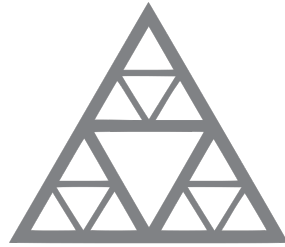


ÉCOLE NATIONALE DES PONTS ET CHAUSSÉES



École des Ponts

ParisTech

C++ : SLV Model

Ismaël BADAROU - Victor BELLISSANT - Raphaël LASRY - Ludovic QUILY

30 juin 2022

[GitHub](#)

1 Introduction

Le but de ce projet est de coder un pricer dit SLV (*Stochastic Local Volatility*), et de comparer ses performances sur un cas jouet avec un modèle d'Heston. Tous les points techniques liés à ce modèle ne seront pas re-détaillés dans ce rapport attendu qu'ils sont démontrés dans ce [papier](#). L'intégralité du code développé (C++) est en libre accès sur ce repos [GitHub](#).

Notre projet s'est construit en plusieurs étapes. Premièrement, nous avons repris le projet réalisé en Décembre qui traitait d'un modèle de Dupire (interpolation/extrapolation d'une surface de volatilité implicite), puis implémenté un modèle d'Heston, ainsi qu'un générateur de scénario via un schéma d'Euler. Ensuite, nous avons calculé une espérance conditionnelle pour finaliser la classe SLV. Dans un dernier temps nous avons travaillé sur une méthode pour inverser le modèle de Black et Scholes afin d'obtenir la volatilité à partir d'un prix de call donné.

Notre framework de test `test_implementation.cpp` s'articule donc de la manière suivante :

- Paramétrisation d'un modèle d'Heston,
- Détermination des prix des calls dans le modèle d'Heston,
- Estimation des volatilités Black-Scholes associées aux prix obtenus,
- Perturbation de la surface de volatilité par un bruit (normal ou uniforme),
- Détermination des prix des calls dans le modèle SLV,
- Visualisation des différences entre les deux pricers.

2 Le modèle d'Heston

Le modèle d'Heston est un modèle dit à volatilité stochastique. Contrairement au modèle de Black et Scholes, la volatilité de ce modèle à deux facteurs est elle même sujette à une équation différentielle stochastique. L'actif sous-jacent S_t suit dans ce modèle les équations suivantes :

$$\begin{cases} dS_t = \mu S_t dt + \sqrt{V_t} S_t dW_t^s \\ dV_t = \kappa(\theta - V_t)dt + \xi \sqrt{V_t} dW_t^v \\ d < W^s, W^v >_t = \rho dt \end{cases} \quad (1)$$

De plus, la condition de Feller impose que si $2\kappa\theta > \xi^2$, alors $V_t > 0$.

Une étude plus précise (et avec plus de moyens) se serait déroulée ainsi :

- Requête d'un set de call pour des maturités et strikes donnés sur un actif liquide (type CAC40 ou S&P500),
 - Calibration du modèle d'Heston,
 - Inversion des prix obtenus dans ce modèle pour obtenir la matrice de volatilité associée (voir 4.1)
- Néanmoins, n'ayant pas de données de marché, et ne sachant pas comment calibrer ce modèle, nous proposons de fixer les paramètres, et de simuler des prix via une méthode de Monte-Carlo.

μ	κ	θ	ξ	ρ
1.00	2.00	0.01	0.10	0.00

TABLE 1 – Paramètres du modèle d'Heston

3 Le modèle SLV

Comme son nom l'indique, le modèle SLV est lui aussi un modèle à volatilité stochastique. Dans ce modèle, le sous-jacent S_t est mû par les équations différentielles stochastiques suivantes :

$$\begin{cases} dS_t = \mu S_t dt + \sigma(t, S_t) \sqrt{V_t} S_t dW_t^s \\ dV_t = \kappa(\theta - V_t) dt + \xi \sqrt{V_t} dW_t^v \\ d \langle W^s, W^v \rangle_t = \rho dt \end{cases} \quad (2)$$

On remarque que la seule différence avec un modèle à volatilité stochastique comme celui d'Heston est l'introduction de la partie *locale* contrôlée par $\sigma(t, S_t)$. Ainsi, la principale difficulté de ce projet réside dans le calcul de cette dernière. L'article explicite la formule pour le calcul de $\sigma(t, S_t)$:

$$\sigma^2(t, K) = \frac{\sigma_{LV}^2(t, K)}{\mathbb{E}[V_t | S_t = K]} \quad (3)$$

3.1 Volatilité locale

Nous avons repris le travail effectué plus tôt dans l'année sur le modèle de Dupire (interpolation/extrapolation de la volatilité). On se donne une matrice de volatilité $(\sigma(T^i, K^j))$, et on cherche à calculer $\sigma_{LV}(T, K)$ quelconque. Les calculs détaillés dans le polycopié du cours aboutissent aux formules suivantes :

Interpolation sur les strikes : $K^1 \leq K \leq K^n$

On définit S_j la fonction d'interpolation cubique telle que :

$$S_j(x) = \alpha_j(x - x_j)^3 + \beta_j(x - x_j)^2 + \gamma_j(x - x_j) + \delta_j$$

On ne re-détaille pas l'algorithme de Thomas qui permet d'obtenir les coefficients $\alpha_j, \beta_j, \gamma_j$ et δ_j . On obtient la volatilité locale alors ainsi :

$$\sigma_{LV}(T^i, K) = S_j(K) \quad \text{avec } K \in [K^j, K^{j+1}] \quad (4)$$

Extrapolation sur les strikes : $K < K^1$ ou $K > K^n$

On introduit les grandeurs suivantes

$$\begin{cases} D_L = \gamma_1 \\ D_R = 3\alpha_{N-1} (K^N - K^{N-1})^2 + 2\beta_{N-1} (K^N - K^{N-1}) + \gamma_{N-1} \end{cases}$$

Les formules d'extrapolation donnent $\sigma_{LV}(T^i, K)$:

$$\begin{cases} \sigma(T^i, K^1) + D_L (K - K^1) & \text{si } K < K^1 \\ \sigma(T^i, K^n) + D_R (K - K^n) & \text{si } K > K^n \end{cases} \quad (5)$$

Interpolation sur les maturités : $T^1 \leq T \leq T^m$

On définit les grandeurs $K^{(i)}$ comme :

$$K^{(i)} = K \int_0^{T^i - T} r_s ds$$

L'interpolation sur les maturités se fait alors via la formule suivante :

$$\sigma_{\text{LV}}(T, K) = \sqrt{\frac{1}{T} \left(\sigma_{\text{LV}}^2(T^i, K^{(i)}) T^i + \frac{\sigma_{\text{LV}}^2(T^{i+1}, K^{(i+1)}) T^{i+1} - \sigma_{\text{LV}}^2(T^i, K^{(i)}) T^i}{T^{i+1} - T^i} (T - T^i) \right)} \quad (6)$$

Extrapolation sur les maturités : $T < T^1$ ou $T > T^m$

On reprend les notations introduites juste au dessus, $\sigma_{\text{LV}}(T, K)$ vaut alors :

$$\begin{cases} \sigma_{\text{LV}}(T^1, K^{(1)}) & \text{si } T < T^1 \\ \sigma_{\text{LV}}(T^m, K^{(m)}) & \text{si } T > T^m \end{cases} \quad (7)$$

3.2 Espérance conditionnelle

On reprend ici les résultats principaux développés dans l'article pour l'estimation de $\mathbb{E}[V_t | S_t = K]$. Étant donné que l'on effectue des simulations Monte-Carlo des paires $(s_{i,j}, v_{i,j})$, on a nécessairement $\mathbb{E}[V_{t_i} | S_{t_i} = s_{i,j}] = v_{i,j}$. Pour y remédier, on va chercher à regrouper les paires de réalisations. On choisit donc l bins disjointes $(b_1, b_2], (b_2, b_3] \dots, (b_l, b_{l+1}]$ qui couvrent les valeurs prises par S . On effectue l'approximation suivante :

$$\mathbb{E}[V_{t_i} | S_{t_i} = s_{i,j}] \approx \mathbb{E}[V_{t_i} | S_{t_i} \in (b_k, b_{k+1}]] \quad (8)$$

Se pose alors la question du choix optimal des bins pour calculer cette espérance conditionnelle. Parmi les N simulations Monte-Carlo réalisées, on ordonne les valeurs des prix spot obtenues pour une time-step t_i tel que $\bar{s}_{i,1} \leq \bar{s}_{i,2} \leq \dots \leq \bar{s}_{i,N}$. On choisit alors les bornes des bins b_k telles qu'elles contiennent approximativement le même nombre de simulations Monte-Carlo :

$$b_{i,1} = \bar{s}_{i,1}, \quad \bar{s}_{i,l+1} = \bar{s}_{i,N}, \quad b_{i,k} = \bar{s}_{i, \frac{(k-1)N}{l}}, \quad \forall k \in \llbracket 2, l \rrbracket \quad (9)$$

Enfin, on obtient une expression de l'approximation de l'espérance conditionnelle :

$$\mathbb{E}[V_{t_i} | S_{t_i} = s_{i,j}] \approx \frac{l}{N} \sum_{j \in \mathcal{J}_{i,k}} v_{i,j} \quad (10)$$

Avec $\mathcal{J}_{i,k} = \{j \mid (s_{i,j}, v_{i,j}) \in \mathcal{B}_k\}$, et \mathcal{B}_k est la $k^{\text{ième}}$ bin.

4 Résultats et comparaison avec le modèle d'Heston

4.1 Inversion du modèle de Black-Scholes

Le but de cette méthode est de partir d'un prix de call donné (C^*), d'un strike (K), d'une maturité (T), d'un taux sans risque (r) et d'un prix de sous-jacent initial (S_0) ; et de retrouver la volatilité (σ^*) telle que le modèle de Black Scholes redonne le prix du call C^* .

On rappelle que dans le modèle de Black-Scholes, le prix du call est donné par la formule :

$$\begin{cases} C = S_0 \mathcal{N}(d_1) - K e^{-rT} \mathcal{N}(d_2) \\ \mathcal{N}(x) = \int_{-\infty}^x \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}t^2} dt \\ d_1 = \frac{1}{\sigma\sqrt{T}} \left[\ln\left(\frac{S_0}{K}\right) + \left(r + \frac{1}{2}\sigma^2\right) T \right] \\ d_2 = d_1 - \sigma\sqrt{T} \end{cases} \quad (11)$$

On rappelle aussi l'expression d'une des grecques dans ce modèle, le *véga* (ν), qui mesure la sensibilité du prix du call aux variations de la volatilité :

$$\begin{cases} \nu = S_0 \sqrt{T} \mathcal{N}'(d_1) \\ \mathcal{N}'(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} \end{cases} \quad (12)$$

On peut alors mettre en place la méthode de Newton-Raphson, détaillée dans ce [papier](#), résumée dans le pseudo-code suivant :

Algorithm 1: Méthode de Newton-Raphson pour l'estimation de σ^* dans le modèle de Black-Scholes

Data: $C^*, K, T, r, S_0, \epsilon > 0$

Result: σ such that $|\sigma^* - \sigma| < \epsilon$

$\tilde{\sigma} \leftarrow \sqrt{\frac{2\pi}{T} \frac{C^*}{S_0}};$

$d_1 \leftarrow \frac{1}{\tilde{\sigma}\sqrt{T}} \left[\ln\left(\frac{S_0}{K}\right) + \left(r + \frac{1}{2}\tilde{\sigma}^2\right) T \right];$

$d_2 \leftarrow d_1 - \tilde{\sigma}\sqrt{T};$

$C \leftarrow S_0 \mathcal{N}(d_1) - K e^{-rT} \mathcal{N}(d_2);$

$\nu = S_0 \sqrt{T} \mathcal{N}'(d_1);$

$\sigma \leftarrow \tilde{\sigma} + \frac{C - C^*}{\nu};$

while $|\sigma - \tilde{\sigma}| > \epsilon$ **do**

$\tilde{\sigma} \leftarrow \sigma;$

$d_1 \leftarrow \frac{1}{\tilde{\sigma}\sqrt{T}} \left[\ln\left(\frac{S_0}{K}\right) + \left(r + \frac{1}{2}\tilde{\sigma}^2\right) T \right];$

$d_2 \leftarrow d_1 - \tilde{\sigma}\sqrt{T};$

$C \leftarrow S_0 \mathcal{N}(d_1) - K e^{-rT} \mathcal{N}(d_2);$

$\nu = S_0 \sqrt{T} \mathcal{N}'(d_1);$

$\sigma \leftarrow \tilde{\sigma} + \frac{C - C^*}{\nu};$

end

Notons que cet algorithme converge en généralement très rapidement, mais des problèmes numériques peuvent apparaître quand le *véga* devient nul. Une courbe probable de cette grecque est la suivante :

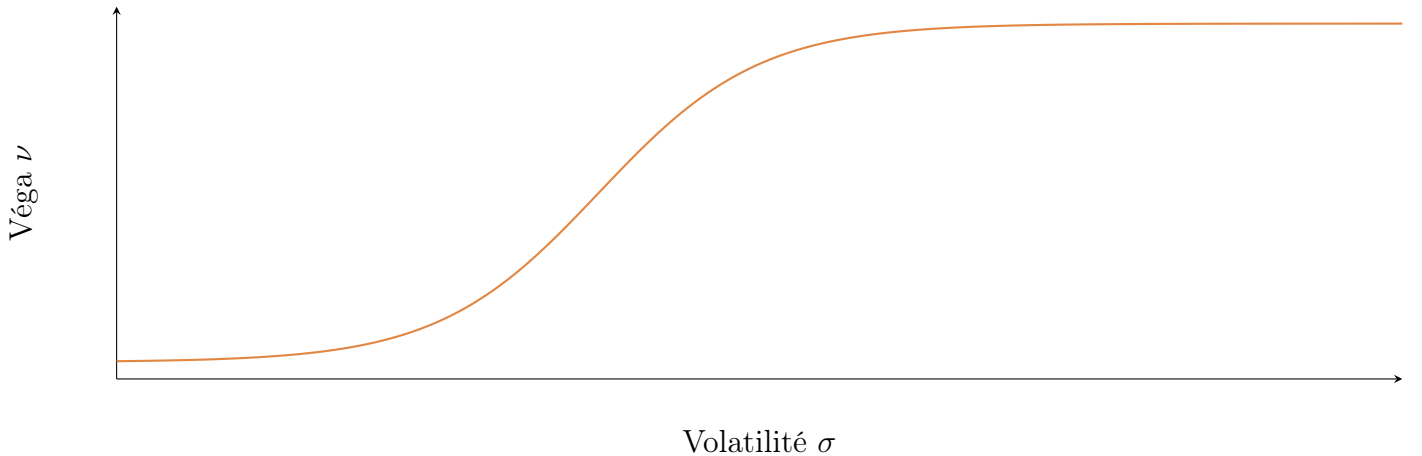


FIGURE 1 – Allure générique du *véga* dans le modèle de Black-Scholes en fonction de la volatilité σ

Il apparaît ainsi que pour les valeurs trop petites de σ , le *véga* s'annule et la méthode de Newton-Raphson ne converge plus.

4.2 Performances relatives

Nous avons comparé notre modèle SLV au modèle d'Heston que nous avons implémenté avec les paramètres décrits en TABLE 1. Nous avons premièrement obtenu les prix des calls dans le modèle d'Heston, inversé la formule de Black-Scholes pour obtenir la volatilité implicite, puis bruité cette même matrice et utilisé le pricer SLV pour tenter de retrouver des prix assez proches.

On présente ici plusieurs tests que nous avons réalisés. L'idée est que cet exercice est artificiel puisque nous ne comparons pas réellement un modèle d'Heston avec notre pricer SLV (les données sont générées via ce même modèle d'Heston). Les bruits ajoutés aux résultats fournis par le modèle d'Heston permettent ainsi de simuler les limites de ce modèle et les améliorations apportées par l'aspect *local* de la volatilité du modèle SLV. Dans les simulations qui suivent, les runs ont été effectués avec les paramètres suivants : 20 bins, 10^4 runs Monte-Carlo et une discretisation temporelle de 100 points.

4.2.1 Bruit uniforme

Pour ce premier test, nous ajoutons le même bruit à tous les coefficients de la matrice de volatilité obtenue avec le modèle d'Heston.

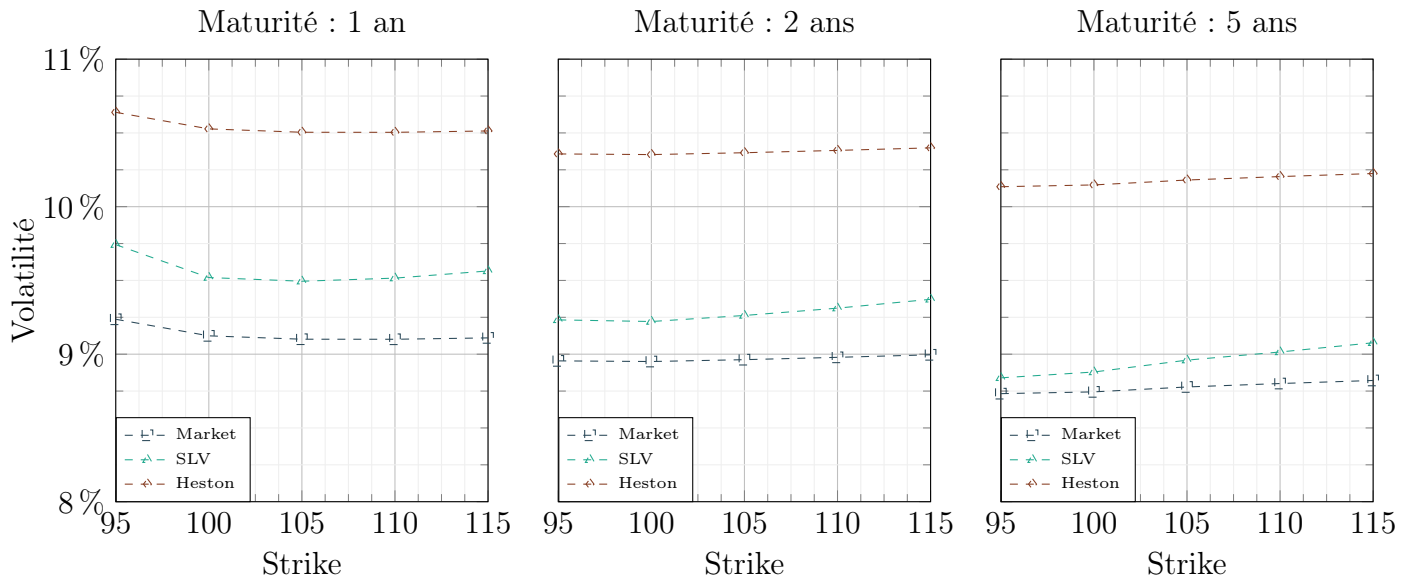


FIGURE 2 – Smile de volatilité avec bruit uniforme

On voit sur ces différentes figures que le modèle SLV nous permet de rattraper le bruit ajouté sur la volatilité, l'écart entre la volatilité du modèle SLV et la volatilité du marché est plus faible que l'écart de volatilité entre le marché et le modèle d'Heston. La méthode de simulation SLV nous permet donc d'améliorer un modèle d'Heston pré-calibré pour mieux correspondre aux données de marchés.

4.2.2 Bruit normal

Le coefficient ajouté à la matrice des volatilités calculé avec le modèle d'Heston n'est plus le même pour chaque coefficient. Nous avons ajouté un bruit gaussien de moyenne nulle et de variance 1%. On perd alors légèrement en cohérence puisque nos volatilités ne suivent plus aucune monotonie, mais on test ainsi la robustesse du modèle SLV à retrouver divers prix.

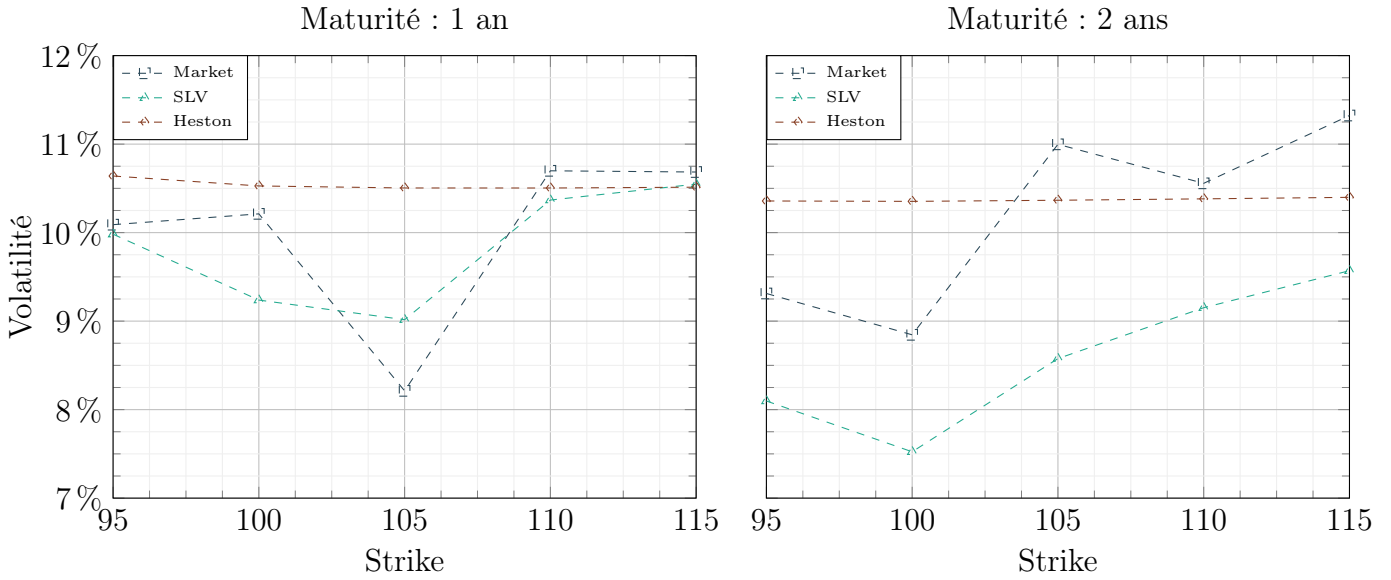


FIGURE 3 – Smile de volatility avec bruit non uniforme

On observe sur le graphe de gauche que le modèle SLV arrive à s'adapter au smile de la volatilité du marché. Sur la figure avec maturité deux ans, le profil de volatilité obtenu avec le modèle SLV correspond mieux au profil des volatilités de marché, mais l'écart est plus grand qu'avec le modèle d'Heston classique pour certaines volatilités.

4.2.3 Absence de bruit

Dans ce test nous regardons ce qu'il se passe lorsque nous ne modifions pas la matrice des volatilités obtenue avec le modèle d'Heston (Market = Heston).

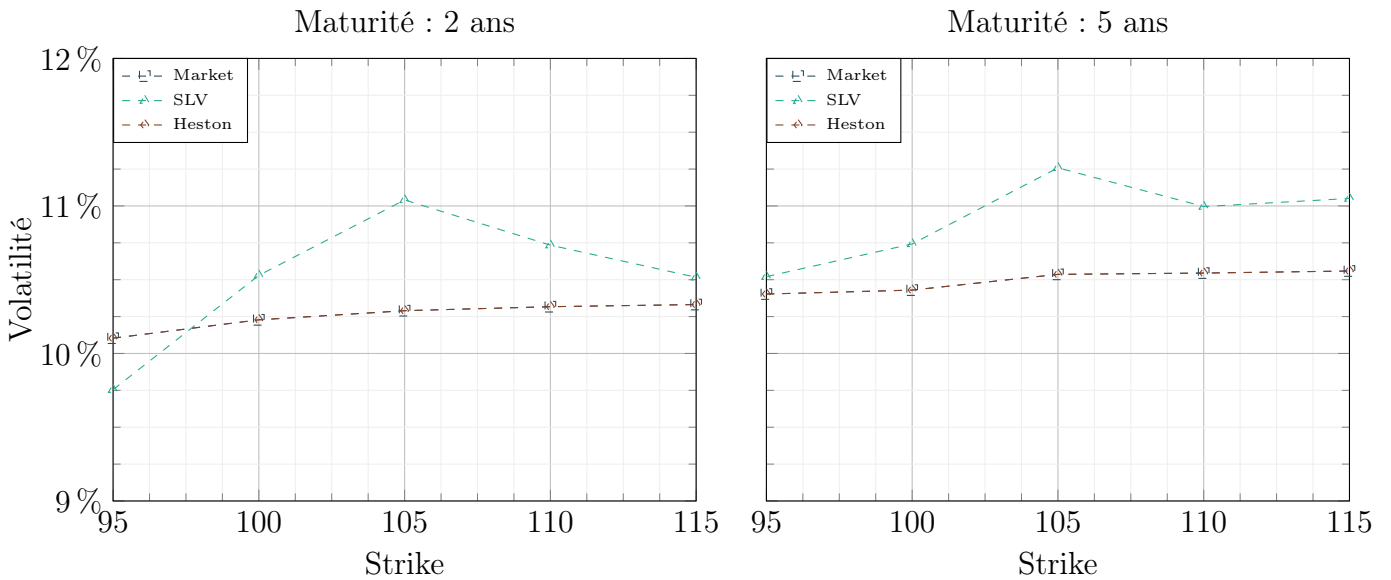


FIGURE 4 – Smile de volatility (Sans bruit)

Ce test nous permet de vérifier que si la matrice de volatilité obtenue avec un modèle d'Heston n'est pas modifiée, l'ajout de la partie volatilité locale ne nuira pas à la calibration. Notre implémentation ne répond pas à ce test car le profil de volatilité obtenu avec le modèle SLV n'est pas le même que

celui obtenu avec le modèle d'Heston classique. De plus, nous avons un écart de volatilité pouvant aller jusqu'à 1% entre les deux modèles. Notre implémentation des simulations Monte-Carlo du modèle SLV semble donc perfectible.

5 Conclusion

Dans ce projet nous avons implémenté un modèle d'Heston ainsi qu'un pricer SLV. Ces deux modèles, bien que proches sur le papier diffèrent uniquement par l'ajout d'une volatilité locale dans le second. Cela entraîne quelques complications techniques liées premièrement à l'estimation de cette volatilité locale (interpolation/extrapolation de Dupire), et ensuite via l'estimation d'une espérance conditionnelle. On opte alors pour une approximation, asymptotiquement exacte, basée sur le regroupement des valeurs simulées en intervalles.

Une étude plus poussée aurait permis de collecter des données de marché, de calibrer le modèle d'Heston sur ces dernières et de comparer l'efficacité relative de l'ajout d'une volatilité locale. Nous avons tout de même pu constater que le pricer implémenté parvient correctement à retrouver les volatilités de marché (celles simulées puis bruitées).