Interest Rates Modeling

Project assignment - SABR calibration

Wednesday, November 23 2022

Team members :

Amal BACHA amalbacha42@gmail.com
Lucas RODRIGUEZ lucas.rodriguez@ensiie.fr

Master in Quantitative Finance (M2QF)

Table of contents

We have used Python as numerical tool for all required computations and to perform model simulations. In order to find our practical approach, please refer to the attached Jupyter Notebook file.

1 General framework

1.1 Introduction

The purpose of this work is to implement and calibrate the SABR volatility model for swaptions in Python. This model provides a close form solution for implied volatility that accurately captures the market smile volatility dynamics. Using market data of forward rates and swaption volatilities, we calibrate the parameters in order to obtain the implied volatilities.

1.2 Theoretical background

1.2.1 Swaptions

A *swaption* is a contract that allows you to enter into an interest rate swap. The time difference between "today" and the option's exercise date is known as *expiry*, while the time difference between the exercise date and the maturity of the swap is known as *tenor*. [Ioa11]

Swaptions introduce a second level of derivative optionality. A European payer swaption is an option to enter, at a date T called the maturity of the swaption, into a payer swap for the period (α, β), of nominal N and strike K. For simplicity, we assume here that $T = T_\alpha$.

At time T_α, the payoff value of the underlying payer swap is written by adapting the equation :

$$\Pi^{SP}(t;\alpha,\beta,N,K) = \sum_{i=\alpha+1}^{\beta} NP\left(T_\alpha, T_i\right)\left(T_i - T_{i-1}\right)\left[L\left(T_\alpha, T_{i-1}, T_i\right) - K\right]$$

where :

- N : the nominal
- $P(t,T)$: the price at time $t \in [0,T]$
- $L(t,T,S)$: the forward rate expiring at date T, with maturity $S > T$ and defined at date t
- K : the swaption's strike

By definition of the swaption, the payoff of the payer swaption is therefore written at date T_α :

$$N\left(\sum_{i=\alpha+1}^{\beta} \tau_i P\left(T_\alpha, T_i\right)\left[L\left(T_a, T_{i-1}, T_i\right) - K\right]\right)^+$$

This payoff at date T_α is rewritten as a function of the swap rate defined as :

$$N\left(\sum_{i=\alpha+1}^{\beta} \tau_i P\left(T_\alpha, T_i\right)\right)\left(S_{\alpha,\beta}\left(T_\alpha\right) - K\right)^+$$

where : $\tau_i = T_i = T_{i-1}$ and $S_{\alpha,\beta}(t)$ is the forward swap rate

The receiving swaptions are symmetrically defined. Swaptions are structurally very sensitive to the correlation of LIBOR rates of different maturities.

1.2.2 SABR model

The SABR model is a stochastic volatility model in mathematical finance that attempts to capture the volatility smile in derivatives markets. The name is an abbreviation for **"Stochastic Alpha, Beta, Rho"**, which refers to the model's parameters. The SABR model is widely used by financial industry practitioners, particularly in interest rate derivative markets.

For a given tenor, we compute the SABR implicit volatilities using :

- Forward rate
- Expiry of the option
- Option strike (for example ATM + 50bps)
- Values for the parameters α, β, ρ, ν

The SABR model [Wik] [Rod19] is a dynamic model in which both F and σ are represented by stochastic state variables, the time evolution of which is given by the following system of stochastic differential equations :

$$dF_t = \sigma_t (F_t)^{\beta} dW_t,$$
$$d\sigma_t = \alpha \sigma_t dZ_t$$

where W_t and Z_t are two correlated Wiener processes with correlation coefficient ρ such that :

$$dW_t dZ_t = \rho dt$$

1.2.3 Calibration of parameters

To calibrate the model with respect to market data, our goal is to minimize the following function :

$$\sqrt{\sum_{i=-x}^{x} (\sigma_{SABR,i} - \sigma_{MKT,i})^2}$$

with :

- i is the strike index
- $\alpha > 0$
- $0 \leq \beta \leq 1$
- $-1 \leq \rho \leq 1$
- $\nu > 0$

$\Longrightarrow$ The sum is computed along the swaption's **volatility smile**.

1.3 Problem statement

Our main objective is studying in detail the SABR model and determining its superior performance. We want to find the parameters (for each smile) that bring model volatilities as close to market volatilities as possible. To do that, we start by computing the implied volatilities for a single swaption, for a given "smile" and for all combinations of swaptions. We also used the *shifted model* to calibrate the whole model.

2 Modeling & Results

2.1 Preliminary pre-processing

The SABR Python script was originally given in Python 2.X. In order to perform the required computations and plotting, we had to transform most of the features to make it valid in Python 3.X.

- The given Excel file **market_data.xlsx** were transformed from a **.xlsx** file to a **.xls** (downgraded version) in order to fit the original code version.
- The **print** statements were also transformed
- Unnecessary disk-writing methods were removed in order to be fully processed

Finally, we have engineered a feature function which maps the real column indices from the dataframe corresponding to the strike columns to indices in $[\![-4, 4]\!]$.

2.2 Calibration with SABR model

2.2.1 SABR Implied Volatilities

The first thing that was done was calculating the SABR implied volatilities using Hagan's formula. Our first function computes the implied volatilities for a **single swaption**, where the input arguments are the model parameters, the forward rate, the strike of the option, its expiry and the market volatility. Output values were set to zero in the case of negative forward rate where the implied volatility can not be computed.

In order to capture the volatility smile in derivative markets, we must compute the implied volatilities for a **given *smile***. The input arguments are all scalars, the same as they were in the previous function, except for the strike K and the market volatility MKT where they were introduced as vectors.

For a general case, we computed the implied volatilities for all combinations of swaptions where the difference here was that the strike and the market volatilities are input into the model as matrices.

2.2.2 Shifted Model

The shifted SABR model is a popular SABR model extension for **negative interest rates**, in which the shifted forward rate is assumed to follow a SABR process. In fact, the strike can become negative when the forward rate is close to zero. In this case, the shifted model allows for parameter calibration and computation of implied volatilities for the entire smile. The size of the shift is determined by the lowest strike, and its value is added to all strikes.

2.2.3 Calibration

For the calibration part, we want to find the parameters, for each smile, that bring model volatilities as close to market volatilities as possible. Hence, we introduce the objective function that computes the volatilities for a single smile using arbitrary parameters, then sums the squares of all market differences. The final quantity that we want to minimize is its square root.

For the optimization part, we used **SLSQP** method (Sequential Least Squares Programming) [Sci22] for constrained minimization of multivariate scalar functions, where the objective function is minimized with parameter starting values.

This function takes, as arguments, the objective function that needs to be minimized, the starting parameters, the other parameters needed for the optimization and the bounds which were specified in the theoretical background.

2.3 Results & Discussion

We begin by creating the output files, which will be overwritten each time the script is run.

- **outvol.csv** : contains the volatilities computed.
- **vol_diff.csv** : the difference between the SABR and market volatilities.
- **parameters.csv** : contains the calibrated parameters.

2.3.1 Calibration of the Market Data

After this, we plotted different graphs showing the evolution of the parameters with respect to both the tenor and the expiry date. [W J19]

From the given figure, we can observe that the instantaneous volatility α fluctuates slowly over time under the value **0.05**. In the first 2 years of tenor, α is basically constant and almost null and from the date where we have 2 years as tenor and the expiry date goes from 2 to 5 years, it increases drastically to its first high value.

Also, we can obviously spot the appearance of an outlier at the end of the plot

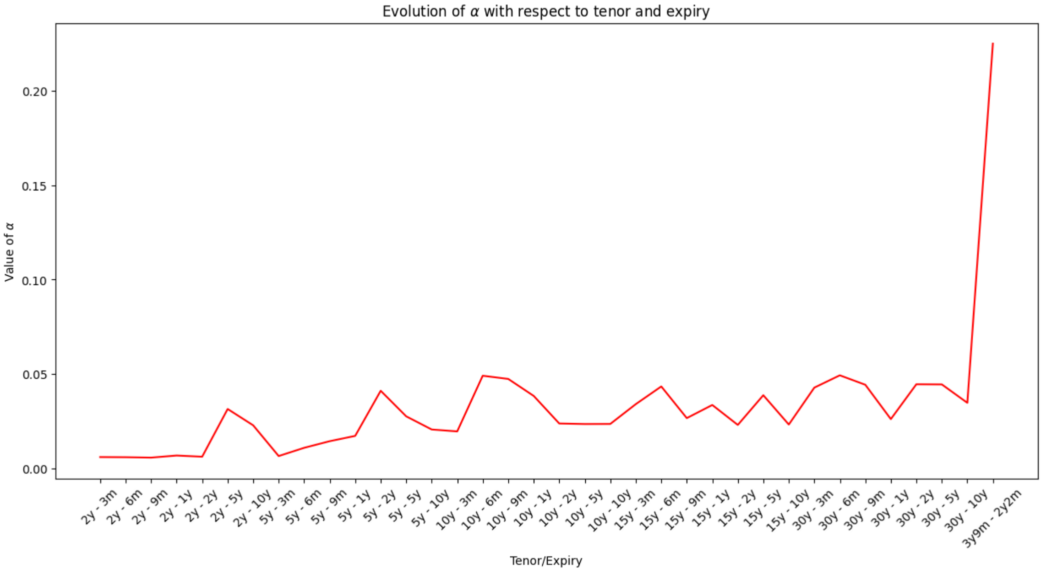

FIGURE 1 – Evolution of α with respect to tenor and expiry

defined by the couple **(3y9m - 2y2m, +0.2)**. Other plots of the rest of the parameters are presented in the appendices.

Graph of the volatility with respect to different strikes

Here, we plot both the SABR and Market MKT volatilities over the time period of tenor going from 2 all the way to 30 years and in each period of tenor, we take different values of the expiry date rangin from 3 months to 10 years.

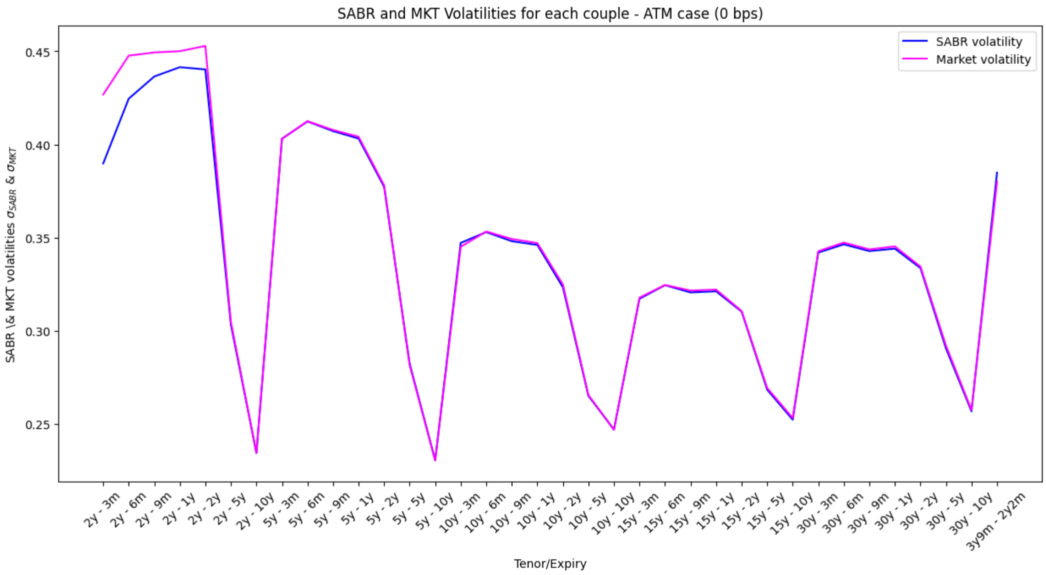

FIGURE 2 – SABR and MKT volatilities for each couple - ATM case

We can notice that our modeled SABR volatility is **nearly equal** to the market volatility we have as an input which shows that our computation is efficient enough to give results that are as close as the market volatility, our main goal.

On the other hand, throughout the time axis, the figure shows cyclic variations of the volatilities σ_{impl} and σ_{MKT}. In each cycle :

- The **maximum** values are reached when the expiry date is **6 months**
- The **minimum** values are reached when the expiry date is **10 years.**

Graph of the absolute error evolution between the two volatilities

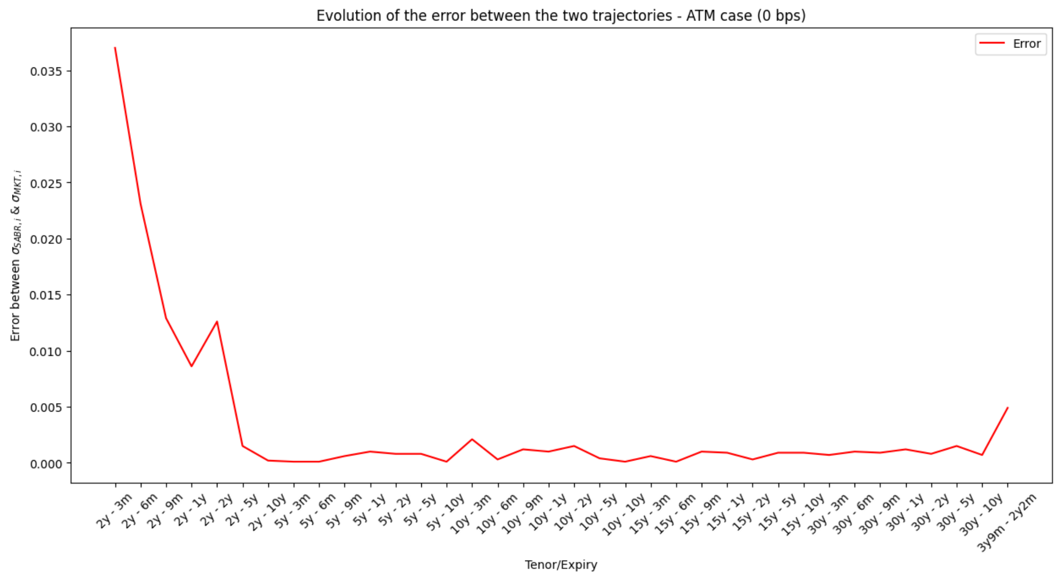

FIGURE 3 – Evolution of the error between the two trajectories

The graph presented above displays the absolute error computed between the two volatilities. Despite the fact that the first value of the error is relatively high and it is over **0.035**, we can evidently notice that it sharply decreases over time and, once the expiry date becomes **5 years**, the error is then constantly null and it remains like that.

Graphs of the Volatility Smile for different time periods

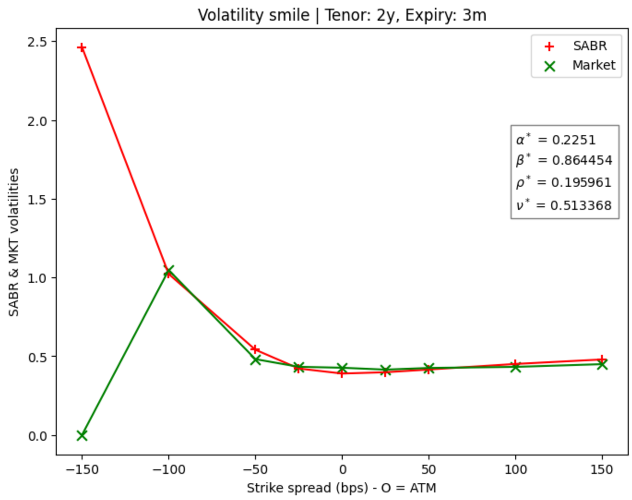

FIGURE 4 – Volatility smile (Tenor : 2 years, **Expiry : 3 months**)

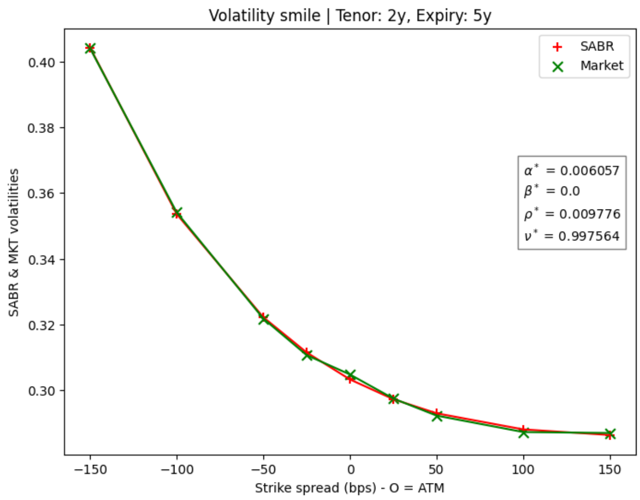

FIGURE 5 – Volatility smile (Tenor : 2 years, **Expiry : 5 years**)

(1) **The first figure** : where the expiry date was just 3 months, the SABR and Market volatilies were almost conflicting. In fact, the SABR volatility was at its highest value and from the initial time, it started decreasing, while the Market volatility was at its lowest value, which is 0, and began increasing throughout time. The two trajectories overlapped at the strike spread **ATM - 100(bps)**.

    Starting from this value, the two volatilities started following the same trajectory which became constant at the strike spread **ATM - 50(bps)**.

(2) **The second figure** : unlike the first figure, we can see here that the two volatilities are overlapped from the start of the time period and they are both decreasing over tim. The second difference is that, from the strike spread **ATM - 50(bps)**, the volatilities are still dropping until they reach a minimum point at the strike spread **ATM + 100(bps)**.

Remark 1 (Additional feature)**.** We have performed an interpolation using the SciPy module between the different volatilities points for each strike spread, in order to plot a trend line. The order of the selected spline interpolator is **cubic**.

For each couple $(\mathbf{tenor}, \mathbf{expiry})$, results between *parameters.csv* and *outvol.csv* are also given ; this helped us map for each case, the corresponding vector of the optimized results $(\alpha^*, \beta^*, \rho^*, \nu^*)$, obtained by the SLSQP minimization implementation.

Remark 2 (See attached notebook)**.** For the other **bps** values of the strike spreads $(0, \pm 25, \pm 50, \pm 100, \pm 150)$, the same plotting procedure is performed.

2.3.2 Impact of the SABR parameters

Now, we want to enlight the impact of the various parameters and it would be interesting to investigate how thesy can influence the shape of the implicit volatility.

Impact of α

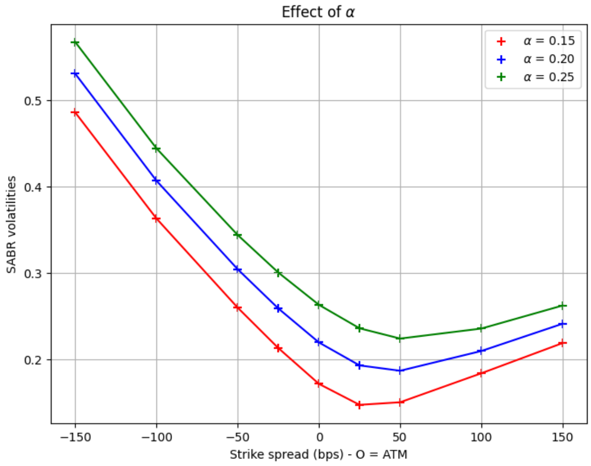

FIGURE 6 – Effect of α with the base parameters : 2 years as tenor, 5 years as expiry, β = 1, ρ = -0.6 and ν = 1.2

$\Longrightarrow$ We can observe that when α decreases, the curve moves downwards and when it increases it moves upwards.

Impact of β

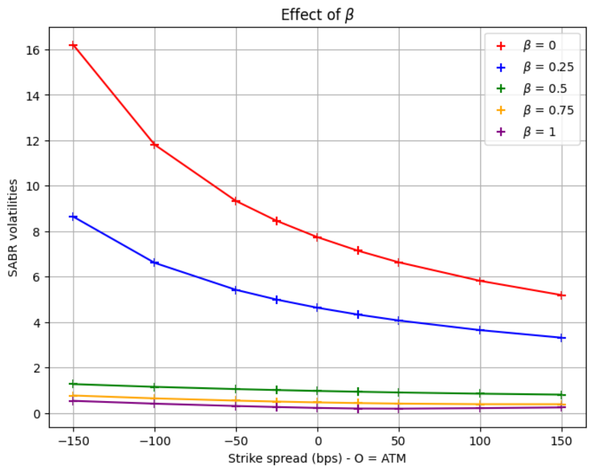

FIGURE 7 – Effect of β with the base parameters : 2 years as tenor, 5 years as expiry, α = 0.2, ρ = -0.6 and ν = 1.2

$\Longrightarrow$ From this figure, we can state that the more the value of β increases and grows towards 1 (upper bound), the more the curve flattens towards 0 and tends towards a straight line. On the contrary, when β approaches 0, the curve tends to explode in a parabolic shape.

Impact of ρ

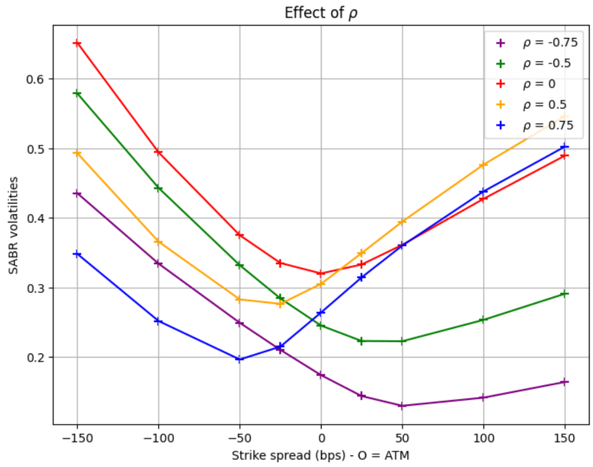

FIGURE 8 – Effect of ρ with the base parameters : 2 years as tenor, 5 years as expiry, $\beta = 1$, $\alpha = 0.2$ and $\nu = 1.2$

As we've seen in the theoretical background, ρ values take range -1 $\leq \rho \leq$ 1. To see how they behave, we try out different positive and negative values of ρ. We can obviously notice the effect of ρ on the volatility skew. In fact, when it has negative values, the smaller it is, the volatility skew increases. The opposite occurs if the value of ρ is positive.

Remark 3. It would be more relevant to have two different plots for positive and negative values of ρ since we can obviously point out two different groups of curves each centered at different points.

Impact of ν

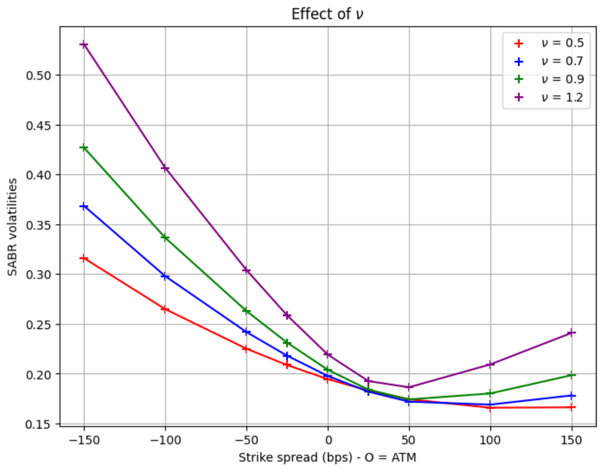

FIGURE 9 – Effect of ν with the base parameters : 2 years as tenor, 5 years as expiry, β = 1, α = 0.2 and ρ = -0.6

We can see here that the effect of ν is evident. As ν increases, the volatility smile is becoming more pronounced. In a particular strike spread [ATM - 50bps, ATM + 0bps], the curves become narrower and then they become more space as the strike spread increases.

The dynamics of the Implied Volatility formula

If we want to study the implied volatility deeper, we can study its dynamics and more precisely, the ***backbone*** of the model.

For strikes close to the forward rate, we can approximate the implied volatility formula by :

$$\sigma_{\text{impl}}(f, K) = \frac{\alpha}{f^{1-\beta}} \left\{ 1 - \frac{1}{2}\Big(1 - \beta - \rho\lambda\Big) \ln\left(\frac{K}{f}\right) \right.$$
$$\left. + \frac{1}{12}\Big((1-\beta)^2 + \big(2 - 3\rho^2\big)\,\lambda^2\Big) \ln^2\left(\frac{K}{f}\right) \right\}$$

where λ compares the *vol-of-vol* strength to the local volatility at the current forward rate and it is given by :

$$\boxed{\lambda := \frac{\nu}{\alpha} f^{1-\beta}}$$

This equation cannot be used to price transactions, but it is precise enough to reveal the qualitative behavior of the SABR model. The first term in the equation is known as ***"the backbone"***.

The backbone is the change in ATM volatility for a change in the forward rate and it is characterised completely by the exponent β.

- $\beta = 0$: we have a stochastic volatility normal model that produces a steeply sloping backbone.
- $\beta = 1$: we have a stochastic volatility log-normal model and we get a nearly flat backbone similar as in the Black & Scholes model.

Hence, we can conclude that other than the direct impact of the β parameter on the implied volatility and its shape, it also has an indirect impact going through the backbone and while varying the forward rate.

3 Conclusion

After a quick pre-processing of the given calibration scripts, we have calibrated the SABR model with the help of the given market dataset.

First, the solutions were saved on the disk, on separate CSV files : the parameters values as results of the minimization problem, the output volatilities and the difference of market and modeled volatilities.

Then, a thorough study of the SABR model and its calibration techniques were performed and various figures were added to present the evolution and the relationship between the different parameters and the volatilities.

Finally, we carried out a study of the impact of the different parameters of the SABR model and we have seen how they influence the shape of the volatility curve.

Extension to other models

The SABR model can be extended by assuming its parameters α, β, ρ & ν to be time-dependent [1]. This, however, complicates the calibration procedure and that is why some advanced calibration methods of the time-dependent SABR model were studied on so-called "effective parameters".

Alternatively, one can show that a time-dependent local stochastic volatility (denoted as SLV) model can be reduced to a system of autonomous PDEs that can be solved using the heat kernel. With that, we can apply generic techniques in order to solve those PDEs ; explicit solutions obtained by some techniques are comparable to traditional Monte Carlo simulations allowing for shorter time in numerical computations.

1. Since then, they were assumed **totally constant** over time.

4 Appendices

Evolution of the SABR model parameters with respect to tenor and expiry

FIGURE 10 – Evolution of β with respect to tenor and expiry

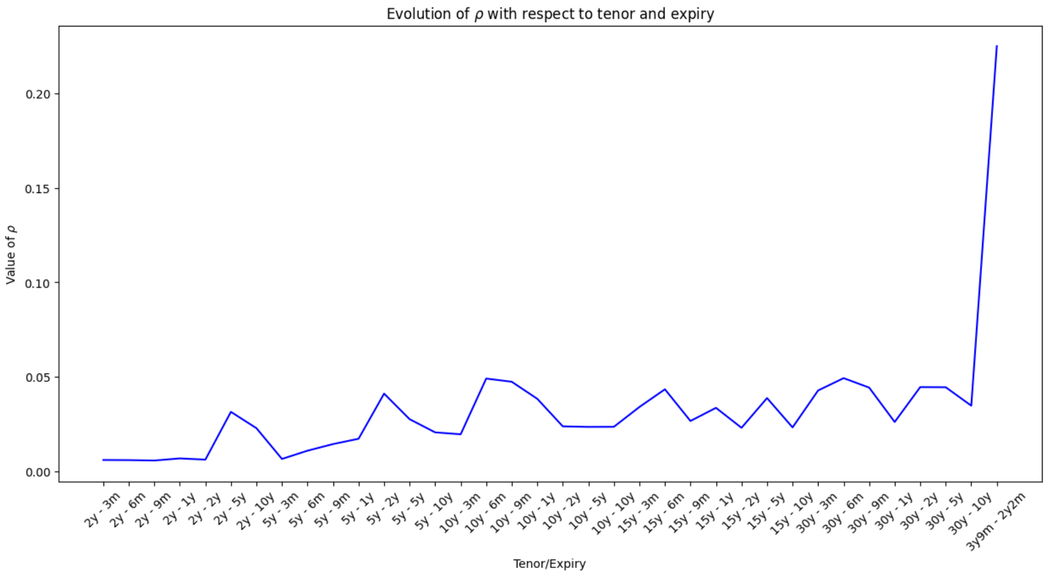

FIGURE 11 – Evolution of ρ with respect to tenor and expiry

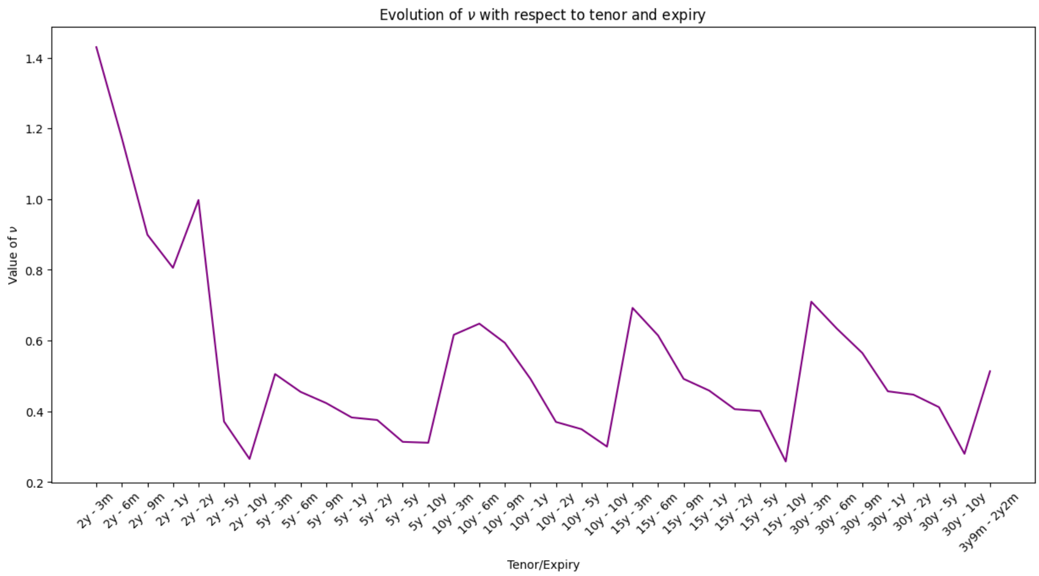

FIGURE 12 – Evolution of ν with respect to tenor and expiry

SABR and MKT volatilities for each (tenor, expiry) couple and the evolution of the absolute error

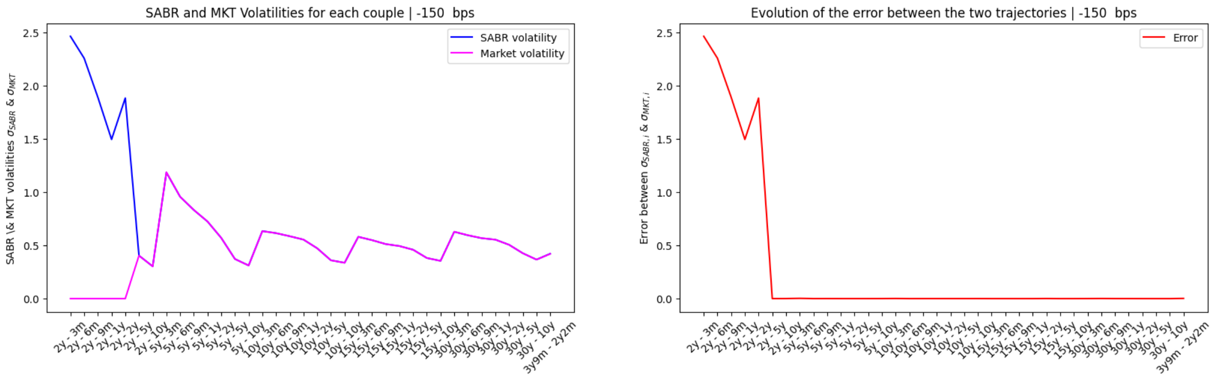

FIGURE 13 – SABR/MKT volatilities for each (tenor, expiry) + Evolution of absolute error (-150 bps)

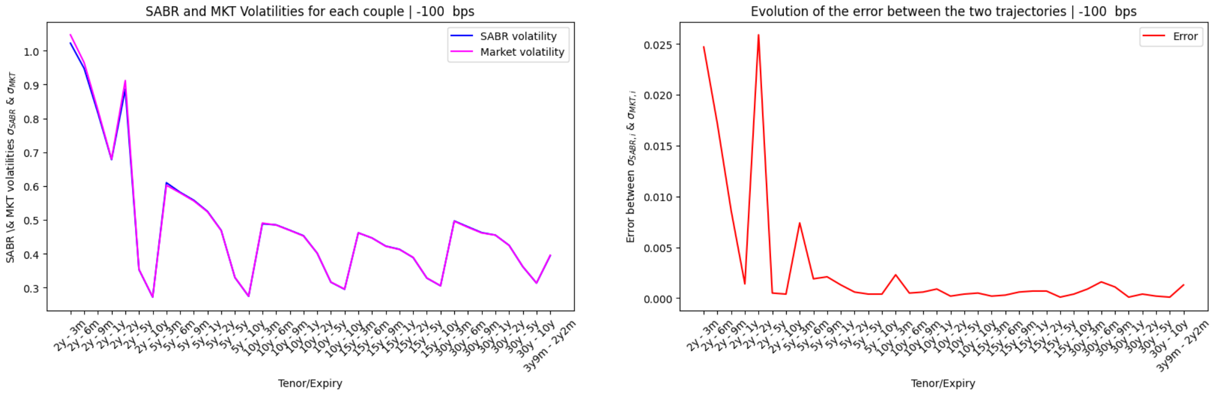

FIGURE 14 – SABR/MKT volatilities for each (tenor, expiry) + Evolution of absolute error (-100 bps)

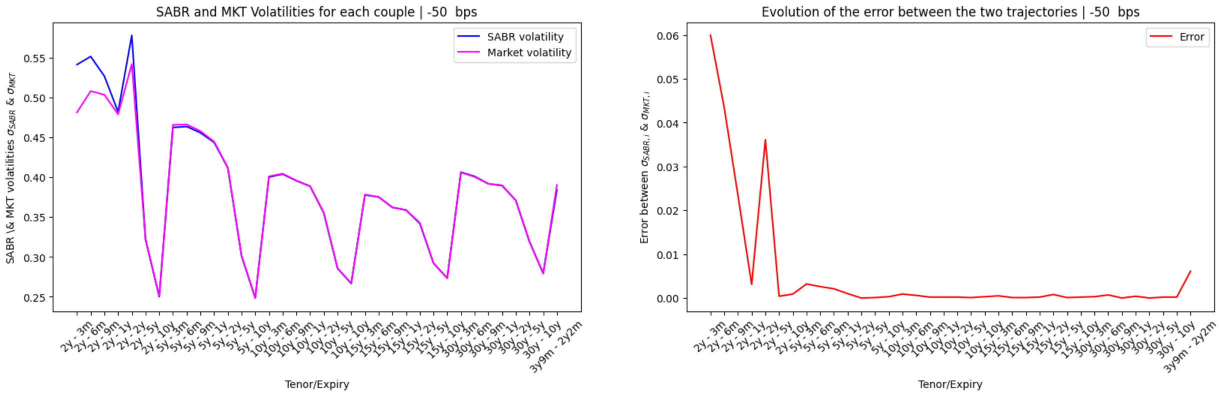

FIGURE 15 – SABR/MKT volatilities for each (tenor, expiry) + Evolution of absolute error (-50 bps)

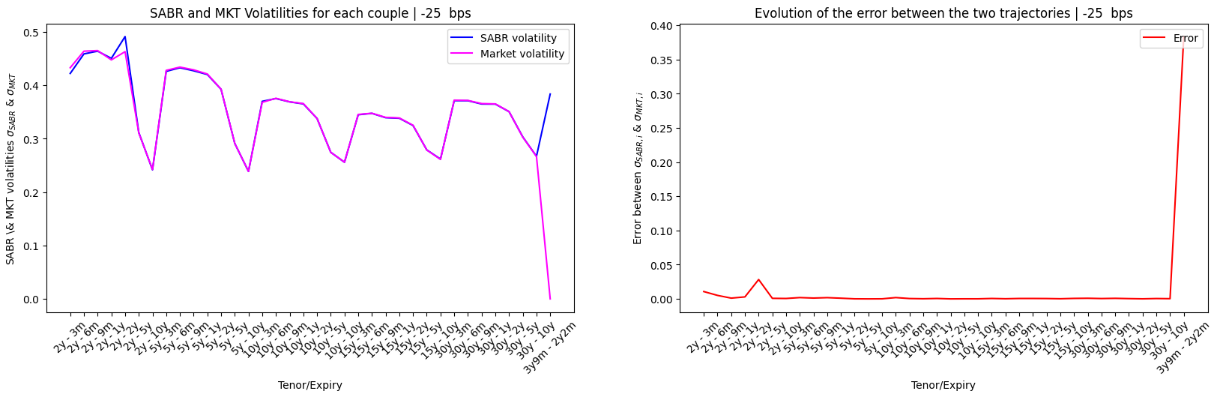

FIGURE 16 – SABR/MKT volatilities for each (tenor, expiry) + Evolution of absolute error (-25 bps)

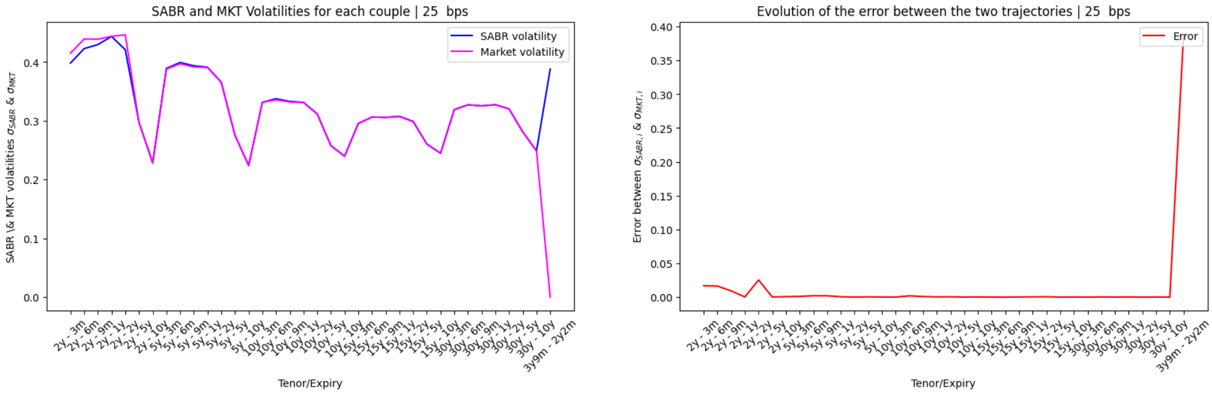

FIGURE 17 – SABR/MKT volatilities for each (tenor, expiry) + Evolution of absolute error (+25 bps)

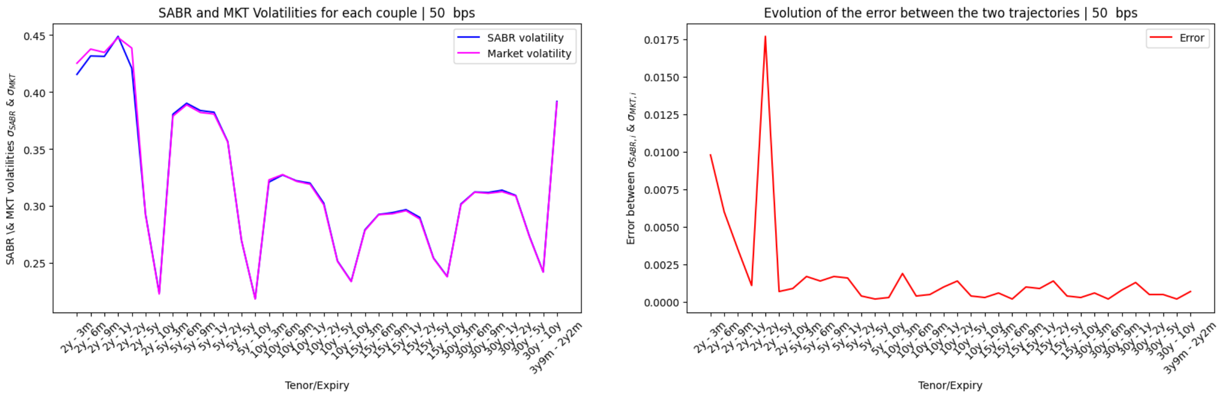

FIGURE 18 – SABR/MKT volatilities for each (tenor, expiry) + Evolution of absolute error (+50 bps)

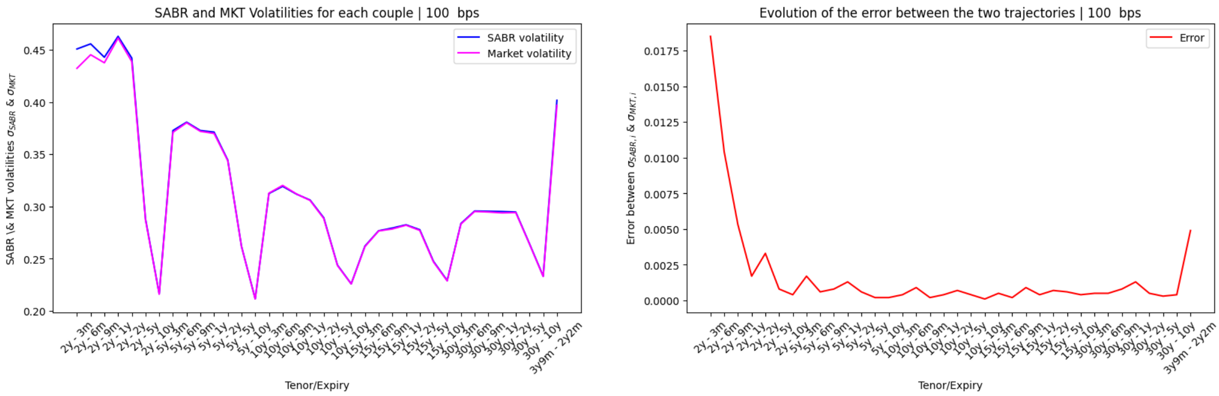

FIGURE 19 – SABR/MKT volatilities for each (tenor, expiry) + Evolution of absolute error (+100 bps)

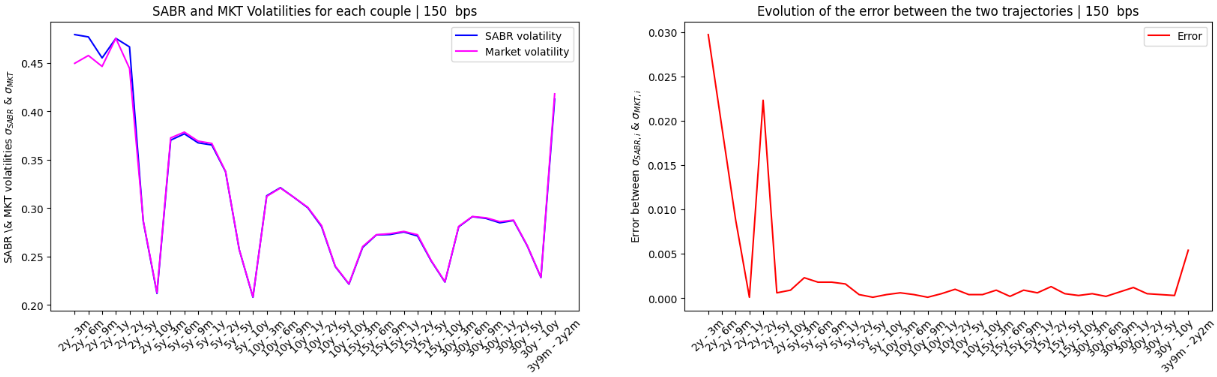

Figure 20 – SABR/MKT volatilities for each (tenor, expiry) + Evolution of absolute error (+150 bps)

List of figures

Bibliography

[Ioa11] Ioane Muni Toke. *Modèles stochastiques de taux d'intérêts*. 2011.

[Rod19] Rodrigo GRANDEZ. “SABR : A Stochastic Volatility Model in practice”. 2019.

[W J19] W. J. Trenberth. *SABR Calibration*. 2019. url : https://github.com/W-J-Trenberth/SABR-calibration.

[Sci22] SciPy contributors. *SciPy Optimize Minimize techical documentation*. 2022. url : docs.scipy.org.

[Wik] WIKIPÉDIA. *SABR Volatility model.* URL : https://en.wikipedia.org/wiki/SABR_volatility_model.