Ecole CentraleSupélec
Projet d'Option, Option Mathématiques Appliquées, 2019-2020

# Rough Volatility Models and Calibration

Jed Houas, Achraf Maalej, Achraf Dridi, Ilyass OUBAIK

Supervisor: Gad Bouaziz

# Contents

# 1 Introduction

Volatility is a crucial variable when modeling the evolution of returns. In fact a very popular model to forecast a derivatives asset price is:

$$dS_t = S_t \sigma_t dB_t$$

with $B_t$ a brownian motion (White noise).
In this diffusion equation, we introduce the volatility parameter $\sigma_t$ which is the most important ingredient in this representation. By understanding the distribution and dynamics of volatility, we intend to add another model to diffuse this parameter: This is the volatility model which therefore is crucial to forecast the absolute magnitude of returns. It may also be used to predict quantiles or the entire distribution. Such forecasts are used in risk management, derivative pricing and hedging, market making, market timing, portfolio selection and many other financial activities.
In addition to being the first feature traders look at in any asset, Volatility has become in itself a traded asset with volatility trades reaching $2 trillion.

This is why now more than ever quantitative researchers all around the sphere are working on developing new ideas to building robust models for volatility.

# 2 Motivation for a new model of volatility

In the Black-Scholes framework, the volatility function is either constant or a deterministic function of time. In Dupire's local volatility model, the local volatility $\sigma(Y_t, t)$ is a deterministic function of the underlying price and time, chosen to match observed European option prices exactly. Such a model by definition is not time-homogeneous; its dynamics are highly unrealistic, typically generating future volatility surfaces completely unlike those we observe in the market.

On the other hand, in the stochastic volatility models, the volatility is modeled as a continuous brownian semi-martingale. Notable amongst such stochastic volatility models are the Hull and White model, the Heston model, and the SABR model. While stochastic volatility dynamics are more realistic than local volatility dynamics, generated option prices are not consistent with observed European option prices.
In this project, We will introduce the Rough volatility model as an alternative model of volatility that will mitigate the problems seen in the other mentioned models (Stochastic volatility, Local volatility):

- The local volatility models will be able to match the value of the smile as of today, but because the smile flattens for long maturities, the model gives an almost constant smile for these maturities, leading to a flattening of the forward smile (i.e. smile in the future), which is unrealistic.
- Stochastic volatility models generate volatility surfaces that are inconsistent with the observed volatility surface.
  - In stochastic volatility models, the ATM volatility skew is constant for short dates and inversely proportional to T for long dates.
  - Empirically, we find that the term structure of ATM skew is proportional to $\frac{1}{T^\alpha}$ for some $0 < \alpha < 1/2$ over a very wide range of expirations.

# 3 How volatility is rough

In this section, we will resume the study in Volatility is Rough, Jim Gatheral, Thibault Jaisson, Mathieu Rosenbaum [1] to show that log-volatility behaves essentially as a fractional Brownian motion with Hurst exponent H of order 0.1, at any reasonable time scale i.e

$$\mathrm{d}log(\sigma_t) = \eta \mathrm{d}W_t^H$$

where $W_t$ is a fractional brownian motion.
In the beginning, we will revisit the question of the smoothness of the volatility process.
Let us first define the momentum of order q using discrete observations of the volatility process, on a time grid with mesh $\Delta$ on [0,T]:

$$m(q, \Delta) = \frac{1}{N} \sum_{k=1}^{N} |log(\sigma_{k\Delta}) - log(\sigma_{(k-1)\Delta})|^q$$

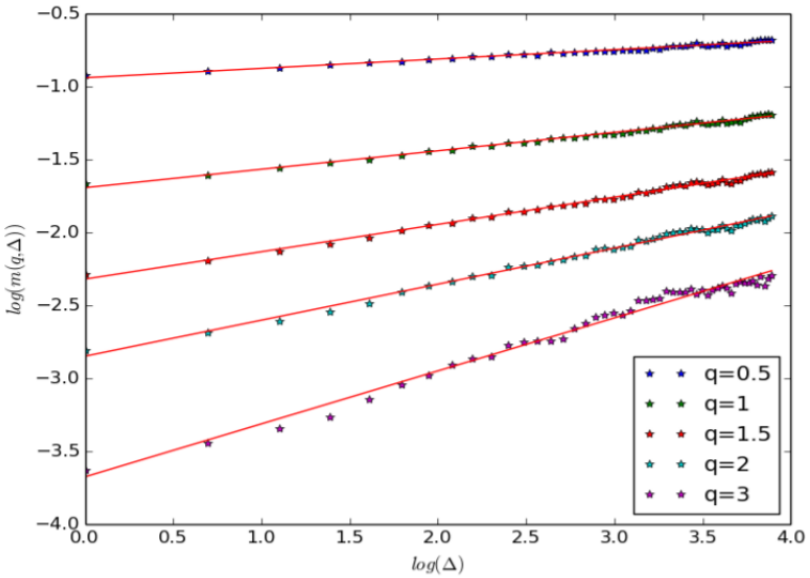
Assuming the increments of the log-volatility process are stationary and that a law of large number can be applied, m(q, $\Delta$) can also be seen as the empirical counterpart of

$$\mathbf{E}[|log(\sigma_\Delta) - log(\sigma_0)|^q]$$

When plotting log m(q, $\Delta$) vs log $\Delta$ for different values of q and for different asset classes, we found out that, for a given q, the points essentially lie on a straight line. Under stationarity assumptions, this implies that the log-volatility increments respect the following scaling property in expectation:

$$\mathbf{E}[|log(\sigma_\Delta) - log(\sigma_0)|^q] = K_q \Delta^{\xi_q}$$

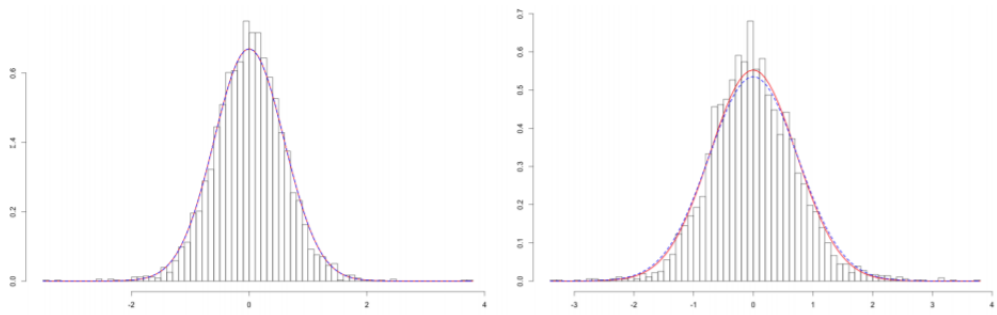
where $\xi_q > 0$ is the slope of the line associated to q and then by plotting $\xi_q$ against q, we obtain that $\xi_q$ is a linear function of q: $\xi_q \sim Hq$.




*log m(q, $\Delta$) as a function of log $\Delta$, DAX*

After studying different assets, we noticed that the smoothness parameter lies systematically between 0.08 and 0.2. Furthermore, we found that increments of the log-volatility are approximately

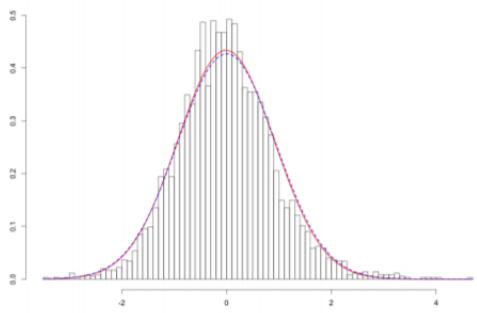
normally distributed and that their moments enjoy a monofractal scaling property. This leads us to model the log of volatility using a fBM with Hurst parameter $H < 1/2$.



(a) $\Delta = 1$ day



(b) $\Delta = 5$ days



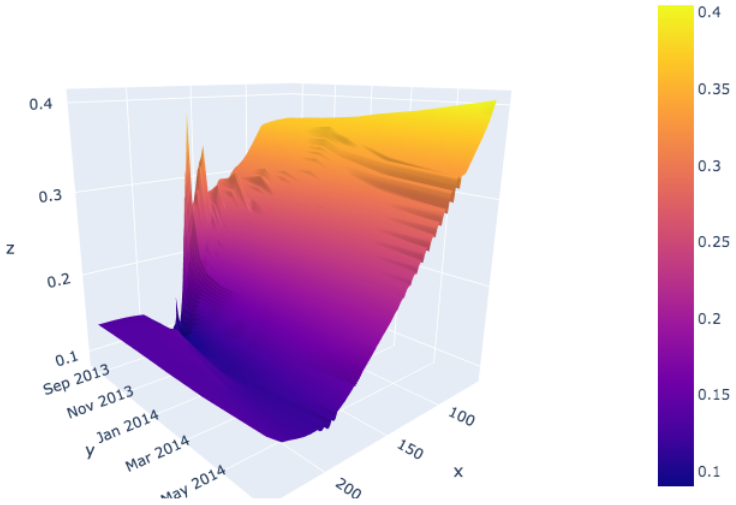
(c) $\Delta = 25$ days

(d) $\Delta = 125$ days

*Histograms for various lags $\Delta$ of the (overlapping) increments $log\sigma_{t+\Delta} - log\sigma_t$ of the SP log-volatility; normal fits in red; normal fit for $\Delta = 1$ day rescaled by $\Delta^H$ in blue.*

The slight deviations from the Normal distribution observed in the figure above are again consistent with the computation of the empirical distribution of the increments of a fractional Brownian motion on a similar number of points.

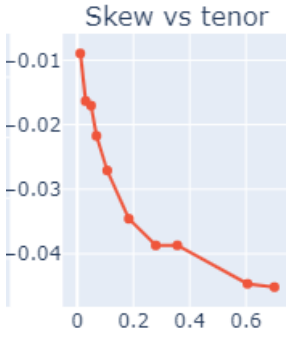
Next, we explored the data and we plotted the surface volatility surfaces for different assets and on multiple dates.




*SPY volatility surface on 14-08-2013*

It is a stylized fact that, at least in equity markets, although the level and orientation of the volatility surface do change over time, the general overall shape of the volatility surface does not change, at least to a first approximation. This suggests that it is desirable to model volatility as a time-homogenous process, i.e. a process whose parameters are independent of price and time. However, conventional time-homogenous models of volatility such as the Hull and White, Heston, and SABR models do not fit the volatility surface.

Using empirical data we plotted the term structure of at-the-money volatility skew and we observed that it is well-approximated by a power-law function of time to expiry . In contrast, conventional stochastic volatility models generate a term structure of at-the-money (ATM) skew that is constant for small and behaves as a sum of decaying exponentials for larger .




*The term structure of at-the-money volatility skew of SPY on 07-01-2013*

To sum up, a volatility model driven by fBM with $H < 1/2$, therefore has the potential to be not only consistent with the empirically observed properties of the volatility time series but also consistent with the shape of the volatility surface.

# 4 Option pricing under Rough Volatility model

In the previous section, we showed that log-volatility behaves mostly as a fractional Brownian motion, that being at any time scale that is reasonable enough.
We will now try to show how to use it to price on both the underlying and integrated volatility. We will first attempt to tackle the pricing model using a method similar to the one used in normal stochastic volatility model. We will explain the problem we faced in this method and propose an implementation of the well documented rBergomi model.

## 4.1 Simplistic representation model

### 4.1.1 Presentation and implementation

The idea here is to use a protocol similar to the one used for stochastic volatility. We have:

$$\mathrm{d}log(S_t) = \sigma_t \mathrm{d}Z_t$$

$$\mathrm{d}log(\sigma_t) = \eta \mathrm{d}W_t^H$$

Where $Z_t$ is a brownian motion and $W_t$ is a fractional brownian motion correlated to $Z_t$.
We then use Monte Carlo to diffuse scenarios/paths for these two random variables using the Cholesky decomposition. Therefore we get scenarios for $\sigma_t$ and $S_t$ and inverse Black-Scholes to get the implied volatility. However Monte Carlo will help us only to diffuse random processes with known covariance matrix, but will not handle the correlation part.
In other models, we used a technique to build a second brownian motion that is correlated to a first one: If $B^1$ and $B^2$ are two independent brownien motions, then $B_t^3 = \rho B_t^1 + \sqrt{1-\rho^2} B_t^2$ is also a brownian motion with a correlation of $\rho$ to $B^1$. We can obviously generalise this to a multi-dimensional brownian motion.

However since the sum of two fractional brownian motion a is not necessarily a fractional brownian motion, this technique can not be used.
Our idea was to generate the two random variables $Z_t$ and $W_t$ together, ie using the same Lower-Triangular matrix L from the Cholesky decomposition:
We consider a time interval [0, T] and fix an equidistant partition $0 = t_0 \leq t_1 \leq ... \leq t_n = T$.

$$\mathbf{E}[Z_{t_i} Z_{t_j}] = \min(t_i, t_j)$$

$$\mathbf{E}[W_{t_i} W_{t_j}] = \frac{1}{2}(t_i^{2H} + t_j^{2H} - \mid t_i - t_j \mid^{2H})$$

H being the Hurst parameter of the fractional brownian motion W. Let's also assume a correlation of $\rho$ between the two processes.
To be able to use the Cholesky decomposition, we need to have a covariance matrix, a symmetric semi definite matrix. This matrix should contain the two covariance matrices of $Z_t$ and $W_t$ as well as the correlation between them.

$$\Sigma = \begin{bmatrix} \Sigma_W & (\rho)_{i,j} \\ (\rho)_{i,j}^T & \Sigma_Z \end{bmatrix}$$

Where $\Sigma_Z$ is the matrix of $(\mathbf{E}[Z_{t_i} Z_{t_j}])_{i,j}$, $\Sigma_W$ is the matrix of $(cov(W_{t_i} W_{t_j}])_{i,j}$ and $(\rho)_{i,j} = \rho t_j^{\frac{1}{2}} t_i^H$
In fact,

$$cov(W_{t_i}, Z_{t_j}) = corr(W_{t_i}, Z_{t_j}) \sigma_{W_{t_i}} \sigma_{Z_{t_j}} = \rho t_j^{\frac{1}{2}} t_i^H$$

Assuming $\Sigma$ is semi-definite positive, we can decompose $\Sigma = LL^T$ where L is lower-triangular. We then generate a 2n size vector $X = (X_1, ..., X_{2n})$ of independent standard Gaussian random variables. The idea is that we should get

$$(W_{t_1}, ..., W_{t_n}, Z_{t_1}, ..., Z_{t_n})^T = LX$$

Now in this construction, we don't guarantee that $\Sigma$ is semi-definite positive. To mitigate this problem, we use a technique presented in [2] N.J. Higham, "Computing a nearest symmetric positive semi-definite matrix" (1988) that uses the Singular Value Decomposition to analytically calculate the nearest semi-definite positive matrix to $\Sigma$, nearest being in terms of the Frobenius norm:

Let $U * diag(s) * V$ be the Single Value decomposition of $\Sigma^* = \frac{1}{2}\Sigma\Sigma^T$. Let $\Sigma^{**} = V^T * diag(s) * V$. We can show that $\Sigma^{***} = \frac{1}{4}(\Sigma^{*T} + \Sigma^* + \Sigma^{**T} + \Sigma^{**})$ is either semi-definite positive, or we iterative add $\xi \min(\lambda)\mathbf{I_n}$ where $\lambda$'s are the eigenvalues of the previous $\Sigma^{***}$ and $\xi$ a factor, until we converge to a semi-definite positive $\Sigma^{***}$.

### 4.1.2 Results and review

After generating paths for $\sigma_t$ ans $S_t$, we can inverse Black-scholes to get:

- plot_iv_skew
- plot_term_structure
- plot_vol_surface

Using this model, and as can be seen in these figures, we got a "no result", we don't fit at all observed volatilities:

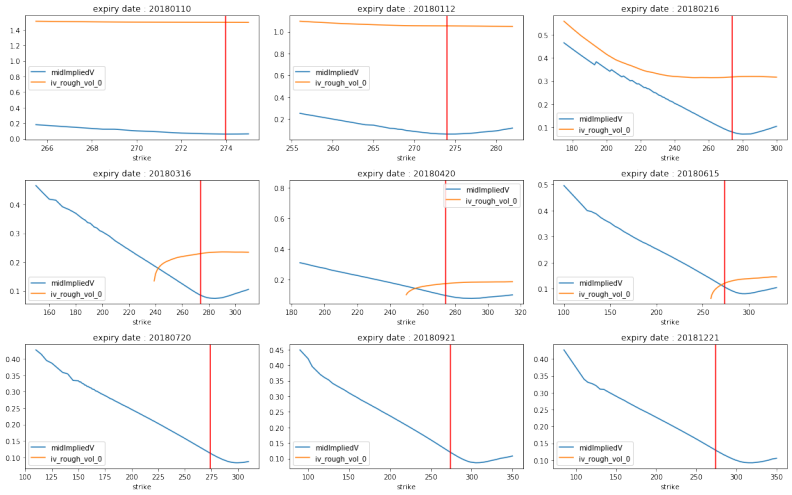



Figure 1: Implied Volatility Skews

In fact, the simplistic aspect of this model comes for the fact that we represented the fractional brownian motion with a simplistic method using the covariance matrix and the Cholesky decomposition, whereas as we will see in the next section, we need a whole new representation for the fractional brownian motion.

On the other hand, using a "close" matrix instead of the real covariance matrix that we need, "close" being in the Frobenius norm sens, gives no guarantees that the resulting processes follow the instructed covariances and inter-correlation. Indeed we have no inequality that bounds the error in covariance by the norm of the difference of the two matrices.

## 4.2 rBergomi Model

### 4.2.1 Presentation

From the studies we showcases about rough volatility, there are two important regularities about the daily spot volatilities we concluded:

- The distributions of increments of log volatility are close to a Gaussian distribution.
- for reasonable timescales of practical interest, the time series of volatility was found to be consistent with the simple model:

$$\log(\sigma_{t+\Delta}) - \log(\sigma_t) = \nu(W^H_{t+\Delta} - W^H_t)$$

  where $W^H$ is fractional brownian motion. This relationship was proven to be consistent across different products from different asset classes (equities, bonds, commodities, ..).

In the rBergomi model, we will consider the Mandelbrot-Van Ness representation of fractional brownian $W^H$ in terms of Wiener integrals:

$$W^H_t = C_H \left\{ \int_{-\infty}^t \frac{dW^{\mathbb{P}}_s}{(t-s)^\gamma} - \int_{-\infty}^0 \frac{dW^{\mathbb{P}}_s}{(-s)^\gamma} \right\}$$

where $\gamma = \frac{1}{2} - H$ and $C_H = \sqrt{\frac{2H\Gamma(3/2-H)}{\Gamma(H+1/2)\Gamma(2-2H)}}$ which implicate the needed formula of the covrariance of a fractional brownian motion of Hurst parameter H.
So then:

$$\begin{aligned}
\log(v_u) - \log(v_t) &= 2(\log(\sigma_u) - \log(\sigma_t)) \\
&= 2\nu C_H \left\{ \int_{-\infty}^u \frac{dW^{\mathbb{P}}_s}{(u-s)^\gamma} - \int_{-\infty}^t \frac{dW^{\mathbb{P}}_s}{(t-s)^\gamma} \right\} \\
&= 2\nu C_H \left\{ \int_t^u \frac{dW^{\mathbb{P}}_s}{(u-s)^\gamma} + \int_{-\infty}^t \frac{dW^{\mathbb{P}}_s}{(u-s)^\gamma} - \frac{dW^{\mathbb{P}}_s}{(t-s)^\gamma} \right\} \\
&= 2\nu C_H \left\{ M_t(u) + Z_t(u) \right\}
\end{aligned}$$

Where $Z_t(u) = \int_{-\infty}^t \frac{dW^{\mathbb{P}}_s}{(u-s)^\gamma} - \frac{dW^{\mathbb{P}}_s}{(t-s)^\gamma}$ is $\mathbf{F}_t$-measurable and $M_t(u) = \int_t^u \frac{dW^{\mathbb{P}}_s}{(u-s)^\gamma}$ is independent of $\mathbf{F}_t$ and Gaussian with mean zero, and variance $(u-t)^{2H}/(2H)$.
Finally,

$$v_u = v_t \exp(\eta \widetilde{W}^{\mathbb{P}}_t(u) - 2\nu C_H Z_t(u))$$

with $\widetilde{W}^{\mathbb{P}}_t(u) = \sqrt{2H} \int_t^u \frac{dW^{\mathbb{P}}_s}{(u-s)^\gamma}$
And so:

$$v_u = \mathbb{E}^{\mathbb{P}}[v_u \mid \mathbf{F}_t] \exp(\eta \widetilde{W}^{\mathbb{P}}_t(u) - \frac{1}{2}\mathbb{E}[\mid \eta \widetilde{W}^{\mathbb{P}}_t(u) \mid^2])$$

This computation reveals that the conditional distribution of $v_u$ depends on $\mathbf{F_t}$ only through the variance forecasts $\mathbb{E}^{\mathbb{P}}[v_u|\mathbf{F}_t]$. In particular, to price options, we don't need to know $\mathbf{F}_t$, ie the entire

history of the brownian motion for $s \leq t$.
Let's consider some general change of measure:

$$dW_s^{\mathbb{P}} = dW_s^{\mathbb{Q}} + \lambda(s)ds$$

where $\lambda(s)$ is a deterministic function of s. We get then for the volatility:

$$\begin{aligned} v_u = v_u &= \mathbb{E}^{\mathbb{P}}[v_u \mid \mathbf{F}_t] \exp(\eta \widetilde{W}_t^{\mathbb{Q}}(u) - 1/2\mathbb{E}[\mid \eta \widetilde{W}_t^{\mathbb{Q}}(u) \mid^2]) \exp(\eta\sqrt{2H} \int_t^u \frac{\lambda(s)ds}{(u-s)^\gamma}) \\ &= \mathbb{E}^{\mathbb{Q}}[v_u \mid \mathbf{F}_t] \exp(\eta \widetilde{W}_t^{\mathbb{Q}}(u) - 1/2\mathbb{E}[\mid \eta \widetilde{W}_t^{\mathbb{Q}}(u) \mid^2]) \end{aligned}$$

We can see that $\mathbb{E}^{\mathbb{Q}}[v_u \mid \mathbf{F}_t]$ is the product of $\mathbb{E}^{\mathbb{P}}[v_u \mid \mathbf{F}_t]$ which depends on the history of the driving brownian motion as explained earlier, and a term which depends on the price of risk $\lambda(s)$. This volatility model is a non-Markovian generalization of the Bergomi model called rough Bergomi (or rBergomi) model. In this model we are replacing the exponential kernels in the Bergomi model with a power-law kernel. We may therefore expect that the rBergomi model will generate a realistic term structure of ATM volatility skew.

As for the pricing model in general, we will use this representation for the volatility, and to reflect the observed anticorrelation between price moves and volatility moves we will use a technique similar to the one used in the "Simplistic model representation".

### 4.2.2 Simulation

The model to be simulated is :

$$S_t := S_0 \exp\left\{\int_0^t \sqrt{V_u}\mathrm{d}B_u - \frac{1}{2}\int_0^t V_u \mathrm{d}u\right\}, \quad B_u := \rho W_u^1 + \sqrt{1-\rho^2}W_u^2$$

$$V_u := \xi \ \exp\left\{\eta \widetilde{W}_u^H - \frac{\eta^2}{2}u^{2H}\right\}, \quad \widetilde{W}_u^H := \sqrt{2H}\int_0^u (u-s)^{-\gamma}\mathrm{d}W_s^1$$

$\widetilde{W}$ is a Volterra process with mean zero and $Var[\widetilde{W}_u] = u^{2H}$

$$\begin{aligned} Var[\widetilde{W}_u] &= Var[\sqrt{2H}\int_0^u (u-s)^{-\gamma}\mathrm{d}W_s^1] = 2H\int_0^u (u-s)^{-2\gamma}\mathrm{d}t \\ &= 2H * \frac{1}{-2\gamma+1}[-(u-s)^{-2\gamma+1}]_0^u = u^{-2\gamma+1} = u^{2H} \end{aligned}$$

#### 4.2.2.1 Simulation of Volterra process with cholesky decomposition :

So far $\widetilde{W}$ behaves just like a fBm. However, the dependence structure is different. Specifically, for $v \geq u$,

$$E[\widetilde{W}_v \widetilde{W}_u] = u^{2H} G(\frac{u}{v})$$

where for $x \leq 1$

$$G(x) = 2H\int_0^1 \frac{1-s^\gamma}{x-s^\gamma} = \frac{1-2\gamma}{1-\gamma}x^\gamma {}_2F_1(1,\gamma,2-\gamma,x)$$

where ${}_2F_1(\Delta)$ denotes the confluent hypergeometric function.

We also need covariances of the Brownian motion Z with the Volterra process $\widetilde{W}$ . With v $\geq$ u, these are given by :

$$E[\widetilde{W}_v Z_u] = \rho D_H (v^{H+\frac{1}{2}} - (v-u)^{H+\frac{1}{2}})$$

and,

$$E[Z_v \widetilde{W}_u] = \rho D_H u^{H+\frac{1}{2}}$$

where,

$$D_H = \frac{\sqrt{2H}}{H+\frac{1}{2}}$$

These two formulae may be conveniently combined as

$$E[\widetilde{W}_v Z_u] = \rho D_H (v^{H+\frac{1}{2}} - (v - min(u,v))^{H+\frac{1}{2}})$$

Lastly, the covariance function of the brownian motion is of course

$$E[Z_u Z_v] = min(u,v)$$

With m the number of time steps and n the number of simulations, our rBergomi model simulation algorithm using cholesky decomposition may then be summarized as follows :

1. Construct the joint covariance matrix for the Volterra process $\widetilde{W}$ and the Brownian motion Z and compute its Cholesky decomposition.
2. For each time, generate iid normal random vectors and multiply them by the lower-triangular matrix obtained by the Cholesky decomposition to get a 2n x m matrix of paths of $\widetilde{W}$ and Z with the correct joint marginals.
3. With these paths held in memory, we may evaluate the expectation under P or Q of any payoff of interest.

This method was implemented in our work but we noticed that the simulation is very slow. This is not surprising because of the high number of matrix-vector multiplications with a lower triangular but otherwise dense matrix.
A more efficient method of simulation was proposed afterwards which we will be described in the next sub-section. All the results and the comments will be presented later.

#### 4.2.2.2 Simulation of Volterra process with the hybrid BSS scheme :

A Brownian Semi-stationary (BSS) process is defined as :

$$X(t) = \int_{-\infty}^{t} g(t-s)\sigma(s)dW(s)$$

where $\sigma = \sigma(t)$; t $\in$ R is an $\mathbf{F}_t$-predictable process with locally bounded trajectories and g : [0, $\infty$) $\rightarrow$ [0,$\infty$) is a Borel measurable kernel function.
The Volterra process is a special case of a Brownian Semistationary (BSS) process with a power law kernel and $\sigma(s) = 1 \; \forall \; s$. It is called truncated Brownian semistationary (T BSS) process, as $\widetilde{W}$ is obtained from the BSS process X by truncating the stochastic integral at 0.
In order to simulate the process $\widetilde{W}_u$ efficiently and accurately, we utilise the first order variant ($\kappa = 1$) of the hybrid scheme (Bennedsen, Lunde and Pakkanen, 2017) [3], which is based on the approximation

$$\widetilde{W}^H_{\frac{i}{n}} \approx \widetilde{\widetilde{W}}^H_{\frac{i}{n}} := \sqrt{(2H)} \int_{\frac{i-1}{n}}^{\frac{i}{n}} (\frac{i}{n} - s)^{-\gamma} \mathrm{d}W^1_u + \sum_{k=2}^{i} (\frac{b_k}{n})^{-\gamma} (W^1_{\frac{i-(k-1)}{n}} - W^1_{\frac{i-k}{n}})$$

where

$$b_k = (\frac{k^{1-\gamma} - (k-1)^{1-\gamma}}{1-\gamma})^{-\frac{1}{\gamma}}$$

We employ the fast Fourier transform to evaluate the sum in the formula, which is a discrete convolution, a skeleton $\widetilde{\widetilde{W}}^H_0, \widetilde{\widetilde{W}}^H_{\frac{1}{n}}, ..., \widetilde{\widetilde{W}}^H_{\frac{\lfloor nt \rfloor}{n}}$ can be generated in $O(nlogn)$ floating point operations.

### 4.2.3 Results and comments

The results that will be presented below are based on simulations with the following parameters :

- H = 0.07
- $\eta = 1.9$
- $v_0 = 0.15^2 = 0.0225$
- rho = -0.9
- T = 1
- $S_0 = 100$
- m = 100
- N = 100000

**4.2.3.1 Methods of simulations comparison :** We implemented both of the two methods of Volterra process simulation. We plot the implied volatility curve obtained by each simulater :

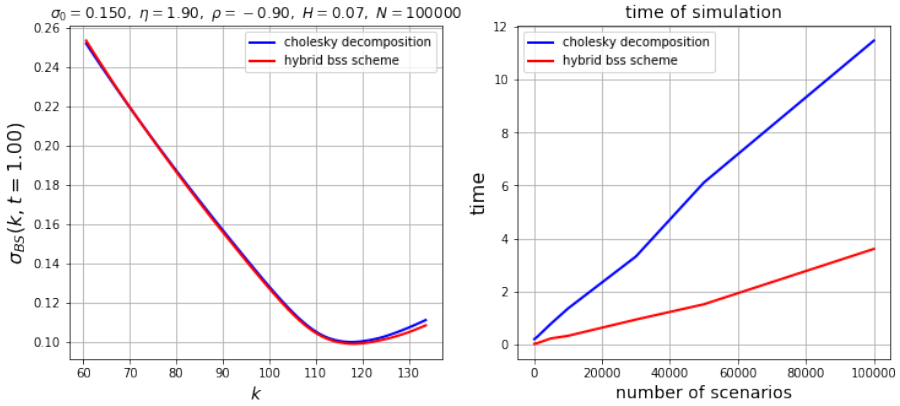



Figure 2: Simulations method comparison

We verify that both methods leads to the same results regarding the implied volatility curve. However, time of simulations of the hybrid scheme method is much lower than the cholesky decomposition method which demonstrates the effeciciency of the first.

**4.2.3.2 Verification of statistical properties :** Now we verify some properties of the processes $\widetilde{W}_t^H$, $V_t$ and $S_t$.
First we compute moments of the Volterra process,

$$\mathbb{E}[\widetilde{W}_t^H] = 0, \quad \mathbb{V}\left[\widetilde{W}_t^H\right] = t^{2H}$$

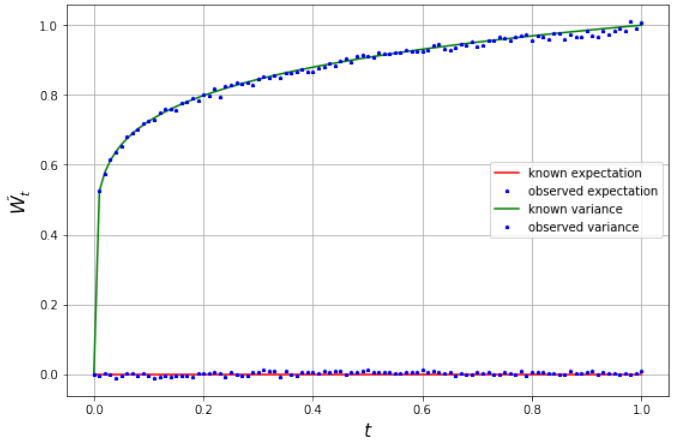



Figure 3: Volterra process properties

Now we check the expectation of the variance process,

$$\mathbb{E}[V_t] = \xi = v_0 = 0.0225$$

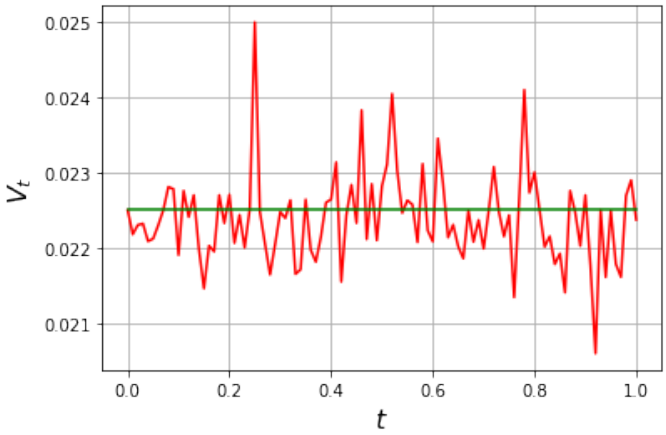



Figure 4: variance process expectation

Finally, we check the log-price process's expectation,

$$-2\mathbb{E}[\log \frac{S_t}{S_0}] = \int_0^t \mathbb{E}[V_t]\mathrm{d}u = \xi * t$$

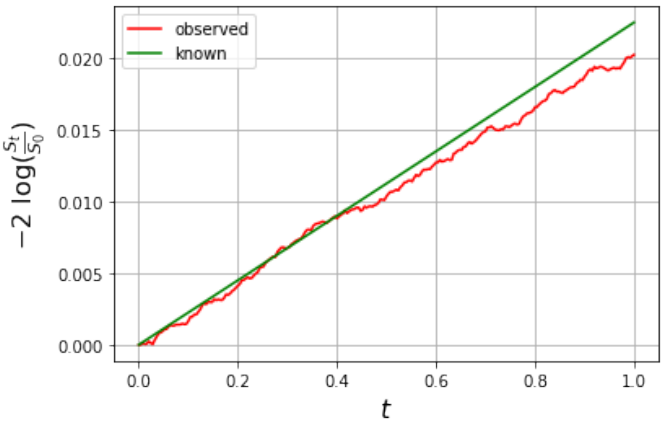



Figure 5: log-price process's expectation

**4.2.3.3 Results and parameters effects :** We demonstrate in the following plot the Volterra process sample paths, which lead directly to the rBergomi variance and price sample paths. We compare the paths for two different values of H : 0.5 and 0.07.

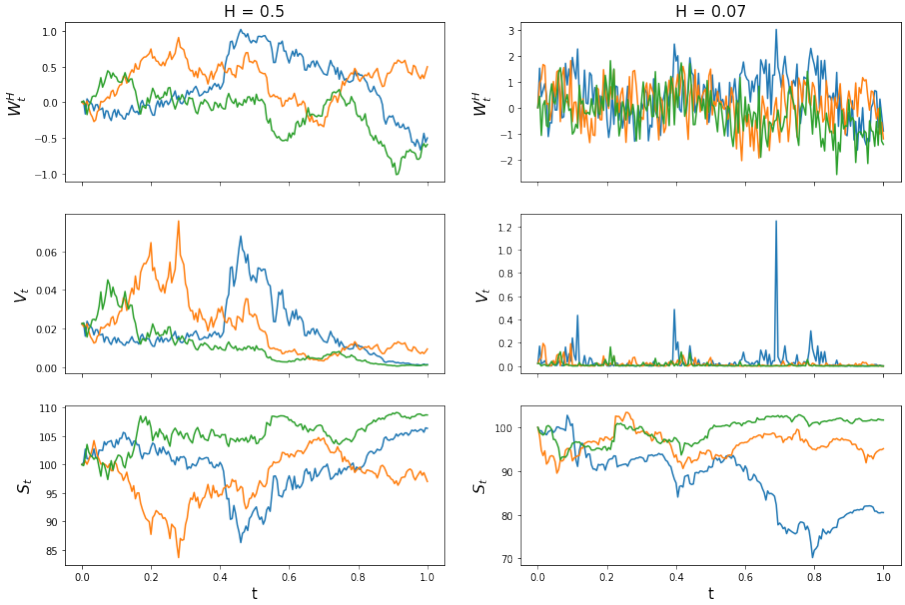



Figure 6: log-price process's expectation

The Volterra process with H = 0.5 corresponds to a standard brownian motion. using $H << 0.5$ demonstrates the roughness of the paths for the Volterra process and the variance process. When H = 0.07, a greater variance is exhibited for short times. This explosive short-time variance, generated when H is close to 0, leads to short-time implied volatilities observed in practice. The price process, despite being a continuous martingale, exhibits jump-like behaviour when the Volterra, thus variance, process peaks.

We will illustrate now the effect of each parameter H, $\eta$, rho and $\sigma_0$ on the implied volatility curve :

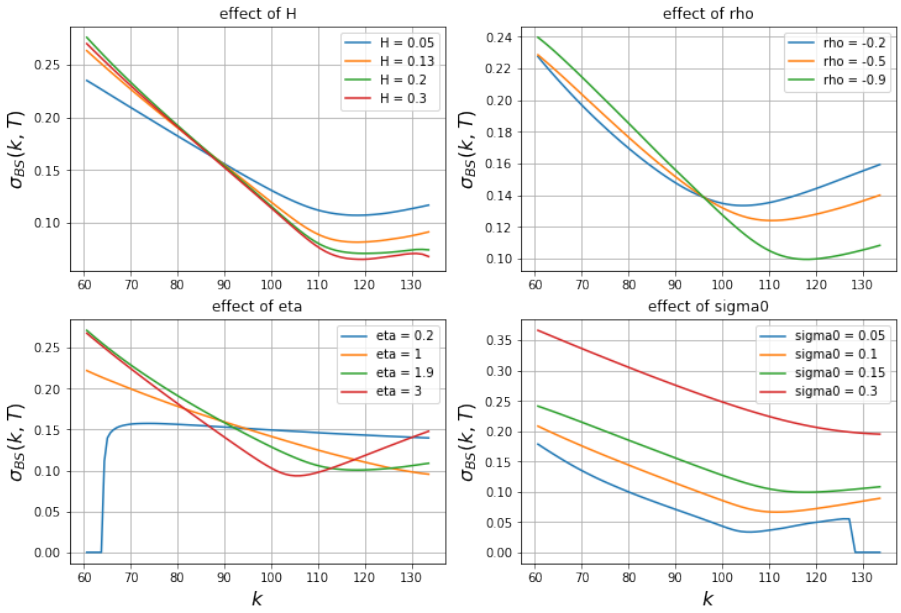



Figure 7: Effect of parameters on implied volatility curve

Our observations are :

- H controls only the skew of the implied volatility curve.
- rho controls the convexity and the skew at the same time.
- $\eta$ controls mainly the convexity of the curve.
- $v_0$ controls only the level of the implied volatility.

The parameters of H $= 0.07$, $\eta = 1.9$ and $\rho = 0.9$ which are used in our simulations are demonstrated by Bayer, Friz and Gatheral (2016) [4] to be remarkably consistent with the SPX market on February, $4^{th}$ 2010. We test these parameters on our real data of the SPY market on the same date to confirm the statement :

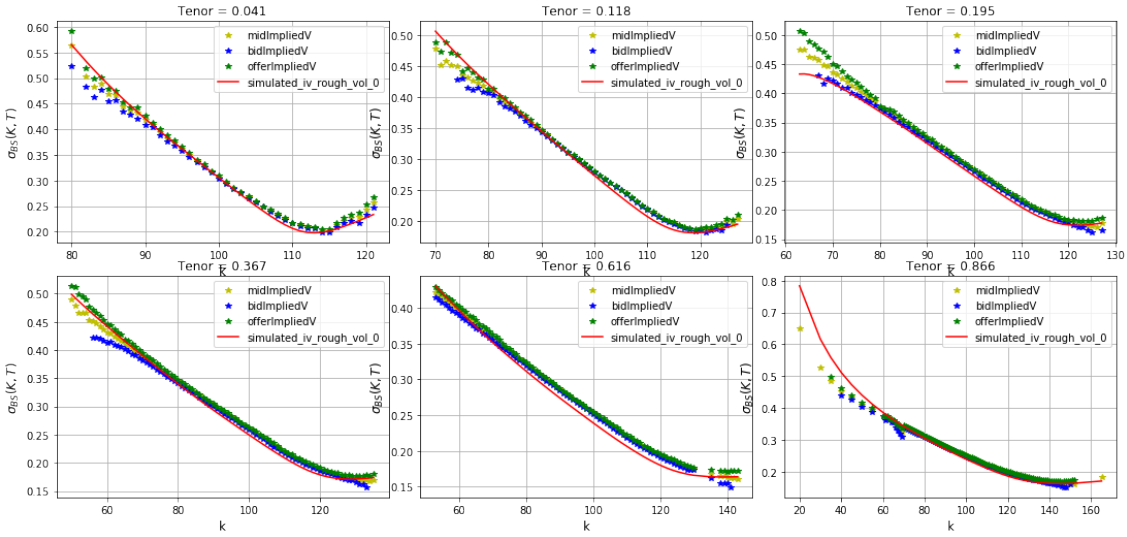



Figure 8: Implied volatilities per tenor for SPY as of 2010-02-04 : Blue, yellow and green points represent bid, mid and offer SPY implied volatilities; red smile is from the rBergomi simulation.

We obtained a good fit of the model on the real implied volatility data using these parameters. In the next section, we will explore the calibration in order to find ourselves the best parameters of the model to fit real data for several dates.

To verify the consistency of the simulated ATM skew term structure with the real one :

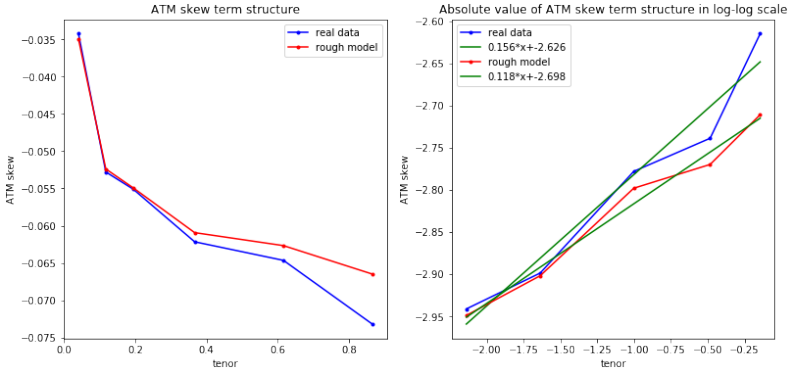



Figure 9: ATM skew term structure for SPY as of 2010-02-04

We compute the ATM skew term structure empirically by computing the derivative on monyness zero with finite difference. We take two points near the monyness 0 and use them to compute the slope of the curve there.
The rough bergomi model fits better the ATM skew term structure than the other stochastic volatility models.
We observe a power law coefficient which is equal to -0.15 which is between 0 and 0.5 unlike the conventional stochastic volatility models whose ATM skew term structures are inversely proportional to T.

# 5 Calibration

The calibration is done on the SPY options data.
In the last section, we saw for fixed parameters H, $\eta$, $\rho$ and $\sigma_0$ we can generate an implied volatility curve. We will try to calibrate our model, i.e, find the best parameters which fit well the implied volatility smiles: we will try to minimise the L2-error between the real implied volatility points and the simulated ones.

For a certain date, we can do a calibration per tenor, ie just the volatility smile (find the best parameters for each tenor) or a calibration on all the volatility surface (a fixed set of parameters for all the tenors)

The problem of optimisation is the followilng :

- Optimisation for a fixed tenor :

$$min_{H,\eta,\rho} \sum_{k} (\hat{\sigma}_{BS}(k, tenor) - MidIv(k, tenor))^2$$

- Optimisation for all the volatility surface (all the tenors) :

$$min_{H,\eta,\rho} \sum_{tenor} \sum_{k} (\hat{\sigma}_{BS}(k, tenor) - MidIv(k, tenor))^2$$

  where,

$$H \in [0.03, 0.3]$$
$$\eta \in [1.5, 3.5]$$
$$\rho \in [-0.6, -0.9]$$

This is a black-box optimization problem since our objective function doesn't have an analytical form.
Regarding the value of $\sigma_0$, we can estimate it by taking the real ATM implied volatility and slightly adjust its value (we added 0.01 to this value for a better fit).

We observed that sometimes we had Null implied volatility values for lower strikes for some set of parameters while the other points fits well the rest of the curve.
To deal with this problem and ensure a complete fit of the volatility smile, we introduced a penalisation term with the inverse of the number of the non null values.
The problem of optimisation becomes as below :

$$min_{H,\eta,\rho} \quad Error + c * (\frac{1}{nb\ points})$$

c was an hyper parameter to tune. We decided to take c equal to 0.01 .

## 5.1 Optimisation methods

Another observation while working on the optimization methods was that for the same set of input parameters there may be changes in the output.

Since we don't use a very big number of scenarios ( N between 10000 and 100000, otherwise the task will be really heavy), we observe more stochasticity as the variance of the Monte Carlo estimator

for the option pricing is higher. Our Objective function (OF) in this case is not deterministic and the classical deterministic optimisation methods might struggle to find the global minimum of our OF.

In the scipy optimisation package we tried some Bound-Constrained minimization methods like SLSQP, COBYLA, L-BFGS-B. However, they didn't really work well due to the stochasticity of our OF.
Some other inconvenients of these methods is that they need a good initial guess, a lot of hyper-parameters tuning, and some of them need the OF to be continuously differentiable, but, our OF is likely not "smooth": there could be discontinuities in the function or its derivatives.

So, we will use another class of optimisation methods which are called stochastic optimisation methods and in particular the Differential Evolution algorithm.
This method is costly in time and needs a lot of OF evaluations, but it showed very promising results.

We will present below the Differential Evolution algorithm :

Formally, let $f : \mathbb{R}^n \to \mathbb{R}$ be the objective function which must be minimized (note that maximization can be performed by considering the function h := -f instead). The function takes a candidate solution as argument in the form of a vector of real numbers and produces a real number as output which indicates the fitness of the given candidate solution. The gradient of f is not known. The goal is to find a solution m for which $f(\mathbf{m}) \leq f(\mathbf{p})$ for all p in the search-space, which means that m is the global minimum.
Let $\mathbf{x} \in \mathbb{R}^n$ designate a candidate solution (agent) in the population. The basic DE algorithm can then be described as follows:

- Initialize all agents x with random positions in the search-space.
- Until a termination criterion is met (e.g. number of iterations performed, or adequate fitness reached), repeat the following:
  - For each agent x in the population do:
    - Pick three agents a,b, and c from the population at random, they must be distinct from each other as well from agent x
    - Pick a random index $R \in \{1, \ldots, n\}$ where n is the dimensionality of the problem being optimized.
    - Compute the agent's potentially new position $\mathbf{y} = [y_1, \ldots, y_n]$ as follows:
      - For each $i \in \{1, \ldots, n\}$, pick a uniformly distributed random number $r_i \sim U(0, 1)$
      - If $r_i < CR$ or i = R then set $y_i = a_i + F \times (b_i - c_i)$ otherwise set $y_i = x_i$. The parameter CR $\in [0, 1]$ is called the crossover probability and the parameter $F \in [0, 2]$ is called the differential weight, both these parameters must be set by the user along with the population size NP $\geq 4$ and may greatly impact the optimization performance.
    - If $f(\mathbf{y}) \leq f(\mathbf{x})$ then replace the agent x in the population with the improved or equal candidate solution y .
- Pick the agent from the population that has the best fitness and return it as the best found candidate solution.

## 5.2 Results

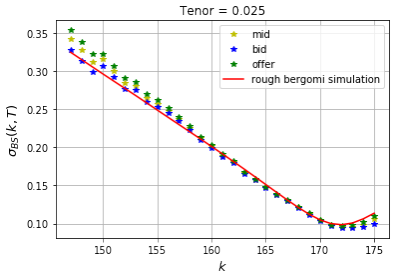
### 5.2.1 Fixed tenor

We chose randomly two dates on which we will calibrate the model per tenor : 2013-08-14 and 2015-03-02. We will summarize the results of the calibration in the following tables :

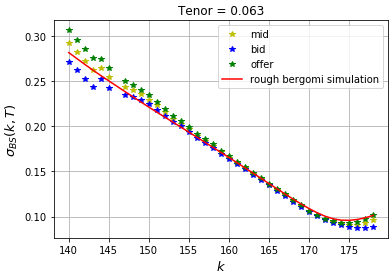
- 2013-08-14 :

| Tenor | H | $\eta$ | $\rho$ | L2_error |
|---|---|---|---|---|
| 0.025 | 0.13965069 | 3.03526396 | -0.8270291 | 5.9914e-05 |
| 0.063 | 0.08782856 | 2.07209507 | -0.84040342 | 2.1480e-05 |
| 0.1 | 0.096831 | 1.96787099 | -0.850588 | 8.8134e-05 |
| 0.427 | 0.14550376 | 1.86616789 | -0.70979479 | 1.0803e-05 |
| 0.849 | 0.10282982 | 1.69869223 | -0.77296747 | 1.6204e-05 |

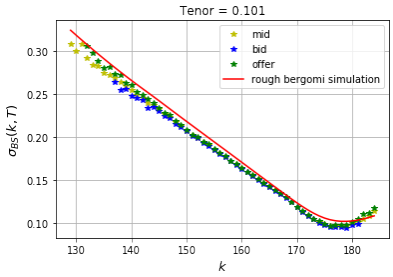
Table 1: Parameters calibration per tenor for 2015-03-02



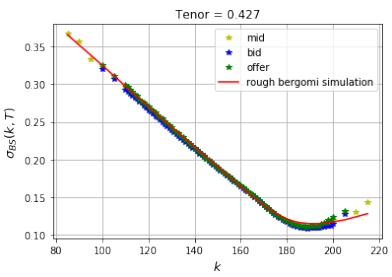

(a) Tenor = 0.03



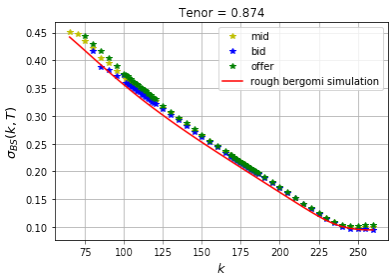

(b) Tenor = 0.085




(c) Tenor = 0.1




(d) Tenor = 0.874

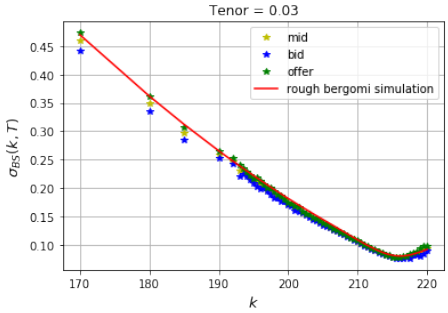



(e) Tenor = 0.874

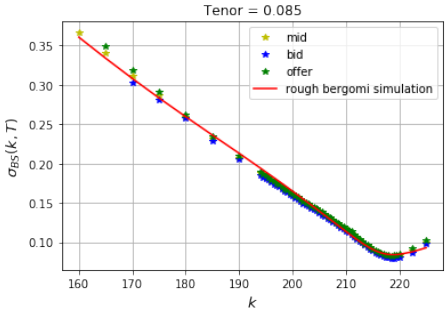
- 2015-03-02 :

| Tenor | H | $\eta$ | $\rho$ | L2_error |
|---|---|---|---|---|
| 0.03 | 0.138426 | 2.65306668 | -0.81658459 | 4.3569e-05 |
| 0.085 | 0.18531902 | 3.08555865 | -0.761033 | 1.4103e-05 |
| 0.3 | 0.15545168 | 2.3363784 | -0.84000553 | 7.5656e-05 |
| 0.874 | 0.12868358 | 1.92031159 | -0.88505979 | 1.34964e-04 |

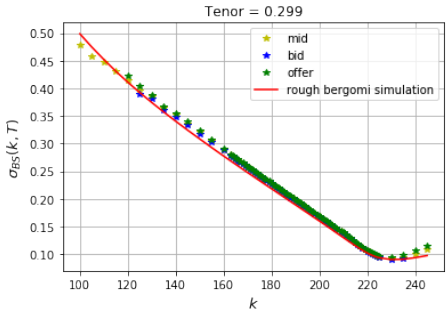
Table 2: Parameters calibration per tenor for 2015-03-02



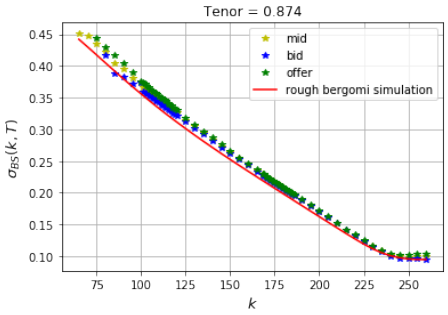

(f) Tenor = 0.03




(g) Tenor = 0.085




(h) Tenor = 0.03




(i) Tenor = 0.874

### 5.2.2 Whole surface calibration

- 2013-08-14 :
  H = 0.08930132, $\eta$ = 2.14441384, $\rho$ = -0.79512602

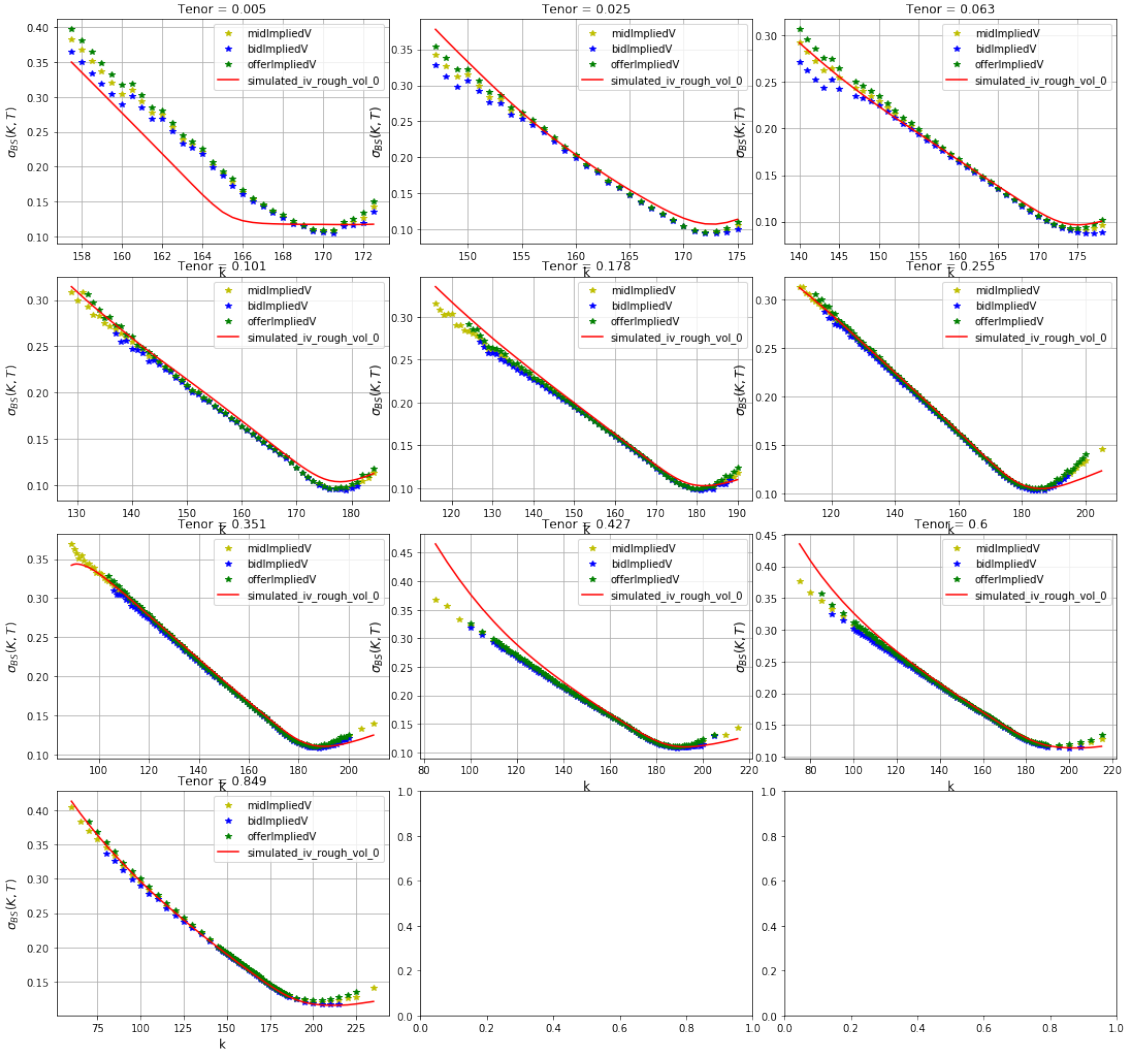



Figure 10: Calibration all volatility surface 2013-08-14

- 2015-03-02 :
  H = 0.12408347, $\eta$ = 2.58549533, $\rho$ = -0.77516331

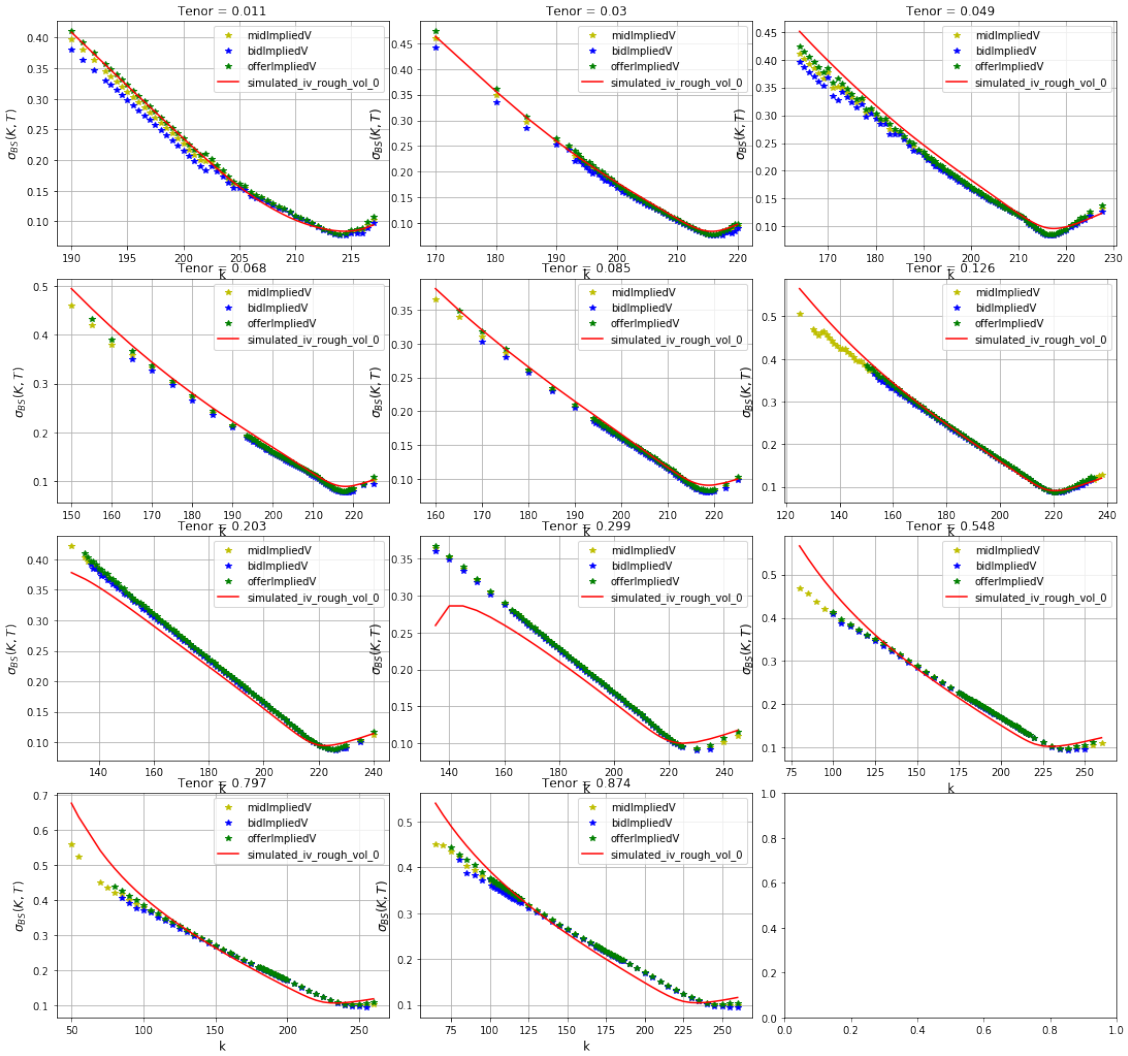



Figure 11: Calibration all volatility surface 2015-03-02

Calibrating all the surface of volatility is a more complicated task than per tenor, the results are less precise. Adding more iterations to the algorithm process may lead to better results.
Other results of calibration on all the volatility surface are in the notebook "rbergomi_calibration.ipynb". We notice that often the early maturities are not well fitted. This is due to the dominance of the other maturities which may have similar curves so that the optimisation tends to fit better those long-time maturities.

# 6 Conclusion

After calibrating the rough volatility model to get the appropriate parameters, we see that we this calibrated model fits the volatility surface better than the conventional stochastic volatility models. For Futher steps, we can analyse the time series of parameters H, $\eta$ and $\rho$ to predict them in the future and thus predict future volatility surfaces.

# References

[1] Mathieu Rosenbaum Jim Gatheral, Thibault Jaisson. Volatility is rough. October 4 2014.

[2] N.J. Higham. Computing a nearest symmetric positive semi-definite matrix. 1988.

[3] Mikko S. Pakkanen Mikkel Bennedsen, Asger Lunde. Hybrid scheme for brownian semistationary processes. Mai 16 2017.

[4] Jim Gatheral Christian Bayer, Peter Friz. Pricing under rough volatility. January 23 2015.