Eklipx.io Quantitative Engineer Technical Test - Solutions

Steve Nguoghia
17/06/2024
Update on 30/06/2024

1) Create a volatility surface from the clean.csv file suitable for use with the pricing exercises below.

Model: SABR model
We choose the SABR model for the implied volatility dynamics, given its strong theoretical foundation and its ability to capture the volatility smile. Additionally, it is more likely to be arbitrage-free, contrary to models such as polynomial regression models. There is of course the issue of mean reversion is a concern, but given that the contract is relatively short dated (expiring on 26 Jul 2024), it should hopefully not become a major one.

Calibration: The SABR parameters are calibrated based on the input curve (clean.csv), using the ***scipy*** function ***minimize***.

Implementation: We create the ***SABRVolatility*** class (in *vol_model_sabr.py*), which inherits from the volatility interface ***VolatilityModel*** (base for potential additional volatility models to be implemented in a latter stage, in *ivol_model.py*).
Objects can be created by providing either SABR parameters, or a volatility curve by strike for a specific maturity, which then allows to calibrate the SABR parameters using the method ***fit_to_volatility_curve***.

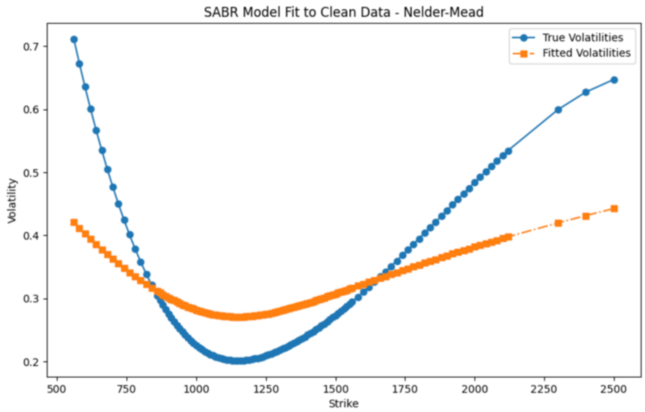
Update: We have improved the class to accept more flexibility in the calibration of the curve, and implemented additional global (*basinhopping*) and local (*minimize*) optimizer methods in ***fit_to_volatility_curve*** for better results.

Results: The SABR parametrisation and calibration of the clean curve is implemented in ***clean_vol_surface.py***. However, the SABR parameters extracted and the curve obtained does not fit the original curve. Further investigation is required to properly calibrate the parameters.

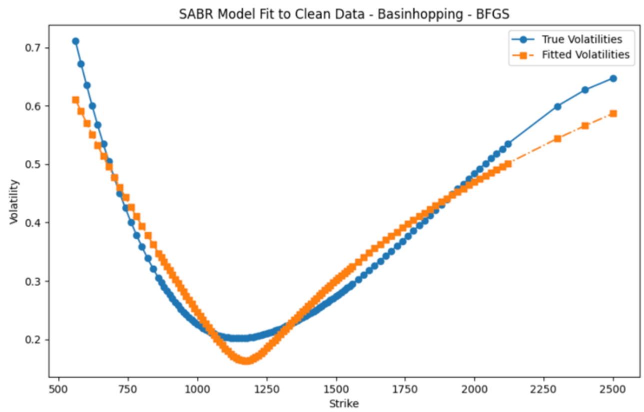
Update: After more investigation, we have managed to find a better fit for the curve. This is mainly due to removing the bounds around the volatility parameters (alpha and nu). We find indeed that fitted parameters for volatilities are above 1. While, for interest rates this might prove very rare, it is not necessarily the case for commodities (e.g.).

All the results are available in the folder:
1. Data results in:
    a. "SABR Clean Data fitted parameters – Basinhopping.txt" for basinhopping
    b. "SABR Clean Data fitted parameters – Minimize.txt" for minimize
2. Figures in: images/basinhopping/ and images/minimize/.

When bounds around the volatilities are in place, we barely obtain acceptable results:



Without bounds, we obtain more acceptable results:



Results are being obtained with the following parameters:

Method = Nelder-Mead

F(solution) = 0.10584973431350013

Fitted parameters:

alpha = 1.6581957304518182

beta = 0.6647995598582886

rho = 0.044076365970625 64

nu = 2.399614316049137

More work could be done around the calibration, including changing the number of points used for calibration, the range for calibration (e.g. more focus around par), which the functions implemented allow to do.

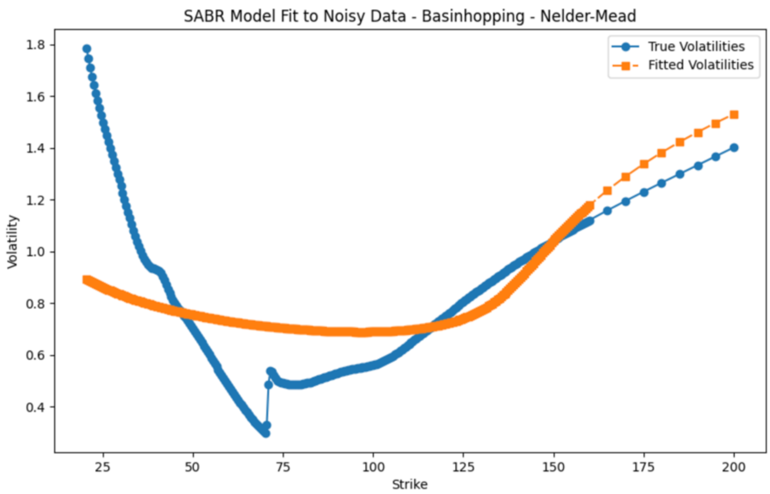
2) Parse the noisy.csv file and based on your knowledge clean and interpolate the curve so it is suitable for pricing (propose and implement an interpolation method).

Here as well we propose to model the dynamics using SABR as above, once the calibration has been sorted. However in this case, outliers need to be removed, in particular for the strike prices between 40 and 100, a part from the minimum, for the strike of K=70.

3) Plot the interpolated curve and comment on any pertinent transformations where you see fit.

Similar process to clean curve.

Update: We have managed to find a fit for the curve, but this needs to be improved. Note that this is with the full data.



Pricing

This has not been possible due to issues encountered during QuantLib installation.