LSV-2: PDE approach to model calibration

Dmitrii Sotnikov
Supervisors: Mikhail Zhitlukhin, Charles-Henri Roubinet
Vega Institute Foundation

July 24, 2023

Model description

By a local stochastic volatility (LSV) model we will mean a model of type

$$\begin{cases} dX_t = -\frac{1}{2}\sigma(t,X_t)^2 V_t dt + \sigma(t,X_t)\sqrt{V_t} dW_t, \\ dY_t = b^Y(t,X_t,Y_t)dt + \sigma^Y(t,X_t,Y_t)dB_t, \end{cases}$$

where

- $X_t = \log F_t$ is a logarithm of the forward price;
- Y_t is a phase variable such that stochastic component of variance $V_t = V(t,Y_t)$;
- W_t and B_t are correlated Brownian motions, $dW_t dB_t = \rho dt$.

The local component $\sigma(t,x)$ will be referred to as a **leverage function**.

For simplicity we assume that discounting, dividend and repo rates are equal to zero.

Why use LSV models?

- With exotic options come exotic risks such as vol-of-vol risk or spot/volatility correlation risk. Local volatility (LVM) models give no control on them. They also cannot price correctly products on forward variance such as forward start options or cliquet due to the flattening of the volatility smile.

- For that reason stochastic volatility models (SVM) can be used. However, they have only a finite number of parameters and cannot be perfectly calibrated to the market volatility smiles.

- A natural way to overcome it is to embed a local volatility component $\sigma(t, x)$ into the SVM. It will allow for a perfect match to the market volatility surface owing to the infinite-dimensional parametrization.

Calibration condition

Calibration of the LSV model involves two steps:

1. Calibration of the stochastic component parameters (approximately once a month) to capture dynamics of the volatility surface.
2. Calibration of the local component (daily) to match exactly market implied volatility surface. We will pay attention to this part.

Proposition 1 (see [1])

The LSV model is calibrated to the market $\iff \sigma^2_{Dup}(t,x) = \sigma^2(t,x)\mathbb{E}[V_t|X_t = x]$.

Here $\sigma_{Dup}(t,x)$ is a market Dupire's local volatility function. Thus,

$$\sigma(t,x) = \frac{\sigma_{Dup}(t,x)}{\sqrt{\mathbb{E}[V_t \mid X_t = x]}} = \frac{\sigma_{Dup}(t,x)}{\sqrt{\frac{\int V(t,y)p(t,x,y)\,dy}{\int p(t,x,y)\,dy}}},$$

where $p(t,x,y)$ is the p.d.f. of (X_t, Y_t) given the distribution $p_0(x,y)$ of (X_0, Y_0).

Kolmogorov forward equation

The function $p(t, x, y)$ of (X_t, Y_t) is unknown. Fortunately, it solves the *Kolmogorov forward equation*

$$\begin{cases} \partial_t p(t,x,y) = \mathcal{L}^* p(t,x,y), \\ p(0,x,y) = p_0(x,y). \end{cases}$$

Here $\mathcal{L}$ denotes the generator of (X_t, Y_t) and $\mathcal{L}^*$ stands for its adjoint. This equation admits a formal solution satisfying

$$p(t+\tau, x, y) = e^{\tau \mathcal{L}^*} p(t,x,y)$$

on the time intervals where $\mathcal{L}$ is time-homogeneous.

Our goal is to implement a reasonable numerical scheme approximating $e^{\tau \mathcal{L}^*}$.

What does the generator look like?

The generator of the considered model can be written as

$$\mathcal{L} = \frac{1}{2} a_{xx} \partial^2_{xx} + \frac{1}{2} a_{yy} \partial^2_{yy} + a_{xy} \partial^2_{xy} + b_x \partial_x + b_y \partial_y,$$

where

$$a_{xx}(t,x,y) = \sigma(t,x)^2 V(t,y), \quad a_{yy}(t,x,y) = \sigma^Y(t,x,y)^2,$$

$$a_{xy}(t,x,y) = \rho \sigma(t,x) \sqrt{V(t,y)} \sigma^Y(t,x,y),$$

$$b_x(t,x,y) = -\frac{1}{2} \sigma(t,x)^2 V(t,y), \quad b_y(t,x,y) = b^Y(t,x,y).$$

The adjoint to it is

$$\mathcal{L}^* = \frac{1}{2} \partial^2_{xx}(a_{xx} \cdot) + \frac{1}{2} \partial^2_{yy}(a_{yy} \cdot) + \partial^2_{xy}(a_{xy} \cdot) - \partial_x(b_x \cdot) - \partial_y(b_y \cdot).$$

Space and time discretization

We consider a uniform space grid

$$\{(x_i, y_j)\}_{i=1,\dots,n_x,\ j=1,\dots,n_y}$$

with steps Δx, Δy and a uniform time grid

$$\{t_k\}_{k=1,\dots,n_t}$$

with step Δt. We approximate the densities $p(t, x, y)$ by discrete distributions

$$\{p(t_k, x_i, y_j)\}_{k=1,\dots,n_t} \equiv \{p_{ij}^k\}_{k=1,\dots,n_t}$$

taking non-zero values only on the space grid. We approximate generator by a piece-wise homogeneous on each interval $[t_k,\ t_{k+1})$ for $k = 1, \dots, n-1$ fixing $t = t_k$ and denote the discrete approximation of this generator by $L_k \in \mathbb{R}^{n_x n_y \times n_x n_y}$.

Density propagation

A solution of the Kolmogorov PDE at t_{k+1} is approximated by

$$p(t_{k+1}, :, :) = e^{L_k^* \Delta t} p(t_k, :, :), \quad k = 1, \ldots, n-1,$$

that allows for the density propagation along the time grid.

Implementation challenges

- Unstable behavior of density propagator.
- Speed of density propagation.
- Loss of the probability mass.
- Violation of the martingale property.
- Numerical stability of leverage calibration.

Stability of the propagator

- A sufficient condition of stability is non-negativity of the off-diagonal values of L_k, so-called **Metzler property**.
- For the generators satisfying Metzler's property, it follows from the Gershgorin circle theorem (see [2]) that the real parts of all its eigenvalues are not positive. In turn, it guarantees the stability of dynamics.
- Note that this stability has nothing to do with numerical stability.
- Moreover, the discretized generator L_k satisfying Metzler's property is exactly a generator of some continuous-time Markov chain taking values in the grid $\{(x_i, y_j)\}_{ij}$. Hence, its elements can be interpreted as transition rates from one grid node to another.
- Thus, while constructing numerical scheme, we will try to keep Metzler's condition fulfilled.

Discretization of the generator

By default, we use the following discretization stencils:

$$\partial^2_{xx} p^k_{ij} \approx \frac{p^k_{i-1,j} - 2p^k_{ij} + p^k_{i+1,j}}{\Delta x^2}, \quad \partial^2_{yy} p^k_{ij} \approx \frac{p^k_{i,j-1} - 2p^k_{ij} + p^k_{i,j+1}}{\Delta y^2},$$

$$\partial_x p^k_{ij} \approx \frac{p^k_{i+1,j} - p^k_{i-1,j}}{2\Delta x}, \quad \partial_y p^k_{ij} \approx \frac{p^k_{i,j+1} - p^k_{i,j-1}}{2\Delta y},$$

$$\partial^2_{xy} p^k_{ij} \approx \begin{cases} \dfrac{p^k_{i+1,j+1} + 2p^k_{ij} + p^k_{i-1,j-1} - p^k_{i+1,j} - p^k_{i-1,j} - p^k_{i,j+1} - p^k_{i,j-1}}{2\Delta x \Delta y}, & \rho \geq 0, \\ -\dfrac{p^k_{i-1,j+1} + 2p^k_{ij} + p^k_{i+1,j-1} - p^k_{i+1,j} - p^k_{i-1,j} - p^k_{i,j+1} - p^k_{i,j-1}}{2\Delta x \Delta y}, & \rho < 0. \end{cases}$$

Note that we use seven-point stencil for mixed derivative as recommended in [3] rather that more common 4-point stencil. All the discretizations are of the second order.

Stencils for mixed derivative

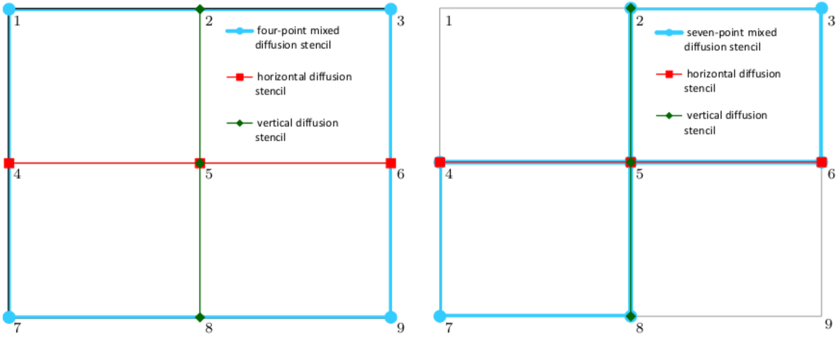

Figure: The nodes of the four- and the seven-point mixed diffusion stencils. Source: [3].

We also prefer to use the transposed matrix L_k^* as a discretization of $\mathcal{L}_k^*$ rather than discretize it directly.

Generator decomposition

At each time step the discretized generation can be decomposed into three simple generators along three directions x, y, and xy:

$$L_k = L_{k,x} + L_{k,y} + L_{k,xy}.$$

This fact is extremely useful to the PDE approximation since it will reduce the problem to the solution of tri- or pentadiagonal linear system.
We denote the diffusion terms of each component by

$$D_{k,x}(x_i, y_j) = \frac{1}{2}\left(\frac{a_{xx}(t_k, x_i, y_j)}{\Delta x^2} - \frac{|a_{xy}(t_k, x_i, y_j)|}{\Delta x \Delta y}\right),$$

$$D_{k,y}(x_i, y_j) = \frac{1}{2}\left(\frac{a_{yy}(t_k, x_i, y_j)}{\Delta y^2} - \frac{|a_{xy}(t_k, x_i, y_j)|}{\Delta x \Delta y}\right),$$

$$D_{k,xy}(x_i, y_j) = \frac{|a_{xy}(t_k, x_i, y_j)|}{2\Delta x \Delta y}.$$

Generator decomposition

Non-zero elements of the components $L_{k,x}, L_{k,y}$, and $L_{k,xy}$ are given by

$$L_{k,x}(i,j,i,j) = -2D_{k,x}(x_i,y_j), \qquad L_{k,y}(i,j,i,j) = -2D_{k,y}(x_i,y_j),$$

$$L_{k,x}(i,j,i+1,j) = D_{k,x}(x_i,y_j) + \frac{b_x(t_k,x_i,y_j)}{2\Delta_x}, \quad L_{k,y}(i,j,i,j+1) = D_{k,y}(x_i,y_j) + \frac{b_y(t_k,x_i,y_j)}{2\Delta_y},$$

$$L_{k,x}(i,j,i-1,j) = D_{k,x}(x_i,y_j) - \frac{b_x(t_k,x_i,y_j)}{2\Delta_x}, \quad L_{k,y}(i,j,i,j-1) = D_{k,y}(x_i,y_j) - \frac{b_y(t_k,x_i,y_j)}{2\Delta_y}.$$

$$\begin{aligned}
L_{k,xy}(i,j,i,j) &= -2D_{k,xy}(x_i,y_j), \\
L_{k,xy}(i,j,i+1,j+\operatorname{sgn}(\rho)) &= D_{k,xy}(x_i,y_j), \\
L_{k,xy}(i,j,i-1,j-\operatorname{sgn}(\rho)) &= D_{k,xy}(x_i,y_j).
\end{aligned}$$

Here we use double-index notation: $L_k(i_1,j_1,i_2,j_2)$ corresponds to the transition rate from the node (i_1,j_1) to (i_2,j_2) and denotes the element on $(i_1(n_y-1)+j_1)$-th row and $(i_2(n_y-1)+j_2)$-th column of the generator matrix.

Approximation of the exponential

We use Strang scheme [5] to speed up the computation of the matrix exponential

$$e^{\tau L_k^*} = e^{\frac{\tau}{2}L_{k,y}^*} e^{\frac{\tau}{2}L_{k,x}^*} e^{\tau L_{k,yx}^*} e^{\frac{\tau}{2}L_{k,x}^*} e^{\frac{\tau}{2}L_{k,y}^*} + \mathcal{O}(\tau^3)$$

and the following second-order implicit scheme (see Padé's work [4])

$$e^{\tau L_{k,w}^*} = \left[I - \tau L_{k,w}^* + \frac{\tau^2}{2} L_{k,w}^{*2} \right]^{-1} + \mathcal{O}(\tau^3)$$

to approximate $e^{\frac{\tau}{2}L_{k,x}^*}$ and $e^{\frac{\tau}{2}L_{k,y}^*}$. By the construction of the generators $L_{k,x}$ and $L_{k,y}$, the matrix $I - \tau L_{k,w}^* + \frac{\tau^2}{2} L_{k,w}^{*2}$ is pentadiagonal and allows for fast inversion. For the cross term of the generator a second-order explicit scheme

$$e^{\tau L_{k,xy}^*} = I + \tau L_{k,xy}^* + \frac{\tau^2}{2} L_{k,xy}^{*2} + \mathcal{O}(\tau^3)$$

was found to demonstrate much more stable numerical results.

Boundary conditions

To keep the probability mass constant and to preserve the martingale property of e^{X_t}, we set the values of L_x and L_{xy} equal to zero for the corresponding x-boundaries to make them absorbing:

$$L_{k,x}(1,\cdot,\cdot,\cdot) = L_{k,xy}(1,\cdot,\cdot,\cdot) = 0,$$

$$L_{k,xy}(1,\cdot,\cdot,\cdot) = L_{k,xy}(n_x,\cdot,\cdot,\cdot) = L_{k,xy}(\cdot,1,\cdot,\cdot) = L_{k,xy}(\cdot,n_y,\cdot,\cdot) = 0,$$

For y-component we use, following [3], *method of images*. For each y-boundary node, let say (x_i, y_1) we add an imaginary node (x_i, y_0), such that $y_1 - y_0 = y_2 - y_1$ and $p(t_k, x_i, y_2) = p(t_k, x_i, y_0)$. Doing so, we ensure that the finite-differencing term for the first derivative is equal to zero, so there is no advection through the y-boundary. Hence, only the diffusion term is taken into account:

$$L_{k,y}(i,1,i,1) = -D_y(x_i,y_1), \qquad L_{k,y}(i,n_y,i,n_y) = -D_y(x_i,y_{n_y}),$$

$$L_{k,y}(i,2,i,2) = D_y(x_i,y_1), \qquad L_{k,y}(i,n_y-1,i,n_y-1) = D_y(x_i,y_{n_y}).$$

Switching to forward/backward stencils

If the one of $L_{x,k}(i,j,i-1,j)$ or $L_{x,k}(i,j,i+1,j)$ becomes negative due to the advection term, we switch to the forward or backward stencil for ∂_x depending on the sign of $b_x(t_k,x_i,y_j)$ to preserve the Metzler's property.

Namely, if $D_x(x_i,y_j) \geq 0$ and $D_{k,x}(x_i,y_j) - \dfrac{b_x(t_k,x_i,y_j)}{2\Delta_x} < 0$, we use forward stencil:

$$\begin{cases} L_{k,x}(i,j,i,j) = -2D_{k,x}(x_i,y_j) - \dfrac{b_x(t_k,x_i,y_j)}{\Delta_x}, \\ L_{k,x}(i,j,i+1,j) = D_{k,x}(x_i,y_j) + \dfrac{b_x(t_k,x_i,y_j)}{\Delta_x}, \\ L_{k,x}(i,j,i-1,j) = D_{k,x}(x_i,y_j). \end{cases}$$

If $D_x(x_i,y_j) \geq 0$ and $D_{k,x}(x_i,y_j) + \dfrac{b_x(t_k,x_i,y_j)}{2\Delta_x} < 0$, we switch to backward stencil.

Similar modifications of generator are made for $L_{k,y}$.

Martingality correction

Let $\bar{F}$ denote a flattened matrix of prices e^{x_i} on the space grid. The martingale property

$$\mathbb{E}\left[e^{x(t+\Delta t)} \,\middle|\, x(t)\right] = \bar{F}^* \cdot e^{\Delta t L^*} p(t,\cdot,\cdot) = e^{x(t)}$$

is satisfied if $\bar{F}^* \cdot L^* = 0$, or equivalently $L \cdot \bar{F} = 0$. For each $k = 1, \ldots, n_t$, we compute drift terms for interior nodes

$$\mu_{ij}^k = (L_k \cdot \bar{F})_{ij}, \quad i = 2, \ldots, n_x - 1, \quad j = 2, \ldots, n_y - 1.$$

and adjust the (i,j)-th row of the $L_{k,x}$ generator to cancel out the drift. If

$$L_{k,x}(i,j,i-1,j) + \frac{\mu_{ij}^k}{e^{x_{i+1}} - e^{x_{i-1}}} \geq 0, \quad L_{k,x}(i,j,i+1,j) - \frac{\mu_{ij}^k}{e^{x_{i+1}} - e^{x_{i-1}}} \geq 0,$$

we add $\dfrac{\mu_{ij}^k}{e^{x_{i+1}} - e^{x_{i-1}}}$ to $L_{k,x}(i,j,i-1,j)$ and subtract it from $L_{k,x}(i,j,i+1,j)$. Otherwise, we switch to the corresponding forward/backward stencil.

Leverage function calibration

The leverage function at time step t_k can be calibrated via the formula

$$\sigma(t_k, x_i) = \frac{\sigma_{Dup}(t_k, x_i)}{\sqrt{\frac{\sum_{j=1}^{n_y} V(t_k, y_j) p(t_k, x_i, y_j)}{\sum_{j=1}^{n_y} p(t_k, x_i, y_j)}}}, \quad i = 1, \ldots, n_x, \quad k = 1, \ldots, n_t,$$

before the density propagation from t_k to t_{k+1}.
However, usage of this formula for x_i with small values of the marginal density

$$p_x(t_k, x_i) = \sum_{j=1}^{n_y} p(t_k, x_i, y_j)$$

may cause numerical instability in leverage function computation. Thus, we set the value $\alpha \in (0, 1)$ and apply formula only for $x_i \in [q_\alpha, q_{1-\alpha}]$, where q_α denotes the quantile of order α of the distribution $p_x(t_k, x)$. For $x_i \notin [q_\alpha, q_{1-\alpha}]$ flat extrapolation of the leverage function $\sigma(t_k, x)$ in x is used. In the experiments we set $\alpha = 0.01$.

Calibration algorithm

1. Calculate the «market» Dupire local volatility function $\sigma_{Dup}(t, x)$ on a predefined (dense enough) grid and interpolate it with cubic splines.
2. At each step $k = 1, \ldots, n_t$ calculate the leverage function

$$\sigma(t_k, x_i), \quad i = 1, \ldots, n_x,$$

   given the distribution $\{p(t_k, x_i, y_j)\}_{i=1,\ldots,n_x,\, j=1,\ldots,n_y}$.
3. Propagate the density $p(t_k, x, y)$ from t_k to t_{k+1} plugging the leverage function $\sigma(t_k, \cdot)$ into the discretized generator L_k.
4. Given the p.d.f. calculate the option prices

$$C(t_k, K) = \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} p(t_k, x_i, y_j) \left(e^{x_i} - K\right)^+$$

   and implied volatility surface.

Numerical results: unit leverage test

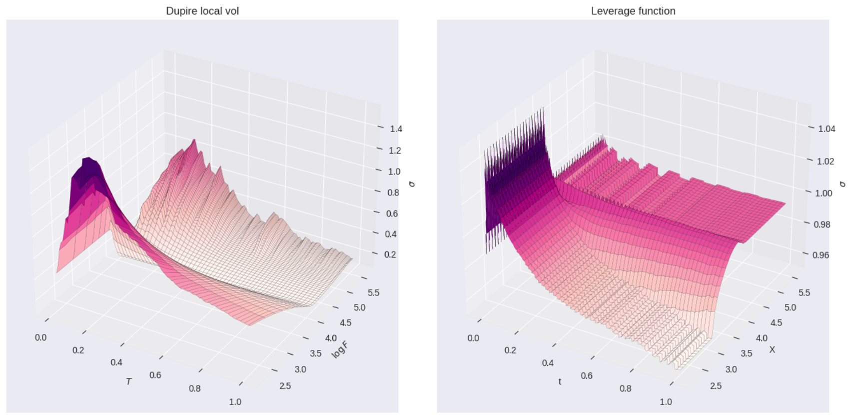

Figure: On the left: local volatility of the 1-factor Bergomi model with $\rho = -0.8$ and $k = 5.35,\ \nu = 1.74,\ \xi_0^T \equiv 0.04$ calculated with $N = 10^7$ Monte Carlo simulations. On the right: calibrated leverage function $\sigma(t, x)$.

Numerical results: unit leverage test

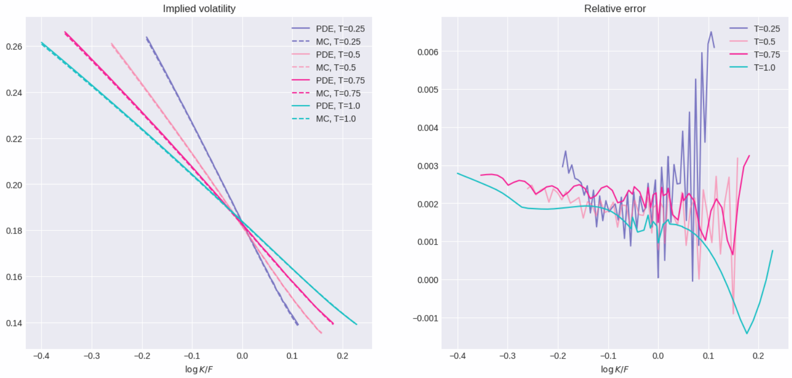

Figure: On the left: volatility smiles for calibrated 1-factor Bergomi model with $\rho = -0.8$ and $k = 5.35,\ \nu = 1.74,\ \xi_0^T \equiv 0.04,\ T \in \{3M, 6M, 9M, 12M\}$ to market generated by this model obatained via PDE (solid) and Monte Carlo method (dashed). Parameters of grids: $n_x = 251,\ n_y = 71,\ n_t = 5001$. On the right: relative difference between smiles.

Numerical results: 2-factor Bergomi market

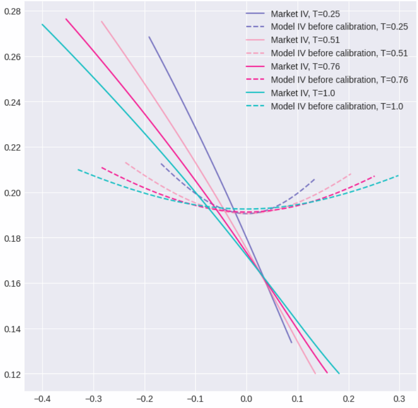

Figure: Implied volatility surface of the uncalibrated 1-factor Bergomi model (dashed) and «market» implied volatility surface generated by 2-factor Bergomi model obtained via Monte Carlo simulation for $T \in \{3M, 6M, 9M, 12M\}$.

Numerical results: 2-factor Bergomi market

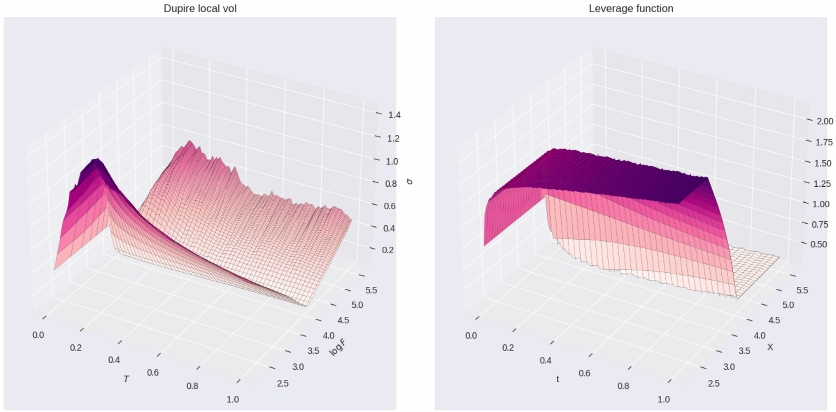

Figure: On the left: local volatility of the 2-factor Bergomi model calculated with $N = 10^6$ Monte Carlo simulations. On the right: calibrated leverage function $\sigma(t, x)$.

Numerical results: 2-factor Bergomi market

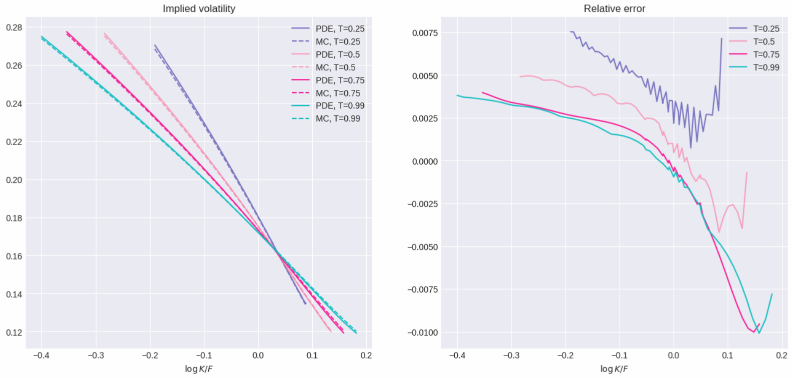

Figure: On the left: volatility smiles for 1-factor Bergomi model calibrated to the market generated by 2-factor Bergomi model, $T \in \{3M, 6M, 9M, 12M\}$ obatained via PDE (solid) and Monte Carlo method (dashed). Parameters of grids: $n_x = 325,\ n_y = 125,\ n_t = 101$. On the right: relative difference between smiles.

Numerical issues: violation of the Metzler's property

For large enough values of $|\rho|$ the diffusion terms $D_{k,x}(x_i, y_j)$ and $D_{k,y}(x_i, y_j)$ may become negative and cause dramatic numerical difficulties while propagation. We found two possible solutions to this problem:

- **Local decorrelation**: artificial reduction of the correlation at the nodes with negative diffusion terms. A similar approach is just to floor the diffusion terms at zero. However, numerical experiments demonstrated that both these approaches significantly change model dynamics and cause large errors in implied volatility for 1-factor Bergomi model with $\rho \approx -0.9$.
- **Choice of grid sizes**: the problem occurs at the step of matrix exponent approximation and vanishes if $\dfrac{\Delta t}{\Delta x^2}$, $\dfrac{\Delta t}{\Delta y^2}$, and $\dfrac{\Delta t}{\Delta x \Delta y}$ are kept small enough. Thus, choice of small enough time-step allows for stable propagation without any changes in the generator, but may be very time-consuming for dense space grids.

Conclusion

- Calibration of the leverage function with PDE method was implemented and validated for 1-factor Bergomi model with reasonable (from a market point of view) parameters.
- The implementation is applicable for a wide class of stochastic volatility models.

Further tasks

- Tests of the implemented model on the real market data.
- Calibration to the volatility surface dynamics.
- Pricing of the exotic forward volatility dependent options.
- Dynamic choice of time step.
- Dynamic space-grid change and transition of probability between grids.
- Calibration of more sophisticated stochastic vol models (pseudo-rough LSV?).

References

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

[2] Horn, R., & Johnson, C. (2012). Matrix Analysis (2nd ed.). Cambridge University Press.

[3] Jäckel, P., & Abellan-Lopez, M. (2014). Implementation of the DHI volatility framework. VTB Capital Quantitative Research.

[4] Jäckel, P. (2013). Finite differencing schemes as Padé approximants. http://www.jaeckel.org/FiniteDifferencingSchemesAsPad%C3%A9Approximants.pdf

[5] Strang, G. (1968). On the Construction and Comparison of Difference Schemes. SIAM Journal on Numerical Analysis. 5.3, pp. 506–517. DOI: 10.1137/0705041.