_{\mathbb{R}} e^{-izx} \hat{f}(z) dz \quad (13)$$

### 2.2.2 • FONCTION CARACTÉRISTIQUE

Nous voulons calculer la valeur présente d'un actif, qui est donnée par  $C_0 = e^{-rT} E[g(S_T)]$  sous l'hypothèse de mesure neutre en risque. Pour évaluer cette espérance, il est suffisant de connaître la densité de probabilité de  $S_T$  issue du modèle (dès qu'elle existe), via la formule suivante :

$$C_0 = e^{-rT} \int_{\mathbf{R}} g(S_0 e^x) P_T(x) dx \quad (14)$$

Ici,  $P_T(\cdot)$  est la densité du log-prix  $\log\left(\frac{S_T}{S_0}\right)$ .

**Fonction caractéristique et Fourier** Le calcul du prix  $C_0$  de l'actif reste ainsi dépendant de la densité  $P_T$ . Une manière de l'obtenir, si l'on a une expression explicite de la fonction caractéristique du modèle, est à travers l'inversion de Fourier. En effet, la fonction caractéristique d'un modèle est donnée par  $E \left[ e^{iv \log \frac{S_T}{S_0}} \right] = \int_{\mathbf{R}_+} e^{ivx} P_T(x) dx$ . Si l'on connaît une formule close pour cela, une transformée inverse de Fourier nous fournit l'expression de la densité  $P_T$ .

De cette manière, il faut se concentrer sur comment calculer, à partir d'un modèle donné, la fonction caractéristique. Pour y procéder de manière constructive, on la calculera tout d'abord pour le modèle de Black-Scholes.

**Black-Scholes : fonction caractéristique** La fonction caractéristique du modèle de Black-Scholes est donnée par :

$$\varphi_{BS}(v, t) = E \left[ e^{iv \ln(S_t/S_0)} \right] = E \left[ e^{iv \left( \ln \int_0^t S_u \sigma dB_u - \ln S_0 \right)} \right] \quad (15)$$

Par construction, le modèle de Black-Scholes satisfait  $d(\ln S_t) = \sigma dB_t$ , d'où  $\ln(S_t)$  suit une loi  $N(\ln(S_0), \sigma^2)$ . Ainsi :

$$\varphi_{BS}(v, t) = E \left[ e^{ivX} \right] \quad (16)$$

où  $X \sim N(0, \sigma^2)$ . On procède par le calcul d'une intégrale gaussienne (en faisant intervenir la densité de probabilité de  $X$ ). On a que :

$$E \left[ e^{ivX} \right] = \int_{\mathbf{R}} e^{-\frac{1}{2\sigma^2}x^2 + ivx} = \frac{e^{-\frac{v^2\sigma^2}{2}}}{2} \quad (17)$$

**Heston : fonction caractéristique** Une fois établi le processus de calcul d'une fonction caractéristique, nous allons reprendre un résultat de référence [5] pour déduire celle du modèle d'intérêt. En particulier, on s'intéressera au cas où la volatilité est une fonction affine du facteur  $x$ , avec  $\xi_0$  constant, indépendant du temps.

On doit ainsi traduire les paramètres de notre modèle à la notation du papier pour établir une correspondance. Pour ce faire, nous utilisons la même procédure que celle du modèle de Vasicek (1977) (ref [2]) - cela nous donnera la dynamique de la volatilité  $\sigma_t$ .

Reprenons les deux équations du modèle de Heston, i.e. (2) et (3), et celles de notre modèle ((5) à (9)). Nous nous concentrons au cas spécifique où (6) est substituée par  $\sigma_t = \xi(t)P(X_t)$ , le noyau est exponentiel, i.e.,  $K(t, s) = \eta e^{-\lambda(t-s)} \mathbf{1}_{s \leq t}$  et, dans (9),  $P(x) = a_0 + a_1x$ .



Dans Heston, l'équation (3) fait référence à la volatilité, donc il nous faut retrouver l'EDS pour la volatilité dans notre modèle. Nous voulons ainsi calculer  $d\sigma_t^2$ , correspondant à  $dv_t$ . On observe que  $\sigma_t^2 = \xi_0^2 (a_0^2 + 2a_0a_1X_t + a_1^2X_t^2)$ , d'où :

$$d(\sigma_t^2) = 0 + 2a_0a_1\xi_0^2dX_t + a_0^2\xi_0^2dX_t^2 \quad (18)$$

Grâce à la correspondance avec Vasicek, nous avons que :

$$dX_t = -\lambda X_t dt + \eta dW_t^1 \quad (19)$$

Il nous reste juste à savoir calculer  $dX_t^2$ . On pose  $\rho_t := e^{\lambda t}X_t$ , d'où  $d\rho_t = \eta e^{\lambda t}dW_t^1$ . En utilisant la formule d'Ito appliquée à la fonction  $f(x, t) = x^2e^{-2\lambda t}$ , on pourra calculer  $dX_t^2$ . Les dérivées partielles pertinentes de  $f$  sont :

$$\frac{\partial f}{\partial x} = 2e^{-2\lambda t}x \quad (20)$$

$$\frac{\partial^2 f}{\partial x^2} = 2e^{-2\lambda t} \quad (21)$$

$$\frac{\partial f}{\partial t} = -2\lambda e^{-2\lambda t}x^2 \quad (22)$$

Il en découle que :

$$dX_t^2 = \left(-2\lambda t e^{-2\lambda t}\rho_t^2 + e^{-2\lambda t}\eta^2 e^{2\lambda t}\right) dt + 2e^{-2\lambda t}\rho_t d\rho_t \quad (23)$$

On remplace  $d\rho_t$  pour obtenir :

$$dX_t^2 = \left(-2\lambda X_t^2 + \eta^2\right) dt + 2\eta X_t dW_t^1 \quad (24)$$

Finalement, on remplace les résultats obtenus dans (18), d'où on obtient l'expression finale pour  $d(\sigma_t^2)$  :

$$dv = d\sigma_t^2 = \left(-2a_1^2\lambda X_t^2 - 2a_0a_1\lambda X_t + a_0^2\eta^2\right)\xi_0^2 dt + \left(2a_0a_1\eta + 2a_1^2\eta X_t\right)\xi_0^2 dW_t^1 \quad (25)$$

Pour établir la correspondance entre modèles, nous devons encore réécrire cette équation en fonction de  $\sigma_t$  à la place de  $X_t$ . Pour ce faire, on utilise la relation  $\xi_0 a_1 X_t = \sigma_t - a_0 \xi_0$ . Cela nous donne les expressions suivantes pour les coefficients de  $dt$  et  $dW_t^1$  :

$$[dW_t^1] = 2\eta\xi_0 a_1 \sigma_t \quad (26)$$

$$[dt] = -2\lambda \left(\sigma_t - \frac{\xi_0 a_0}{2}\right)^2 + a_0^2 \xi_0^2 \left(\eta^2 + \frac{\lambda}{2}\right) \quad (27)$$

Finalement, une comparaison directe avec le modèle de Heston nous donne la correspondance suivante :

$$\kappa = 2\lambda \quad (28)$$

$$\theta = a_0^2 \xi_0^2 \left( \frac{\eta^2}{2\lambda} + \frac{1}{4} \right) \quad (29)$$

$$\omega = 2\eta \xi_0 a_1 \quad (30)$$

$$\mu(t) = 0 \quad (31)$$

**Fonction caractéristique** Dans [5], on nous fournit l'expression de la fonction caractéristique du modèle de Heston :

$$\phi(u) = \exp(iu \ln(S_T) + A(u, \tau) + B_\nu(u, \tau)v(0)) \quad (32)$$

Ici,

$$B_\nu(u, \tau) = \frac{\beta(u) - D(u)}{\omega^2} \frac{1 - e^{-D(u)\tau}}{1 - G(u)e^{-D(u)\tau}} \quad (33)$$

$$A(u, \tau) = \kappa \theta \omega^{-2} ((\beta(u) - D(u)) \tau - 2 \ln \psi_2(u, \tau)) \quad (34)$$

où :

$$G(u) = \frac{\beta(u) - D(u)}{\beta(u) + D(u)} \quad (35)$$

$$D(u) = \sqrt{\beta(u)^2 - 4\hat{\alpha}(u)\gamma} \quad (36)$$

$$\hat{\alpha}(u) = -\frac{1}{2}u(i + u) \quad (37)$$

$$\beta(u) = \kappa - \rho \omega u i \quad (38)$$

$$\gamma = \frac{1}{2}\omega^2 \quad (39)$$

$$\psi_2(u, \tau) = \frac{c(u)e^{D(u)\tau} - 1}{c(u) - 1} \quad (40)$$

$$c(u) = \frac{1}{G(u)} \quad (41)$$

Remarque : pour implémenter le calcul de cette fonction caractéristique, on commence par la traduction de notre modèle en analyse à la notation de Heston, en utilisant les équations (28) à (31). Par la suite, on procède avec le calcul direct grâce à l'expression (32) ci-dessus.

**Prix d'une call par Fourier** Le but de cette méthode est d'arriver à calculer le prix d'un actif d'intérêt. Elle nous permet de le faire sans utiliser beaucoup de puissance de calcul, ce qui peut devenir nécessaire pour les méthodes de Monte Carlo.

L'introduction d'une variable de contrôle peut être utile pour l'algorithme de *pricing* en question. Un résultat de Cont et Tankov (2004) nous permet d'utiliser Black-Scholes comme variable de contrôle puis, à l'aide de la fonction caractéristique du modèle d'intérêt, calculer le prix par inversion de Fourier (et Gil-Paleaz).

Ce résultat établit que, dès qu'il existe  $\alpha > 0$  tel que  $E \left[ e^{(1+\alpha)(\log(S_T/S_0) - rT)} \right] < \infty$ , alors le prix d'une call avec *log-moneyness*  $k = \log \left( \frac{K}{S_0 e^{rT}} \right)$  est donné par :

$$C_0(k) = C_0^{BS}(k, \sigma_{BS}) + \frac{S_0}{\pi} \int_0^\infty \operatorname{Re} \left( e^{-izk} \frac{\varphi_T(z-i) - \varphi^{BS}(z-i)}{iz(1+iz)} \right) dz \quad (42)$$

Ici,  $\varphi_T$  est la fonction caractéristique du modèle ;  $\varphi^{BS}$ , celle de Black-Scholes.

Remarque : en pratique, l'intégrale sera calculée par discrétisation entre 0 et une valeur  $M$  fixée.

Cela conclut la méthode de *pricing* par inversion de Fourier de la fonction caractéristique.

## 2.3 JUSTIFICATION THÉORIQUE

### 2.3.1 • CALCUL DE LA COVARIANCE DU PROCESSUS $X_t$

La simulation de trajectoires (discrétisées) à partir de l'un des noyaux requiert la connaissance de la matrice de covariance entre les plusieurs points simulés. Une fois qu'elle est obtenue, il nous reste à appliquer une décomposition de Cholesky pour simuler nos données, de façon similaire à la simulation typique d'un mouvement brownien. Ainsi, concentrons-nous d'abord à calculer  $\operatorname{Cov}(X_s, X_t)$ . On précise que, puisque  $X_t$  est l'intégrale d'une fonction déterministe, sa loi est gaussienne ; comme  $x_0$  est centrée,  $X_t$  le sera aussi. On revient donc à calculer  $E[X_s X_t]$ .

$$X_t = \int_0^t K(t, s) dW_s^1$$

En supposant que  $s < t$ , on a :

$$\begin{aligned} E[X_s X_t] &= E \left[ \int_0^t K(t, u) dW_u^1 \int_0^s K(s, u) dW_u^1 \right] \\ &= E \left[ \int_0^s K(t, u) dW_u^1 \int_0^s K(s, u) dW_u^1 \right] + E \left[ \int_s^t K(t, u) dW_u^1 \int_0^s K(s, u) dW_u^1 \right] \\ &= E \left[ \int_0^s K(t, u) dW_u^1 \int_0^s K(s, u) dW_u^1 \right] + 0 \quad (\text{Caractéristiques du mouvement Brownien}) \\ &= \int_0^s K(t, u) K(s, u) du \quad (K \text{ fonction déterministe, isométrie}) \end{aligned}$$

Lorsque  $s \geq t$ , nous sommes en mesure d'effectuer un calcul similaire. Le résultat final est :

$$E[X_s X_t] = \int_0^{\min(s,t)} K(t,u)K(s,u)du \quad (43)$$

**1. Modèle exponentiel** Pour le modèle exponentiel,  $K(t,s) = \eta e^{-\lambda(t-s)} \mathbb{1}_{\{s \leq t\}}$ , en supposant que  $s \leq t$ , on obtient :

$$\begin{aligned} E[X_s X_t] &= \int_0^s K(t,u)K(s,u)du \\ &= \int_0^s \eta e^{-\lambda(t-u)} \mathbb{1}_{\{u \leq t\}} \eta e^{-\lambda(s-u)} \mathbb{1}_{\{u \leq s\}} du \\ &= \eta^2 \int_0^s e^{-\lambda(t+s-2u)} du \\ &= \frac{\eta^2}{2\lambda} e^{-\lambda(t_i+t_j)} (e^{2\lambda \min(t_i,t_j)} - 1) \end{aligned}$$

Donc, d'une manière générale :

$$E[X_s X_t] = \frac{\eta^2}{2\lambda} e^{-\lambda(t_i+t_j)} (e^{2\lambda \min(t_i,t_j)} - 1) \quad (44)$$

**2. Modèle fractionnaire** Pour le modèle fractionnaire,  $K(t,s) = c(t-s)^{H-\frac{1}{2}}$ , on obtient que :

$$\begin{aligned} E[X_s X_t] &= \int_0^s K(t,u)K(s,u)du \\ &= \int_0^s c(t-u)^{H-\frac{1}{2}} c(s-u)^{H-\frac{1}{2}} du \\ &= c^2 \left( f \left( \min(t,s) - \frac{s+t}{2}, \frac{(s-t)^2}{4}, H - \frac{1}{2} \right) - f \left( -\frac{s+t}{2}, \frac{(s-t)^2}{4}, H - \frac{1}{2} \right) \right) \end{aligned} \quad (45)$$

où

$$f(x,k,a) := x k^a {}_2F_1 \left( \frac{1}{2}, -a, \frac{3}{2}, \frac{-x^2}{k} \right) \quad (46)$$

avec  ${}_2F_1$  la fonction hypergéométrique.

Pour  $s = t$  l'expression ci-dessus n'est plus valable et doit être remplacée par :

$$E[X_t^2] = 2c^2 \frac{\left(\frac{t}{2}\right)^{2a+1}}{2a+1} \quad (47)$$

**Discretisation et matrices de covariance** Une fois défini un horizon temporel  $T$  et un niveau de discretisation  $N$ , on pose  $t_n := \frac{T}{N}$ , de sorte à pouvoir construire une matrice de covariance  $N$  par  $N$  ayant comme entrées  $\Sigma_{i,j} = E[x_{t_i} x_{t_j}]$ . L'on pourra appliquer la décomposition de Cholesky à  $\Sigma$  afin d'obtenir une simulation des réalisations de prix des modèles.

# 3

## SIMULATION NUMÉRIQUE

### 3.1 MÉTHODOLOGIE

La simulation numérique sera divisée en deux étapes principales : (i) générer un échantillon de  $S_t$  sous un modèle donné par une série de méthodes de Monte Carlo. (ii) effectuer la calibration du modèle sur la base des données du marché réel et comparer les résultats. La section code de ce rapport utilise principalement les bibliothèques numpy et scipy pour générer des variables gaussiennes et effectuer des calculs numériques.

#### 3.1.1 • MÉTHODE DE MONTE-CARLO

##### Génération d'espaces probabilisés

On détermine à l'avance la répartition  $n$  dans le temps  $(t_i)_{i=0\dots n}$  avec  $t_0 = 0, t_n = T$ , et le nombre d'échantillons  $N$  pour la méthode de Monte Carlo. Ainsi, nous pouvons utiliser numpy pour générer  $N$  ensembles d'échantillons de  $(Z_i^1)_{i=0\dots n}, (Z_i^2)_{i=0\dots n}$  deux vecteurs gaussiens centrés réduits mutuellement indépendants.

Après cela, nous pouvons générer les processus browniens correspondants  $W^1, W^2$  par  $Z^1, Z^2$  selon l'équation suivante :

$$W_{t_k} = \sqrt{\frac{T}{n}} \sum_{i=0}^k Z_i \quad (48)$$

##### Génération d'échantillons $X_t$

On sait que  $(X_t)_{0 \leq t \leq T} = (\int_0^t K(t, s) dW_s^1)_{0 \leq t \leq T}$  est un processus gaussien. On définit

$$\Sigma_{i,j} = Cov(X_{t_i}, X_{t_j}) \quad (49)$$

Alors selon les calculs de chapitre 2.3.1 surtout les équations (44) et (45), on a :

$$\Sigma_{i,j}^{exp} = \frac{\eta^2}{2\lambda} e^{-\lambda(t_i+t_j)} (e^{2\lambda \min(t_i, t_j)} - 1) \quad (50)$$

$$\Sigma_{i,j}^{frac} = c^2 \left( f \left( \min(t_i, t_j) - \frac{t_i + t_j}{2}, \frac{(t_i - t_j)^2}{4}, H - \frac{1}{2} \right) - f \left( -\frac{t_i + t_j}{2}, \frac{(t_i - t_j)^2}{4}, H - \frac{1}{2} \right) \right) \quad (51)$$

La matrice de covariance de  $(X_{t_i})_{i=0\dots n}$  peut être calculée directement à partir du choix de noyau, i.e.

$$(X_{t_i})_{i=0\dots n} = LZ^1 \quad (52)$$

avec

$$\Sigma = LL^T \quad (53)$$

Nous pouvons utiliser la méthode de décomposition de Cholesky propre à numpy pour obtenir L.

En particulier, lorsque K est sous forme exponentielle. On pose

$$Y_{t_i} = e^{\lambda t_i} X_{t_i} = \eta \int_0^{t_i} e^{\lambda s} dW_s^1 \quad (54)$$

Alors

$$Y_{t_{i+1}} = Y_{t_i} + \eta \int_{t_i}^{t_{i+1}} e^{\lambda s} dW_s^1 \quad (55)$$

Par sommation successive :

$$Y_{t_k} = Y_{t_0} + \eta \sum_{i=0}^k \int_{t_i}^{t_{i+1}} e^{\lambda s} dW_s^1 \quad (56)$$

Puisque les  $(W_s, t_i \leq s \leq t_{i+1})_i$  sont indépendants les uns des autres, on a :

$$\int_{t_i}^{t_{i+1}} e^{\lambda s} dW_s^1 \sim \mathcal{N}(0, \int_{t_i}^{t_{i+1}} e^{2\lambda s} ds) \quad (57)$$

On peut donc calculer directement  $(X_{t_i})_{i=0\dots n}$  par une série de sommations :

$$X_{t_k} = e^{-\lambda t_k} \eta \sum_{i=0}^k G_i \quad (\text{on sait que } X_{t_0} = 0) \quad (58)$$

avec  $G_i \sim \mathcal{N}(0, \int_{t_i}^{t_{i+1}} e^{2\lambda s} ds)$  mutuellement indépendants.

**Génération d'échantillons  $B_t$**  Sur la base du coefficient de corrélation  $\rho$  spécifiés, générer  $(B_{t_i})_{i=0\dots n}$ .

$$B_{t_i} = \rho W_{t_i}^1 + \sqrt{1 - \rho^2} W_{t_i}^2 \quad (59)$$

**Génération d'échantillons  $\sigma_t$**   $\sigma(t, X_t) = \xi \frac{P(X_t)}{\sqrt{E(P(X_t))}}$ , le numérateur est calculé à partir de l'échantillon  $(X_t)$  et le dénominateur peut être calculé directement à partir du moment de  $(X_t)$  (puisque  $(X_t)$  suit une distribution gaussienne).

En particulier, dans notre simulation, on a deux choix de  $P(x) = \sum_n \alpha_n x^n$  :

$$P(x) = a_1 x + a_0 \quad (\text{affine}) \quad (60)$$

et

$$P(x) = e^{bx} \quad (\text{exponentiel, i.e. } P(x) = \sum_{n=0}^{\infty} \frac{b^n}{n!} x^n) \quad (61)$$

**Génération d'échantillons**  $S_T$   $S_T = S_0 \exp(-\frac{1}{2} \int_0^T \sigma_s^2 ds + \int_0^T \sigma_s dB_s)$ , pour cela, nous utilisons l'approche d'intégrale de Riemann (sous forme discrète) pour calculer les deux intégrales :

$$\int_0^T \sigma_s^2 ds \approx \sum_{i=0}^{n-1} \frac{T}{n} \left( \frac{\sigma_{t_i} + \sigma_{t_{i+1}}}{2} \right)^2 \quad (62)$$

$$\int_0^T \sigma_s dB_s \approx \sum_{i=0}^{n-1} \sigma_{t_i} (B_{t_{i+1}} - B_{t_i}) \quad (63)$$

**Estimer le prix de "call option"** Enfin, après avoir obtenu la méthode de génération d'échantillons  $S_T$ , nous générons  $N$  échantillons (généralement  $N > 10000$ ) et déterminons le prix de "call option" par la méthode de Monte-Carlo :

$$call(S_0, T, K) = E[(S_T - K)^+] \approx \frac{1}{N} \sum_{i=0}^N (S_T^i - K)^+ \quad (64)$$

### 3.1.2 • CALIBRATION

Pour le modèle que nous avons choisi, il y aura généralement 5 à 8 paramètres :  $\rho$ ,  $\xi$ ,  $a_0$ ,  $H$  (ou  $\lambda$ ),  $c$  (ou  $\eta$ ) etc.

Nous utilisons ici la fonction `mean_squares` de `scipy.optimize`. Étant donné un ensemble d'attributs  $(X_i)_{i=1\dots d}$ , un ensemble de valeurs  $(Y_i)_{i=1\dots d}$ , et une fonction  $f_\theta$  où  $\theta$  est le vector des paramètres que nous devons calibrer et une fonction de perte *loss*, `mean_squares` recherchera à minimiser  $\frac{1}{d} \sum_{i=1}^d loss(f_\theta(X_i), Y_i)$  en ajustant  $\theta$  par itération.

Dans ce rapport, nous considérons trois types de pertes :

1.  $perte_1 = \frac{1}{d} \sum_{i=1}^d |prix_{modèle} - prix_{marché}|^2$
2.  $perte_2 = \frac{1}{d} \sum_{i=1}^d \left| \frac{prix_{modèle} - prix_{marché}}{e^{\frac{T}{2}} 2} \right|$  où  $T$  est la échéance.
3.  $perte_3 = \frac{1}{d} \sum_{i=1}^d \left| \frac{prix_{modèle} - prix_{marché}}{vega} \right|^2$
4.  $perte_4 = \frac{1}{d} \sum_{i=1}^d |\sigma_{modèle} - \sigma_{marché}|^2$  où  $\sigma$  est la volatilité implicite (calculée par la méthode de Newton ou la méthode de bisection).

En fin de compte, nous avons abandonné le 4<sup>e</sup> en raison de problèmes d'efficacité.

### 3.1.3 • DONNÉES ET DIFFÉRENTES VERSIONS DU MODÈLE

Dans cet article, nous nous concentrerons sur deux types de calibrations. L'un est une calibration pour une seule échéance et l'autre est une calibration pour un mélange de données avec différentes échéances.

Avant de procéder au réglage des paramètres spécifiques, nous rappelons le modèle que nous avons utilisé :

$$dS_t = S_t \sigma_t dB_t$$

$$\sigma_t = \xi(t) \frac{P(X_t)}{\sqrt{E[P(X_t)^2]}}$$

$$X_t = \int_0^t K(t, s) dW_s^1$$

$$B_t = \rho W_t^1 + \sqrt{1 - \rho^2} W_t^2$$

Nous pouvons voir que dans ce modèle, nous avons besoin du choix approprié de  $\rho$ ,  $K$ ,  $\xi$  et  $P$ .

**Différentes versions du modèle** Nous donnons ici les noms de code de toutes les versions du modèle utilisé dans cet article et les fonctions spécifiques correspondantes :

Modèle	$K(t, s)$	$\xi(t)$	$P(x)$	Paramètres
<b>exp</b>	$\eta e^{-\lambda(t-s)} \mathbb{1}_{\{s \leq t\}}$	$\xi_0$	$a_1 x + a_0$	$\eta, \lambda, \xi_0, a_0, \rho$
<b>frac</b>	$c(t-s)^{H-\frac{1}{2}} \mathbb{1}_{\{s \leq t\}}$	$\xi_0$	$a_1 x + a_0$	$H, c, \xi_0, a_0, \rho$
<b>iso</b>	$v \epsilon^{H-\frac{1}{2}} e^{\frac{1}{\epsilon}(H-\frac{1}{2})(t-s)} \mathbb{1}_{\{s \leq t\}}$	$\xi_0$	$a_1 x + a_0$	$H, \epsilon, v, \xi_0, a_0, \rho$
<b>exp1</b>	$\eta e^{-\lambda(t-s)} \mathbb{1}_{\{s \leq t\}}$	$\xi_0$	$e^{bx}$	$\eta, \lambda, \xi_0, b, \rho$
<b>ksit</b>	$\eta e^{-\lambda(t-s)} \mathbb{1}_{\{s \leq t\}}$	$\alpha_0 e^{-\beta^2 t} + \alpha_1 (1 - e^{-\beta^2 t})$	$a_1 x + a_0$	$\eta, \lambda, \alpha_0, \alpha_1, \beta, a_0, \rho$
<b>ksit_frac</b>	$c(t-s)^{H-\frac{1}{2}} \mathbb{1}_{\{s \leq t\}}$	$\alpha_0 e^{-\beta^2 t} + \alpha_1 (1 - e^{-\beta^2 t})$	$a_1 x + a_0$	$H, c, \alpha_0, \alpha_1, \beta, a_0, \rho$
<b>ksit_iso</b>	$v \epsilon^{H-\frac{1}{2}} e^{\frac{1}{\epsilon}(H-\frac{1}{2})(t-s)} \mathbb{1}_{\{s \leq t\}}$	$\alpha_0 e^{-\beta^2 t} + \alpha_1 (1 - e^{-\beta^2 t})$	$a_1 x + a_0$	$H, \epsilon, v, \alpha_0, \alpha_1, \beta, a_0, \rho$

### Remarques :

1. Les modèles **exp** et **frac** sont bien deux modèles de bases qui utilisent respectivement le noyau exponentiel et le noyau fractionnaire.
2. On constate que lorsque  $P(x) = a_1 x + a_0$ , la liste des paramètres à calibrer n'inclut pas  $a_1$ . Ceci est dû au fait que la fonction de  $a_1$  peut être remplacée par la combinaison de  $a_0$  et  $\xi$  dans le résumé final de l'expression  $\sigma$ . Plus précisément,  $\xi(a_1 x + a_0) = \xi a_1 x + \xi a_0 = param_1 x + param_0$ , en ajustant les valeurs de  $\xi$  et de  $a_0$ , on peut obtenir toutes les combinaisons possibles.
3. Le modèle **iso** est essentiellement un modèle de noyau exponentiel. Mais en établissant une correspondance entre les paramètres  $\eta = v \epsilon^{H-\frac{1}{2}}$ ,  $\lambda = -\frac{1}{\epsilon}(H - \frac{1}{2})$ , nous sommes en mesure d'obtenir des changements plus rapides (théoriquement) lors du calibrage. Le mappage permet également la corrélation et la comparaison entre les modèles **exp** et **frac**.
4. Les trois derniers modèles commençant par **ksit** sont des versions optimisées des trois modèles du début pour des combinaisons de données à échéance multiple. Ici, nous faisons évoluer  $\xi$  en fonction du temps, renforçant ainsi la limite supérieure de l'ajustement du modèle pour les données mixtes.



**Données** Dans cet article, nous avons principalement utilisé les données contenues dans le fichier *UnderlyingOptionsEODCalcs\_2022-09-02.csv*. Ce fichier contient principalement les attributs suivants :

- `quote_date`
- `expiration (date)`
- `strike`
- `option_type` : C ou P
- `root` : SPX ou SPXW
- `trade_volume`
- `bid_1545`
- `ask_1545`

Tout d'abord, une série de pré-traitements a été effectuée sur les données. Nous avons supprimé toutes les données dont le volume de transaction réel était égal à 0 et les avons classées en fonction des échéances. En outre, nous avons principalement traité les "call options".

Ensuite, la échéance est calculée à partir de la différence entre la date d'expiration et la date de cotation, et les données sont classées selon la échéance. Après avoir obtenu l'échéance  $T$ , nous devons effectuer certains filtres sur la base de  $T$ . Par exemple, lorsque  $T = 14$  jours, nous pensons que  $S_T$  doit être compris entre  $[0.95S_0, 1.05S_0]$ . Et quand  $T=49$  jours, nous sélectionnons les données pour  $S_T \in [0.9S_0, 1.1S_0]$ .

Le prix du marché est défini comme

$$price\_market = \frac{1}{2}(bid\_1545 + ask\_1545) \quad (65)$$

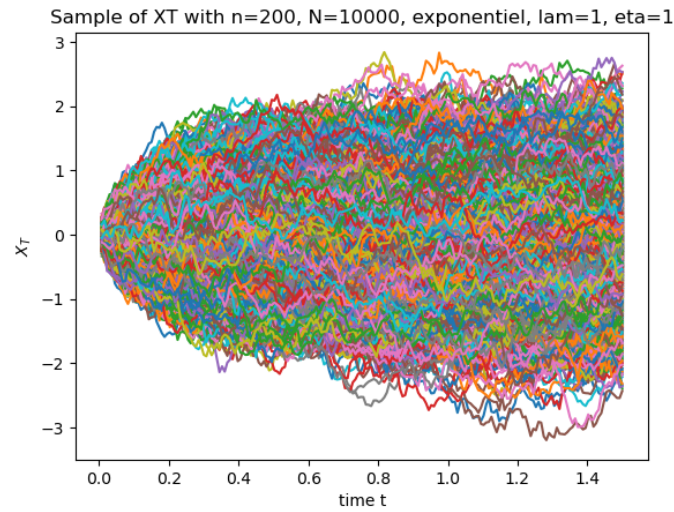
Enfin, pour toutes les simulations traitées dans cet article, le nombre de fractionnements temporels  $n=200$ , et le nombre d'itérations de la méthode de Monte Carlo  $N=10\,000$ . En outre, pour l'optimisateur `/scipy.optimize.least_squares`, nous devons donner trois limites de tolérance. Les limites prises dans cet article sont :

- $ftol = 1.0 \cdot e^{-8}$  : tolérance d'arrêt par la modification de la fonction de coût ;
- $xtol = 1.0 \cdot e^{-8}$  : tolérance d'arrêt par le changement des variables indépendantes ;
- $gtol = 1.0 \cdot e^{-8}$  : tolérance pour la terminaison par la norme du gradient.

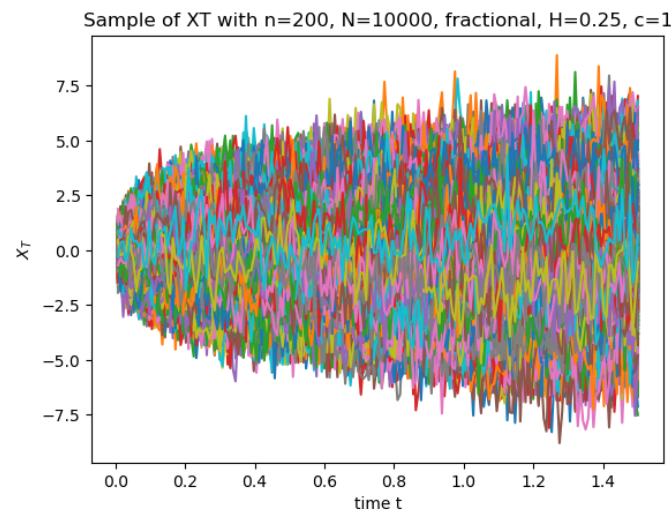
## 3.2 RÉSULTATS

### 3.2.1 • TESTS INTERMÉDIAIRES

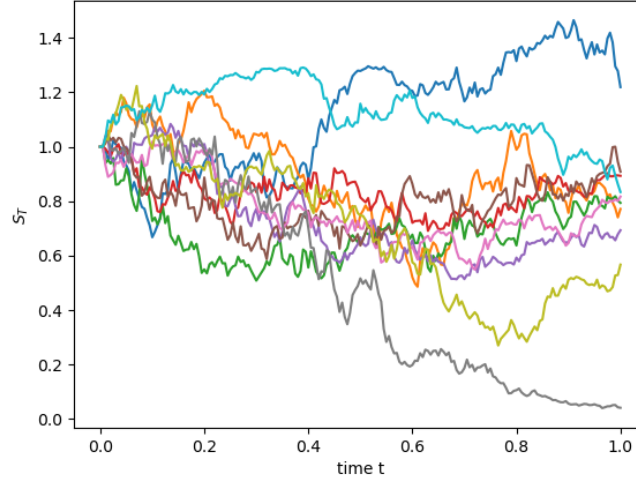
Dans cette section, nous testons certains processus intermédiaires  $(X_t)$ ,  $(S_t)$ . L'exactitude du code est confirmée en vérifiant certaines propriétés connues (moyenne, variance, intervalle de confiance).


FIGURE 2 – Test sur  $(X_t)$  avec noyau exponentiel

$E(X_T) = -0.003$ ,  $Var(X_T) = 0.469$  (*théorie* = 0.475),  $IC(95\%) = [-0.016, 0.010]$ . On peut voir que la valeur moyenne de  $X_T$  s'arrête autour de 0. La variance n'est pas non plus très différente de la valeur théorique donnée par l'équation (44), et l'intervalle de confiance contient bien 0.


FIGURE 3 – Test sur  $(X_t)$  avec noyau fractionnaire

$E(X_T) = 0.008$ ,  $Var(X_T) = 3.493$  (*théorie* = 3.464),  $IC(95\%) = [-0.028, 0.045]$ . On peut voir que la valeur moyenne de  $X_T$  s'arrête aussi autour de 0. La variance n'est pas non plus très différente de la valeur théorique donnée par l'équation (45), et l'intervalle de confiance contient bien 0.

Sample of  $S_T$  (exp kernel) with  $n=200$ ,  $N=10000$ ,  $\text{lam}=1$ ,  $\text{eta}=1$ ,  $T=1$ ,  $\text{rho}=-0.8$ ,  $S_0=1$ FIGURE 4 – Test sur  $(S_t)$  avec noyau exponentiel

$E(S_T) = 1.004$ ,  $Var(S_T) = 0.184$ ,  $IC(95\%) = [0.996, 1.013]$ . On sait que  $(S_t)$  est une martingale. On peut voir que la valeur moyenne de  $S_T$  s'arrête bien autour de  $S_0 = 1$ . Et l'intervalle de confiance contient bien  $S_0$ .

### 3.2.2 • CALIBRATION À ÉCHÉANCE UNIQUE

Dans cette section, nous laissons d'abord les quatre premiers modèles être testés pour une seule échéance. Pour les figures suivantes, nous obtenons d'abord les paramètres du modèle par calibration, puis nous entrons à notre tour les  $S_0$ ,  $T$ ,  $K$  fournis dans le fichier csv pour générer les prix de "call option" pour les prédictions du modèle. Enfin, nous calculons l'implicite de volatilité correspondant à ce prix et le comparons à l'implicite de volatilité correspondant au prix du marché original. Pour quantifier cette comparaison, nous avons utilisé l'erreur quadratique moyenne :

$$MSE(\sigma_{market}, \sigma_{model}) = \frac{1}{d} \sum_{i=1}^d |\sigma_{market}^i - \sigma_{model}^i|^2 \quad (66)$$

où  $\sigma$  est la volatilité implicite et  $d$  représente le nombre de données.

**Fonction de perte** Pour une seule échéance, nous n'avons pas besoin d'effectuer des corrections pour le temps. En d'autres termes, dans ce cas, il n'y a pas de différence entre les fonctions de perte 1 (moyenne de la différence de prix au carré) et 2 (moyenne de la différence de prix au carré divisée par  $e^T$ ). Nous commençons par comparer les fonctions de perte 1 et 3 (moyenne de la différence de prix au carré divisée par vega).

T	frac-I	frac-III	exp-I	exp-III
28	1.47E-08	9.86E-09	2.93E-08	1.97E-08
49	6.07E-09	5.13E-09	3.40E-09	2.88E-09
119	5.13E-07	5.14E-07	5.03E-07	5.14E-07

FIGURE 5 – Erreur quadratique moyenne pour différentes fonctions de perte - échéance unique

T	frac-I	frac-III	exp-I	exp-III
28	63.2	72.1	15.1	16.1
49	72.9	62.6	14.4	13.6
119	46.5	37.2	12.3	9.2

FIGURE 6 – Durée totale de fonctionnement en seconde - échéance unique

Comme le montrent les figures 5 et 6, la fonction de perte 3 n'a pratiquement aucun effet sur le moment de la calibration, mais peut améliorer considérablement la précision de la calibration. Par conséquent, dans les futures simulations de ce document, les fonctions de perte 3 et 2 seront principalement considérées.

**Performance des différents modèles** Nous comparons maintenant les performances des différents modèles dans le cas d'une seule échéance. Ici, nous avons pris  $T=49$  jours et filtré toutes les données avec  $S_T \in [0.93S_0, 1.07S_0]$ . Il y avait 69 entrées.

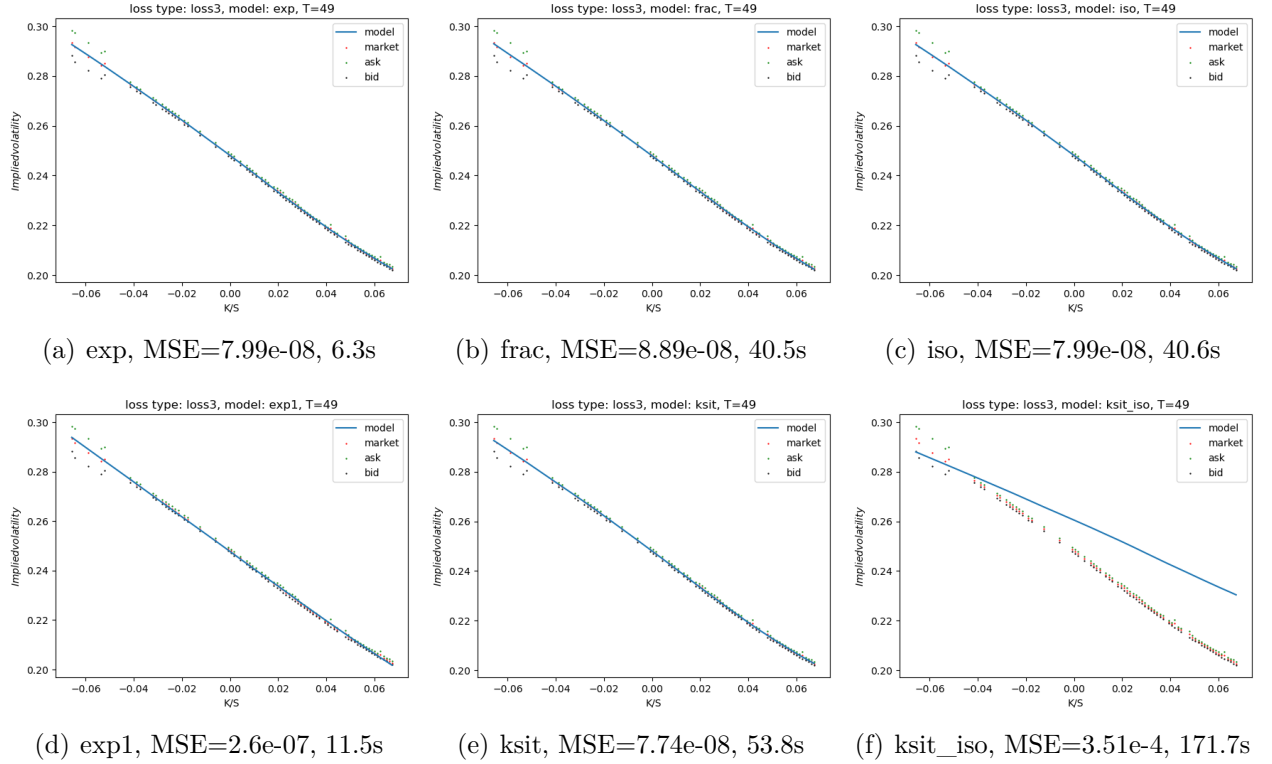


FIGURE 7 – Performance des différents modèles

## Remarques

1. Nous avons abandonné le modèle **ksit\_frac** en raison de problèmes d'efficacité. En effet, il faut plus de 15 minutes pour effectuer la calibration (exp ne prend que quelques secondes) et la précision est presque indiscernable de exp, ce qui est une vitesse inacceptable.
2. L'image montre clairement que nous devons abandonner **ksit\_iso**. Le lecteur se souvient peut-être que c'est le modèle le plus raffiné de tous ceux que nous avons utilisés, avec 8 paramètres. Cependant, la pratique a montré que le modèle ksit\_iso ne peut pas être bien calibré pour la tolérance donnée car il a trop de paramètres, et ne peut donc pas être mis en pratique.
3. Parmi ceux-ci, le modèle **ksit** est le plus performant car il possède le plus de paramètres pratiques et est le modèle le plus complexe. **exp1** est le modèle le moins performant, et c'est aussi le modèle le plus simple avec le moins de paramètres.
4. Le modèle **exp** et le modèle **iso** ont exactement la même erreur MSE, ce qui est prévisible. Ceci est dû au fait que exp et iso sont simplement des représentations différentes du même modèle. Nous avons initialement ajouté le modèle iso pour faire varier les paramètres de manière plus sensible par le biais de la relation de appage ( $x$  passe de 0,1 à 0,2 et  $y$  passe de 10 à 5 dans  $y = \frac{1}{x}$ ). En pratique, cependant, le modèle iso est beaucoup plus lent que exp, car il a un paramètre de plus à calibrer que exp.

5. Nous pouvons voir que les modèles restants (exp, frac, exp1, iso, ksit) ont tous une bonne précision. En particulier, nous avons étudié les conditions d'arrêt de l'optimisateur, qui sont toutes  $ftol$ , c'est-à-dire que la valeur de la fonction de perte réduite pour un tour d'itérations est inférieure à la valeur spécifiée. Cela suggère qu'avec les paramètres de  $n=200, N=10000, ftol=xtol=gtol=1e-8$ , la précision des résultats dépend plus de l'optimisateur que du modèle lui-même.

### 3.2.3 • CALIBRATION À MULTI-ÉCHEANCES

Pour explorer davantage les forces et les faiblesses des différents modèles et leur valeur en pratique, nous utiliserons des données composées d'un mélange de différentes échéances pour la calibration.

**Fonction de perte** Pour le cas multi-écheances, nous commençons par comparer les fonctions de perte 2 (moyenne de la différence de prix au carré divisée par  $e^T$ ) et 3 (moyenne de la différence de prix au carré divisée par vega). Nous avons d'abord sélectionné les données combinées d'échéance  $T=105, 140$  et  $167$ , avec les fenêtres de filtrage supérieure et inférieure de 20%, 25% et 30% pour  $S_T$ . Les trois échéances contiennent respectivement 66, 41 et 47 points.

TABLE 1 – Erreur quadratique moyenne - multi-échéances

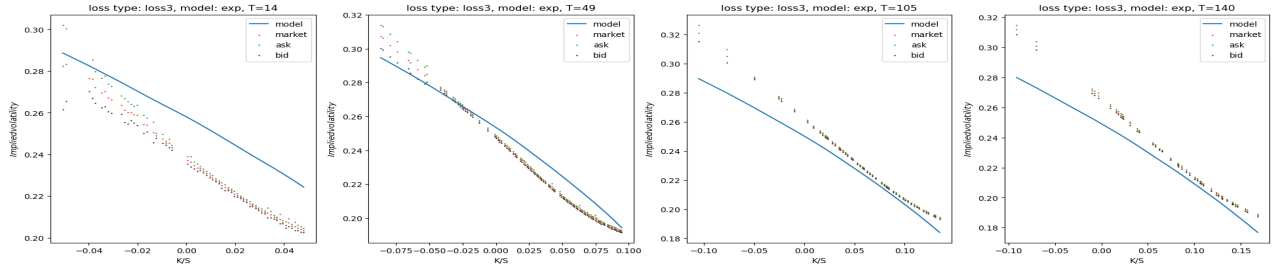
T	frac-II	frac-III	exp-II	exp-III
105, 140, 168	1.2e-05	1.15e-05	8.42e-06	7.99e-06

TABLE 2 – Durée totale de fonctionnement (s) - multi-échéances

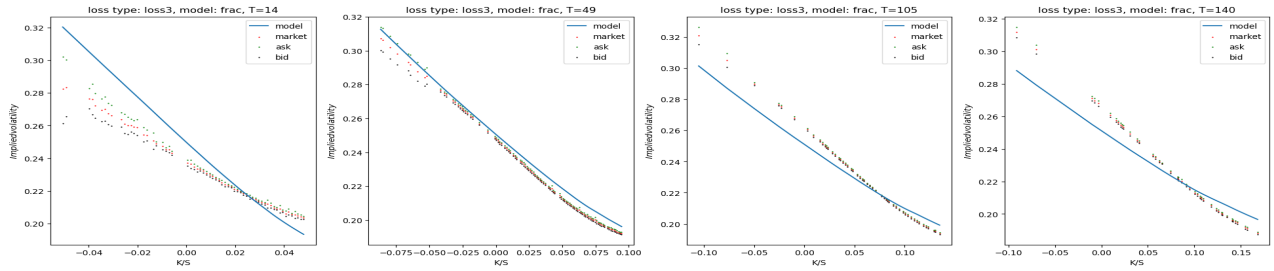
T	frac-II	frac-III	exp-II	exp-III
105, 140, 168	149.5	184.3	83.9	26.4

Nous pouvons voir que la fonction de perte de type 3 est plus performante que la fonction de perte de type 2, à la fois en termes de temps et d'erreur. Nous utilisons donc uniquement la fonction de perte de type 3.

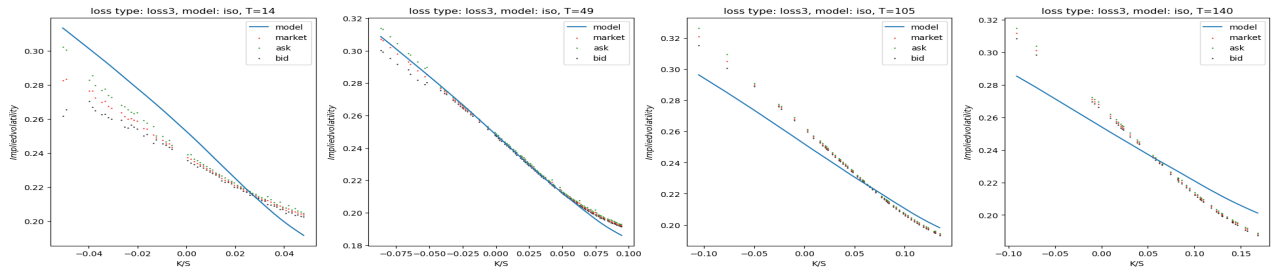
**Calibration avec 4 échéances** Pour examiner la performance de notre modèle dans le cas où plusieurs échéances se calibrent ensemble, nous avons sélectionné 4 échéances ( $T=14, 49, 105, 140$ ) présentant des différences relativement importantes, avec les fenêtres de filtrage supérieure et inférieure de 5%, 10% et 15% et 20% pour  $S_T$ . Les 4 échéances contiennent respectivement 62, 96, 51 et 42 points.



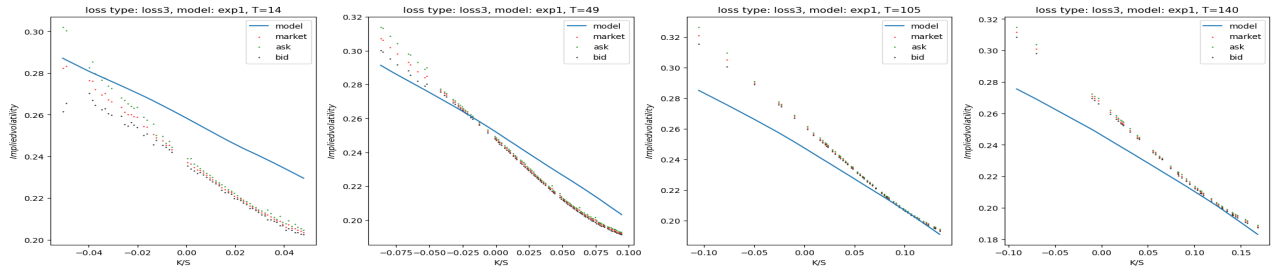
(a) exp, MSE=1.6e-04, 29.7s



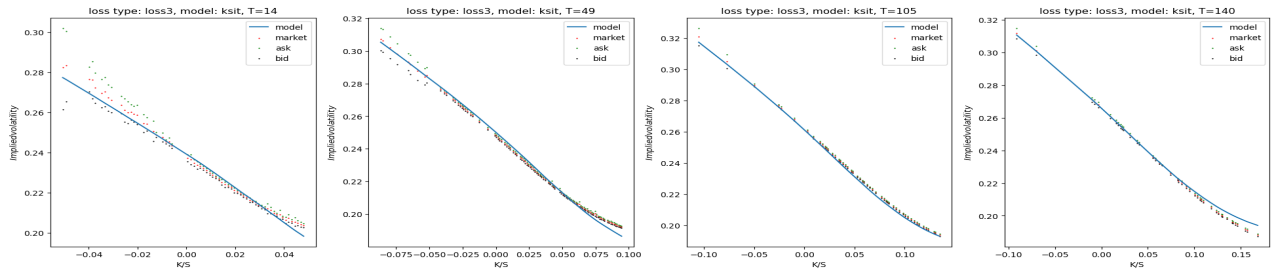
(b) frac, MSE=9.21e-05, 229.1s



(c) iso, MSE=8.68e-05, 37.2s



(d) exp1, MSE=2.04e-04, 27.4s



(e) ksit, MSE=5.83e-06, 45.4s

FIGURE 8 – Performance des différents modèles - multi-échéances

## Remarques

1. Nous pouvons voir que l'écart entre les modèles se creuse progressivement dans le cas de multi-échéances. Le modèle exp a 74% d'erreur en plus que le modèle frac. iso est également meilleur que exp en raison d'un ajustement plus sensible des paramètres. Le modèle exp1 est totalement inadapté à la situation en raison du manque de paramètres.
2. Le modèle ksit est le meilleur car il est plus finement optimisé spécifiquement pour T. Le modèle ksit utilise moins de 1,3 fois le temps utilisé par iso, réduisant l'erreur de plus de 90 %.



## 4

# CONCLUSION

---

Théoriquement, le modèle frac devrait être supérieur au modèle exp que des résultats plus raffinés et meilleurs. En pratique, on constate que le modèle frac utilise la fonction hypergéométrique, et que ce calcul est si complexe que l'efficacité du frac est fortement réduite (c'est-à-dire que l'amélioration de la précision du modèle frac est loin de compenser sa longueur en temps).

En même temps, les modèles sont bons ou mauvais en termes relatifs. Pour le cas d'une seule échéance, le modèle le plus efficace est exp. Ce n'est que lorsque les données sont suffisamment complexes et que les échéances sont suffisamment variées que ksit peut être plus efficace que exp. C'est aussi le cas de iso. Cela nous indique également que le modèle le plus complexe avec le plus de paramètres n'est pas nécessairement le meilleur, et que nous devons prendre en compte le scénario d'application réel.

De plus, nous pouvons voir que tous les modèles, à l'exception de ksit, sont peu performants dans le cas de la multi-échéances. Mais si nous ne regardons pas la volatilité implicite mais comparons directement le prix de "call option" donné par le modèle avec le prix du marché, la différence est très faible. Cela nous indique que les modèles des marchés financiers sont très sensibles aux petites différences de prix. Nous travaillons dans un environnement où les erreurs peuvent être amplifiées à l'extrême.

En outre, nous pouvons constater que la plupart du temps, ce n'est pas le modèle lui-même qui affecte réellement les performances, mais l'algorithme d'optimisation et les réglages des paramètres associés. Cela nous rappelle également que le modèle n'est pas tout, et que le travail de génération associé, le traitement des données, est tout aussi important.

Pour la suite des travaux, je pense qu'il est possible de commencer par le nombre de fractionnements temporels  $n$ .  $n$  devrait être positivement corrélé avec l'échéance. Cela nous permettra d'optimiser davantage les performances du modèle pour la même quantité de calcul.

Deuxièmement, nous pouvons essayer différentes combinaisons de paramètres initiaux, ainsi que différents intervalles de paramètres. Cela a également un impact important sur le temps du processus d'optimisation.

## 5 APPENDIX

---

### 5.1 INFORMATION DU LOGICIEL ET DE L'ORDINATEUR UTILISÉS

---

Les sections de ce rapport traitant des comparaisons de temps et de performances ont été exécutées sur un ordinateur de bureau. Les caractéristiques informatiques sont les suivantes :

- Système d'exploitation : Windows 10 Home Edition 64-bit (Version 21H1 / DirectX 12)
- Processeur : Intel Core i9-10850K @ 3.60GHz
- Nombre de cœurs : 10
- Nombre de threads : 20
- Carte graphique : AMD Radeon RX 6900 XT (16 GB)
- RAM : 32 GB (Kingston DDR4 3600MHz)

## RÉFÉRENCES

---

- [1] Mikhailov, S. & Nögel, U. Heston's stochastic volatility model : Implementation, calibration and some extensions. (John Wiley,2004)
- [2] Tankov, P. & Touzi, N. Calcul stochastique en finance. *Ecole Polytechnique Paris, Département De Mathématiques Appliquées*. pp. 146 (2010)
- [3] Gatheral, J., Jusselin, P. & Rosenbaum, M. The quadratic rough Heston model and the joint SP 500/VIX smile calibration problem. *ArXiv Preprint ArXiv :2001.01789*. (2020)
- [4] Abi Jaber, E. The characteristic function of Gaussian stochastic volatility models : an analytic expression. *Finance And Stochastics*. 26, 733-769 (2022)
- [5] Lord, R. & Kahl, C. Why the rotation count algorithm works. (Tinbergen Institute Discussion Paper,2006)