LSV-3: Monte Carlo Approach

Victor Antipov, Nikita Fediashin
Supervisors: Mikhail Zhitlukhin, Charles-Henri Roubinet
Vega Institute Foundation

July 23, 2023

Outline

Calibration condition

If deterministic interest, dividend and repo rates are given, LSV model is

$$df_t = a_t \sigma(t, f_t) f_t dW_t, \tag{1}$$

where f_t is forward on maturity T and a_t is a stochastic process.

Proposition 1 (Nonlinear Option Pricing, 2013, Section 11.2)

The model (1) is calibrated to the market smile $\iff \sigma^2_{Dup}(t,f) = \sigma^2(t,f)\mathbb{E}^{\mathbb{Q}}[a_t^2 | f_t = f]$.

$$\sigma(t,f) \equiv \sigma(t,f,\mathbb{Q}_t) = \frac{\sigma_{Dup}(t,f)}{\sqrt{\mathbb{E}^{\mathbb{Q}}[a_t^2 | f_t = f]}},$$

$$df_t = a_t \sigma(t, f_t, \mathbb{Q}_t) f_t dW_t. \tag{2}$$

Outline

Particle method

The idea is to replace the unknown marginal distribution $\mathbb{Q}_t$ of a McKean SDE at time t by its empirical distribution $\mathbb{Q}_t^N = \frac{1}{N}\sum_{i=1}^N \delta_{f_t^{i,N}}$,

$$df_t^{i,N} = a_t^{i,N}\sigma(t, f_t^{i,N}, \mathbb{Q}_t^N) f_t^{i,N} dW_t^i. \tag{3}$$

As $\mathbb{Q}_t^N$ is atomic and admits no density $p_N(t,f,a)$, the Dirac function $\delta(\cdot)$ is replaced by regularizing kernel $K(\cdot)$.
Define

$$\sigma_N(t,f) := \sigma_{Dup}(t,f)\left(\sqrt{\mathbb{E}^{\mathbb{Q}_t^N}[a_t^2|f_t = f]}\right)^{-1} \approx \sigma_{Dup}(t,f)\sqrt{\frac{\sum_{i=1}^N K\left(f_t^{i,N} - f\right)}{\sum_{i=1}^N \left(a_t^{i,N}\right)^2 K\left(f_t^{i,N} - f\right)}}$$

and simulate

$$df_t^{i,N} = a_t^{i,N}\sigma_N(t, f_t^{i,N}) f_t^{i,N} dW_t^i. \tag{4}$$

The G-HL algorithm

Let $\{t_k\}$ denote a time discretization of $[0, T]$. The particle algrorithm is the following:

1. Initialize $k = 1$ and $\sigma_N(t, f) \equiv \sigma_{Dup}(0, f)/\sqrt{v_0}$ for all $t \in [0, t_1]$, where v_0 is initial variance.
2. Simulate the N particles $\{f_t^{i,N}, a_t^{i,N}\}$ from t_{k-1} to t_k using appropriate scheme assuming that $\sigma_N(t, f) \equiv \sigma_N(t_{k-1}, f)$ for each $t \in [t_{k-1}, t_k]$ for all f.
3. For each f compute

$$\sigma_N(t_k, f) = \sigma_{Dup}(t_k, f) \sqrt{\frac{\sum_{i=1}^N K(f_{t_k}^{i,N} - f)}{\sum_{i=1}^N (a_{t_k}^{i,N})^2 K(f_{t_k}^{i,N} - f)}}.$$

   Interpolate $\sigma_N(t_k, f)$ as a function of f using cubic splines and extrapolate flat.
4. Set $k = k + 1$ and iterate until the latest expiration time T.

Regularization and acceleration techniques

- Select a subset $G_{f,t}$ of particles such that $\mathbb{E}[1_{f_t > \max G_{f,t}}] = \mathbb{E}[1_{f_t < \min G_{f,t}}] = \alpha$, where $\alpha = 10^{-3}$.
- Pick $N_{f,t} = \max(30\sqrt{t}, 15)$ backbone equidistant (in order) particles from $G_{f,t}$.
- Select $K(x) := \frac{1}{h_{t,N}} K\left(\frac{x}{h_{t,N}}\right)$ where K is a fixed symmetric kernel with a bandwidth $h_{t,N}$.
- To save computational time, quadratic kernel can be used. A bandwidth parameter can be chosen proportional to $\kappa f_0 \sigma_{VS,t} \sqrt{\max(t, t_{min})} N^{-1/5}$, where $\kappa \simeq 3/2$ and $t_{min} = 1/4$.

Limitations of G-HL algorithm

- A rule of thumb choice of bandwidth parameter h and kernel $K(\cdot)$.
- Non-parametric kernel regression methods suffer from a bias of order $O(h^2)$ (see [**HarBow1988**]).
- The variance order also heavily depends on the correct choice of the bandwidth h.
- The need for a prespecified parameter h poses an inherent risk and makes it inclined to potential instabilities.

Muguruza's approach

Assumption 1

The leverage function has the following structure: $\sigma(t,f) = \sum_{k=1}^{n} \sigma_k(f)\mathbb{I}_{\{t\in[t_k,t_{k-1})\}}$.

Theorem 1 (Muguruza, 2019)

Given assumption above, we have

$$Law(\log f_{t_k}|\mathbb{F}^W_{t_{k-1}} \cup \mathbb{F}^Z_{t_k}) \sim N(\mu_k, (1-\rho^2)\sigma_k^2),$$

where (W,Z) *are BM with correlation* ρ*, generating* (f,a)*, and*

$$\begin{aligned}
\mu_k &:= \log f_{t_{k-1}} + \rho \int_{t_{k-1}}^{t_k} a_u \sigma(f_{t_{k-1}},u) dZ_u - \frac{1}{2}\int_{t_{k-1}}^{t_k} a_u^2 \sigma^2(f_{t_{k-1}},u) du,\\
\sigma_k^2 &:= \int_{t_{k-1}}^{t_k} a_u^2 \sigma(f_{t_{k-1}},u) du.
\end{aligned}$$

The exact formula

Corollary 2 (Muguruza, 2019)

Given assumption above, we have

$$\mathbb{E}^{\mathbb{Q}}[a_{t_k}^2 | f_{t_k} = f] = \frac{\mathbb{E}^{\mathbb{Q}}\left[a_{t_k}^2 \frac{e^{-\frac{1}{2}d_k^2(f)}}{\sigma_k}\right]}{\mathbb{E}^{\mathbb{Q}}\left[\frac{e^{-\frac{1}{2}d_k^2(f)}}{\sigma_k}\right]}, \quad d_k(f) = \frac{\mu_k - \log(f)}{\sqrt{(1-\rho^2)\sigma_k^2}}.$$

Hence, in Monte Carlo setting we get

$$\mathbb{E}^{\mathbb{Q}_{t_k}}[a_{t_k}^2 | f_{t_k} = f] = \frac{\sum_{i=1}^N a_{t_k}^2 \frac{e^{-\frac{1}{2}(d_k^i(f))^2}}{\sigma_k^i}}{\sum_{i=1}^N \frac{e^{-\frac{1}{2}(d_k^i(f))^2}}{\sigma_k^i}}.$$

Remark. If the variance a_t^2 is lognormal, more precise formulas for $\mathbb{E}^{\mathbb{Q}}[a_t^2 | f_t = f]$ can be obtained (see [**AM**], Proposition 4.1).

Pros and cons of Muguruza's approach

- Allows one to automate the selection of the kernel, reducing such parameters as the bandwidth.
- As a consequence of the previous bullet, the algorithm is more robust.
- We pay for increased accuracy with the reduced computation speed. In particular, now we spend more time calculating kernel parameters and using squared exponential kernels instead of quadratic in G-HL.
- Now the quality of the method depends on our ability to approximate the corresponding integrals $\int_{t_{k-1}}^{t_k} a_u \sigma(f_{t_{k-1}}, u) dZ_u$ and $\int_{t_{k-1}}^{t_k} a_u^2 \sigma^2(f_{t_{k-1}}, u) du$.

Bins method

Let us divide the range of f_{t_i} into l mutually exclusive bins $(b_1, b_2], (b_2, b_3], \ldots, (b_l, b_{l+1}]$, with $b_1 \geq 0$ and $b_{l+1} < \infty$. For $f \in (b_j, b_{j+1}]$ we will use the following approximation:

$$\mathbb{E}^{\mathbb{Q}_t^N}[a_t^2 | f_t = f] \approx \mathbb{E}^{\mathbb{Q}_t^N}[a_t^2 | f_t \in (b_j, b_{j+1}]] = \frac{\mathbb{E}\left[a_t^2 \mathbb{1}_{f_t \in (b_j, b_{j+1}]}\right]}{\mathbb{Q}_t^N[f_t \in (b_j, b_{j+1}]]} \approx$$

$$\frac{\frac{1}{N}\sum_{i=1}^N \left[(a_t^{i,N})^2 \mathbb{1}_{f_t^{i,N} \in (b_j, b_{j+1}]}\right]}{\mathbb{Q}_t^N[f_t \in (b_j, b_{j+1}]]}.$$

Define

$$\mathcal{I}_{k,j} := \left\{ i | f_{t_k}^{i,N} \in (b_j, b_{j+1}] \right\} \text{ and } \alpha_k(j) := \mathbb{Q}_{t_k}^N[f_{t_k} \in (b_{k,j}, b_{k,j+1}]].$$

Bins method

To specify the bin boundaries $\{(b_{k,j}, b_{k,j+1}]\}$ such that each bin contains an approximately equal number of Monte Carlo paths, the following formulas are used:

$$b_{k,1} = \overline{f}_{t_k}^{1,N}, b_{k,l+1} = \overline{f}_{t_k}^{N,N}, b_{k,j} = \overline{f}_{t_k}^{(j-1)N/l,N}, j = 2, \dots, l$$

where $\{\overline{f}_{t_k}^{i,N}, \overline{a}_{t_k}^{i,N}\}$ – particles sorted according to the asset price. These formulas allow us to use $\alpha(j) = 1/l$.
Now one can sort the particles according to the asset price and select a subset G_k. For each $f \in G_k$ find j such that $f \in (b_{k,j}, b_{k,j+1}]$ and compute

$$\mathbb{E}^{\mathbb{Q}_{t_k}}[a_{t_k}^2 | f_{t_k} = f] \approx \frac{\sum_{i \in \mathcal{I}_{k,j}} (a_{t_k}^{i,N})^2}{N \alpha_k(j)}.$$

Pros and cons of bins method

- The method is very simple both from the point of view of implementation and from the point of view of computational costs.
- The main parameter l of this algorithm is the number of bins used. Its selection is done manually.
- There are concerns that if more complex volatility model is used, the method will show significantly worse results, even with a relatively nice choice of the parameter l. In other words, the robustness of the algorithm is questionable.

Gaussian mixture model

Assumption 2

At each fixed moment of time t random variable $x = (f_t, a_t^2)$ has following distribution:

$$p(x) = \sum_{m=1}^{M} \pi_m \mathcal{N}(x|\mu_m, \Sigma_m), \quad \pi_m > 0, \quad \sum_{m=1}^{M} \pi_m = 1.$$

This distribution is called Gaussian mixture with M gaussians.

Here we omitted the dependence of parameters $\{(\pi_m, \mu_m, \Sigma_m)\}_{m=1}^{M}$ on time.
It's well-known that knowing $\mathcal{N}(f_t, a_t^2|\mu_m, \Sigma_m)$ one can calculate $p_m(a_t^2|f) = \mathcal{N}\left(a_t^2|\mu_{a_t^2|f;m}, \Sigma_{a_t^2|f;m}\right)$ analytically.

Gaussian mixture model

Using the fact from the previous slide in combination with Bayes rule, one can easily write conditioned mixture density:

$$p(a_t^2|f) = \sum_{m=1}^{M} \overline{\pi}_m p_m(a_t^2|f), \quad \overline{\pi}_m = \frac{\pi_m \mathcal{N}\left(f|\mu_{f;m}, \Sigma_{f_t,f_t;m}\right)}{\sum_{l=1}^{M} \pi_l \mathcal{N}\left(f|\mu_{f;l}, \Sigma_{f_t,f_t;l}\right)}.$$

Since we are again looking at a mixture of Gaussians, we can calculate the expectation of such a distribution using the following formula:

$$\mu_{a_t^2|f} = \sum_{m=1}^{M} \overline{\pi}_m \mu_{a_t^2|f;m}.$$

In order to calculate the parameters of the corresponding mixture of Gaussians at each time step, we use the celebrated EM-algorithm.

Pros and cons of GMM approach

- The mixture of Gaussians is a fairly rich family of distributions, so we can achieve a good approximation even in complex stochastic volatility models.
- On the other hand usually mixtures of Gaussians are used to approximate multimodal distributions. And this is not the case for Heston and Bergomi models.
- For this approach to work, it is necessary to perform at least several iterations of the EM-algorithm. It turns out to be much more computationally expensive than in all previous models.

Outline

Numerical schemes

For **Heston LSV**:

$$\begin{aligned}\log f_{t_{i+1}} &= \log f_{t_i} - \frac{1}{2}V_{t_i}\Delta t + \frac{\rho}{\gamma}(V_{t_{i+1}} - \kappa\bar{v}\Delta t + V_{t_i}(\kappa\Delta t - 1)) \\ &\quad + (1-\rho^2)^{1/2}(V_{t_i}\Delta t)^{1/2}\sigma(t_i, f_{t_i})Z_{N(0,1)}, \\ V_{t_{i+1}} &\sim c\chi^2(\delta, \bar{\kappa}),\end{aligned}$$

where $c = \frac{\gamma^2}{4\kappa}(1 - e^{-\kappa\Delta t})$, $\delta = \frac{4\kappa\bar{v}}{\gamma^2}$, $\bar{\kappa} = \frac{4\kappa e^{-\kappa\Delta t}}{\gamma^2(1-e^{-\kappa\Delta t})}V_{t_i}$.

For **1F Bergomi LSV**:

$$\begin{aligned}\log f_{t_{i+1}} &= \log f_{t_i} - \frac{1}{2}\xi_{t_i}\sigma(t_i, f_{t_i})^2\Delta t + \sqrt{\xi_{t_i}^{t_i}\Delta t}\sigma(t_i, f_{t_i})\left(\rho Z^1_{N(0,1)} + \sqrt{1-\rho^2}Z^2_{N(0,1)}\right), \\ X_{t_{i+1}} &= X_{t_i}\exp(-\kappa\Delta t) + \sqrt{(1-e^{-2\kappa\Delta t})/(2\kappa)}Z^1_{N(0,1)}, \\ \log \xi_{t_{i+1}}^{t_{i+1}} &= -\omega^2\left(1 - \exp(-2\kappa t_{i+1})\right)/(4\kappa) + \omega X_{t_{i+1}}.\end{aligned}$$

Methods of quality assessment

To evaluate the performance, we use the following procedure:

1. Calculate implied volatility of the market and implied volatility generated with calibrated LSV model.
2. Calculate absolute difference at points of some grid $\{K_j\}$
3. Compute weighted average of these differences

To produce weights we use one of the following two formulas:

- $w_j = \frac{\bar{w}_j}{||\bar{w}||}$, where $\bar{w}_j = P(f_T > K_j) \cdot I[K_j > f_0] + P(f_T < K_j) \cdot I[K_j < f_0]$,
- $w_j = \frac{\bar{w}_j}{||\bar{w}||}$, where $\bar{w}_j = g_{f_T}(K_j)$,

where $g_{f_T}(\cdot)$ is (estimated) density function of f_T.

LSV and Heston market IVs

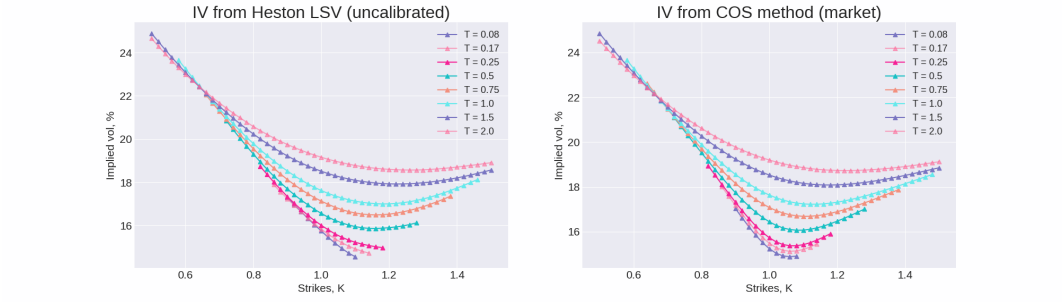

Figure: Heston market IV and IV generated with calibrated LSV model. The G-HL method with a quadratic kernel with 2^{18} particles was used.

LSV and Heston market IV differences

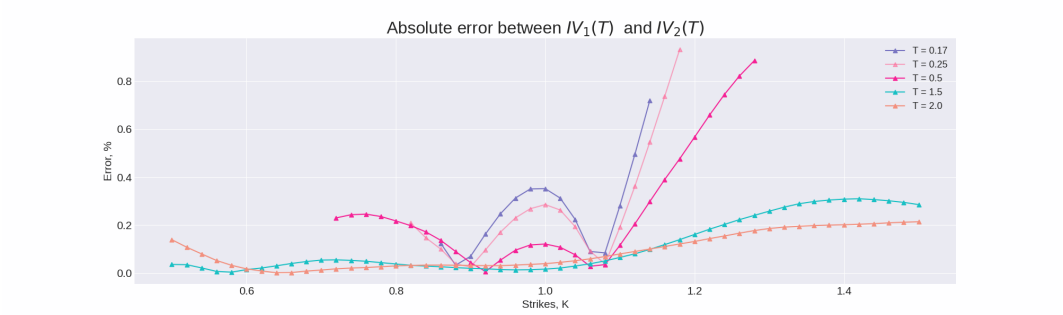

Figure: Absolute difference between Heston market IV and IV generated with calibrated LSV model. The G-HL method with a quadratic kernel with 2^{18} particles was used.

Local component effect

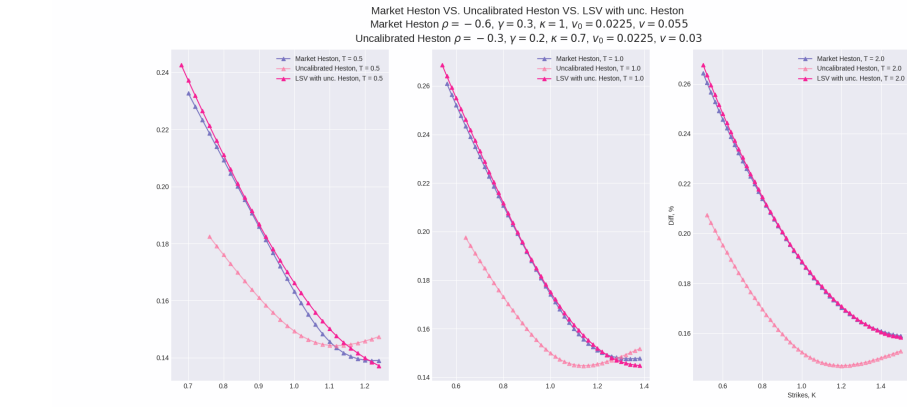

Figure: The G-HL method with a quadratic kernel with 2^{18} particles was used to calibrate Heston market model. Here one can see the effect of local component (leverage function) in case of poorly calibrated stochastic volatility model

Weighted convergence of implied volatility for Heston model

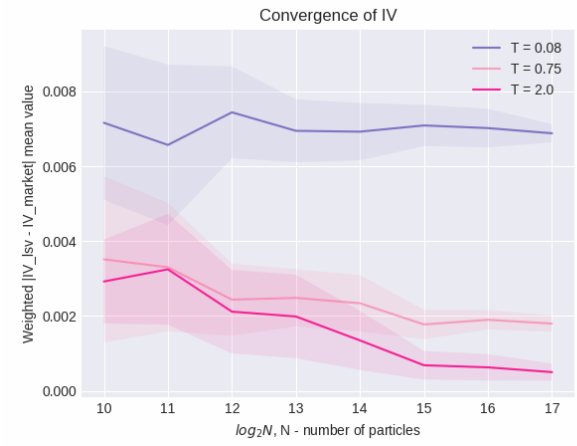

Figure: IV convergence for Heston model with quadratic kernel

LSV and Bergomi market IVs

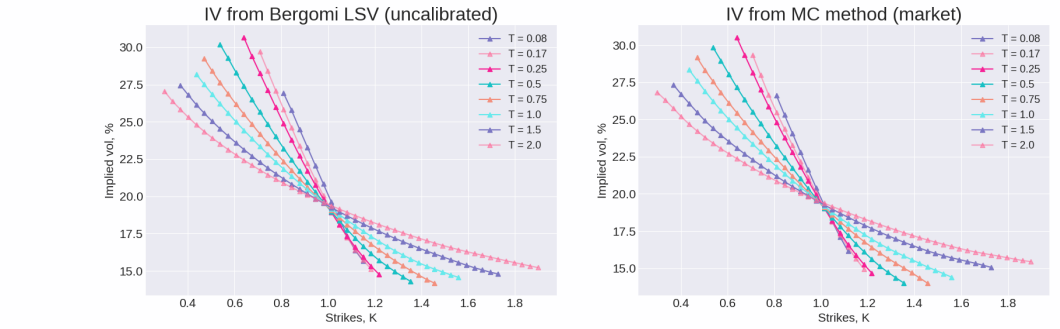

Figure: Bergomi market IV and IV generated with calibrated LSV model. The G-HL method with a quadratic kernel with 2^{18} particles was used.

LSV and Bergomi market IV differences

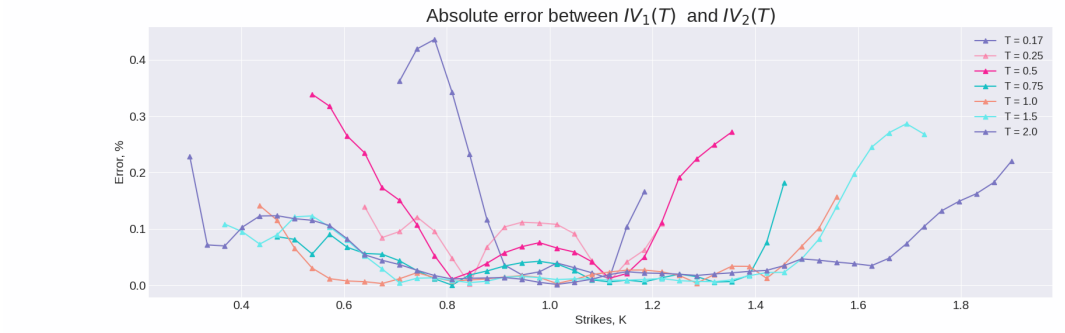

Figure: Absolute difference between Bergomi market IV and IV generated with calibrated LSV model. The G-HL method with a quadratic kernel with 2^{18} particles was used.

Local component effect

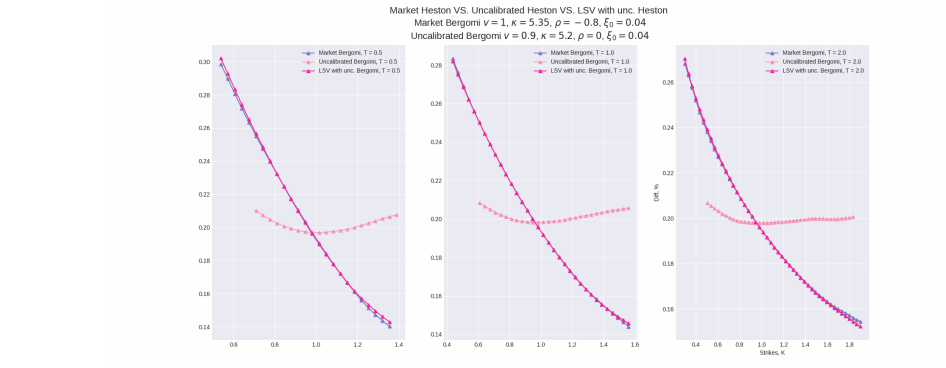

Figure: The G-HL method with a quadratic kernel with 2^{18} particles was used to calibrate Bergomi market model. Here one can see the effect of local component(leverage function) in case of poorly calibrated stochastic volatility model

Weighted convergence of implied volatility for Bergomi model

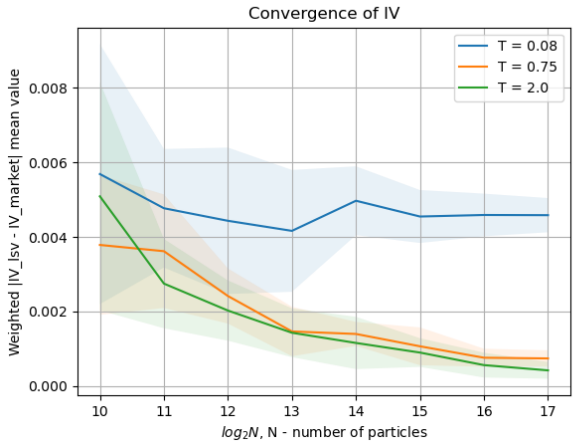

Figure: IV convergence for Bergomi model with quadratic kernel

Different kernels comparison

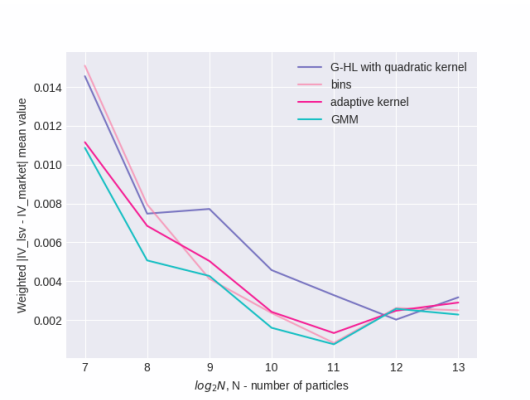

Figure: Convergence of LSV IVs to market ones in terms of weighted mean absolute error in case of Heston model and market. Here different methods for conditional expectation calculation are presented. Tenor $T = 0.75$

Calibration speed

Since the qualitative results for different kernels turned out to be very similar, we also compared the calibration speed. The machine with Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz was used.

Calibration time for 2^{16} particles, seconds				
Stochastic volatility model	quadratic kernel	bins(20)	adaptive kernel	GMM(4 gaussians)
Heston model	34.280	12.395	38.560	297.728
Bergomi model	17.356	11.872	–	258.780

Future work

Possible directions and improvements are as follows:

- To calibrate stochastic volatility and LSV models on real-word market data
- To resolve problem with small tenors. Possible directions are grid shredding and additional verification of the local volatility function at the corresponding time intervals
- To do backtests in order to understand whether implied volatility dynamics is correct
- Detailed stress-tests of algorithms to understand if some of them are more robust or effective in case of different stochastic volatility models
- To compare particle and PDE approaches
- To add different interest rate models: deterministic and stochastic

References

[1] [NOP] Guyon, J., and Henry-Labordere, P. (2013). Nonlinear Option Pricing (1st ed.). CRC Press.

[2] [AM] Aitor Muguruza (2019). Not so Particular about Calibration: Smile Problem Resolved.

[3] [SVM] Lorenzo Bergomi (2015). Stochastic Volatility Modeling. Chapman and Hall/CRC.

[4] [HB88] W. H¨ardle and A. Bowman., Bootstrapping in Nonparametric Regression: Local Adaptive Smoothing and Confidence Bands., 1988