HEC
PARIS

Major in International Finance

MASTER RESEARCH PAPER

Academic Year 2024–2025

Calibrating a Singular Local Stochastic Volatility Model using Gaussian Process Regression

Davi Méaille

Under the supervision of Prof. Ahmed Kebaier

President of the Jury: Prof. Tjeerd De Vries

Keywords: Volatility Surface, Local Stochastic Volatility Model, McKean-Vlasov Processes, Gaussian Process Regression

✓ Public report
❑ Confidential report

Abstract

Recent advances are incorporating machine learning techniques into pricing techniques. In particular, we use the framework developed by Bayer et al. [2024] together with Gaussian process regression in order to calibrate a singular local stochastic volatility model. We show that using Gaussian process regressions allows us to develop a confidence interval around our estimated volatility surface, leading to our concept of "confidence surface volatility".

Contents

1 Introduction **3**

2 Literature Review **6**

3 Framework from Bayer et al. [2024] **6**

4 Gaussian Process Regressions **9**

5 Algorithm Description **13**

6 Results **15**

7 Further Comments **18**

8 Conclusion **19**

9 Appendix **20**

A Dupire Formula Derivation 20

B Gaussian Process Regression 22

C Backward Kolmogorov and Fokker-Planck Equations 22

D Supplementary Material 23

1 Introduction

Recent development in finance and computational finance are aiming at utilizing machine learning in forecasting or model calibration. Among the several technics used, Gaussian Process regression, first introduced by Williams and Rasmussen [2006], can be relevant as they provide confidence intervals, thus allowing to measure the uncertainty of the calibration or the estimation.

One particular area of interest concerns the calibration of the volatility surface and the possibility to generate a smile that doesn't vanish with time. To describe it briefly, the volatility surface gives, for each spot and maturity, the correct volatility to plug in the standard Black-Scholes formula in order to retrieve the market price of the option (here calls and puts). This is an efficient work around to go beyond the limitations of the assumption of constant volatility. For more precise developments, see Gatheral [2011]. This in in particular important in the equity world, where the smile, that is the implied volatility for a given maturity, is asymmetrical, that is is skewed on the left., thus causing higher prices of far out-of-the money (OTM) puts than for OTM calls.

A first break through in calibrating the volatility surface came from Dupire et al. [1994], who provided a formula in order to calibrate a local volatility model. As a reminder, the local volatility model assumes that volatility is a function of both the spot and the time, and that the underlying evolves according to:

$$\frac{dX_t}{X_t} = rdt + \sigma(X_t, t)dW_t \tag{1}$$

where r is the risk-free rate and W_t is a Brownian motion. Starting from the price C of a call with strike K and maturity T, $C(,T) = \int_0^\infty \max(S-K,0)\phi_T(S)dS$, where $\phi_T(S)$ is the risk-neutral probability associated with the spot S and assuming $r = 0$ for simplicity, the author derives the implied formula:

$$\frac{\partial C}{\partial T} = \frac{\sigma^2 K^2}{2}\frac{\partial^2 C}{\partial K^2} \tag{2}$$

$$\iff \sigma^2_{\text{Dupire}} = \frac{\dfrac{\partial C}{\partial T}}{\dfrac{1}{2}K^2\dfrac{\partial^2 C}{\partial K^2}} \tag{3}$$

See Appendix A for more detail on the derivation.

As is well known in the finance literature (see for instance Henrotte [2003]), even though the Dupire formula allows to replicate the prices available on the market for out-of-the-money (OTM) and in-the-money (ITM) calls and puts, on which it is calibrated, it lacks in its predictions for future smile patterns and can't generate reasonable prices for exotic products. As such, several other models have been proposed to generate a smile using some parameters, that are to be calibrated to match the market prices.

One example of such alternative models is to use Poisson diffusion instead of simply Brownian diffusion (see for instance Derman [2003] for a pedagogical approach to the topic). Poisson jumps and jump to default can be used to generate asymmetric and sizable jumps that are useful when generating a short and long-term smile. However, as highlighted in Dupire et al. [1994], introducing non-traded sources of risk (jumps, stochastic volatility) leads to the incompleteness of the model and thus the ability to hedge using traded instruments, as there are not liquid instruments to hedge against jumps.

Another alternative is stochastic volatility, that consists in adding a coupled process for volatility on top of the process for the underlying, with:

$$dX_t = rX_t dt + \sqrt{\nu_t} X_t dW_t^X \tag{4}$$

$$d\nu_t = \alpha(X_t, \nu_t, t)dt + \eta\beta(X_t, \nu_t, t)\sqrt{\nu_t} dW_t^\nu \tag{5}$$

with correlation between the two processes $\langle dW_t^X, dW_t^\nu \rangle = \rho$ (Gatheral [2011]). This proves to be a good diffusion model to generate a realistic diffusion and manages to replicate realistic volatility surfaces. It is also capable of replicating asymmetric volatility smiles, based on the sign and the magnitude of the correlation between stock and volatility. For instance, for equity, it is done by setting the correlation $\rho < 0$, thus creating periods of higher volatility with negative returns for the underlying and on the contrary periods of lower volatility with positive returns for the underlying, which is a reason for the asymmetric smile in equity. Another advantage of stochastic volatility is the more and more liquid market for volatility swap (Gross [2004]), allowing for better replication and hedging.

However, one last issue remaining is the difficulty to calibrate stochastic volatility models, as they need a lot of parameters. A simple example is the Heston model for stochastic volatility,

where the volatility follows an Ornstein-Uhlenbeck process, according to:

$$dX_t = \mu X_t dt + \sqrt{\nu_t} X_t dW_t^X,$$
$$d\nu_t = \kappa(\theta - \nu_t)dt + \zeta\sqrt{\nu_t} dW_t^\nu$$

where κ is the speed of the mean reversion, θ is the long term mean variance, ζ is the volatility of the volatility "vol of vol" and ρ is the correlation of the two Wiener processes (see Heston [1993]).

In this context, new types of models are combining the advantages of local volatility, that is the easy calibration, as well as the advantages of stochastic volatility, that is realistic diffusion and ability to generate a smile and a volatility surface. In particular, models such as Bayer et al. [2024] that we will study more extensively in Section 3 have the important feature that they manage to reproduce a given dataset volatility surface thanks to their local volatility component, while maintaining a realistic diffusion due to their stochastic components. This is the case, even though the parameters of the stochastic part may be quite far off the ones of the real diffusion. This result is possible in our case of interest thanks to the Markovian projection, that we will develop a bit later (See Equation 7).

Following Bayer et al. [2024], we propose to use machine learning techniques to calibrate one specific type of local stochastic volatility model. Our novelty is to use Gaussian Process Regressions (GPR) in the calibrating framework. This has three main advantages:

- GPR operates well under small datasets
- GPR uses kernels that are adaptable to the priors on the functions to be learned (see Duvenaud [2014])
- GPR is well known for their ability to estimate not only the mean of their predictions, but also the variance, thus providing a confidence interval to have an idea of the error rate around the prediction.

See Section 4 for more details about how GPR operates.

The main drawback we are going to face is the computational load t=when computing the Gram matrix $K(X^*, X^*)$ in Equation 18. Williams and Rasmussen [2006] Addresses some potential computational improvements that we leave for further development.

We now turn to a short literature review followed by a detailed explanation of the paper that we are building upon, that is Bayer et al. [2024].

2 Literature Review

Local stochastic volatility models are becoming more and more studied. For instance, Cuchiero et al. [2020] calibrate directly a leverage function using market options prices instead of using the interpolation method from the Dupire's formula, using neural network, in order to simulate the diffusion and establish that they manage to reproduce the volatility surface. Another practice, which the paper Bayer et al. [2024] takes its inspiration from, comes from the inversion of the Markovian projection of the volatility to enrich the Dupire's formula. One example of such paper is given by Lacker et al. [2020] or Henry-Labordère [2009], that requires to estimate numerically a conditional expectation term. Another example is the paper by Van der Stoep et al. [2014] where they develop a Monte-Carlo approach to simulate the diffusion, that doesn't rely on the Markovian projection and they learn the local component directly during the diffusion.

There are also some interesting applications of GPR to computational finance problems. For example, Han et al. [2016]develop a Gaussian process regression stochastic volatility (GPRSV) model to extend the capabilities of the stochastic volatility model to capture more time-varying dynamics of the financial market. Petelin et al. [2011] showed that GPR can be a good complement to more traditional Bayesian Vector Autoregression technics in estimationg and calibration in finance. Liu et al. [2020] use multi variate GPR to forecast volatility of FX returns, showing that it can reach great accuracy. Gonzalvez et al. [2019] show how GPR can be use to model the term structure of interest rates and to build trend following strategies. Some other applications may concern the estimation of the Greeks, after calibrating the GPR on a pricing function, as in Herfurth [2020]. A last example is given by Bank et al. [2023], that develops a methodology which derive a time-inhomogeneous Markov process and the adjoing Feynman-Kac equation to develop a rough partial differential equation, which is used for the pricing.

3 Framework from Bayer et al. [2024]

From now one, we assume $r = 0$ for simplicity. Note that assuming $r > 0$ will simply affect the Dupire calibration formula that we would have to chose, where the relevant term in the

Fokker Planck equation doesn't vanish (see Appendix A) as well as the Euler scheme for the numerical simulation of the diffusion process.

We will now delve into the paper by Bayer et al. [2024]. In their model, they study a stochastic volatility model of the form:

$$\frac{dX_t}{X_t} = \sqrt{\nu_t}\sigma(X_t, t)dW_t \tag{6}$$

combining thus the stochastic volatility component $\sqrt{\nu_t}$ and the local volatility component $\sigma(X_t, t)$. They use another expression for the Dupire's local volatility formula, that is:

$$\sigma^2_{\text{Dupire}}(x, t) = \sigma^2(x, t)\mathbb{E}\left[\nu_t | X_t = x\right] \tag{7}$$

Please see Appendix A for more information in the derivation of the formula. This equation is the reason why the distribution our local stochastic volatility model manages to mimic the one of our data, even though the parameters of our stochastic volatility component may be far off. Indeed, this equation corresponds to the one dimensional Markovian projection of the distribution of our diffusion X_t into the distribution of our dataset (See Gyöngy [1986] for further details). As such, even though the overall diffusion may follow a different law, its one-dimensional distribution is, at each step, identical to the original distribution, explaining why we are capable of reproducing the observed data.

As a consequence, Equation 6 rewrites:

$$dX_t = \sigma_{\text{Dupire}}(X_t, t)X_t\frac{\sqrt{\nu_t}}{\sqrt{\mathbb{E}[\nu_t|X_t]}}dW_t \tag{8}$$

The particularity of this diffusion model comes from the term $\mathbb{E}[\nu_t|X_t]$, which will also justify the use of machine learning techniques. In particular, this term means that the diffusion of each underlying of our Monte-Carlo simulation will depend on a common term, that is computed using the entire distribution of underlying stocks. We will now refer to each of these underlyings as particles, following the literature on McKean-Vlasov equations.

So-called McKean-Vlasov partial differential equations come from the study of certain non-linear parabolic equations (see McKean Jr [1966]). Diffusion processes, through their Backward Kolmogorov and Fokker-Planck equations, are tightly linked to certain classes of partial

differential equations. For more details, please refer to Appendix C. The particularity of these McKean-Vlasov equations is that the diffusion of each particle features a term that depends on the overall distribution of all the particles. A simplest example could be for instance the following diffusion:

$$dX_t = \mathbb{E}[X_t]dt + \sigma dW_t \tag{9}$$

A general result (See Lacker [2018], **Lemma 3.2**) is that, for a system of N interacting particles $(X_t^{N,1}, X_t^{N,2}, \ldots, X_t^{N,N}$ to admit a strong solution, with:

$$dX_t^{N,i} = b(X_t^{N,i}, \mu_t^n)dt + \sigma(X_t^{N,i}, \mu_t^n)dW_t^i,\ X_0^{N,i} = Z^i \tag{10}$$

$$\mu_t^n = \frac{1}{N}\sum_i \delta_{X_t^{N,i}} \tag{11}$$

one typically needs that b and σ be Lipschitz (b following a standard Euclidian metric, σ following the Wasserstein $\mathcal{W}_2$ metric (See again Lacker [2018], Chapter 2). This is linked to the convergence of the particle scheme, when there are more and more particles, to the true distribution of the diffusion.

In our case of interest, however, Bayer et al. [2024] study systems of the form:

$$dX_t = H\left(t, X_t, Y_t, \mathbb{E}[A_1(Y_t)|X_t]\right)dt + F\left(t, X_t, Y_t, \mathbb{E}[A_1(Y_t)|X_t]\right)dW_t^X \tag{12}$$

$$dY_t = b\left(t, Y_t\right)dt + \sigma\left(t, Y_t, \right)dW_t^Y \tag{13}$$

The problem is that, as they argue, for $\mu_t = \text{Law}(X_t, Y_t)$, the following functional is not Lipschitz, even for A_i smooth enough:

$$(x, \mu_t) \to \mathbb{E}[A_i(Y_t)|X_t = x] \tag{14}$$

To circumvent this problem, they approximate the conditional expectation above using Reproducing Kernel Hilbert Spaces (RKHS) and Ridge regression. In particular, they replace the conditional expectation with the result from the following optimization problem:

$$m_A^\lambda(\cdot;\nu) := \arg\min_{f\in\mathcal{H}}\left(\int_{\mathcal{X}\times\mathcal{X}} |A(y) - f(x)|^2\nu(dx, dy) + \lambda\|f\|_{\mathcal{H}}^2\right) \tag{15}$$

This is a Hilbert generalization of the discrete case Ridge regression, searching on an infinite dimensional Hilbert space instead. The estimated function has the property of being smooth

enough to be Lipschitz and allows the numerical Euler scheme to converge and to replicate the volatility smile present in a given set of prices.

In the following, we briefly introduce Gaussian Process Regression (GPR) and we show how using GPR allows to reproduce the results obtained by Bayer et al. [2024] but also allows us to construct a "Confidence Volatility Surface", that informs us about the accuracy of the results of our pricing. This might be less relevant in the cases studied by Bayer et al. [2024], where they simulate the prices and thus have a very large number of grid points to simulate their volatility surface, but could be important in cases when ones uses sparse market data.

4 Gaussian Process Regressions

We follow Williams and Rasmussen [2006] for a brief presentation of GPR (See Appendix B for more details).

The main idea behind GPR is to give a distribution over possible functions that would fit the data points available. Using a prior over the shape of the function expected, one computes the conditional expectation over the set of new points, conditional on the couple of points and observations available.

Formally, from a set of points and observations $\{X_t, Y_t\}$, we want to find the value $\{Y_t^*\}$ expected on a set of points $\{X_t^*\}$. Theoretically, the number of test points $\{X_t^*\}$ could be infinite, hence the idea of estimating a distribution of functions conditional on observed points.

In the most simple case, this comes down to estimating the following model, using however the information of normality for the ε term to deduce the variance on top of the mean:

$$Y = X^T w + \varepsilon \tag{16}$$

We then have the possibility to enrich the data by projecting our dataset into a richer space, called the feature space. This is for instance done by replacing X in our model by $\phi(X)$, resulting in a new model: $Y = \phi(X)w + \varepsilon$ (we find our simplest model by setting $\phi(X) = X$).

This gives a conditional expectation for Y^* values for the points X^*:

$$Y^*|X, Y, X^* \sim \mathcal{N}\left(\phi(X^*)^T A^{-1}\phi(X)Y, \phi(X^*)^T A^{-1}\phi(X^*)\right) \tag{17}$$

where $A = \left[\Sigma_p^{-1} + \frac{\phi(X)\phi(X)^T}{\sigma_n^2}\right]$.

This is the well known kernel trick in Support Vector Machine field. An equivalent representation of doing so consists in using kernels, that allows to go from the feature space to the function space. Simply put, kernels are a practical way of computing inner products in richer spaces. This allows to represent angles and similarities between vectors in higher spaces and allows to capture distances with non-linearities. See again Appendix B for more details.

We then compute the posterior distribution for the predicted points $\{Y_t^*\}$, using the given kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$:

$$Y^*|X, Y, X^* \sim \mathcal{N}\left(K(X^*, X)\left[K(X, X) + \sigma_n^2 I_d\right]^{-1} Y, \Sigma\right) \tag{18}$$

with $\Sigma = K(X^*, X^*) - K(X^*, X)\left[K(X, X) + \sigma_n^2 I_d\right]^{-1} K(X, X^*)$.

The kernel traditionally used is the radial basis function kernel, defined as:

$$K_{\text{RBF}}(X, X') = \sigma^2 \exp\left(-\frac{1}{2}\frac{\|X - X'\|^2}{l^2}\right) \tag{19}$$

However, we prefer to use the Matérn covariance kernel, defined as:

$$K_{\text{Matérn},\nu}(X, X') = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\frac{\sqrt{2\nu}|X - X'|}{l}\right)^{\nu} K_\nu\left(\frac{\sqrt{2\nu}r}{l}\right) \tag{20}$$

with K_ν the modified Bessel function. The Matérn kernel (see Matérn [2013]) is chosen over the RBF kernel because the absence of the quadratic term in the exponential makes it numerically more robust over grids where points are very close to one another: The Euclidian distance is less robust numerically than the Manhattan distance. In our case with $\nu = 3/2$, we use:

$$K_{3/2} = \sigma^2\left(1 + \frac{\sqrt{3}|X - X'|}{l}\right)\exp\left(-\frac{\sqrt{3}|X - X'|}{l}\right) \tag{21}$$

Another advantage of the Matérn kernel vs the squared exponential is the fact that the latter projects onto a space of functions that are much more "sinusoïdal", thus creating some arbitrary "waves" at the edges of the estimated functions. Figure 1 shows the estimated conditional expectation when using both kernels. We see that the conditional expectation using the Matérn Kernel is closer to what we would expect a smile to look like, as it doesn't have unjustified waves at both ends.

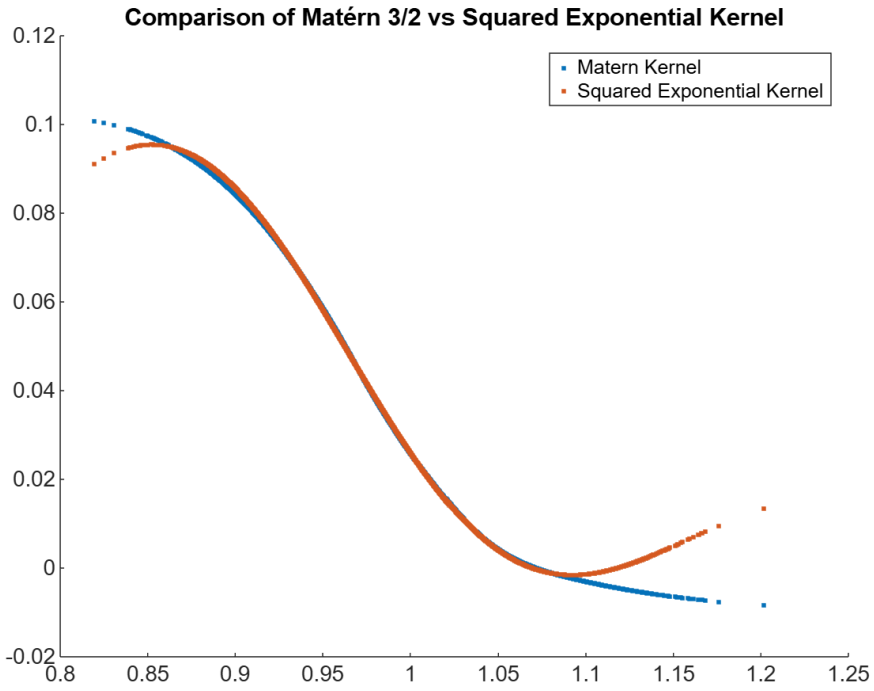

Figure 1: Comparison of Kernels

As we explained earlier, a strong argument in favor of using GPR rather than Ridge regression, despite their computational similarities and the gains in time when using Ridge regression, is the fact that the conditional expectation also gives us a confidence interval around our prediction (cf the Σ in Formula 18). As a matter of fact, this allows us to compute not only the predicted mean of our data points, but also some upper and lower bounds around them. For the explanation behind their analytical identity for the prediction of the Ridge regression and the mean prediction of the GPR, please see Kanagawa et al. [2018].

This is a point of interest to us, as we can translate this to our McKean-Vlasov diffusion in Equation 8, so that it ultimately yields confidence bounds over our calibrated volatility surface. We start with point-wise bounds, that is bounds for our estimation at each step t

and for each particle X_t^i.

To see how, let's denote $\mathcal{F}_t = \mathbb{E}[\nu_t|X_t]$. Our diffusion thus rewrites:

$$dX_t = \sigma_{\text{Dupire}}(X_t, t)X_t\frac{\sqrt{\nu_t}}{\sqrt{\mathcal{F}_t}}dW_t \tag{22}$$

and in particular, if we rewrite it in the discrete time version, under its Euler scheme:

$$X_{t+1} = X_t + \sigma_{\text{Dupire}}(X_t, t)X_t\frac{\sqrt{\nu_t}}{\sqrt{\mathcal{F}_t}}B_t \tag{23}$$

where B_t is a random walk term.

Because $\mathcal{F}_t$ is the result of our GPR, we have that $\mathcal{F}_t \sim \mathcal{N}(\mu_t, \sigma_t)$, where μ is the output of the classical Ridge regression or the mean prediction of the GPR and Σ denotes the uncertainty around μ in the GPR case. Since $\mathcal{F}_t$ is gaussian, we can denote $\mathcal{G}_t = \frac{1}{\sqrt{\mathcal{F}_t}}$, differentiable over $\mathbb{R}^{+*}$. Using the delta method on $\mathcal{G}_t$, since $\frac{\partial g(x)}{\partial x} = -\frac{1}{2}x^{-3/2}$ for $g(x) = \frac{1}{\sqrt{x}}$, we get that:

$$\mathcal{G}_t(\mathcal{F}_t) \sim \mathcal{N}\left(\mathcal{G}(\mu_t), \frac{\sigma^2}{4\mu^3}\right) \tag{24}$$

This allows us to compute some point-wise bounds, that is: the uncertainty delivered by our estimation at each step t and for each particle X_t^i.

Ignoring the randomness induced by the Brownian motion term, as the error of the estimation can be considered independent from it, we get the following bounds for X_t^i:

$$\begin{cases} X_{t+1}^{up} = X_t + X_t\left(\frac{1}{\sqrt{\mu}} + 1.96 * \sqrt{\frac{\sigma^2}{4\mu^3}}\right) \\ X_{t+1}^{low} = X_t + X_t\left(\frac{1}{\sqrt{\mu}} - 1.96 * \sqrt{\frac{\sigma^2}{4\mu^3}}\right) \end{cases} \tag{25}$$

We can also try to determine some path-wise accumulated errors, similar to a Kalman-style variance update for the error (See Kalman [1960], Simon [2006]). Indeed, considering that X_t is a random variable, although of a complex distribution, we can rewrite the diffusion in

the discretized version as:

$$X_t = X_{t-1}(1 + \mathcal{G}(\mathcal{F}_t))$$
$$\text{Var}(X_t) = \text{Var}(X_{t-1})\text{Var}(1 + \mathcal{G}(\mathcal{F}_t)) + \text{Var}(X_{t-1})\mathbb{E}[1 + \mathcal{G}(\mathcal{F}_t))]^2 + \text{Var}(1 + \mathcal{G}(\mathcal{F}_t))\mathbb{E}[X_{t-1}]$$

In both the point-wise and path-wise methods, one issue that arises is linked to the numerical stability of the computation of the variance of $\mathcal{G}(\mathcal{F}_t))$. Indeed, as we are predicting conditional expectation of the variance of a stock, this predicted variance can be very slow for very high value of the stock, due to their inverse correlation. As a consequence, even though we are capping the conditional expectation of the variance, as in Bayer et al. [2024], to a minimum of $1e-3$ in the diffusion scheme, this still creates issue in our error propagation estimations. This is due to the presence of the $\frac{1}{\sqrt{\mu^3}}$ term in the component of the variance due to $\mathcal{G}$ that can become extremely large for very small values of μ, linked to the fact that $x \mapsto \frac{1}{\sqrt{x}}$ is not differentiable at $x = 0$. Thus, here as well, we will cap the minimum value that μ^3 can take in order no to get values of the variance that would be too large and would throw the computations off.

5 Algorithm Description

The algorithm follows closely the one described in Bayer et al. [2024], except that we use GPR instead of a Ridge Regression.

We test different specifications of prices that yield similar results with regards to the quality of the approximation of the volatility surface. We also use two ways to generate prices: Either using the Black-Scholes formula for $\sigma = 0.3$ or the Heston pricing formula with Fourier approximation with a varying set of parameters for the initial and mean variance, the volatility of volatility and speed of mean-reversion.
We first use the author's calibration in Bayer et al. [2024] to show that we manage to reproduce their results but then move on to some arbitrary calibration.

We initialize the diffusion with $X_0 = 1$. We keep a strongly negative correlation between volatility and spot levels in all cases, in order to mimic the important observable feature of the equity world of strongly opposite correlations between spot movements and volatility of a stock. Other parameters' values are detailed in Table 1. Again, we have first the parameters

Parameter	Value	Description
r	0	Interest rate
S_0	1.0	Initial spot
v_0	0.0045	Initial Volatility
σ	0.3	Volatility for Black-Scholes
κ	2.19	Speed of mean-reversion
θ	0.17023	Long-term mean
ζ	1.04	Volatility of volatility
ρ	-0.83	Correlation
Diffusion Parameters		
X_0	1.0	Initial Spot
Y_0	0.0144	Initial Volatility
μ	0.0144	Long-term mean
λ	1	Speed of mean-reversion
η	0.5751	Volatility of volatility
ρ_{sto}	-0.9	Correlation
`cap`	$1e-3$	Cap on minimal value for $\mathbb{E}[\nu_t \vert X_t]$

Table 1: Author's Calibration

of the true Heston diffusion and second the parameters of the one we use within our Local Stochastic volatility diffusion.

We also use the aforementioned Dupire's formula 3 in order to calibrate the model, once the prices are generated.

We then proceed as follows. At each timestep t, we first compute the conditional expectation for the variance, using the variance at t and the spots at $t-1$. Since the conditional expectation doesn't change much from one time step to the next, it is numerically accurate to also use the variance at time $t-1$.

Then, we interpolate the current value of the local volatility using our pre-computed grid to retrieve σ_{Dupire} and we compute X_t according to Formula 6.

Our addition consists in including a measure of the uncertainty of our resulting volatility surface. We proceed with two different approaches.

A first possibility in simply to account for the uncertainty point-wise, that is to compute

upper bounds and lower bounds at each step, and let it move onto the volatility surface computation.

When choosing this option, we perform the computation detailed in Equation 25. A point of caution: As the term interacts with the Brownian motion sign in the discretized Euler scheme, we add in the simulation the following specifications:

$$\begin{cases} X_{t+1}^{\text{upper bound}} = \max(X_{t+1}^{\text{up}}, X_{t+1}^{\text{low}}) \\ X_{t+1}^{\text{lower bound}} = \min(X_{t+1}^{\text{up}}, X_{t+1}^{\text{low}}) \end{cases} \tag{26}$$

A second possibility is to try and accumulate this uncertainty over time. At each step, in order to carry forward the uncertainty and account for the compounding effect of uncertainty, we need to account for this accumulation of uncertainty. We propose to do so using some intuitions behind the Kalman filter (see Kalman [1960] and Simon [2006] for more details on the calculations of the Kalman filter).

6 Results

We display below our results to several cases, first using a Black Scholes pricer, then a Heston pricer (thus following Bayer et al. [2024]). We then compare the algorithm used in Bayer et al. [2024] to ours for sparse data grids.

We first start by comparing our results for the Black-Scholes and the Heston simulations. We can obesrve in Figure 2 that our volatility surface calibrations are more accurate for longer maturities.

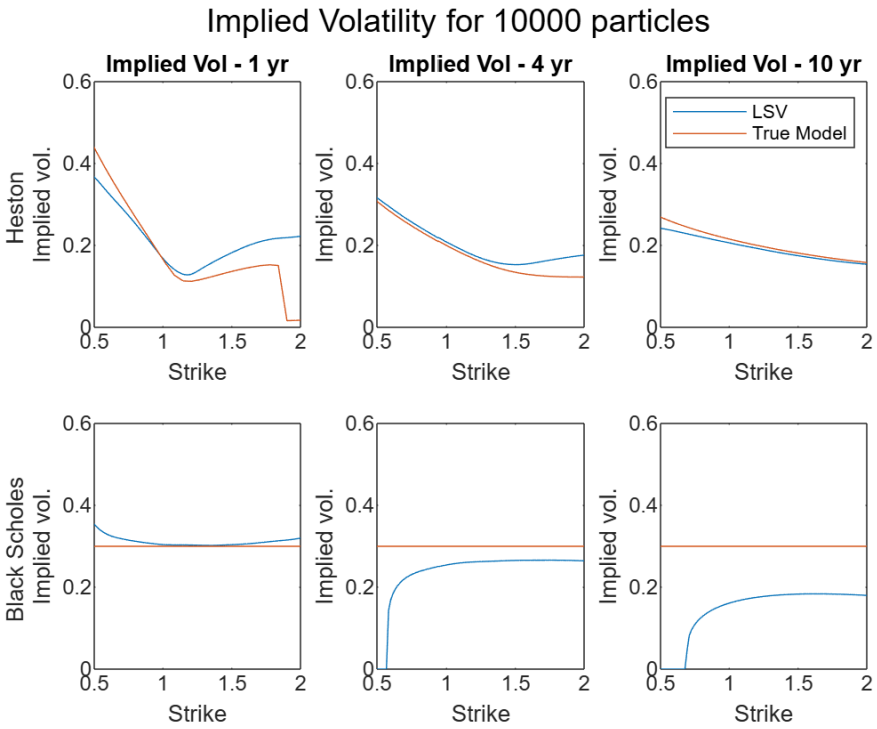

Figure 2: Comparison of performance to replicate the prices from Heston vs Black-Scholes

We can then observe in Figure 3 that we get very similar results to what Bayer et al. [2024] obtain. This is mainly due to the proximity between Ridge regression on Hilbert Spaces and GPR when it comes to the estimation of the mean. The observable differences come in large part because we replicated Bayer et al. [2024] faithfully, leaving a hyperparameter for the gaussian kernel fixed at $\sigma_f = 0.1$, whereas we are constantly optimizing when running the GPR regression.

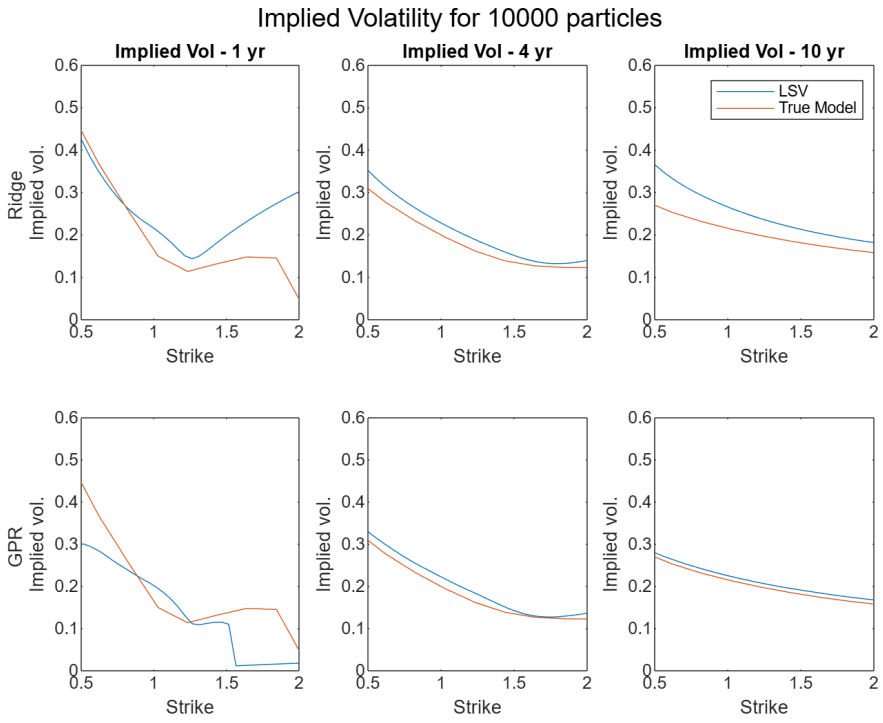

Figure 3: Comparison between Ridge and GPR methods for Smile calibration

Additionally, we observe that we manage to get some point wise upper and lower confidence bounds for the volatility surface, as in Figure 4. This is an important improvement on top of what Bayer et al. [2024] manage to obtain. It is also interesting to note that there is an apparent systematic upward bias in the estimations of the smile. This is only due to the too small number of particles used to simulate. Indeed, when we use up to 100,000 particles in the Ridge scheme, we obtain very close results. The advantage of Bayer et al. [2024] being that they compute no large a Gram matrix than $100 \times 100,000$, we managed to reproduce these results. In our GPR case, we didn't implement optimization method such as Nyström approximation of SUbset selections and thus our computational capabilities were limited by the large Gram matrix of size $N \times N$, with N the number of particles. We restricted ourselves to only computing up to 10,000 particles, explaining the more approximate results in the Heston case.

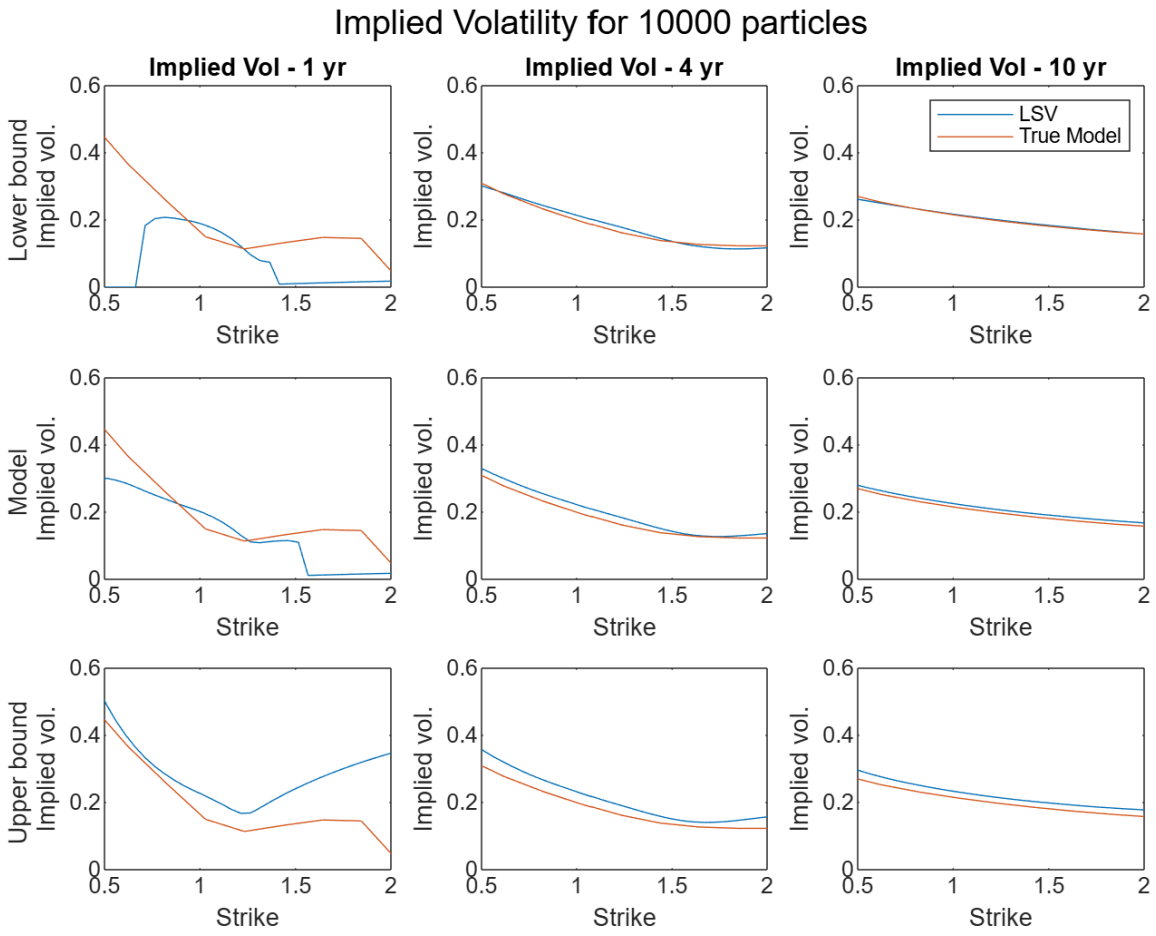

Figure 4: Point wise Lower and Upper bounds with GPR

7 Further Comments

We can see that the point-wise uncertainty sometime doesn't comprise the true smile. This is mainly due to numerical limits that we face, making it hard to go to 100,000 particles, as would have been necessary.

Our result for the path-wise accumulated error aren't very conclusive yet. We display below in Figure 5 the computed volatility surface. We didn't manage to circumvent the numerical issue of the uncertainty becoming tremendously large in the path-wise case and we thus leave that for further research.

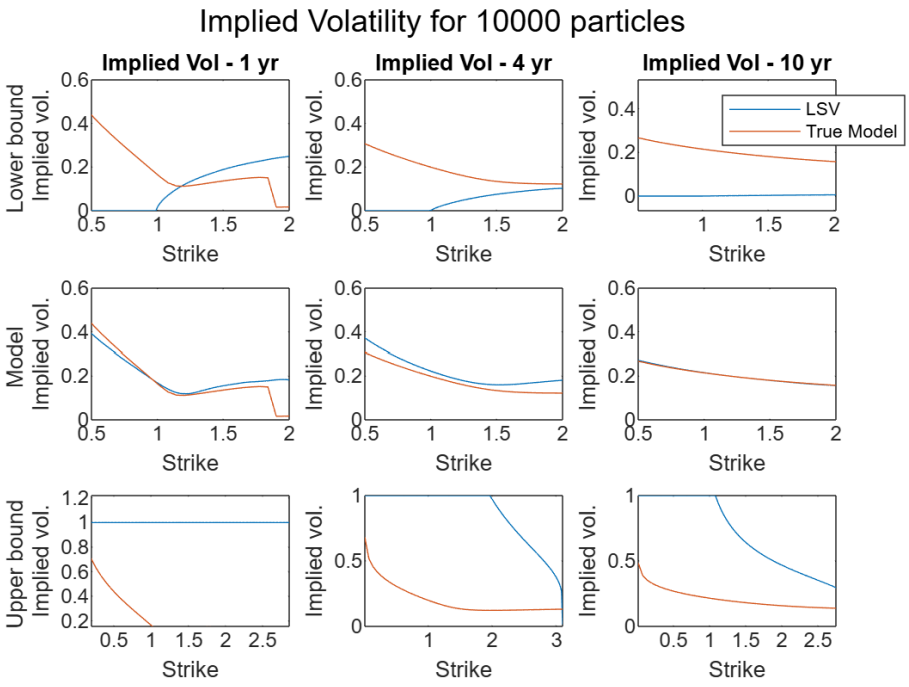

Figure 5: Accumulated Uncertainty Calibration

The main contribution of the article will thus be mainly the pointwise version of the uncertainty with the GPR implementation.

8 Conclusion

GPR are a very versatile and useful techniques in machine learning, with interesting applications in finance. We have developed its use in the context of a Local Stochastic Volatility model to estimate a conditional expectation term. This conditional expectation terms renders the corresponding McKean-Vlasov diffusion analytically singular, but using kernel methods such as Ridge in Bayer et al. [2024] or GPR in our current paper allows to circumvent this difficulty.

We first showed that we managed to reproduce the original authors' results using GPR instead of Ridge regression. This is standard, as both method are equivalent when considering Reproducing Kernel Hilbert Space (Kanagawa et al. [2018]). We get better results with our algorithms are we reoptimize the hyperparameters at each steps of the diffusion, unlike the original paper.

We then move on to show that, utilizing the covariance predicted by the GPR, we can build upper and lower bounds for the estimated volatility surface. This can be very useful, in contexts where the uncertainty is larger.

9 Appendix

A Dupire Formula Derivation

We derive here the Dupire formula, that we use extensively in our code. For more reference, see Dupire et al. [1994], Gatheral [2011].

Assume the stock S_t diffuses with risk-neutral drift $\mu = r = 0$:

$$\frac{dS_t}{S_t} = \sigma(S_t, t)dW_t$$

This yield the undiscounted risk-neutral value $C(S_0, K, T)$ of a call:

$$C(S_0, K, T) = \int_K^\infty (S_T - K)\phi(S_t, T; S_0)dS_T$$

where we weight the payoff $S_T - K$ when $S_T > 0$ by their risk-neutral probabilities $\phi(S_t, T; S_0)$. These risk-neutral probabilities are the state prices derived from the market and are what we are interested in in order to calibrate our model, as they give how much the market prices each future state for the stock S_T. To retrieve them, do the following derivations:

$$\begin{aligned}
\frac{\partial C}{\partial K} &= -\int_K^\infty \phi(S_t, T; S_0)dS_T = \Phi(S_t, T; S_0) - 1 \\
\frac{\partial^2 C}{\partial K^2} &= \phi(S_t, T; S_0)
\end{aligned}$$

We also know that $\phi(S_t, T; S_0)$ evolves according to the Fokker-Planck equation of the diffusion, as they express the distribution of the diffusion over time, and thus:

$$\frac{\partial \phi}{\partial t} = \frac{1}{2}\frac{\partial^2}{\partial S_T^2}(\sigma S_T^2 \phi) - S\frac{\partial}{\partial S_T}(rS_T\phi) = \frac{1}{2}\frac{\partial^2}{\partial S_T^2}(\sigma S_T^2 \phi)$$

Using this, we get:

$$\begin{aligned}\frac{\partial C}{\partial T} &= \int_K^\infty (S_T - K)\frac{\partial \phi(S_t, T; S_0)}{\partial T} dS_T \\ &= \int_K^\infty (S_T - K)\left[\frac{1}{2}\frac{\partial^2}{\partial S_T^2}(\sigma S_T^2 \phi)\right] dS_T \\ &= \frac{\sigma^2 K^2}{2}\phi = \frac{\sigma^2 K^2}{2}\frac{\partial^2 C}{\partial K^2}\end{aligned}$$

This rewrites:

$$\frac{\partial C}{\partial T} = \frac{\sigma^2 K^2}{2}\frac{\partial^2 C}{\partial K^2} \iff \sigma^2 = \frac{\dfrac{\partial C}{\partial T}}{\dfrac{K^2}{2}\dfrac{\partial^2 C}{\partial K^2}}$$

To see how to get Equation 7 that corresponds to the Markovian projection studied by Gyöngy [1986], note that:

$$\begin{aligned}C(S_0, K, T) &= \mathbb{E}[(S_T - K)^+] \\ \frac{\partial C}{\partial K} &= -\mathbb{E}[H(S_T - K)], \quad H \text{ being the heaviside function} \\ \frac{\partial^2 C}{\partial K^2} &= \mathbb{E}[\delta(S_T - K)], \quad \delta \text{ the dirac function}\end{aligned}$$

since $\max(S - K, 0) = (S - K)H(S - K)$. Applying Itô's formula to the terminal payoff:

$$d(S_T - K)^+ = H(S_T - K)dS_T + \frac{1}{2}\sigma^2 S_T^2 \delta(S_T - K)dT$$

and taking conditional expectations yields:

$$\begin{aligned}\frac{dC}{dT} = \frac{d\mathbb{E}[(S_T - K)^+]}{dT} &= \frac{1}{2}\mathbb{E}[\sigma^2 S_T^2 \delta(S_T - K)] \\ &= \mathbb{E}[\sigma^2 | S_T = K]\frac{K^2}{2}\delta(S_T - K) \\ &= \mathbb{E}[\sigma^2 | S_T = K]\frac{1}{2}\frac{\partial^2 C}{\partial K^2} = \frac{\partial C}{\partial T}\end{aligned}$$

And thus this implies:

$$\sigma^2(K, T; S_0) = \mathbb{E}[\sigma^2 | S_T = K]$$

with the local variance on the left hand side.

B Gaussian Process Regression

We already developed the basics of GPR in the main text. The full details are in Williams and Rasmussen [2006]. We just add some more precisions here for the sake of auto-comprehensiveness.

Let $\mathcal{X} \subseteq \mathbb{R}^n$ be the input space. A kernel is a function

$$k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$$

such that there exists a Hilbert space $\mathcal{H}$ and a mapping $\phi : \mathcal{X} \to \mathcal{H}$ satisfying:

$$k(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{H}} \quad \text{for all } x, x' \in \mathcal{X}.$$

The idea is to project the data into a richer higher dimensional space, that allows to compute distance in a richer way.

Given data points $x_1, \ldots, x_n$, the kernel matrix or Gram matrix $K \in \mathbb{R}^{n \times n}$ is defined by:

$$K_{ij} = k(x_i, x_j)$$

A kernel function must produce a kernel matrix that is symmetric and positive semi-definite (PSD), meaning:

$$\forall a \in \mathbb{R}^n, \quad a^\top K a \geq 0$$

This represents exactly the idea of a generalized dot product.

C Backward Kolmogorov and Fokker-Planck Equations

For more reference, see Pavliotis [2014].

For a continuous-time stochastic process $\{X_t\}_{t \geq 0}$ that models the random evolution of a system under continuous random fluctuations following a stochastic differential equation (SDE) of the form:

$$dX_t = \mu(X_t, t)\, dt + \sigma(X_t, t)\, dW_t,$$

We can derive the backward and forward Kolmogorov equations (the foward equation is the same as the Fokker-Planck equation).

Let $p(x,t)$ be the probability density function of X_t, that is,

$$p(x,t) = \mathbb{P}(X_t \in dx).$$

The Kolmogorov forward equation describes the evolution of this density over time:

$$\frac{\partial p}{\partial t} = -\frac{\partial}{\partial x}[\mu(x,t)p(x,t)] + \frac{1}{2}\frac{\partial^2}{\partial x^2}[\sigma^2(x,t)p(x,t)].$$

This is a partial differential equation (PDE) that describes how the probability distribution "spreads out" over time. It is basically computed by rewriting the diffusion Itô's lemma for $f(X_t)$, then taking the expectation term $\mathbb{E}[f(X_t)]$ and doing integration by part, we can isolate the probability density term, as we have done in Appendix C to retrieve the risk-neutral density.

In contrast, the Kolmogorov backward equation describes the evolution of the expected value of a function of the future state, given the current state.

Let $u(x,t) = \mathbb{E}[f(X_T) \mid X_t = x]$ be the conditional expectation of a terminal payoff $f(X_T)$, where $T > t$. Then u satisfies:

$$\frac{\partial u}{\partial t} + \mu(x,t)\frac{\partial u}{\partial x} + \frac{1}{2}\sigma^2(x,t)\frac{\partial^2 u}{\partial x^2} = 0, \quad u(x,T) = f(x).$$

This is a backward parabolic PDE, propagating information from the future to the present. his is simply derived by taking the time derivative of the previously mentioned $\mathbb{E}[f(X_t)]$, after applying Itô's lemma.

D Supplementary Material

See my github: https://github.com/dmeaille

References

Peter Bank, Christian Bayer, Peter K Friz, and Luca Pelizzari. Rough pdes for local stochastic volatility models. *Mathematical Finance*, 2023.

Christian Bayer, Denis Belomestny, Oleg Butkovsky, and John Schoenmakers. A reproducing kernel hilbert space approach to singular local stochastic volatility mckean–vlasov models. *Finance and Stochastics*, 28(4):1147–1178, 2024.

Christa Cuchiero, Wahid Khosrawi, and Josef Teichmann. A generative adversarial network approach to calibration of local stochastic volatility models. *Risks*, 8(4):101, 2020.

Emanuel Derman. Laughter in the dark-the problem of the volatility smile. In *Euronext Options Conference (Amsterdam)*, pages 1–13, 2003.

Bruno Dupire et al. Pricing with a smile. *Risk*, 7(1):18–20, 1994.

David Duvenaud. *Automatic model construction with Gaussian processes*. PhD thesis, 2014.

Jim Gatheral. *The volatility surface: a practitioner's guide*. John Wiley & Sons, 2011.

Joan Gonzalvez, Edmond Lezmi, Thierry Roncalli, and Jiali Xu. Financial applications of gaussian processes and bayesian optimization. *arXiv preprint arXiv:1903.04841*, 2019.

Leon Gross. Recent developments in the variance swap market, 2004. URL `https://www.risk.net/sites/default/files/import_unmanaged/db.riskwaters.com/data/risk/pdf/special/1204_sr_citigroup.pdf`.

István Gyöngy. Mimicking the one-dimensional marginal distributions of processes having an itô differential. *Probability theory and related fields*, 71(4):501–516, 1986.

Jianan Han, Xiao-Ping Zhang, and Fang Wang. Gaussian process regression stochastic volatility model for financial time series. *IEEE Journal of Selected Topics in Signal Processing*, 10(6):1015–1028, 2016.

Philippe Henrotte. The case for time homogeneity. *The Best of Wilmott*, page 211, 2003.

Pierre Henry-Labordère. Calibration of local stochastic volatility models to market smiles. *Risk*, 22(9):112, 2009.

Hugues Herfurth. Gaussian process regression in computational finance, 2020.

Steven L Heston. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *The review of financial studies*, 6(2):327–343, 1993.

Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960.

Motonobu Kanagawa, Philipp Hennig, Dino Sejdinovic, and Bharath K Sriperumbudur. Gaussian processes and kernel methods: A review on connections and equivalences. *arXiv preprint arXiv:1807.02582*, 2018.

Daniel Lacker. Mean field games and interacting particle systems. *preprint*, 2018.

Daniel Lacker, Mykhaylo Shkolnikov, and Jiacheng Zhang. Inverting the markovian projection, with an application to local stochastic volatility models. 2020.

Bingqing Liu, Ivan Kiskin, and Stephen Roberts. An overview of gaussian process regression for volatility forecasting. In *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pages 681–686. IEEE, 2020.

Bertil Matérn. *Spatial variation*, volume 36. Springer Science & Business Media, 2013.

Henry P McKean Jr. A class of markov processes associated with nonlinear parabolic equations. *Proceedings of the National Academy of Sciences*, 56(6):1907–1911, 1966.

Grigorios A Pavliotis. Stochastic processes and applications. *Texts in applied mathematics*, 60, 2014.

Dejan Petelin, Jan Šindelář, Jan Přikryl, and Juš Kocijan. Financial modeling using gaussian process models. In *Proceedings of the 6th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems*, volume 2, pages 672–677. IEEE, 2011.

Dan Simon. *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006.

Anthonie W Van der Stoep, Lech A Grzelak, and Cornelis W Oosterlee. The heston stochastic-local volatility model: Efficient monte carlo simulation. *International Journal of Theoretical and Applied Finance*, 17(07):1450045, 2014.

Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.