

# Neural Network assisted Option Pricing under Rough Volatility: An Empirical Validation

This master thesis dives into the concept of so-called “Rough Volatility” and its meaning for both realized and implied volatility. We begin by estimating roughness of volatility from daily OHLC data via time series analysis and end with assessing how rough volatility models fit to implied volatility surfaces in a practical context. With the help of the rough Bergomi model we calibrate to observed option data both by Monte Carlo Simulation as well as via novel Neural Networks and find that this model can achieve decent to almost perfect fits to observed data depending on the asset under consideration and the type of calibration. With a fixed set of parameters, we observe good fits across the board, disregarding results due to poor data quality. Calibration by each maturity independently showed both highly accurate results and was achieved in a matter of seconds using neural networks. Difference in estimated parameters per maturity highlights that the model parameters do vary through time, which constitutes this work’s main contribution to existing literature. Due to poor quality of data our further-going studies aren’t as clear as we had preferred. Nevertheless, we looked at persistence of estimated parameters through time and relation between parameters estimated on both realized and implied volatility. We found that while the term structure of volatility was not well replicated, estimated model parameters do persist to some degree for OTM option contracts until close to expiration, also indicating that model parameters fluctuate. Parameters estimated from realized volatility achieved only somewhat decent fits for select expirations. We uncovered that for more volatile assets the fit from parameters estimated using realized volatility was best for short to maturity contracts, while stable assets achieved their best fit for maturities far into the future. This might indicate valuations dictate volatility expectations in the market. The code making such analyses feasible is publicly available via our [GitHub here](#).

**Keywords:** *Rough Volatility, Time series analysis, stochastic volatility models, rough Bergomi model, option pricing, option calibration, Monte Carlo simulation, neural networks*

## Table of Contents

Directory of Abbreviations .....	III
Directory of Figures.....	V
Directory of Tables.....	XI
Introduction .....	1
Literature Review.....	3
Rough Volatility .....	11
Empirical Time Series Observations .....	15
The RFSV Model .....	21
Validating the Results by Daily OHLC Data .....	27
The rBergomi Model .....	34
The Black Scholes Model and Implied Volatility Dynamics .....	34
The Bergomi Model .....	38
Pricing under the physical probability measure .....	40
Change of Measure, Introducing the risk-neutral measure.....	42
Pricing using the rBergomi Framework with Monte Carlo Simulation .....	47
Turbo Charging and the Hybrid Scheme.....	47
Pricing an observed Option Chain.....	53
Automating the Calibration: Minimization Techniques .....	58
Neural Network assisted Calibration .....	64
Image based Implementation.....	79
Calibrating to an observed Option Chain .....	83
Empirical Evaluation .....	91
Methodology .....	91
Assessment of Fit to the Implied Volatility Surface.....	92
Persistence of rBergomi Parameters.....	96
Historical Time Series Implied Parameters .....	97
Quality of Data .....	102
Conclusion .....	103
References .....	A
Papers .....	A
Online Sources .....	G
Appendix .....	I
Appendix A:.....	I
Appendix B: .....	M
Appendix C: .....	P
Appendix D: .....	Q
Appendix E: .....	V

# Directory of Abbreviations

Abbreviation	-	Term	-	Translation/Explanation
AAPL	-	Apple	-	Stock market symbol
Adj. Close	-	Adjusted Close	-	Incorporates price changes after the exchange has officially closed for the day
AMC	-	AMC	-	Stock market symbol
approx.	-	Approximately	-	
ATM	-	At-the-money	-	Often indicated by k=0, signifies that an options contract is neither worthless, nor providing any return. Its strike price is equal to the current spot price of the stock
BS	-	Black and Scholes	-	Refers to the Black Scholes Option Pricing Model
Bps	-	Basis points	-	Financial speak for decimal numbers.
CET	-	Central European Time	-	
C.p.	-	Ceteris paribus	-	All else being equal
E.g.	-	Exempli gratia	-	For example
Et al.	-	Et alii/et aliae	-	And others
Etc.	-	Et cetera	-	And so forth
EV	-	Electric Vehicles	-	
fBm	-	Fractional Brownian motion	-	
fOU	-	fractional Ornstein-Uhlenbeck	-	
FRED	-	Federal Reserve Bank of St. Louis	-	Publicly available economic data
FSV	-	Fractional stochastic volatility	-	
GKYZ	-	Garman, Klass, Yang, and Zhang	-	Refers to a variance estimator based on work of Garman, Klass that has been extended by Yang, and Zhang
GME	-	GameStop	-	Stock market symbol
I.e.	-	Id est	-	That is, that means
ITM	-	In the money	-	Indicating that an option is paying out a return

<b>Abbreviation</b>	-	<b>Term</b>	-	<b>Translation/Explanation</b>
IV	-	Implied Volatility $\sigma_{BS}$	-	Option contract's implied volatility
LM	-	Levenberg-Marquardt	-	Minimization technique
Market Cap	-	Market Capitalization	-	Measure to indicate the total worth of all stocks outstanding for a company
MC	-	Monte Carlo	-	Simulation Technique
m.m.	-	mutatis mutandis	-	with the necessary modifications
MRW	-	Multifractal Random Walk	-	A volatility model
MSE	-	Mean square error	-	
MSFT	-	Microsoft	-	Stock market symbol
NM	-	Nelder Mead	-	Minimization Method
NN	-	Neural Network	-	A new tool for calibration of volatility models
OHLC	-	Open, High, Low, Close	-	Used to indicate different observation times within one time interval, e.g., one day.
OTC	-	Over the counter	-	Non-public traded financial assets
OTM	-	Out of the money	-	Indicating that an option is not providing any return
P&L	-	Profit and Loss	-	
PCA	-	Principal Component Analysis	-	
rBergomi	-	Rough Bergomi	-	A novel approach to option pricing, presented in this thesis
RFSV	-	Rough fractional stochastic volatility	-	
RMSE	-	Root mean square error	-	
S&P 500	-	Standard and Poor's 500 Index	-	One of the most important stock market indices in the world
SPX	-	S&P 500 trading symbol	-	Index trading ticker
s.t.	-	Such that	-	
T	-	AT&T	-	Stock market symbol
TESLA	-	Tesla	-	Stock market symbol
VIX	-	CBOE Volatility Index	-	An index that tracks S&P implied volatility
wrt	-	With respect to	-	

# Directory of Figures

Figure 1: Synthetic implied volatility surface, generated by the so-called rBergomi model with parameters: $\alpha=-0.35$ , $\eta=2.5$ , $\rho=-0.7$ , $\xi=0.15$ .	1
Figure 2: Excerpt of an option chain of Peloton for expiration on August 27, 2021. Observed on 27.08.2021 from yahoo finance.	2
Figure 3: Fractional Brownian Motion for different Hurst Parameters	12
Figure 4: SPX Daily Adjusted (Adj.) Close Data between 01.01.2000 and 05.12.2017. All data derived from yahoo finance	18
Figure 5: SPX Daily realized kernel estimates for variance between 01.01.2000 and 05.12.2017. All data derived from the Oxford-Man Institute of Quantitative Finance	18
Figure 6: Regressing $\log(\Delta)$ against $\log(m(q, \Delta))$ yields us the intercept $\alpha_q$ and the slope $\zeta_q$ . Results for SPX data. Data derived from the Oxford-Man Institute of Quant. Fin.	19
Figure 7: Regressing each $q$ against its corresponding $\zeta_q$ value, yields us the slope $s_q$ and the intercept $\beta_q$ . Results for SPX data. All data derived from the Oxford-Man Institute of Quantitative Finance	19
Figure 9: Distribution of increments of log-volatility (1-, 5-, 25, and 125-days). SPX data derived from the Oxford-Man Institute of Quant. Fin.	20
Figure 10: Displays the kernel variance estimates vs the 12-month rolling estimate for the Hurst parameter. Results for SPX data. All data derived from the Oxford-Man Institute of Quantitative Finance	21
Figure 11: Shows the empirical $\sigma_{BS}$ implied volatility surface for AAPL on the 25.06.2021. Data gathered at Close from yahoo finance.	22
Figure 12: From (Gatheral et al., 2018, p. 3). Black dots: observed ATM volatility skews (11) for SPX as of June 20th, 2013. Red Curve: Fit generated by $\psi\tau \sim A\tau - 0.4$	23

Figure 13: X-axis: Predicted variances. Y-axis: Observed variances for SPX. Estimates of H and v from values reported in Table 11. Data from the Oxford-Man Institute of Quant. Fin. 26

Figure 14: The x-axis shows the timescale in days, and the y-axis plots the level of volatility for the observed SPX data (blue), as well as the predicted values (red). Estimates of H and v rely on the values reported in Table 11.. All data derived from the Oxford-Man Institute of Quantitative Finance .....26

Figure 15: X-axis: timescale in years. Y-axis: level of variances. Plots in order: Kernel Variance Estimates used by (Gatheral et al., 2018), GKYZ and standard return Variance estimators. SPX Data derived from the Oxford-Man Institute of Quant. Fin. and from close data from yahoo finance.....29

Figure 16: X-axis: log of the step-size  $\Delta$ . Y-axis: level of log of  $m(q, \Delta)$ . Plots in order: Kernel Variance Estimates used by (Gatheral et al., 2018), GKYZ and standard return Variance. SPX Data derived from the Oxford-Man Institute of Quant. Fin.# and from close data from yahoo finance.....30

Figure 17: X-axis: Order of q. Y-axis: level of  $\zeta_q$  for observed SPX data. Data derived from the Oxford-Man Institute of Quantitative Finance and from close data from yahoo finance..30  
Figure 18: Highlights the spike in trading volume for SPX at around the COVID incident. Data derived from yahoo finance close data. .....32

Figure 19: X-axis: Strike Price. Y-axis: IV. Smile for AAPL with spot=\$133.11 and 1.95 years to expiration on the 25.06.2021. Data from yahoo finance at close. .....36

Figure 20: SPX smiles observed on 14.08.2013: blue and red points represent ask and bid IVs. The orange lines are from the rBergomi simulation, calibrated by (Bayer et al, 2015). See (Bayer et al, 2015, p 897).....46

Figure 21: Simulation of  $Wt\alpha, Vt$ , and  $St$  for different parameter combinations. .....50

Figure 22: Implied volatility smile for arbitrary rBergomi parameters, and spot and expirations values. .....52

Figure 23: Showing before and after manual calibration of the rBergomi model to an observed option chain, given by the bid and ask bounds in red and blue respectively. ....56

Figure 24: Comparison between rBergomi and empirical log option prices. Illiquid options are priced via calibrated rBergomi parameters (red stars), compared to the illiquid price (grey dashed line). In magenta color are imaginary option contracts, also priced using the calibrated model.....	57
Figure 25: Implied price distribution of the rBergomi simulation given the calibrated parameters .....	58
Figure 26: Comparing fit to mid-prices. Left: Nelder-Mead; Right: Trust Region Reflective	60
Figure 27: Showing the Nelder-Mead calibration (yellow) to bid and ask implied volatilities. ....	62
Figure 28: Showing the Trust Region Reflective calibration (yellow) to bid and ask implied volatilities.....	62
Figure 29: An example neural network as used in the work by (Horvath, et al., 2019). Taken from (Horvath, et al., 2019, p 15) .....	65
Figure 30: Basic construct of a neuron as part of a neural network. Taken from (Horvath, et al., 2019, p 11) .....	65
Figure 31: Verifying that Levenberg-Marquardt provides the greatest speed in average calibration time. ....	70
Figure 32: Implied Volatility for different expirations (i.e., different option chains) both stemming from artificial generated test-set data and neural network approximations, using the data and NN design employed by (Horvath, et al., 2019).....	75
Figure 33: Strike/Maturity relative error (computed as a percentage) between validation set and NN estimation. ....	76
Figure 34: Relative parameter errors after calibration via LM in the rBergomi model.....	77
Figure 35: Showing the parameter relative errors for the rBergomi model, as well as the RMSE between approximated IVs and the true (synthetic) data .....	78

Figure 36: Evolution of model parameters for SPX under Q using the rBergomi framework taken from (Horvath, et al., 2019, p 25).....	78
Figure 37: Showing the strike/maturity grid used in the deep neural network by (Rømer, 2020) .....	81
Figure 38: Filtering option contracts. Original data in blue, orange indicates option contracts that the NN can be calibrated to.....	83
Figure 39: Option contracts for AAPL on 25-06-2021 for shortest (0.019) and longest (1.975) maturity in years .....	84
Figure 40: Neural network calibration to all available 16 option chains for AAPL, observed on 25-06-2021 .....	85
Figure 41: Neural Network calibration to every smile independently. Parameters are reported for each chain separately.....	86
Figure 42: Neural Network fit with floating Forward Variance Curve to entire implied volatility surface .....	88
Figure 43: Forward Variance Curve estimation under Q through time .....	89
Figure 44: rBergomi fit to the AAPL volatility surface as observed on the 05.08.2021 .....	92
Figure 45: rBergomi fit to the GME volatility surface as observed on the 05.08.2021.....	93
Figure 46: rBergomi fit to the MSFT volatility surface as observed on the 05.08.2021 .....	93
Figure 47: rBergomi fit to the NIO volatility surface as observed on the 05.08.2021 .....	94
Figure 48: rBergomi fit to the SPX volatility surface as observed on the 05.08.2021 .....	94
Figure 49: rBergomi fit to the T volatility surface as observed on the 05.08.2021 .....	94
Figure 50: rBergomi fit to the TSLA volatility surface as observed on the 05.08.2021 .....	95
Figure 51: Tracking maturity 'September 17, 2021' for SPX through time. Orange: true volatility term structure. Blue: Estimated term structure.....	96

Figure 52: Tracking maturity 'September 17, 2021' for SPX through time. Difference between observed and predicted IVs.....	96
Figure 53: rBergomi fit to the AAPL volatility surface with time series implied parameters as observed on the 05.08.2021 .....	98
Figure 54: rBergomi fit to the GME volatility surface with time series implied parameters as observed on the 05.08.2021 .....	99
Figure 55: rBergomi fit to the MSFT volatility surface with time series implied parameters as observed on the 05.08.2021 .....	99
Figure 56: rBergomi fit to the NIO volatility surface with time series implied parameters as observed on the 05.08.2021 .....	100
Figure 57: rBergomi fit to the SPX volatility surface with time series implied parameters as observed on the 05.08.2021 .....	100
Figure 58: rBergomi fit to the T volatility surface with time series implied parameters as observed on the 05.08.2021 .....	100
Figure 59: rBergomi fit to the TSLA volatility surface with time series implied parameters as observed on the 05.08.2021 .....	101
Figure 60: Kernel Variance Increment Distribution by lag .....	N
Figure 61: GKYZ Variance Increment Distribution by lag .....	N
Figure 62: Log Return Variance Increment Distribution by lag.....	O
Figure 63: Shows the forecasted Variance Swap Curve (dashed line) vs the actual observed Variance Swap Curve(continuous) for the dates of 15.11.2008 (blue) and 12.11.2008 (red) ...P	
Figure 64: Using manual calibrated parameters (yellow) to fit to the full volatility surface. IVs that show values of zero reflect market prices for which the Black Scholes option pricing model is not able to compute corresponding values. ....	R

Figure 65: Nelder Mead calibration result for different starting parameters .....	S
Figure 66: Showing fit from calibration to mid-prices with the Nelder-Mead minimizer .....	T
Figure 67: Showing the calibration by prices to the whole surface by optimization slice. Minimizer used: Nelder-Mead.....	U
Figure 68: The Hurst parameter through time as estimated by (Rømer, 2020). Taken from (Rømer, 2020, p18).....	V
Figure 69: Correlation and Volatility of Volatility through time as estimated by (Rømer, 2020). Taken from (Rømer, 2020, p19) .....	V
Figure 70: rBergomi parameters through time as extracted from our individual calibration exercise .....	W
Figure 71: Calibration with floating forward variance and meaned parameters as extracted from individual calibration exercise .....	X
Figure 72: Calibration with floating forward variance and parameters as extracted from calibration to whole surface .....	Y
Figure 73: Neural Network VS Monte Carlo Simulation: Accuracy in mapping parameters to IVs.....	Z
	X

# Directory of Tables

Table 1: List of financial assets and their corresponding stock symbols.....	10
Table 2: Hurst Parameter estimates for SPX following different variance estimation methods (kernel, GKYZ, log-return variance). Data from the Oxford-Man Institute of Quant. Fin. and from close data from yahoo finance.....	31
Table 3: H, v and $\rho$ estimates for 7 stocks following methodology by (Gatheral et al., 2018) using the GKYZ variance estimates for the period of the past 3 years, snapshot at 2021-08- 23. Using yahoo finance close data.....	33
Table 4: Showing all available maturities (option chains) for AAPL, observed on 25.06.21	58
Table 5: Comparing different minimization techniques for their accuracy and speed .....	60
Table 6: Fit to the full implied volatility surface. Comparison between Nelder-Mead and Trust Region Reflective. ....	61
Table 7: Neural Network fit vs traditional MC calibration.....	84
Table 8: Neural Network maturity-wise calibration vs calibration by MC simulation .....	87
Table 9: Comparing MSE of Neural Network with different calibration techniques.....	88
Table 10: Time Series estimated Parameters for alpha, eta, and rho .....	97
Table 11: Estimates for H and v with q=2 for various indices in the Oxford-Man Institute of Quantitative Finance dataset. The name extension ".rk" signifies realized kernel estimates for realized variance. ....	M
Table 12: Showing probability of moneyness in percent for each available strike price. ....	Q
Table 13: Nelder Mead Minimization Single Expiration: Calibration by implied volatilities vs option prices.....	T
Table 14: Nelder Mead Minimization Full Surface: Calibration by implied volatilities vs option prices.....	U
Table 15: MSE for different calibration exercises with the neural network by Rømer.....	Y

# Introduction

From hedging risk to speculation, option trading has never been more popular. The availability of discount brokers such as Robinhood or Trade Republic has made it easy for the common retail investor to engage in option trading and leveraged speculation. This regained interest of the public in the stock market, beginning in early 2020, was only fueled by the COVID pandemic when consumer consumption fell to all-time lows and private savings reached new highs<sup>1</sup>. If traded option volume is any indication<sup>2</sup>, it is likely that many private households poured their hard-earned money into such derivatives.

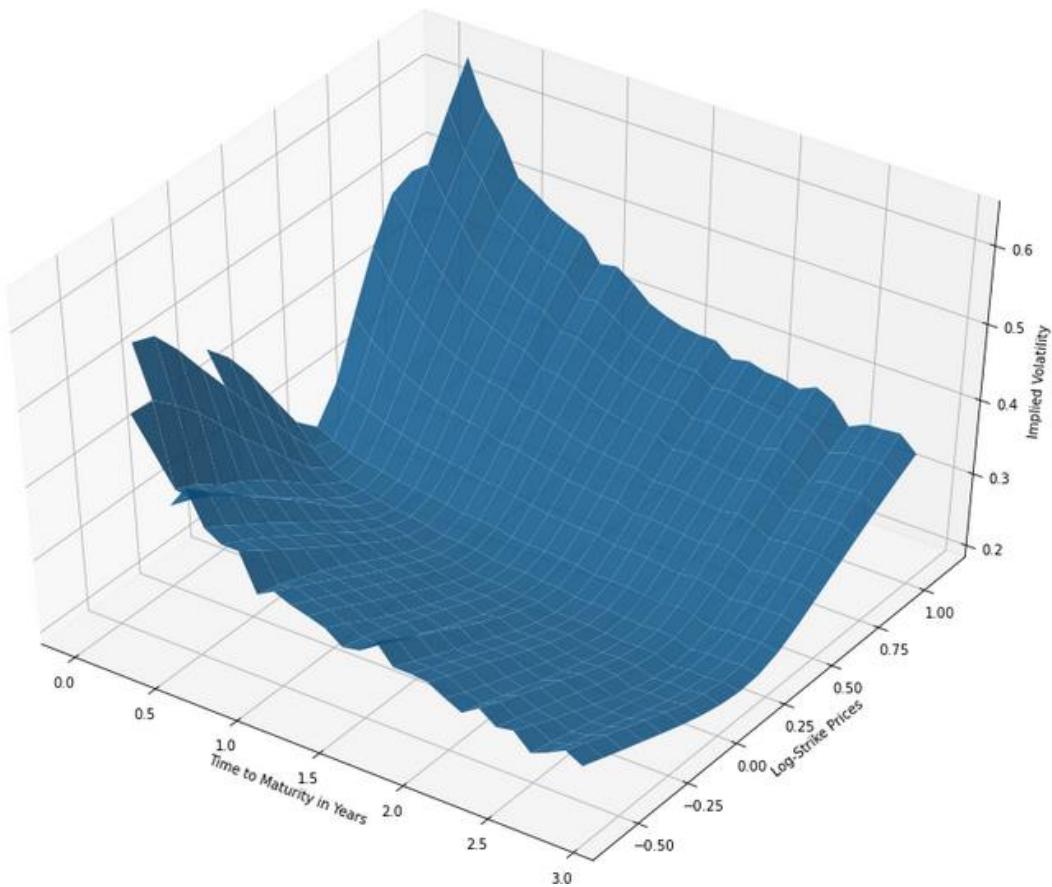


Figure 1: Synthetic implied volatility surface, generated by the so-called rBergomi model with parameters:  $\alpha=-0.35$ ,  $\eta=2.5$ ,  $\rho=-0.7$ ,  $\xi=0.15$ .

With an increase in heavy speculation, that surely reached their pinnacle (for now) in the reddit-driven short-squeeze attempt<sup>3</sup>, hedging risk for market makers becomes of central concern, as

<sup>1</sup> For data on the savings rate for US households and their spending see <https://fred.stlouisfed.org/series/PSAVERT> and <https://fred.stlouisfed.org/series/DPCCRV1Q225SBEA>

<sup>2</sup> Barrons noted that the stock option volume increased by 68% for 2020 year-over-year: <https://www.barrons.com/articles/stock-option-volume-exploded-in-2020-51610017270>

<sup>3</sup> See for example <https://markets.businessinsider.com/news/stocks/Reddit-traders-scared-short-sellers-away-amid-meme-stock-squeezes-2021-6>; and [https://en.wikipedia.org/wiki/GameStop\\_short\\_squeeze](https://en.wikipedia.org/wiki/GameStop_short_squeeze)

options are generally “*marked-to-market*”, and thus daily profit and loss (P&L) are settled (incurred) at end of day (and may even require margin adjustment). Market makers engage in frequent buying and selling of financial assets, to increase liquidity across markets and such assets. Enhancing liquidity is most important to avoid market dislocations and provide low price spreads for all market participants. For options in particular, financial institutions such as market makers and exchanges, but also investment banks, hedge funds and accounting firms need to be able to assess the fair-traded prices across all available strikes. This corresponds to knowing the so-called implied volatility (IV)<sup>4</sup> surface, shown in Figure 1.

**Calls** for August 27, 2021

Contract Name	Last Trade Date	Strike ▾	Last Price	Bid	Ask	Change	% Change	Volume	Open Interest	Implied Volatility
PTON210827C00055000	2021-08-25 10:40AM EDT	55.00	58.35	0.00	0.00	0.00	-	-	0	0.00%
PTON210827C00060000	2021-08-27 9:37AM EDT	60.00	46.65	43.75	44.60	-3.45	-6.89%	6	0	490.63%
PTON210827C00065000	2021-07-27 11:45AM EDT	65.00	52.85	38.65	39.75	0.00	-	1	1	439.06%
PTON210827C00070000	2021-08-20 1:42PM EDT	70.00	33.80	33.90	34.55	-4.80	-12.44%	3	7	387.50%
PTON210827C00075000	2021-08-20 3:49PM EDT	75.00	33.57	28.75	29.70	0.00	-	3	3	328.13%
PTON210827C00080000	2021-08-24 9:49AM EDT	80.00	30.60	0.00	0.00	0.00	-	-	0	0.00%
PTON210827C00085000	2021-08-27 10:48AM EDT	85.00	18.95	18.90	19.45	-10.61	-35.89%	5	2	204.69%
PTON210827C00090000	2021-08-27 9:42AM EDT	90.00	14.74	13.90	14.70	-7.36	-33.30%	19	7	179.49%
PTON210827C00095000	2021-08-27 10:03AM EDT	95.00	10.00	8.85	9.85	-9.90	-49.75%	102	146	130.86%
PTON210827C00098000	2021-08-25 2:25PM EDT	98.00	17.45	0.00	0.00	0.00	-	-	0	0.00%
PTON210827C00099000	2021-08-27 10:47AM EDT	99.00	5.45	5.00	6.00	-10.90	-66.67%	12	28	96.68%

Figure 2: Excerpt of an option chain of Peloton for expiration on August 27, 2021. Observed on 27.08.2021 from yahoo finance.

With this surface, and a theoretical model that is able to replicate the same, all available (and hypothetical) options at any maturity date can be priced<sup>5</sup>. Moreover, (Bali, Hovakimian, Volatility Spreads and Expected Stock Returns, 2009) showed that IV surfaces also harbor a significant amount of information, which may be extracted via so-called principal component analysis (PCA). PCA allows to decompose a volatility surface into factors useful to predict expected returns, uncover market expectations, or future volatility behavior. The need of being able to price options across any strike/maturity combination may be due to manifold reasons. It may be to hedge risks, assess (counterparty) risk exposure, price “over-the-counter” (OTC) products such as exotic options, or even price the simple option contract in the first place<sup>6</sup>.

<sup>4</sup> Implied volatility is a concept from option pricing and relates to an input into the Black Scholes option pricing model. This concept will be explored in detail in chapter 3.

<sup>5</sup> The idea here is that with a theoretical model (which matches most closely to observed data) all possible option contracts may be interpolated or extrapolated, even those that are not actively traded or even exist.

<sup>6</sup> While these are the benefits for large financial institutions, also the common retail investor may gain from advanced option pricing, such as finding the fair price for options that are only lightly traded.

With high volatility and increased trading activity in the market this becomes even more difficult, as not all strikes are traded equally often. An example can be observed in Figure 2, where the “Last Trade Date” digresses for options reported and for some contracts even the price information seems to be missing<sup>7</sup>. Now, calculation of fair prices for such illiquid options may not always be practically feasible while relying on standard procedures such as the renowned Black Scholes model. In particular, a theoretical framework and corresponding model is needed that mimics observed option’s behavior. Such a model may then be used to assess fair prices for all contracts, whether liquid or not.

## Literature Review

Modeling option prices has a long history, beginning as early as 1900 when Louis Bachelier, a French PhD student, published his thesis (**Louis Bachelier, Théorie de la speculation, 1900**) on modeling a stochastic process, today known as the Brownian motion. There he explores how Brownian motions may be used to value stock options. It constitutes the first paper using advanced math in finance and remained, with only minor modifications, the go-to choice for modeling option prices for many years to come. Bachelier’s concept then reached its pinnacle when Fisher Black and Myron Scholes published their famous work **“The Pricing of Options and Corporate Liabilities” in 1973**. However, Black and Scholes’ framework was not without its faults either. The so-called delta-hedging<sup>8</sup> exacerbated the stock market crash in 1987 later known as the “Black Monday” (**Jean-Philippe Bouchaud, A step closer to the perfect volatility model, 2020**). Within their model framework perfect delta hedging is possible, however as the Black Scholes model does not allow for price jumps and thus crashes are seen as impossible the model fell short in this tragic event. Its assumptions are clearly at odds with the real world as shown by Mandelbrot in (**Benoit Mandelbrot, The Variation of Some Other Speculative Prices, 1963**). He found that large fluctuations in price tend to be followed by equally great changes, while small changes are more often followed by small changes yet again<sup>9</sup>. Necessitating a new modeling approach, the Heston model was born: (**Steven L. Heston, A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options, 1993**). While this model incorporates clustering of volatility it assumes, much like Garch, that volatility regimes decay over some fixed time horizon. This

---

<sup>7</sup> See Appendix A1 for a full example option chain for Apple stock, observed on 25.06.2021

<sup>8</sup> Delta-hedging describes a strategy in which the value of an (option) portfolio is “hedged” (i.e., secured) against moves of the underlying asset (e.g. a stock).

<sup>9</sup> This empirical fact is today known as volatility clustering, which was addressed by the financial literature with Garch type volatility models.

again is not compatible with empirical observations, where decay can be anything between hours to years (**Jean-Philippe Bouchaud, 2020**). An entirely different direction was taken by Mandelbrot. Today being famous as the founder of fractal geometry, Mandelbrot studied “roughness” in nature. According to him (geometric) fractals are of fragmented or rough shape, which can be divided into parts. Each such part then is again approximately a smaller copy of the whole<sup>10</sup>. (**Benoît Mandelbrot, The fractal geometry of nature 1983**). So called multifractal systems are universal to the world. These phenomena can be found in the topography of mountain ranges, heartbeat dynamics, human brain activity, coastlines, fluid dynamics, internet traffic, finance, and in many more fields. He concluded that this roughness can be measured (**Benoit Mandelbrot, The Fractalist: Memoir of a Scientific Maverick, 2014**). Being not less interested in quantitative finance, Mandelbrot understood financial markets as “wildly random”. While he developed many original attempts at modelling financial timeseries (**Mandelbrot, Benoit, Encyclopedia of Quantitative Finance, 2010**) the insights about fractals facilitated a new understanding of market price fluctuations only with the availability of high frequency data<sup>11</sup> in the 1980s. Mandelbrot summarized his research in (**Benoit Mandelbrot, Fractals and Scaling in Finance, 1997**) and (**Mandelbrot; Hudson, The (Mis)Behaviour of Markets: A Fractal View of Risk, Ruin and Reward, 2004**). In time series data of fractal systems, the so-called Hurst exponent (denoted by H in honor of Harold Edwin Hurst by Mandelbrot) measures their fractal dimension D<sup>12</sup>. In other words, it measures the “degree of chaos” in a timeseries. It relates directly to “long memory” of such series, i.e., at which rate the autocorrelations between one datapoint and all previous datapoints decays. This concept was originally developed and used by the British hydrologist H. E. Hurst himself when studying the flooding and droughts of the Nile in 1913 (**Hurst, The Nile Flood of 1913, 1913**). Hurst used this measure to find the path dependence or persistence of trend that occurred at the river; where seemingly random floods were followed by heavier overflows and dry seasons by even worse droughts. His work led directly to the construction of the Aswan High Dam between 1954 and 1960 earning him the title “Father of the Nile”. This concept now reappears today in quantitative finance in a what is called “rough volatility”. Most prominently in the works of Jim Gatheral and Mathieu Rosenbaum in 2014 (**Gatheral, Jaisson, 2014**).

---

<sup>10</sup> This concept it also known as self-similarity

<sup>11</sup> Made possible by data from Olsen & Associates in Zurich. See (**Clive Davidson, Wildly Random Market Moves, 1997**)

<sup>12</sup> The fractal dimension, denoted by D, is a ratio that indexes how complexity in a pattern changes with the scale of measurement. E.g., switching from daily to 5 min intervals of observation.

**Rosenbaum, Volatility is Rough, 2014**<sup>13</sup> where the authors estimate the influence that past changes have on future moves and try to capture volatility behavior with only a few parameters. These parameters not only are able to assess time series characteristics of realized volatility, but also capture stylized facts of implied volatility surfaces. In fact, the model proposed by Gatheral and Rosenbaum can be seen as an extension the Multifractal Random Walk (MRW) model by **(Muzy, Delour, Bacry, Modelling fluctuations of financial time series: from cascade process to stochastic volatility model, 2000)**<sup>14</sup>, who themselves used insights by Mandelbrot to incorporate the fractal nature of volatility (**Jean-Philippe Bouchaud, 2020**). While roughness was set constant in the MRW model (**Gatheral, et al, 2014**) incorporates the Hurst parameter, allowing the roughness to be tuned. Ever since the seminal work by Gatheral, Jaisson, and Rosenbaum literature in the field of rough stochastic volatility has sprout, trying to improve upon itself constantly. Thus, the site <https://sites.google.com/site/roughvol/home> was established, dedicated to keeping track of related literature. Over 130 works are listed since the landmark paper by Gatheral, including works on realized and implied volatility. Shortly after publication, in 2015 Bayer, Friz, and Gatheral propose the rough Bergomi (rBergomi) model incorporating the concept of rough volatility in an option pricing framework: **(Bayer, Friz, Gatheral, Pricing under rough volatility, 2015)**. Based on the work by Lorenzo Bergomi, a quant at Squarepoint Capital, the authors tune the model s.t. it allows for rough volatility paths. With only four parameters that all carry an explicit meaning manual calibration to bid/ask observed volatility is possible with relative ease according to the authors. Within the same month January, Fukasawa incorporated the concept of rough fractional volatility via the Hurst parameter in a model that is able to replicate the term structure of implied volatility skew and is consistent with the observed power law behavior of options in time to maturity **(Masaaki Fukasawa, Short-time at-the-money skew and rough fractional volatility, 2015, Finance)**. Especially examining short to maturity behavior, he develops an asymptotic expansion of IVs as time to expiration approaches zero. Half a year later, Bennedsen applied the novel concept of roughness in modelling electricity spot prices and found evidence of roughness in five of six energy markets under consideration **(Bennedsen, A Rough Multi-Factor Model of Electricity Spot Prices, 2015)**. In these markets the “rough component” improved short term forecasting of prices. In the same year still **(Garnier, Sølna, Correction to Black--Scholes Formula Due to Fractional Stochastic Volatility, 2015)** provide further evidence that IV has

---

<sup>13</sup> At this time only a working version was published. The official publication was in 2018

<sup>14</sup> Moreover, this model captures important empirical facts, such as fat-tailed distribution of log-returns and long memory of changes in volatility

a term structure that can be approximated by a power law and apply this fact to the Black Scholes framework. Then in 2016, (**El Euch, Rosenbaum, The characteristic function of rough Heston models, 2016**) experiment with applying a Heston model under assumption of rough volatility via a fractional Riccati equation. In (**Bennedsen, Lunde, Pakkanen, Decoupling the short- and long-term behavior of stochastic volatility, 2016**) the authors find evidence that volatility is indeed rough and very persistent. They report using roughness in forecasting volatility can indeed improve performance as explored by (**Gatheral, et al, 2014**). More, they find that there exists a component of persistence that seems to improve forecasting further. Only two months later, the same authors publish the work (**Bennedsen, Lunde, Pakkanen, Hybrid scheme for Brownian semistationary processes, 2016**), which finally allows relatively quick estimation of the rough Bergomi model without relying on the slow calculation process, originally proposed by (**Gatheral, et al, 2014**). This is achieved with what the authors term the “Hybrid scheme”. Based on this concept, (**McCrickerd, Pakkanen, Turbocharging Monte Carlo pricing for the rough Bergomi model, 2018**) devise a ready to use framework to estimate the rough Bergomi model given some arbitrary parameter values. More, the authors provide publicly available code with which the rBergomi model can be estimated<sup>15</sup> and (theoretical) options priced<sup>16</sup>, as we are exploring in Chapter 4 later. In 2017 (**Livieri, Mouti, Pallavicini, Rosenbaum, Rough volatility: Evidence from option prices, 2017**) explore parameter estimation in the physical probability space as well as in the risk neutral world. They find that the Hurst parameter estimated from at-the-money (ATM) options of the S&P 500 index is slightly greater than what is obtained from realized volatility data. They explain this by the additional smoothing of IV due to remaining time to expiration of options considered. Shortly thereafter (**El Euch, Rosenbaum, Perfect hedging in rough Heston models, 2017**) show how (at least theoretically) perfect hedging is possible in rough Heston models with the forward variance curve and the underlying asset alone. With applications of the rough Heston model already published, curiously only in 2019, (**El Euch, Gatheral, Rosenbaum, 2019, Roughening Heston, Risk, pp. 84-89**) present the rough Heston model. The authors here apply the high tractability and fast calibration times of the standard Heston model within the context of rough volatility, facilitating great fits to the IV surface. In 2018 and 2019 Bayer, Horvath, et. al, experiment using neural networks to aid the

---

<sup>15</sup> In their framework, the model is not directly calculated but estimated, thereby providing significant speed improvements

<sup>16</sup> Pricing is an ambiguous term in financial literature, as well as in practice. Here we mean that one is able to find corresponding implied volatilities for a given option contract. Once the implied volatility is known, a “simple” framework, such as the Black and Scholes model, can find corresponding fair prices for such contracts.

computationally expensive calibration of rough volatility models such as the rBergomi model. Both works (**Bayer, Stemper, Deep calibration of rough stochastic volatility models, 2018**) and (**Horvath, Muguruza, Tomas, Deep Learning Volatility, 2019**) are being discussed in detail in Chapter 5, exploring distinct neural network architectures of point and grid-wise deep learning. The latter concept is then applied via the neural network that was originally built by (**Rømer, Historical Analysis of Rough Volatility Models to the SPX Market, 2020**). Instead of neural nets, (**Zeron, Ruiz, Tensoring volatility calibration, Calibration of the rough Bergomi volatility model via Chebyshev Tensors, 2020**) explore using tensors for the purpose of calibration. The authors find that the tensors provide additional efficiency over deep neural nets, while offering similar accuracy. In 2019, Takaishi studies roughness of log-volatility in Bitcoin. He finds that from time series analysis the Hurst parameter in Bitcoin to be less than  $\frac{1}{2}$  and changing in time. Moreover, he uncovers that log volatility has a multifractal property that comes at least partly from the distributional property (**Tetsuya Takaishi, Rough volatility of Bitcoin, 2019**). A little later in 2019, (**Dandapani, Jusselin, Rosenbaum, From quadratic Hawkes processes to super-Heston rough volatility models with Zumbach effect, 2019**) developed what (**Jean-Philippe Bouchaud, 2020**) calls the most realistic financial time series model to date. The key factor of distinction presents the so-called Zumbach effect. In 2009 (**Zumbach, Time reversal invariance in finance, 2009, Quantitative Finance, 9(5):505–515**) found that empirical time series of financial prices are not invariant to reversal of order of time. Thus, present, future, and past are not equivalent. What seems an obvious fact, poses a problem from a modeling perspective. Even in physics the author quotes that Newton's model for mechanics, the model by Maxwell for electromagnetism, the Einstein model for general relativity, and more are all invariant to reversal of time. Also in finance, models typically do not distinguish future from the past. Zumbach instead found that realized price trends strongly influence future volatility. This effect finally solves the famous riddle in quantitative finance of finding one model able to calibrate to implied volatilities from both the S&P 500 as well as the VIX volatility index (**Gatheral, Jusselin, Rosenbaum, The quadratic rough Heston model and the joint S&P 500/VIX smile calibration problem, 2020**). Most recently and inspired by the works from Bayer and Horvath on neural networks, (**Rosenbaum, Zhang, Deep calibration of the quadratic rough Heston model, 2021**) show how the quadratic rough Heston model can be applied using neural networks, validating its fit and also showing the networks capabilities for hedging. In (**Fukasawa, Gatheral, A rough SABR formula, 2021**), for the well-known SABR model, Gatheral once again assists in applying the concept of rough volatility to yet another popular

stochastic volatility model, showing the model is at least theoretically able to achieve great fits to IVs. (**Garcin, Forecasting with fractional Brownian motion: a financial perspective, 2021**) develops a forecasting strategy based on the assumption that log-prices follow a fractional Brownian motion (fBm) and validates its application on high-frequency FX data.

With a time-tested foundation, rough volatility models, although based on a concept that is over 100 years old, seem to be a new direction for academia. Models such as Black Scholes and SABR necessitate error-prone calculations, while rough volatility models are quicker and even said to be more accurate and can generate an implied volatility surface (see Figure 1) for different option strikes within one calculation (**Kris Devasabai, Rough volatility's steampunk vision of future finance, 2021**). In practice these models are given much more scrutiny however (**Devasabai, 2021**): According to Devasabai many deem the models still largely untested and expensive to implement as budgets are going towards other machine learning projects. He notes this might be a mistake however, as Société Générale estimate that bid/ask spreads for options and futures of the volatility index VIX would decrease by up to 20% with adoption of rough volatility models and that arbitrage strategies exploiting asymmetries between traditional and rough volatility models may be a lucrative endeavor. A recent study from March 2021 points at the use of rough volatility models in the context of deep hedging (**Cesa, Mannix, Deep hedging strays when volatility gets rough – study, 2021**): As such approaches are reported to have difficulties when markets show memory effects, rough volatility models capture those effects and can alleviate related issues by cross checking and validating output of black box (machine learning) algorithms (**Horvath, Teichmann, Zuric, Deep Hedging under Rough Volatility, 2021**). Moreover, the authors note that the reverse, using machine learning to calibrate rough volatility models and check their output against Black Scholes implied volatilities is a useful application as well. Nevertheless, rough volatility may also bring significant benefits to standard hedging approaches as shown by (Fukasawa, Horvath, Tankov, Hedging under rough volatility, 2021).

While the journey of finding the “perfect” volatility model is not yet over, the most recent developments at least give hope that we are not far. With the overwhelming evidence that volatility is rough and the availability of stochastic volatility models that are able to take stock of this effect, better forecasts can be achieved, and calibration becomes more accurate. It thus seems fruitful to study roughness in the market and how such models can be used in a practical environment for calibration and pricing of options. In this master thesis we will focus

specifically on verifying that volatility is rough and on evaluating the power of the rBergomi model to fit to observed and liquid, i.e., recently and highly traded European Call options. Moreover, we will explore two different methods of parameter estimation for the rBergomi model: Namely we will employ **(a)** a more “conventional” Monte Carlo (MC) simulation approach by (McCrickerd, Pakkanen, 2018), as well as **(b)** a Neural Network assisted calibration technique pioneered by Bayer (2018) and Horvath (2019) and realized by Sigurd Rømer (2020). Here we will discuss architecture of such neural networks, benefits and potential problems, as well as how such a deep learning approach can be used in practice.

Furthermore, for the former MC simulation, we will compare two distinct minimization techniques in respect to how reliable they are at finding global minima for the problems under consideration. Under the assumption that rough volatility models fit good quality institutional grade empirical data adequately well, one implicit question becomes to what degree retail investor grade data is able to replicate the same observed stylized behavior of rough volatility models. This is analyzed for **(i)** backward looking realized volatility data, for which we compare two different variance estimation techniques, based on daily open, high, low, close (OHLC) data, as well as for **(ii)** forward looking option implied volatility data, for which daily close data of option chains is used. This we analyze as a byproduct to our empirical results. To make the necessary computations feasible, we rely on the GitHub repositories and other sources from various authors, namely Gatheral, McCrickerd, Horvath, and Sigurd Rømer<sup>17</sup>. With necessary adaptations, the accompanying code for this thesis was developed in Python 3.8 and can be found in our GitHub repository here: <https://github.com/srich86/Rough-Volatility-and-rBergomi-Model>. This also includes manually downloaded option price data for 87 trading days obtained for the following seven stock symbols<sup>18</sup> summarized in Table 1 from yahoo finance. This set of data is used to evaluate the rBergomi model and price European call options in the subsequent chapters. Please see Appendix A2 for a list of dependencies used. The methodology of the program code is discussed in the respective sections. All figures and plots in this work are self-created unless stated otherwise.

---

<sup>17</sup> See the following source codes:

[https://tpq.io/p/rough\\_volatility\\_with\\_python.html](https://tpq.io/p/rough_volatility_with_python.html)

[https://github.com/ryanmccrickerd/rough\\_bergomi](https://github.com/ryanmccrickerd/rough_bergomi)

<https://github.com/amuguruza/NN-StochVol-Calibrations>

[https://github.com/sigurdroemer/rough\\_volatility](https://github.com/sigurdroemer/rough_volatility)

<sup>18</sup> The data was manually scraped from yahoo finance at close over the observation period from 28-04.2021 till 01.10.2021 and is provided as is. We make no claim about accuracy or completeness.

<b>Company / Asset Name</b>	<b>Stock Symbol</b>
Apple	AAPL
AT&T	T
Gamestop	GME
Microsoft	MSFT
Nio	NIO
S&P 500	SPX
Tesla	TSLA

*Table 1: List of financial assets and their corresponding stock symbols*

This thesis is structured as follows: Firstly, we will explore the concept of rough volatility from a time-series perspective (past realized data), using high-frequency intraday volatility data and daily OHLC data in **Chapter 2**. Verifying that the model framework of rough volatility is able to explain empirical implied volatility (IV), this work will rely on the rough Bergomi model, introduced in **Chapter 3**. Specifically, we are interested in how well the model generated IVs fit to the full implied volatility surface. To practically analyze and price options in the following, we will use two distinct methods to obtain model-driven IVs and assess the accuracy of the rBergomi model: Namely, we introduce calibration by simulation (**Chapter 4**) and by means of using a deep neural network (**Chapter 5**). Both methods are then used in **Chapter 6** to assess the persistence of the corresponding parameters needed to calibrate the model, how well the parameters capture stylized features of empirical implied volatility and how well the theoretical fit coincides with observed data. As an addition we will also explore how the parameters from time series analysis and from calibration to empirical IVs compare, before concluding in **Chapter 7**.

# Rough Volatility

The foray into modeling volatility stems from the desire to understand risk and how changes are driven in spot prices of financial assets, but also in applications in physics or other natural sciences. This enables us to make sound predictions regarding the evolution of any time series. “Rough Volatility” is a term coined by Jim Gatheral in his seminal paper (*Gatheral, Jaisson, Rosenbaum, Volatility is Rough, 2018*) to describe how volatility evolves through time. Volatility is expressed as the standard deviation of a given time-series; in technical terms it is the square root of variance<sup>19</sup>:  $\sigma = \sqrt{\sigma^2}$ . In order to model the evolution of any series that features unpredictability to some degree (and its corresponding volatility), relying on a Brownian motion<sup>20</sup> to drive stochastic changes within this process is the current modus operandi. While much of the research into how volatility behaves has relied on assuming a long memory (“smooth”) process<sup>21</sup>, (*Gatheral et al., 2018*) break with this concept, or at least relieve it of its importance. As empirical evidence points to the presence of long memory<sup>22</sup>, literature and volatility estimation models have tried to incorporate this well proven fact. However, (*Gatheral et al, 2018*) take a wildly different approach: As opposed to assuming long memory and smoothness of the process, they introduce roughness or “jaggedness” to it; hence the title of their work “Volatility is Rough”. (*Gatheral et al, 2018*) take this bold step after empirically estimating the smooth- or jaggedness of real-world volatility data, finding that the process is indeed not smooth. More precisely, they find that the autocorrelation function of volatility does not exhibit long memory, i.e., it does not decay as a power law with an exponent smaller than 1. With this information, they also generate new volatility paths under rough volatility. Curiously, here classical statistical analysis also suggests the presence of long memory, for volatility paths that are generated under rough volatility. Hence, (*Gatheral et al, 2018*) conclude that the study into long memory does not contribute much to the understanding of volatility overall.

---

<sup>19</sup> While the standard measure of variance is simply  $Var(X) = \frac{\sum X - \bar{X}}{N-1}$ , other more sophisticated estimators have been developed in the financial literature, of which some are introduced in this chapter.

<sup>20</sup> Prof. John Cochrane has provided publicly available YouTube videos on Brownian motions and asset pricing: [https://www.youtube.com/playlist?list=PLAXSVuGaw0KxTEN\\_cy-RCuEzzRdnF\\_xtx](https://www.youtube.com/playlist?list=PLAXSVuGaw0KxTEN_cy-RCuEzzRdnF_xtx)

<sup>21</sup> Long memory is a concept from traditional time series analysis, indicating that today’s value is correlated not only to yesterday’s value, but also to many days prior to this. Smoothness relates to the decay of this correlation through time, suggesting correlation does not fall off a cliff, but decays slowly. (*Gatheral et al., 2018*) formalizes this concept as  $\sum_{k=0}^{+\infty} Cov [W_1^H, W_k^H - W_{k-1}^H] = +\infty$ , where W is a fractional Brownian Motion and H is the so-called Hurst parameter.

<sup>22</sup> See e.g. (Bollerslev, Mikkelsen, Modeling and pricing long memory in stock market volatility, 1996)

Subsequently to incorporate their observations, the authors build their model on the well-established fractional stochastic volatility (FSV) model by (*Comte, F. and Renault, E., Long memory in continuous-time stochastic volatility models, 1998*) and call it the Rough Fractional Stochastic Volatility (RFSV) model. Specifically, they use its framework to incorporate the fractional nature of volatility, meaning that volatility is not a smooth but rough process. Crucially, the FSV model relies on a process called fractional Brownian Motion, or fBm for short, developed by (*Mandelbrot, B.B. and Van Ness, J.W., Fractional Brownian motions, fractional noises and applications, 1968*).

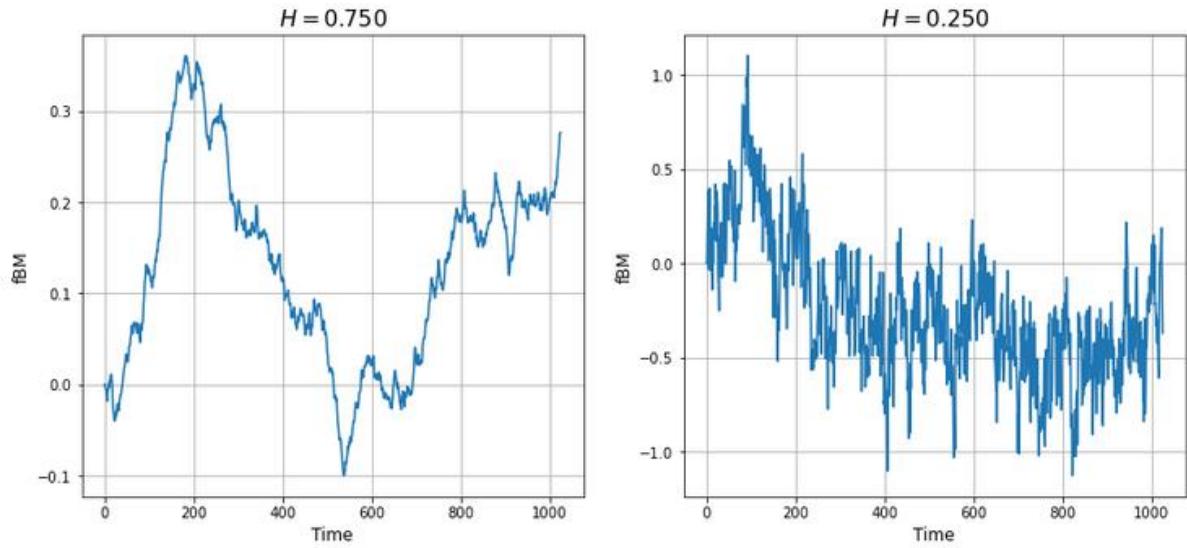


Figure 3: Fractional Brownian Motion for different Hurst Parameters

Following (*Mandelbrot et al., 1968*) a fBm is defined as  $W_t^H, H \in (0, 1)$ , where W is a standard Brownian motion and H is the so-called Hurst-Parameter, which controls the smooth or roughness of the process.

Literature on the subject of fractional Brownian motions is quite extensive as its wide application to physics, biology and finance make it interesting for various fields. (Biagini, Hu, Øksendal, Zhang, Stochastic calculus for fractional Brownian motion and applications, 2008) explain how calculus for fractional Brownian motions can be developed, as fBm is neither a Markov process nor a semi-martingale, thus rendering classical stochastic Ito calculus infeasible. Among others<sup>23</sup>, further reading material is available from (Nourdin, Selected

---

<sup>23</sup> Further literature and more comprehensive guides are available from:

- (Nualart, The Malliavin calculus and related topics, 2006)
- (Mishura, Stochastic calculus for fractional Brownian motion and related processes, 2008).
- (Coeurjolly, Simulation and identification of the fractional Brownian motion: a bibliographical and comparative study, 2000).

Aspects of Fractional Brownian Motion, 2012), who provides extensive proofs for almost all results in his book. Here, we follow (Shevchenko, Fractional Brownian Motion in a Nutshell, 2014) for an introduction and how to construct simulation of the process, which can be observed in Figure 3<sup>24</sup>.

A fBm (see Figure 3) is a self-centered Gaussian process with its auto-covariance function ( $I$ ) defined as:

$$\mathbb{E}[W_t^H W_s^H] = \frac{1}{2} \{t^{2H} + s^{2H} - |t - s|^{2H}\}, \quad (1)$$

where  $t$  and  $s$  denote different points in time. Since a Gaussian process is entirely defined by its covariance and mean, therefore also the distribution of  $W_t^H$ , for a fixed Hurst parameter, is determined by (1). With  $H = \frac{1}{2}$  the fBm  $W_t^H$  becomes a standard Brownian motion again, and (1) becomes the simple covariance of a standard Brownian motion between the increment of  $t$  and  $s$ . Thus, a fBm can be seen as a generalization of Brownian motions by allowing the Hurst parameter to differ from values of  $\frac{1}{2}$ . Crucially, for Hurst parameter  $H \neq 1/2$  the fBm allows for a short to long memory property, also known as persistence, which make it an ideal tool for modeling non-trivial properties as (Nourdin, 2012) states. In fact, Figure 3 shows this (non-) persistence clearly. For a Hurst parameter larger than  $1/2$ , the series persists (is trending) for some time, while there is no clear trend visible for Hurst parameter smaller than  $1/2$ . Now thinking that each fBm from (1) is actually an increment in itself, as time is a continuous process, leads us to:

$$\begin{aligned} & \mathbb{E}[(W_{t_1}^H - W_{s_1}^H)(W_{t_2}^H - W_{s_2}^H)] \\ &= \frac{1}{2} \{|t_1 - s_2|^{2H} + |t_2 - s_1|^{2H} - |t_2 - t_1|^{2H} - |s_2 - s_1|^{2H}\}, \end{aligned} \quad (2)$$

which helps uncover the properties of a fractional Brownian motion, as given by (Shevchenko, 2014):

- **Stationary increments:** For any increment of length  $k$  between two fBm, s.t.  $Y_t = W_{t+k}^H - W_t^H$ , it can be deduced from (2) that both  $Y_t$  and  $W_t^H$  share the same covariance

---

<sup>24</sup> Namely we use a simulation technique that relies on modeling fractional Gaussian noise via the Cholesky method by (Asmussen, Stochastic simulation with a view towards stochastic processes, 1998), it is realized via the python package found here: <https://pypi.org/project/fbm/>

function. Which implies that the distribution of increments is the same for any point in time. (Shevchenko, 2014) notes that such properties are desirable when modeling the evolution of time-homogeneous processes<sup>25</sup>.

- **Self-similarity** in effect means invariance of time of scale of the process. In simple terms, that the behavior for any time interval is the same. Formally for  $Z_t = W_{\alpha t}^H$  and  $\alpha$  constant,  $Z_t$  has the same covariance as in (1). Therefore also the same distribution as given by  $\alpha^H W_t^H$ .
- **Dependence of increments:** Under the assumption that  $s_1 < t_1 < s_2 < t_2$  the two intervals  $[W_{t_1}^H - W_{s_1}^H], [W_{t_2}^H - W_{s_2}^H]$  in (2) do not intersect and we can distinguish the covariance function depending on the regime of the Hurst parameter<sup>26</sup>:

$$\mathbb{E}[(W_{t_1}^H - W_{s_1}^H), (W_{t_2}^H - W_{s_2}^H)] > 0, \quad H \in \left(0, \frac{1}{2}\right), \quad (3)$$

and

$$\mathbb{E}[(W_{t_1}^H - W_{s_1}^H), (W_{t_2}^H - W_{s_2}^H)] < 0, \quad H \in \left(\frac{1}{2}, 1\right). \quad (4)$$

This means that the Hurst parameter is able to control the (anti-) persistence of the time series of the fBm: With  $H \in (0, 1/2)$ , the process would exhibit non-persistence: If the process was decreasing in the past, it becomes more likely to be increasing in the future. Whereas for  $H \in (1/2, 1)$  the fBm would be highly persistent, see Figure 3.

(Comte et al, 1998) force the observed long memory by choosing  $H > 1/2$ , as motivated above, whereas  $H < 1/2$  constitutes the main proposition by (Gatheral et al, 2018). A lower value of  $H$  corresponds to greater roughness, indicating more erratic up and down movements in volatility<sup>27</sup>, while a higher value for  $H$  corresponds to greater smoothness, meaning volatility tends to be flatter, or not move as much over the observed time horizon<sup>28</sup> (see Figure 3). Given this information, we will now examine the estimation of the true roughness parameter.

---

<sup>25</sup> Which according to (Zumbach, 2009) is not the case.

<sup>26</sup> For a proof refer to Appendix B1

<sup>27</sup> This can be thought of as days with high volatility are followed by days of low volatility.

<sup>28</sup> This was also explored in (Bennedsen, Lunde, Pakkanen, Hybrid scheme for Brownian semistationary processes, 2017)

The rest of this chapter is organized as follows: Firstly, this master thesis discusses the empirical findings of (*Gatheral et al, 2018*) and subsequently it introduces the model suggested by the authors in the next step. This chapter concludes by validating their findings with simple daily Open, High, Low, Close (OHLC) data for selected assets.

## **Empirical Time Series Observations**

(*Gatheral et al, 2018*) set out to understand the rough- or smoothness of the process that controls variance. While the volatility process and instantaneous variance is not observable, they approximate the volatility of various financial assets via two different volatility measures, of which the second will be of interest here:

For DAX and Bund Futures contracts they estimate the so-called integrated variance between 9 and 10 am Central European Time (CET). For estimation they rely on a model with uncertainty zones proposed by (*Robert, C.Y. and Rosenbaum, M., A new approach for the dynamics of ultra-high-frequency data: The model with uncertainty zones, 2011*) and (*Robert, C.Y. and Rosenbaum, M., Volatility and covariation estimation when microstructure noise and trading times are endogenous, 2012*). This approach allows (*Gatheral et al, 2018*) to use all available data over the 1-hour window to estimate the variance in a proper manner, as the authors report.

Secondly, for various stock market indices<sup>29</sup> (*Gatheral et al, 2018*) rely on the cleaned so-called realized kernel estimates<sup>30</sup> for integrated variance<sup>31</sup> of one day reported by the Oxford-Man Institute of Quantitative Finance<sup>32</sup>. (*Gatheral et al, 2018*) note however that the resulting roughness estimations are expected to be upward biased as the whole trading day is being considered for these estimates. This thesis will here replicate the author's findings<sup>33</sup> for the second proposed estimator with the same data and later show the same conclusion also holds for uncleaned daily OHLC data available from publicly available sources such as yahoo

<sup>29</sup> Indices considered are SPX, FTSE2, N2252., GDAXI2, RUT2, AORD2, DJI2, IXIC2, FCHI2, HSI2, KS11, AEX, SSMI, IBEX2, NSEI, MXX, BVSP, GSPTSE, STOXX50E, FTSTI, and FTSEMIB

<sup>30</sup> Kernel estimates are reported to be robust to noise and are defined as

$$RMt = \sum_{h=-H}^H k\left(\frac{h}{H+1}\right) \gamma h \quad \text{with } \gamma h = \sum_{j=|h|+1}^n x_{j,t} x_{j-|h|,t}$$

See here for details: <https://realized.oxford-man.ox.ac.uk/documentation/econometric-methods>

<sup>31</sup> Which constitutes an approximation to instantaneous variance and m.m., to instantaneous volatility.

<sup>32</sup> Data publicly available from <https://realized.oxford-man.ox.ac.uk/data/download>

For further reading on the performance of such volatility estimates, see (*Gatheral, J. and Oomen, R.C., Zero-intelligence realized variance estimation, 2010*)

<sup>33</sup> See [https://www.tpq.io/p/rough\\_volatility\\_with\\_python.html](https://www.tpq.io/p/rough_volatility_with_python.html)

finance<sup>34</sup>. In general, the same results hold for the volatility estimators reported for the DAX and Bund Futures contracts, but as the volatility data is not publicly available, this work here relies on the introduced kernel estimates.

In order to make the connection between the Hurst parameter, which measures roughness, and the empirical data, (*Gatheral et al, 2018*) decide on the following process:

1. Take a suitable measure for realized volatility. For this we will use the log of the square root of variance, i.e., the log-volatility<sup>35</sup>:  $\log(\sigma_t)$ .
2. Define a suitable array (here 1 to 100 days)<sup>36</sup> for the step size  $\Delta$  between discrete observations of log-volatility
3. Regress the log of the  $\Delta$  array against the log of the moment generating function  $m(\Delta, q)$  for each moment  $q$ , given by

$$m(q, \Delta) = \frac{1}{N} \sum_{k=1}^N |\log(\sigma_{k\Delta}) - \log(\sigma_{(k-1)\Delta})|^q. \quad (5)$$

We call it the moment generating function, as  $m(q, \Delta)$  in (5) defines the behavior of the distribution of the log-volatility increments c.p. for different values of  $q$ . The step size  $\Delta$  then simply defines the increment<sup>37</sup> between the observations. In the following we will feed  $m(q, \Delta)$  with estimates of realized volatility, thus also our results will be estimates.

4. For the regression in point 3, we define the slope as  $\zeta_q$  and the intercept as  $\alpha_q$  for each moment  $q$ .
5. We extract  $\nu_{est}$  (called the level of, or volatility of volatility) as one of the required parameters for our later volatility model in the same step.
6. Finally, we regress each moment  $q$  against  $\zeta_q$  to extract the slope, which we will call  $H_{est}$  as the estimate for the Hurst parameter.

This methodology relies on the central assumption (6) that the volatility process (which we are trying to measure) characterized by the moment generating function  $m(q, \Delta)$  belongs to a

---

<sup>34</sup> <https://finance.yahoo.com/>

<sup>35</sup> The log is simply used to make the series more normal, allowing for easier handling of the volatility time-series

<sup>36</sup> (*Gatheral et al., 2018*) note that modelling volatility may be only relevant at scales of more than one day

<sup>37</sup> E.g., 1 day, 5 days, 30 days, 1 year, etc

certain smoothness space, which is mathematically characterized by a Besov space  $\mathcal{B}_{q,\infty}^{s_q}$ <sup>38</sup> with  $N = T/\Delta$ .

$$N^{qs_q} m(q, \Delta) \rightarrow b_q, \quad s_q > 0 \text{ and } b_q > 0 \text{ as } \Delta \rightarrow 0. \quad (6)$$

In other words that our (theoretical) volatility process has a characterizing feature that determines its smoothness  $s_q$  which can be extracted (estimated). The example by (Gatheral et al., 2018) gives a clear idea of the concept:

For a volatility process  $\log(\sigma_t)$  that is a fBm with Hurst exponent  $H$ , it holds that

$$s_q = H, \quad (7)$$

for any moment  $q$ . This central assumption of the estimation procedure means we are able to extract the Hurst parameter directly from the slope of the regression between  $\zeta_q$  and each moment  $q$ .

Given these definitions, we are now able to replicate the observations first shown in (Gatheral et al., 2018). For the general behavior and roughness estimation, we will focus exclusively (for now) on the Standard and Poor's (S&P) 500 stock market index, as similar behavior has been observed for other indices as well, see Table 11 and also (*Bennedsen, Lunde, Pakkanen, Decoupling the short and long-term behavior of stochastic volatility, 2021*), who validate the existence of rough volatility for over 2000 financial assets.

---

<sup>38</sup> For further readings on smoothness spaces, please refer to (Rosenbaum, M., First order p-variations and Besov spaces, 2009) and (Rosenbaum, M., A new microstructure noise index, 2011)

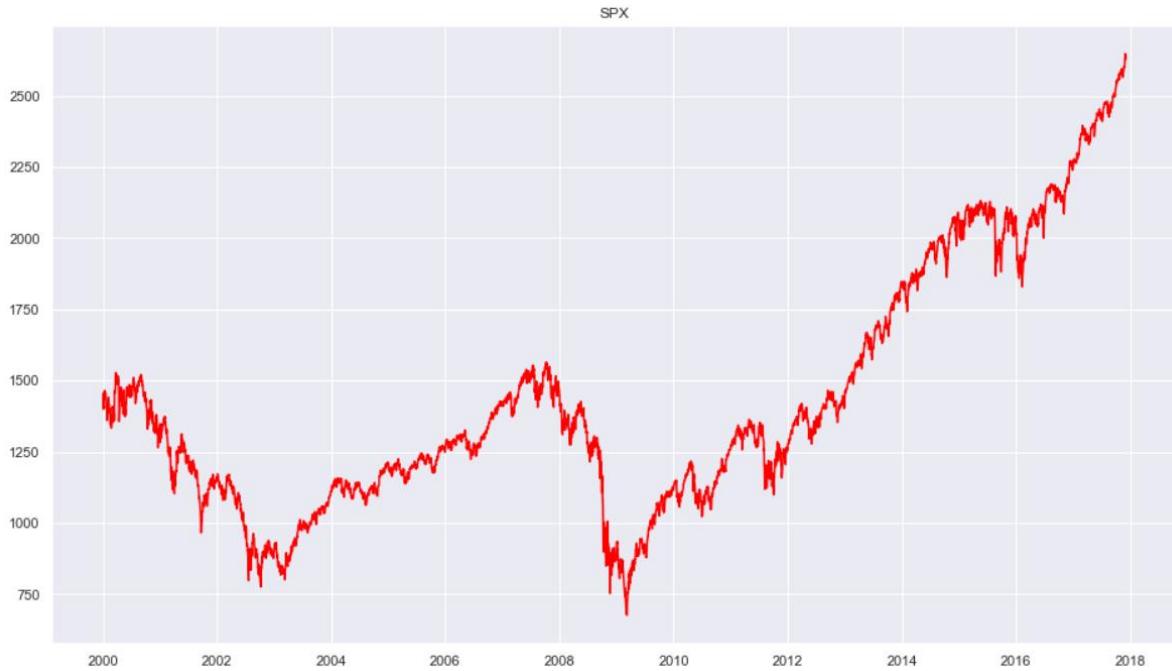


Figure 4: SPX Daily Adjusted (Adj.) Close Data between 01.01.2000 and 05.12.2017. All data derived from yahoo finance

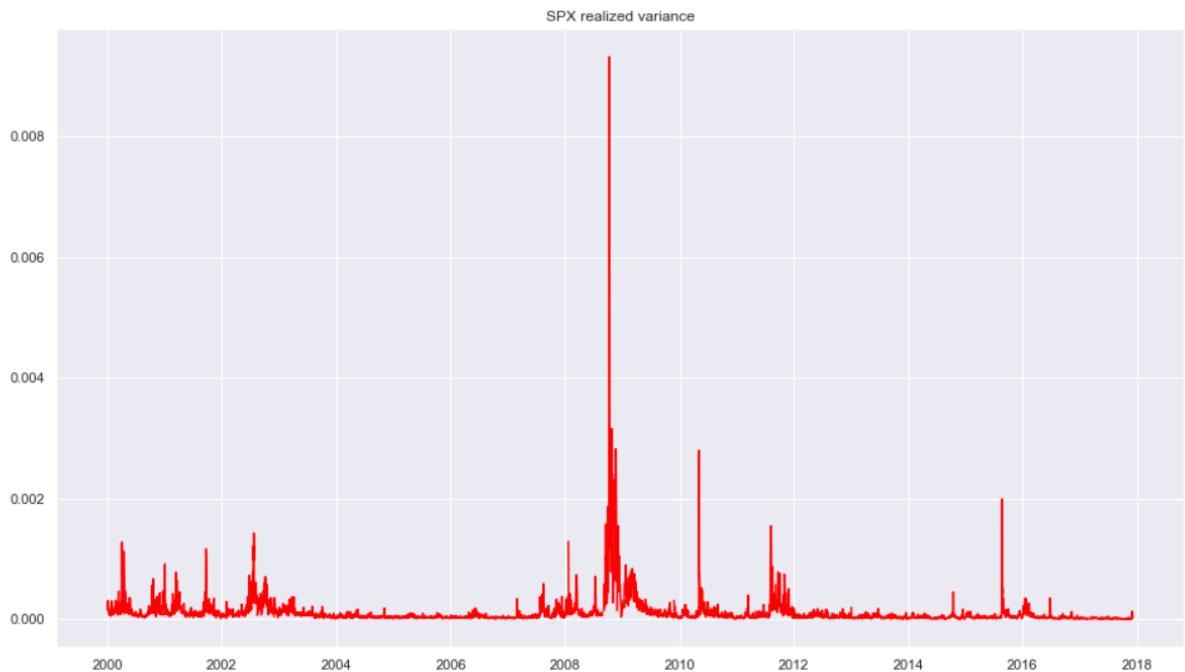


Figure 5: SPX Daily realized kernel estimates for variance between 01.01.2000 and 05.12.2017. All data derived from the Oxford-Man Institute of Quantitative Finance

In Figure 4 we can see the behavior of the S&P 500 index throughout the observation interval used in (Gatheral *et al.*, 2018). Figure 5 then gives a view of the daily realized estimates for volatility over the same period. For our task of understanding and modeling volatility the

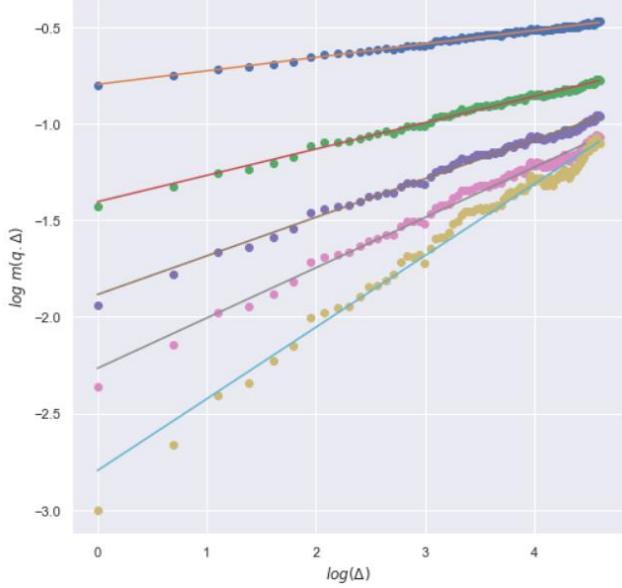


Figure 6: Regressing  $\log(\Delta)$  against  $\log(m(q, \Delta))$  yields us the intercept  $\alpha_q$  and the slope  $\zeta_q$ . Results for SPX data. Data derived from the Oxford-Man Institute of Quantitative Finance.

Using a simple linear regression of the form

$$\log(m(q, \Delta)) = \alpha_q + \zeta_q \log(\Delta) + \epsilon, \quad (8)$$

we obtain  $\zeta_q$  for each value of  $q$  (see Figure 6). Within the same step, we also estimate

$$v_{est} = \sqrt{e}^{\alpha_{q=2}}, \quad (9)$$

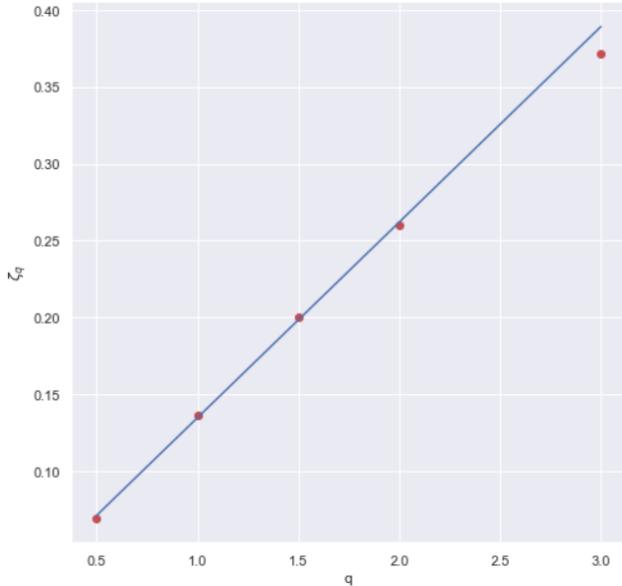


Figure 7: Regressing each  $q$  against its corresponding  $\zeta_q$  value, yields us the slope  $s_q$  and the intercept  $\beta_q$ . Results for SPX data. All data derived from the Oxford-Man Institute of Quantitative Finance

second figure is of main interest: Our goal in this sub-section is to extract  $H_{est}$ , an estimate of the roughness of realized volatility. Therefore, we follow the structure set out in the beginning and start by obtaining the volatility data: Given the publicly available dataset from the Oxford-Man Institute of Quantitative Finance we extract the kernel variance estimates  $\sigma_t^2$  and compute  $\log(\sigma_t)$ . Next, for a predefined list of step size  $\Delta$  and a vector of moments  $q$ , we obtain estimates for  $m(q, \Delta)$  by means of (5).

which will become relevant when we discuss the RFSV in the next part of this chapter.

Next, regressing  $q$  against  $\zeta_q$  with

$$\zeta_q = \beta_q + s_q q + \epsilon, \quad (10)$$

we find that  $q$  is directly proportional to  $\zeta_q$ . Thus, we obtain the slope  $s_q$ , which equates to the estimate for the Hurst parameter  $H_{est}$  following (Gatheral et al., 2018) in their main assumption (6).

Remarkably we estimate  $H_{est} \sim 0.127$ . This result is clearly at odds with the assumptions made within the FSV framework by (Comte et al., 1998).

(Gatheral et al., 2018) also report  $H_{est}$  and  $\nu_{est}$  values for various other indices in Table 11 in Appendix B2, norming  $q = 2$  and calculating  $H_{est} \approx \zeta_2/2$  presents a much simpler estimate<sup>39</sup> (made possible, since  $\zeta_q \sim Hq$ ). Comparing  $h_{est}$  of 0.129 for the SPX from Table 11 with the  $H_{est}$  value of 0.127 obtained from regression (10), we see both estimates very close together. This is because  $\zeta_q$  exhibits constant (i.e., linear) scaling in  $q$ , as is evident from Figure 7 Thus, also the much simpler estimate for the Hurst parameter under  $q = 2$  can be seen as valid. The evidence that all estimates of the Hurst parameter are significantly smaller than 0.5

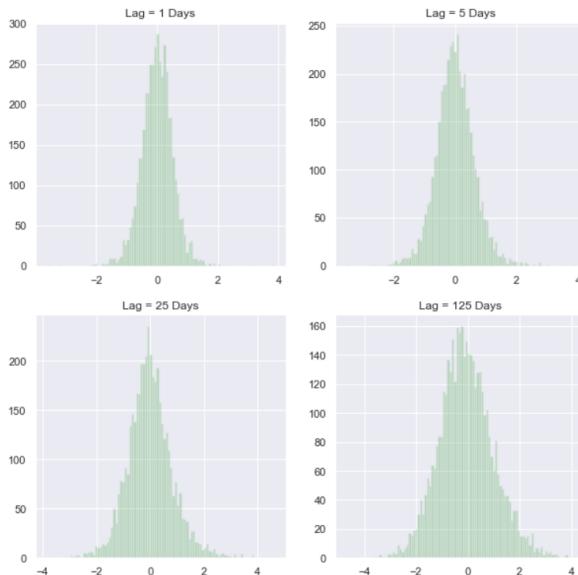


Figure 8: Distribution of increments of log-volatility (1-, 5-, 25, and 125-days). SPX data derived from the Oxford-Man Institute of Quant. Fin.

is of great importance when choosing an appropriate model to mimic the observed behavior.

(Gatheral et al., 2018) also raise the question whether  $H$  changes through time. One of the contributions of this work is that we find  $H$  not only to vary, but that the time series of roughness of volatility is influenced by the price disagreement in the stock market<sup>41</sup>. Estimating  $H$  with the simpler algorithm under  $q = 2$ , the estimate for  $H$  for SPX is computed over a

<sup>39</sup> This behavior comes from the scaling property of the increments of log-volatility, defined as  $\mathbb{E}[|\log(\sigma_{k\Delta}) - \log(\sigma_{(k-1)\Delta})|^q] = b_q \Delta^{\zeta_q}$  with  $\zeta_q = qs_q$ . Thus, recalling assumption (6) that the volatility process has a characterizing feature that determines its smoothness  $s_q$ , the Hurst parameter is simply  $s_q = H$  as already emphasized in (7)

<sup>40</sup> A Gaussian distribution for increments of log-volatility is expected as already explored by (Torben G Andersen, Tim Bollerslev, Francis X Diebold, and Heiko Ebens, The distribution of realized stock return volatility, 2001).

<sup>41</sup> Price disagreement in the stock market here reflects the buying and selling activity, the larger the disagreement the higher the degree of volatility at that time. More erratic spikes in volatility go hand in hand with higher estimates of  $H$

puts the assumption of long memory by (Comte et al., 1998) further into question.

The last graph (Gatheral et al., 2018) look at in this section is the distribution of increments of log-volatility. In Figure 8 they observe that the increments of log-volatility approximately follow a Gaussian distribution<sup>40</sup>. This feature coincides with the increments a fractional Brownian motion generates. Thus, a model relying on a fBm to drive the evolution of log-volatility can be seen as suitable. This

rolling time window of one year. For shorter durations the estimations for H might display nonsensical, i.e. negative, values. Each point of the blue line is estimating H for all 252 previous (working) days.

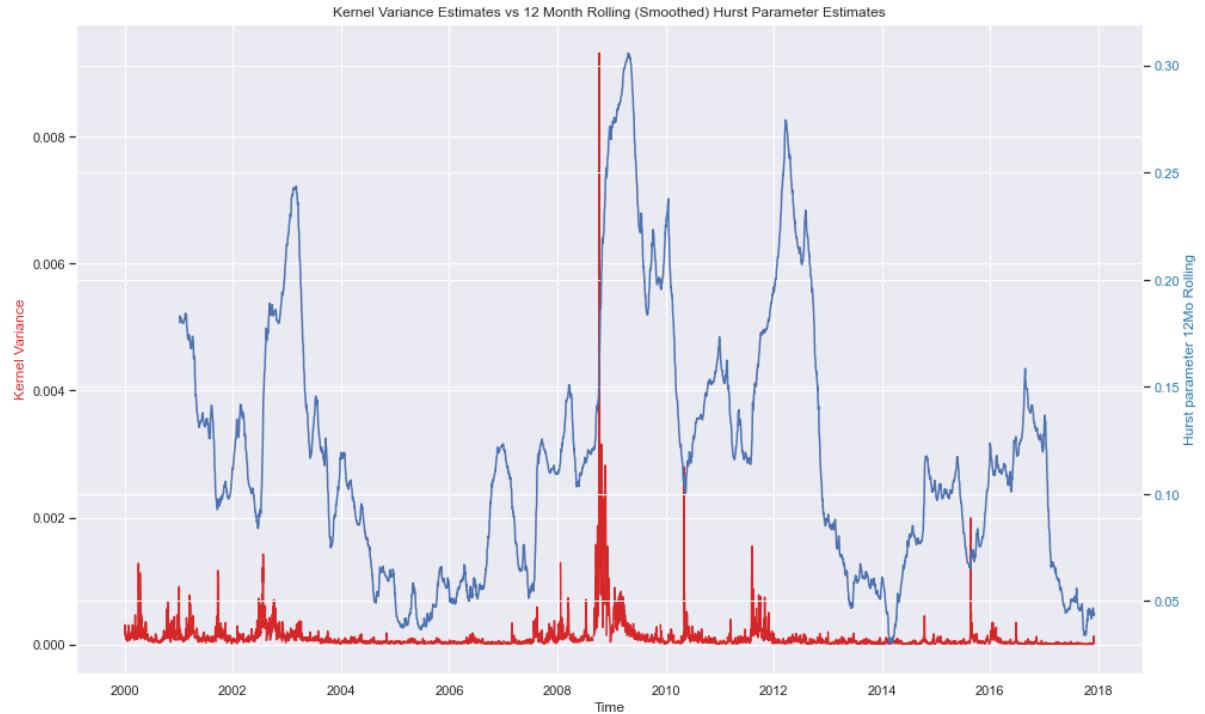


Figure 9: Displays the kernel variance estimates vs the 12-month rolling estimate for the Hurst parameter. Results for SPX data. All data derived from the Oxford-Man Institute of Quantitative Finance

Here one can see H clearly varying within the interval [0.3, 0.03]. Thus, lower volatility environments seem to go hand in hand with lower estimates for H, while a more “active” market prompts a corresponding higher estimation for the Hurst parameter. This observation is in line with results Gatheral showed in a presentation from 2017<sup>42</sup>. Crucially, this may give rise to the notion that estimating implied volatility through multi-year periods for a fixed H may lead to spurious results, as expectations for larger future swings in the stock market should necessitate larger estimates for the Hurst parameter<sup>43</sup>.

## The RFSV Model

When deciding on a model for volatility, many things must be considered. Thus far we have looked at volatility for financial spot prices and that the evolution between two observations (the increments) are approximately normal and may be driven by a fBm with Hurst parameter H. Moreover, we found that the smoothness parameter H for all indices reported by the Oxford-

<sup>42</sup> Namely that high instances of H are correlated with real financial events and that when markets are calm, also H is close to zero. Recorded here: <https://www.youtube.com/watch?v=gW073Tnx7CE>

<sup>43</sup> It could also explain why there exists disagreement in the literature on the magnitude of H.

Man Institute of Quantitative Finance is roughly between 0.2 and 0.08. Additionally, (Gatheral et al., 2018) look at the implied volatility<sup>44</sup>  $\sigma_{BS}$  for option contracts and its behavior through time. This makes sense, as building a model estimating volatility should not only take into account realized (i.e., past) volatility observation, but also future expectations of volatility, which implied volatility tracks to. Combining all option implied volatility observations for all maturities from one observation day for a given symbol, yields the so-called volatility surface, see (Gatheral, J., The Volatility Surface: A Practitioner's Guide, 2006), stretching across all option contracts and all available maturities see Figure 10.

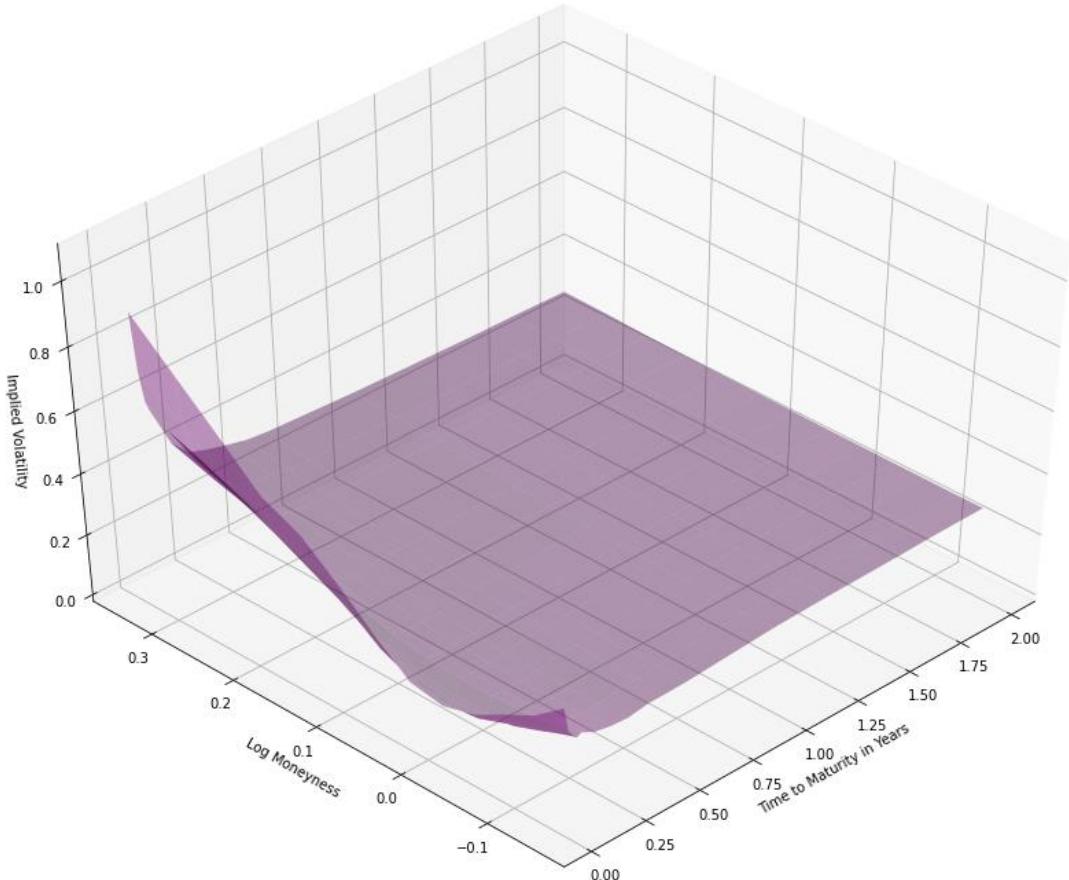


Figure 10: Shows the empirical  $\sigma_{BS}$  implied volatility surface for AAPL on the 25.06.2021. Data gathered at Close from yahoo finance.

Two things are of interest to the authors here:

---

<sup>44</sup> Black Scholes (BS) Implied volatility is a volatility measure used in option pricing. Technically speaking it is the volatility needed as input into the Black & Scholes model to match the observed option price. As options mature in the future, it is by design a forward-looking measure. The next chapter will explain the concept in greater detail

- a) The behavior through time (the so-called “term-structure) of at-the-money (ATM) options contracts, where log-moneyness<sup>45</sup> is equal to zero, and
- b) The volatility surface as a whole and other factors which influence its shape.

The behavior (i.e., the volatility skew) of ATM contracts through time, i.e., their term structure, is defined as

$$\psi(\tau) := \left| \frac{\delta}{\delta k} \sigma_{BS}(k, \tau) \right|_{k=0}, \quad (11)$$

where  $\tau$  is the time until expiration ( $T - t$ ) and  $k$  imposes the log-moneyness equal to zero, i.e., ATM. This term-structure captures the underlying dynamics of observed volatility surfaces well, as Gatheral notes<sup>46</sup>. Moreover, its observed power function can be well approximated by

$$\psi(\tau) \sim \tau^{H-\frac{1}{2}} \sim \frac{1}{\sqrt{\tau}}, \quad (12)$$

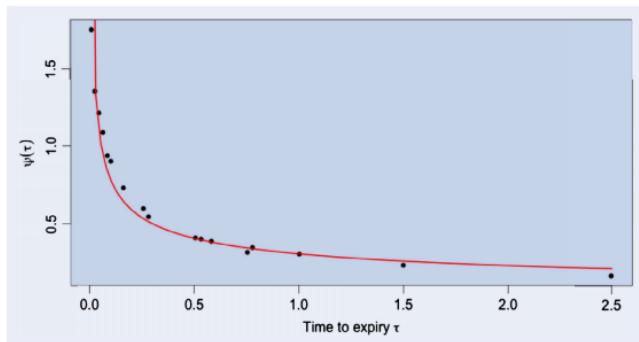


Figure 11: From (Gatheral et al., 2018, p. 3). Black dots: observed ATM volatility skews (11) for SPX as of June 20th, 2013. Red Curve: Fit generated by  $\psi(\tau) \sim A\tau^{-0.4}$

observed term structure has proven futile in the past. Traditional stochastic volatility models, such as SABR or Heston produce term-structures of volatility skew that are flat for small time to expiration and act as summed up decaying exponentials for longer maturities. Also, the study by (Fukasawa, 2011) provides a challenge to the idea that the drastic increase in volatility skew  $\psi(\tau)$  towards the end of life for an options contract necessitates a model that incorporates jump-like behavior (Carr, P. and Wu, L., What type of process underlies options? A simple

which was explored by (Fukasawa, M., Asymptotic analysis for stochastic volatility: Martingale expansion, 2011). For volatility models that rely on a fBm to drive its stochastic changes, Fukasawa shows that (12) generates a fit that matches empirical observations, see Figure 11. Finding a suitable model, which replicates this

<sup>45</sup> Computed as  $\log(S/K)$ , where  $S$  is the spot price of the financial asset and  $K$  reports the strike price of the option contract. Moneyness may also be used in a more informal way to indicate the profitability of an option, meaning how much ITM (in the money i.e., profitable) or OTM (out of the money i.e., not profitable) the option is. ATM or at-the-money option contracts are then on the verge of (un-) profitability, where the spot price is equal to the strike of the option contract.

<sup>46</sup> See his presentation here: <https://www.youtube.com/watch?v=gW073Tnx7CE>

robust test, 2003). This alone already constitutes the motivation to seek a better model of volatility.

While the volatility skew of ATM option contracts is an important feature of the volatility surface to consider, (Gatheral et al., 2018) are also considering the surface as a whole. The authors note that although level, skew and bias may be changing, implied volatility surfaces typically feature a common shape. This fact prompts them to build a volatility model with parameters that do not depend on price nor on time. From these observations the authors start with proposing the simple although non-stationary<sup>47</sup> model

$$\log \sigma_{t+\Delta} - \log \sigma_t = \nu(W_{t+\Delta}^H - W_t^H). \quad (13)$$

Here the log-volatility increments  $\Delta$  are driven by the fractional Brownian motion at different points in time ( $W_{t+\Delta}^H$  and  $W_t^H$ ) at a level, determined by  $\nu^{48}$  in (13), which aligns with our observations, where  $\nu_{est}$  determined the level of our regression (8). As the increments of a fBm process are also Gaussian, as explained previously, this simple model may be a good start. To impose stationarity however and rely on a well-established framework (Gatheral et al., 2018) suggest using a stationary fractional Ornstein-Uhlenbeck (fOU) process<sup>49</sup> of the form (14) with a Riemann-Stieltjes integral, see (Cheridito, et al., 2003)

$$X_t = \nu \int_{-\infty}^t e^{\alpha(t-s)} dW_s^H + m, \quad (14)$$

satisfying the following corollary:

**Corollary 2.0:** *Let  $X_t$  be defined as in (14), then for  $q > 0, t > 0, \Delta > 0$ , and  $\alpha \rightarrow 0$*

$$\mathbb{E}[|X_{t+\Delta}^\alpha - X_t^\alpha|^q] \rightarrow \nu^q K_q \Delta^{qH}, \quad (15)$$

which implies that the observed scaling behavior of the fBm is approx. mimicked by the fOU process  $X_t$ , when  $\alpha$  tends to zero. The fOU process then defines the volatility process

<sup>47</sup> Stationarity is a desirable feature when dealing with time series in general. It signifies that such a series is directionless (without a trend) and oscillates around a constant mean. This is important as trend or explosive behavior makes it difficult to estimate the properties of the series in a concise manner.

<sup>48</sup>  $\nu$  can be seen as a measure for the volatility of volatility as (Gatheral et al, 2018) notes.  $\nu_{est}$  is then the estimation we performed in (9) from regression (8)

<sup>49</sup> The authors note, the choice for the fractional Ornstein–Uhlenbeck process is not mandatory as other options exist, however they found it to be the simplest available option. This freedom may be exploited by other authors to use different processes to incorporate rough volatility.

$$\sigma_t = \exp(X_t), t \in [0, T], \quad (16)$$

in turn driving the price process

$$\frac{dS_t}{S_t} = \mu_t dt + \sigma_t dZ_t, \quad (17)$$

which constitutes the rough fractional stochastic volatility (RFSV) model for  $\nu > 0, \alpha > 0, m \in \mathbb{R}$ , and  $H < \frac{1}{2}$ . With only four parameters  $(\nu, \alpha, H, m)$  the straight-forward model describes the observed volatility process well. Setting  $\alpha = 0$  recovers the simple non-stationary model (13). The model (16) is then a tweak to the FSV used by (Comte et al., 1998). While they forced long memory by choosing  $H > \frac{1}{2}$ , (Gatheral et al., 2018) define  $H < \frac{1}{2}$  and  $\alpha \ll 1/T$ . Applying the logic by (Fukasawa, 2011) to the FSV for  $H > \frac{1}{2}$ ,  $\psi(\tau)$  would be increasing in maturity length  $\tau$ , which is greatly at odds with the empirical behavior observed in Figure 11. Instead, using the more realistic property  $H < \frac{1}{2}$  and “forgoing” the empirical observation of long memory, enables the user of the RFSV model to reproduce the term structure shown in Figure 11. The choice to restrict  $\alpha$  to small values ensures that the process (14) is close to that of a fBm, which is in agreement with the observed distribution of volatility increments (Figure 8).

Using the model to assess the quality of the forecasts it is able to produce, (Gatheral et al., 2018) define the variance forecasting formula as

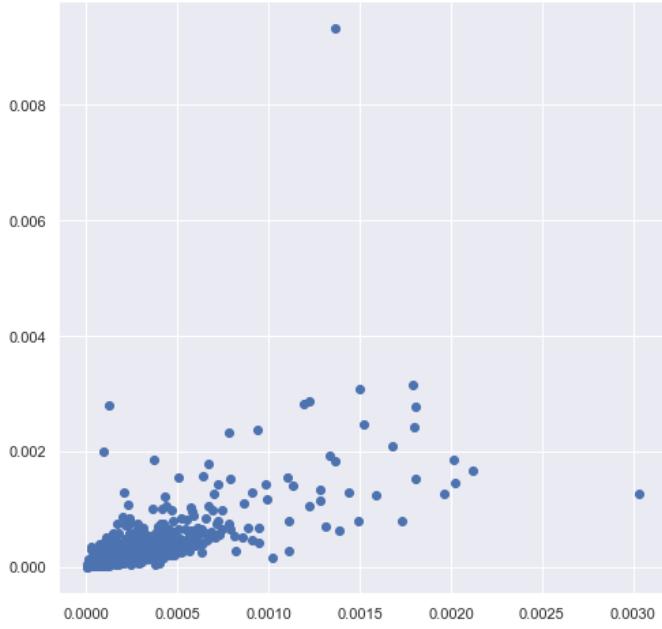
$$\mathbb{E}^{\mathbb{P}}[\log(\sigma_{t+\Delta}^2)|\mathcal{F}_t] = \exp\{\mathbb{E}^{\mathbb{P}}[\log(\sigma_{t+\Delta})|\mathcal{F}_t] + 2c\nu^2\Delta^{2H}\}, \quad (18)$$

with (19), where  $r$  is the smallest real number:

$$\mathbb{E}^{\mathbb{P}}[\log(\sigma_{t+\Delta}^2)|\mathcal{F}_t] = \frac{\cos(H\pi)}{\pi} \int_0^r \frac{\log(\sigma_{t-\Delta u}^2)}{(u+1)u^{H+\frac{1}{2}}} du, \quad (19)$$

and

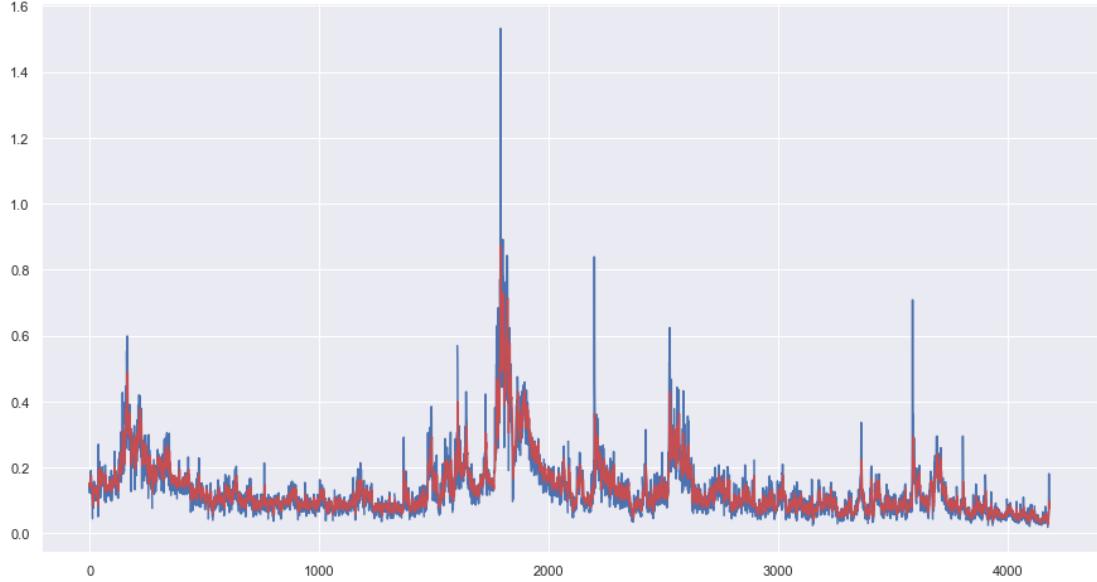
$$c = \frac{\Gamma(3/2 - H)}{\Gamma(H + 1/2)\Gamma(2 - 2H)}. \quad (20)$$



*Figure 12: X-axis: Predicted variances. Y-axis: Observed variances for SPX. Estimates of  $H$  and  $v$  from values reported in Table 11. Data from the Oxford-Man Institute of Quant. Fin.*

Given the kernel estimates for realized variance from the Oxford-Man Institute of Quantitative Finance, Gatheral<sup>50</sup> shows the scatter plot (Figure 12) with actual vs predicted variance for delta=1 days ahead. While some outliers exist, the majority of points seem to cluster at very small variances. In Figure 13 Gatheral superimposes the predicted variances on top of the observed dataset. For sudden large spikes the RFSV seems to underestimate the variance to some

degree, however the moves and roughness of the variance paths seem to be replicated quite well. (Gatheral et al., 2018) also consider other estimation methods and compare their perf-



*Figure 13: The x-axis shows the timescale in days, and the y-axis plots the level of volatility for the observed SPX data (blue), as well as the predicted values (red). Estimates of  $H$  and  $v$  rely on the values reported in Table 11.. All data derived from the Oxford-Man Institute of Quantitative Finance*

ormance by computing the ratio of mean square error (MSE) to approximate variance of log-variance. For comparison they consider the predictive power of the RFSV for the log-

---

<sup>50</sup> See [https://www.tpq.io/p/rough\\_volatility\\_with\\_python.html](https://www.tpq.io/p/rough_volatility_with_python.html)

variance (18) against estimates from both a simple autoregressive (AR(p)) model of order p and from a heterogeneous autoregressive (HAR(p)) model of order p for different values for p. (Gatheral et al., 2018) forecast for all indices in the Oxford-Man Institute of Quantitative Finance dataset for 1,5, and 20 days ahead. The authors observe that the simple RFSV model beats both (more computationally expensive) models in all but one estimate, where the HAR(3) model manages to outperform the RFSV just slightly.

### Validating the Results by Daily OHLC Data

As high-quality estimates of realized variance for other assets are difficult to come by for the common retail investor, an interesting research question is how well the results can be replicated by noisy and uncleaned daily OHLC data, which is freely available from sources such as yahoo-finance<sup>51</sup>. Therefore, firstly we compare two different variance estimators for their robustness to estimate the Hurst parameter H for SPX Close data. Subsequently, after selecting a suitable variance estimator we estimate the roughness  $H$  and level  $\nu$  for a select range of stock symbols from Table 1 and discuss the findings.

For the same time range as used in (Gatheral et al., 2018), we compare two estimators of volatility, or mutatis mutandis (m.m.), the variance for SPX data against the realized kernel estimates reported by the authors with the goal of identifying a variance estimator that is close to the kernel estimates reported. The first estimator (21) is the standard volatility estimate which estimates the volatility of daily realized log-returns. The second, much more sophisticated estimator (22) is an extension by Yang and Zhang of the highly regarded Garman-Klass volatility estimator, see (*Marcin Faldziński, Magdalena Osińska, Volatility Estimators in Econometric Analysis of Risk Transfer on Capital Markets, 2016*) or (*Dikshita Wadhawan, Harjit Singh, Estimating and Forecasting Volatility Using Arima Model: A Study on NSE, India, 2019*).

$$\sigma_{standard} = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2}, \quad (21)$$

with  $x_i = \log\left(\frac{c_i}{c_{i-1}}\right)$ .

---

<sup>51</sup> See <https://finance.yahoo.com>

$$\sigma_{GKYZ} = \sqrt{\frac{1}{N} \sum_{i=1}^N \left[ \log\left(\frac{o_i}{c_{i-1}}\right) \right]^2 + \frac{1}{2} \left[ \log\left(\frac{h_i}{l_i}\right) \right]^2 - (2 \log(2)) \left[ \log\left(\frac{c_i}{o_i}\right) \right]^2}, \quad (22)$$

where  $x$  equals the daily log-realized return for the respective asset, and  $o_i, h_i, l_i, c_i$  correspond to the open, high, low and close price for day  $i$ . Yang and Zhang's contribution here amount to including overnight volatility between yesterday's close and today's open, which is captured by the first term  $\left[ \log\left(\frac{o_i}{c_{i-1}}\right) \right]^2$ .

In Figure 14 the variance data from the kernel estimates can be easily compared to what the estimators produce which rely on daily OHLC data only. Looking at the differences between kernel and GKYZ estimates we can see the lower spikes in variance are more pronounced in the GKYZ estimates, while for larger spikes the kernel estimates provide higher variance estimates. Overall, while differences are clearly visible, they are not as starkly pronounced as for kernel vs log return estimates. For the variance estimates that rely on daily log returns, the variance estimates are far noisier at low levels with more erratic spikes for volatility events. Therefore, judging preliminary from the observed variance data only, we expect the GKYZ estimator to perform more closely to the kernel estimates than the variance data coming from simple log-returns. This is also evident from comparing the MSE between the two OHLC estimates against the better kernel estimate: The GKYZ estimate achieved a MSE of 7.354e-08, comparing to the more than twice as large MSE of the log return estimates with 2.235e-07

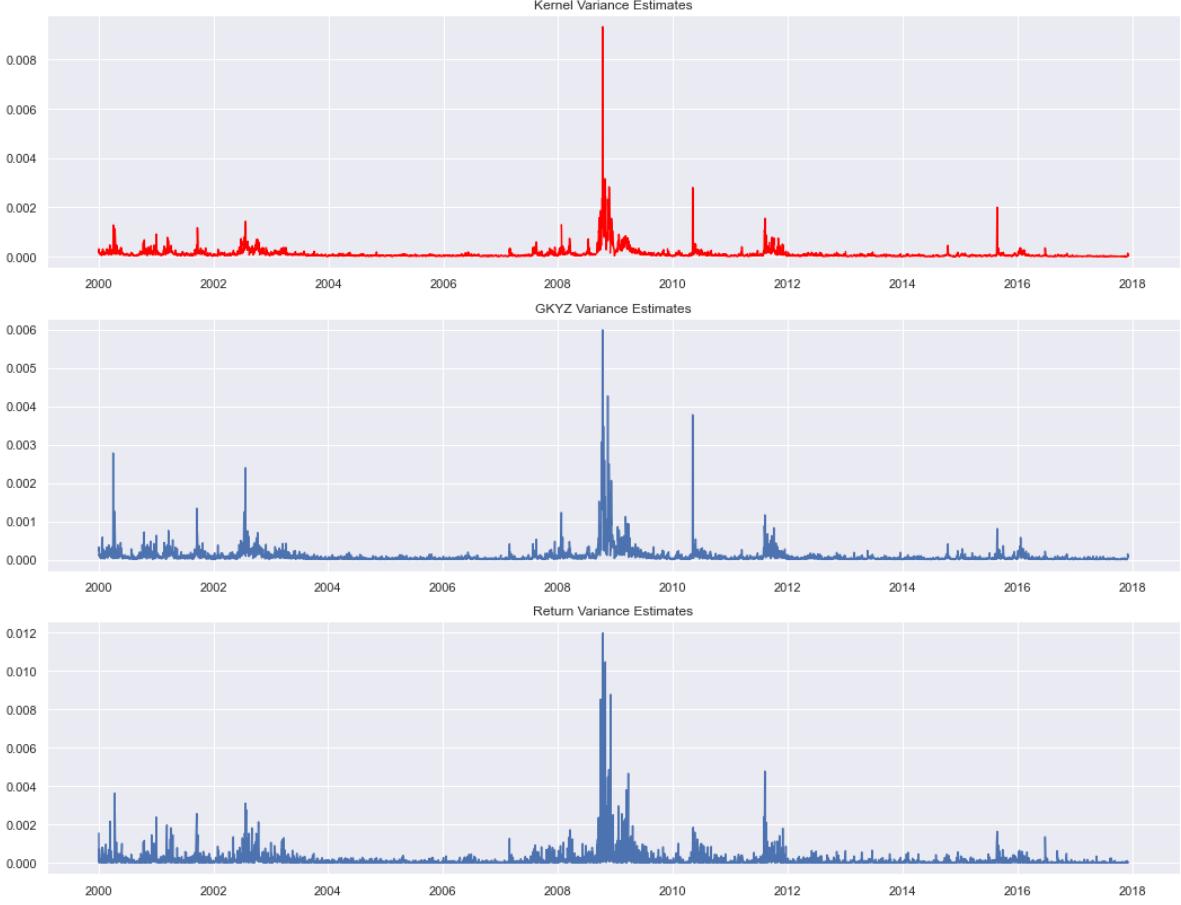


Figure 14: X-axis: timescale in years. Y-axis: level of variances. Plots in order: Kernel Variance Estimates used by (Gatheral et al., 2018), GKYZ and standard return Variance estimators. SPX Data derived from the Oxford-Man Institute of Quant. Fin. and from close data from yahoo finance.

Next, we want to understand the scaling behavior of  $\log(m(q, \Delta))$  in  $\log(q)$ . Recall that for the kernel estimates we have observed a linear behavior in  $\log(q)$ . While the regression in Figure 15 is just an intermediate step, we can already make preliminary observations on the difference of the two estimators in question. The GKYZ estimates seem to mimic the kernel estimates decently. The differences are pronounced for  $q=3$ , where the lines intersect for the GKYZ estimate and for the general level of  $\log(m(q, \Delta))$  which is between -0.5 and -1.75, whereas the kernel estimates span the range of -0.5 to -3. For the log return variance estimates the differences are rather large; the level of  $\log(m(q, \Delta))$  tends into positive territory, and also the order of the  $q$  seems reversed: While for the upper two estimators increasing  $q$  reduces the level of  $\log(m(q, \Delta))$ , the opposite is the case for the third estimator in question. Lastly, we can also observe that the slope for the log-return variance estimate is far lower than what we observed previously, corresponding to what we observed in Figure 14 as noisy behavior in low volatility environments. As such, the log-return variance seems to not capture the scaling properties well.

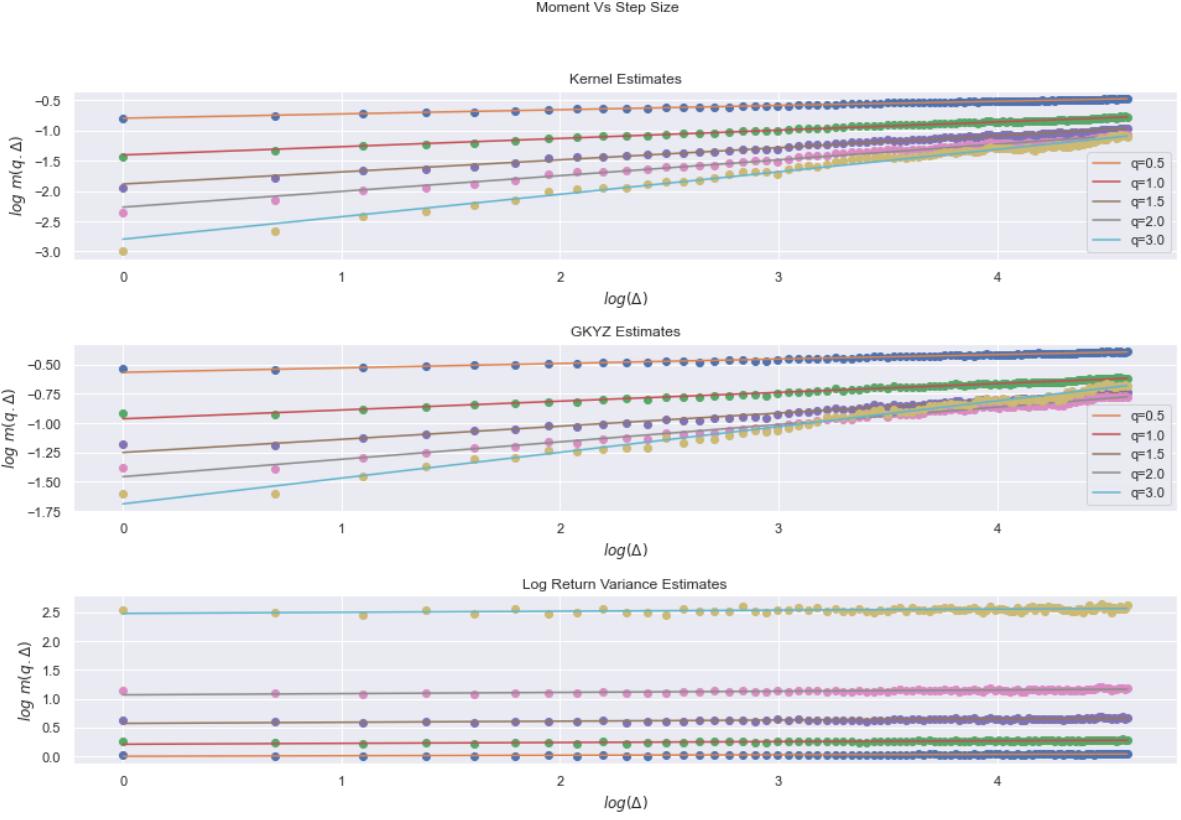


Figure 15: X-axis:  $\log$  of the step-size  $\Delta$ . Y-axis: level of  $\log m(q, \Delta)$ . Plots in order: Kernel Variance Estimates used by (Gatheral et al., 2018), GKYZ and standard return Variance. SPX Data derived from the Oxford-Man Institute of Quant. Fin.# and from close data from yahoo finance.

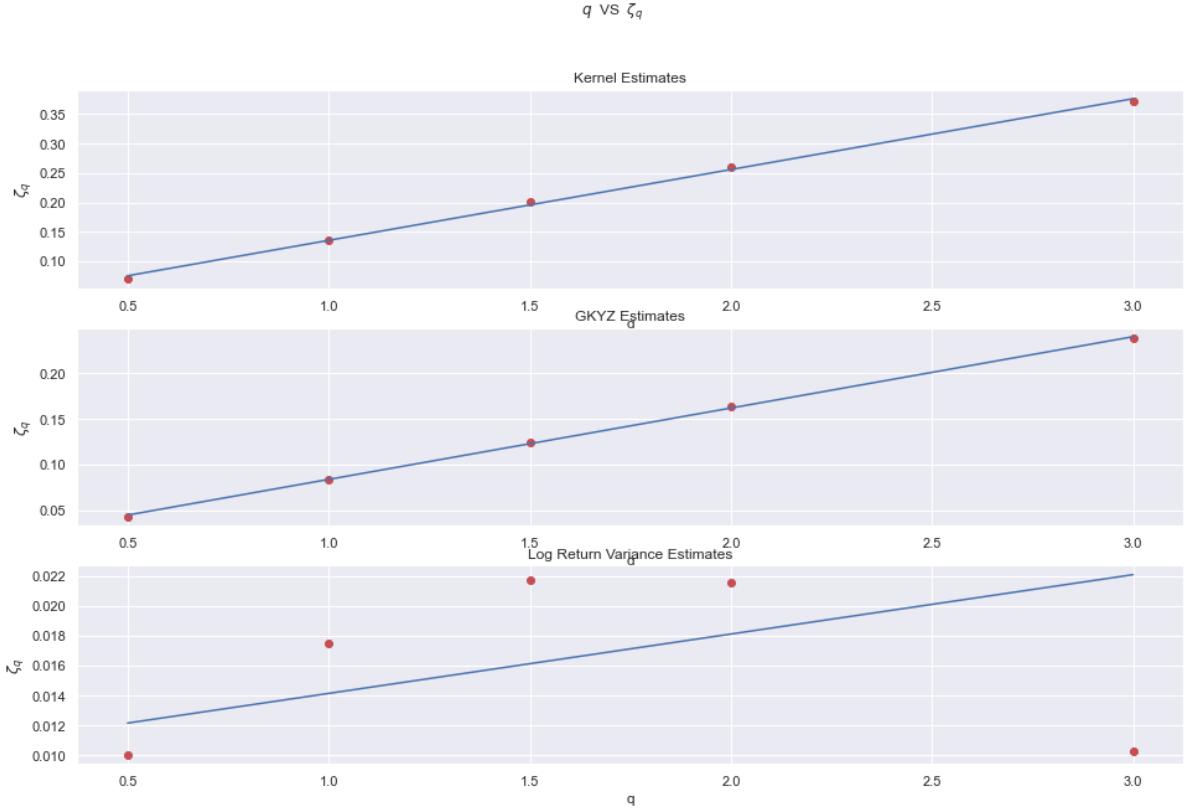


Figure 16: X-axis: Order of  $q$ . Y-axis: level of  $\zeta_q$  for observed SPX data. Data derived from the Oxford-Man Institute of Quantitative Finance and from close data from yahoo finance.

Finally, turning to Figure 16 lets us extract the slope  $s_q$ , which we defined as the Hurst parameter  $H$ . The individual graphs plot the regression of  $\zeta_q$  as a function of  $q$  and the individual  $\zeta_q$  estimates in red dots, for kernel variance, GKYZ and standard return variance estimates. We can see similar scaling behavior in the first two plots for kernel and GKYZ estimates, while for the log-return variance estimates the scaling does not follow any clear direction. The data in Table 2, is the final “nail in the coffin” for the Log Return estimate. While GKYZ estimates the volatility characteristic to exhibit far greater roughness than the kernel estimate<sup>52</sup>, the log-return estimate is magnitudes<sup>53</sup> farther off from the kernel estimate, indicating a very rough volatility process, which corresponds to the greater jaggedness in low volatility environments observed in Figure 14.

Estimation Technique	Kernel Estimate	GKYZ Estimate	Log Return Estimate
H - Estimates	0.120321	0.078073	0.003973

Table 2: Hurst Parameter estimates for SPX following different variance estimation methods (kernel, GKYZ, log-return variance). Data from the Oxford-Man Institute of Quant. Fin. and from close data from yahoo finance

Lastly in Appendix B3, we highlight the distribution of increments for the estimators in question. Again, the differences between kernel and GKYZ estimates seem minor and distributions are approx. Gaussian, while the distribution of increments stemming from the log-return variance estimates approx. follow a t-Student distribution. See our preliminary statistical analysis by distribution fit. This is clearly insufficient as the choice of modeling volatility increments with a fBm relies on the increments being approximately Gaussian distributed.

With these insights we can make the following conclusion: The log-return variance estimates are clearly insufficient to capture the roughness of volatility, as for every benchmark we presented the estimates deviated strongly from what was observed for the kernel variance estimates. For the GKYZ estimate, the results are not as clear. While using daily OHLC it is clearly the better estimator of the two we looked at, it underestimates the Hurst parameter by a significant amount compared to the kernel estimates. However, in its defense it clearly has less data to work with<sup>54</sup>. The data used by Oxford-Man Institute of Quantitative Finance rely on a high-frequency estimation method and come from Refinitiv (formerly Thomson Reuters) one of the most reputable data vendors. Nevertheless, the GKYZ estimate managed to exhibit

---

<sup>52</sup> Note that the estimate for the kernel variance, presented in Table 2, is slightly lower than  $H_{est}$  reported above. This is because in the original code by Gatheral the last point  $q=3$  was excluded from the regression, while it was considered here.

<sup>53</sup> By a factor of 30x to be precise.

<sup>54</sup> Using daily OHLC data results in only 4 datapoints per day, while high-frequency data amounts to 84 tick observations (assuming 5min tick intervals) or more.

similar behavior as the more advanced estimation method across all benchmarks. While we understand the limitations of this estimator, the conformity in behavior and the fact that (Gatheral et al., 2018) note the kernel estimates to overstate the Hurst parameter make us confident in using the GKYZ estimate to estimate the roughness of daily realized volatility, at least for a first approach. Having the GKYZ variance estimator established as a reliable estimator for the Hurst parameter, we follow the

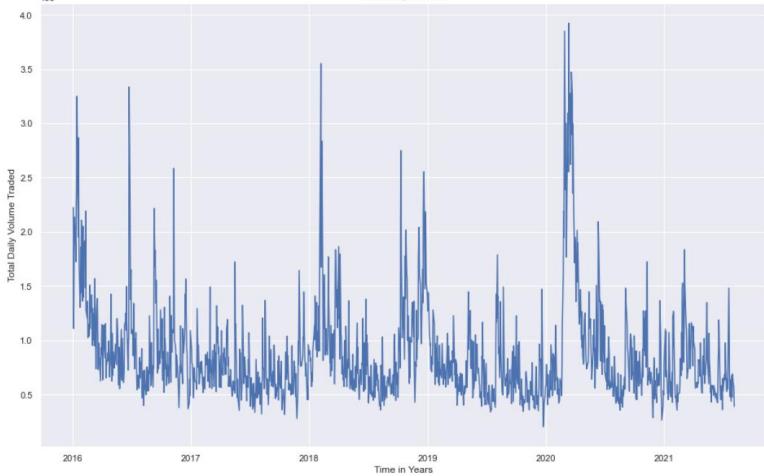


Figure 17: Highlights the spike in trading volume for SPX at around the COVID incident. Data derived from yahoo finance close data.

methodology set out in the beginning to estimate  $H_{est}$  and  $\nu_{est}$  over the course of the past 3 years<sup>55</sup>. We believe this to be a good range, as the trading activity in the stock market increased over the course of the COVID pandemic (see Figure 17), compared to the years prior to that. However, to be prudent we decide to split the dataset, such that (s.t.) approximately half of the data stems from the period before, and half from the period since the COVID incident. This is to take into account that trading activity might come back down again, as we enter a “return from home” phase globally, but also to mitigate (smooth out) the spike in trading activity at around the COVID incident. In Table 3, the estimates for the Hurst parameter,  $\nu$  and the correlation between log returns and observed volatility  $\rho$  are summarized<sup>56</sup>, where  $\rho$  is defined:

$$\rho = \frac{\text{cov}(x_i, \sigma_i^{GKYZ})}{\sigma_{x_i} \sigma_{\sigma_i^{GKYZ}}} \quad (23)$$

Here  $x_i$  denotes again the log-return and  $\sigma_i^{GKYZ}$  the corresponding volatility estimate for day  $i$ . We see “Mega-Cap” stocks such as Apple (AAPL), AT&T (T) and Microsoft (MSFT) to exhibit among the highest levels of  $H$ , with only the index SPX to even show a greater value for the Hurst parameter. As a larger Hurst parameter corresponds to a smoother volatility

<sup>55</sup> From 2018-08-07 until 2021-08-05. This is to take the varying nature of the Hurst parameter into account.

<sup>56</sup> The apparent universality of roughness of volatility for financial assets in Table 11 and Table 3 may come from order flow or market-microstructure as explored by (El Euch, Fukasawa, and Rosenbaum, The microstructural foundations of leverage effect and rough volatility, 2016). As buy orders stimulate more buys and vice-versa for sell orders, order flow is highly autocorrelated. Based on this they develop a generalization of the Heston model, which looks highly similar to the RFSV model for the evolution of the volatility process.

process, one might say this is to be expected. Typically, larger market capitalization (market cap) stocks trade with less volatility as it becomes more expensive to trade large amounts of stock (relative to the market capitalization) in and out again. As such also the volatility increments are smaller between two observations and with this the process can be seen as “smoother”, compared to “rougher” volatility environments. For smaller and more retail-investor oriented stocks, such as Tesla (TSLA), Gamestop (GME), and Nio (NIO) we see lower levels of  $H$ , corresponding to rougher volatility. As electric mobility and vehicles (EVs) are very popular, and the reddit-craze<sup>57</sup> around so called “meme-stocks” (e.g., GME and NIO) took off in the beginning of the year 2021, the higher estimates for roughness are easily explained. Also  $\nu_{est}$ , the volatility of volatility, shows figures that align with this theory. For more volatile stocks, also  $\nu_{est}$  displays higher values corresponding to a higher level of “volatility of volatility”. Lastly, we can observe the empirical anti-correlation between log returns and log volatility for *most* stocks, also reported by (Gatheral et al., 2018).

<b>Symbol</b>	<b><math>H_{est}</math></b>	<b><math>\nu_{est}</math></b>	<b><math>\rho_{est}</math></b>
AAPL	0.110573	0.577031	-0.127195
GME	0.092626	0.626438	0.128399
MSFT	0.146319	0.520883	-0.091008
NIO	0.039228	0.637915	0.153142
SPX	0.170284	0.568015	-0.120013
T	0.110647	0.568868	-0.089833
TSLA	0.089998	0.607813	-0.009374

Table 3:  $H$ ,  $\nu$  and  $\rho$  estimates for 7 stocks following methodology by (Gatheral et al., 2018) using the GKYZ variance estimates for the period of the past 3 years, snapshot at 2021-08-23. Using yahoo finance close data.

Curiously however we observe positive correlation for stocks such as GME and NIO. This may be explained by the exuberant volatility observed for these stocks as part of the reddit driven option speculations mentioned above and the resurgent interest of retail trading with the beginning of the COVID incident. In fact, if we exclude the last two years 2020 and 2021 from the analysis, these stocks exhibit negative  $\rho_{est}$ , as expected. This may signal market dislocations<sup>58</sup> and that those volatility related “facts” (such as anticorrelation) do not hold in every case. They should be subject to high levels of scrutiny, especially when dealing with stocks that are traded in high volume with similar high levels of volatility. This leads us to the conclusion for this chapter that while daily OHLC data may be used to estimate the parameters  $H$ ,  $\nu$ , and  $\rho$ , they do change, sometimes quite significantly so. Hence, it may be wrong to assume they are fixed through time.

<sup>57</sup> The reddit forum “wallstreetbets”(<https://www.reddit.com/r/wallstreetbets/>) launched a wild speculation attempt at prompting a short- and gamma-squeeze involving stocks such as GME and AMC. The increased trading activity prompted increasingly high volatility and options volume.

<sup>58</sup> See for example (Pasquariello, Financial market dislocations, 2014)

# The rBergomi Model

As (Gatheral et al., 2018) built their RFSV model with both implied volatility and the dynamics of the volatility surface in mind, it is only natural to use their framework to build an option pricing model under the rough volatility assumption. This chapter is dedicated to the first such model, the rough Bergomi (rBergomi) model, developed by Bayer, Friz, and Gatheral in their paper “Pricing under rough volatility”, Quantitative Finance, 2015.

Today, the most used option pricing model remains the Black Scholes model. Introduced over 45 years ago, its popularity stems from the fact that it not only presents closed form solutions to pricing European Put or Call options, but also to “The Greeks”. These are first and second order derivatives which help understand how option prices change ceteris paribus (c.p.) for one input parameter at a time.

## The Black Scholes Model and Implied Volatility Dynamics

Before the pricing step, Black Scholes first needed to translate real world (observed) behavior into a tangible mathematical concept, i.e., how prices evolve through time. With this concept Black and Scholes were then able to build their option pricing model. Their theoretical framework on how a given spot price  $S_t$  evolves is given by the model

$$dS_t = \mu S_t dt + \sigma S_t W_t , \quad (24)$$

where  $\mu$  and  $\sigma$  are constants and  $W_t$  is a geometric Brownian motion. While the Black-Scholes framework presents a straightforward computation, it comes at the expense of presuming constant volatility of log returns  $\sigma$  for the underlying asset. This assumption would imply a flat volatility surface, which is clearly at odds with empirical data, see e.g., Figure 10 where volatility smiles and skews are clearly observable. Also recall the term-structure of ATM volatility skew reported by (Fukasawa, 2011) in (12) and visualized by Figure 11. Based on (24) and imposing an absence of arbitrage condition, the authors were able to come up with a closed form solution, the so called standard Black Scholes formulae for the call/put option price, given by

$$C(S, \tau) = S\Phi(d_1) - Ke^{-r\tau}\Phi(d_2) , \quad (25)$$

$$P(S, \tau) = Ke^{-r\tau}\Phi(-d_2) - S\Phi(-d_1) , \quad (26)$$

with

$$d_1 = \frac{\log\left(\frac{S}{K}\right) + (r + \frac{\sigma_{BS}^2}{2})\tau}{\sigma_{BS}\sqrt{\tau}}, \quad (27)$$

$$d_2 = d_1 - \sigma_{BS}\sqrt{\tau}.$$

Where C and P are denoting the Call and Put option premium (the option price) respectively. Both are dependent on current spot price  $S$  and time to maturity  $\tau = T - t$ , where T denotes the time at option expiration and  $t$  the time today.  $\Phi(\cdot)$  is the standard normal cumulative distribution function (CDF) given by  $\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{z^2}{2}} dz$ . The CDF here is simply used to define the probability of moneyness for a given option. The strike price K (the threshold of moneyness) is discounted by the exponential  $e^{-r\tau}$ , where  $r$  is the appropriate discount rate<sup>59</sup>. Lastly,  $\sigma_{BS}$  defines the so-called implied volatility which drives both (25) and (26) through (27). It constitutes the main focus of any option pricing approach. As all other variables ( $C, P, S, \tau, K, r$ ) are observable and known, the implied volatility is the only value which needs to be computed. In the following we will restrict our attention to option Call prices since the put-call parity (28) given by

$$C + Ke^{-r\tau} = P + S, \quad (28)$$

allows us to translate call into put prices and vice versa easily. With this we want to now focus a little more on the concept of implied volatility: Given an *observed* Call-option price in the market  $C^*$ , the term “implied” originates from the fact that it is the single *unknown* input needed in the formula (25) to match the observed call premium  $C^*$ . Thus, the *observed* call option premium  $C^*$  *implies* a certain value for  $\sigma_{BS}$ . In technical terms, IV is defined as<sup>60</sup>

$$\sigma_{BS} = \text{argmin}\{C^*(S, \tau) - C(S, \tau)\}. \quad (29)$$

Implied volatility is expressed as standard deviation in percent normed over a period of one year. See Figure 18, for an example of implied volatility for AAPL with 1.95 years to maturity, observed on the 25.06.2021<sup>61</sup>. Notice how the volatility smile is rather a smirk (as it is not

<sup>59</sup> We use the effective federal funds rate, published by the Federal Reserve Bank of St. Louis (FRED), available here: <https://fred.stlouisfed.org/series/DFF>

<sup>60</sup> Practically, a minimizer is used to find the value for  $\sigma_{BS}$ . Popular choices are the brent-q minimization, or a root finding approach such as the so-called Newton-Method.

<sup>61</sup> Also compare to the option chain in Appendix A1 and the surface shown in Figure 10, reported for the same asset on the same day. The smile (smirk) is really just a cross-section of the whole surface along the y-axis (Log-moneyness)

symmetric), with higher volatility for in-the-money (ITM) options and lower implied volatility for out of the money (OTM) options. Also, we observe what is known as implied volatility bias: The lowest point of implied volatility does not correspond with at-the-moneyness (i.e., strike price equal to the spot price of 133.11 as observed on the same day), but it is in fact shifted to the right. These observations relate to demand and supply dynamics, future expectation by market participants, or their general risk aversion. Greater supply reduces the price of an option and thus ceteris paribus the corresponding implied volatility. This may be due to large financial institutions writing call options<sup>62</sup> to hedge their long positions or market

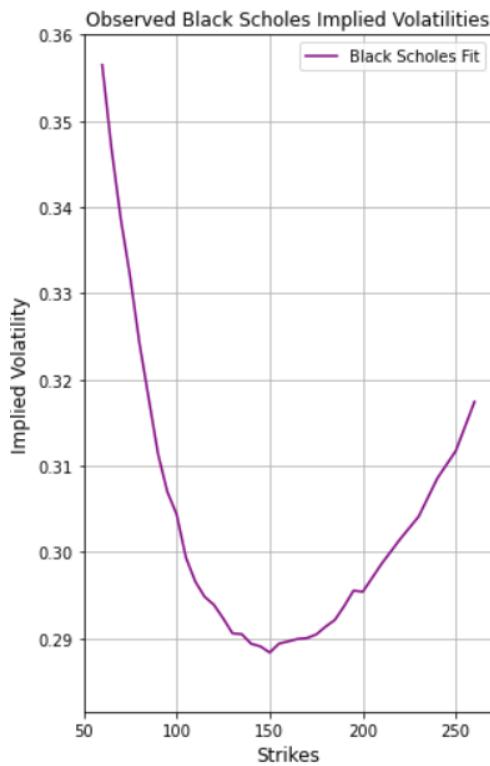


Figure 18: X-axis: Strike Price. Y-axis: IV. Smile for AAPL with spot=\$133.11 and 1.95 years to expiration on the 25.06.2021. Data from yahoo finance at close.

making activity. It may also be explained by the expectation that the spot price will go down, necessitating a lower price and, m.m., a lower value for implied volatility. These smile-characteristics are an intrinsic and *unique* feature of each option chain. Now under the hypothesis that a theoretical model exists that is able to extract these intrinsic smile-features, we could replicate the smile at any point, and thus be able to find the implied volatility (and m.m. the respective option price) for *any* option contract, whether in existence or not. The smirk (i.e., non-symmetric smile) is simply explained by risk aversion of market participants. People are willing to "overpay" for ITM options (left hand side of at-the-moneyness). Thus, the market is giving comparatively less volatility to the upside than to the downside. This may be an indication that downside-

risk protection is seen as more valuable than speculating for hypothetical upside-potential in the market.

Implied volatility is really a measure of future volatility<sup>63</sup>, in other words, implied volatility uses the observed option premium to calculate what people are saying about the expected future volatility of the option's underlying stock. While IV is affected by the general state the market

<sup>62</sup> Writing means selling of call options which given call/put parity is equivalent to buying puts.

<sup>63</sup> However, evidence exists that implied volatility may be a biased estimator for future volatility, e.g.

(Fleming, *The quality of market volatility forecasts implied by S&P 100 index option prices*, 1998) and (Ruas, Curto, Nunes, *The Implied Volatility Bias: A No-Arbitrage Approach for Short-Dated Options*, 2012).

finds itself in (market regime), one of its main drivers constitute supply/demand mechanics<sup>64</sup>. It is straight forward to see that supply and demand directly affects option volume and hence IV for traded option contracts; similar mechanics can be observed in the spot market: As spot-prices rise in demand ceteris paribus, also IVs increase, in turn leading to higher option-premia, as all option contracts increased in moneyness and payoff potential<sup>65</sup>. With implied volatility, volume (i.e., liquidity) is most crucial. Since liquidity is generally the greatest for ATM options, these are typically used to calculate the forward-looking implied volatility on an underlying. Time to expiration is another major factor influencing IV. As observed from the term-structure of ATM implied volatility in Figure 11, IVs typically increase disproportionately the closer an option is to expiration. As the so-called Time Value goes to zero, option volume and thus  $\sigma_{BS}$  tends to increase, leading to dynamics explained above. The total value of an option can then be split into two factors: The Intrinsic Value and the Time Value, where

$$\text{Total Value} = \text{Intrinsic Value} + \text{Time Value}.$$

The Intrinsic Value constitutes the discrepancy between the underlying spot price  $S$  and the strike price  $K$ . However, it cannot be negative, as for  $K > S$  the option is never exercised. The Time Value can be thought of time the option contract has left to become (more) profitable or ITM until option expiration. As such the total value of an option contract is decreasing by the square root of time. Thus for a *fixed* Total Value, the intrinsic value then needs to increase disproportionately to make up for lost Time Value. An option premium is then “fair”, if it assumes the total value an option has at the current time of observation<sup>66</sup>. Ceteris paribus, the Call option premium increases in  $\sigma_{BS}$ ,  $S$ ,  $\tau$ , and  $r$  and decreases in  $K$ .

While in practice the Black-Scholes model can be used for liquid traded options that are “marked to market”, so called Exotic-Options require a pricing framework that is able to price European options across (and in-between) strikes and maturities, whether liquid or not (*Benjamin M. Stemper, Rough volatility models: Monte Carlo, Asymptotics and Deep Calibration, 2019*). So-called diffusion Models that have been introduced since<sup>67</sup>, have tried to alleviate some of the restrictions of the Black Scholes model, however fail to capture the volatility skew (12) adequately, as they assume constant behavior as  $\tau \rightarrow 0$  (*Gatheral, The*

<sup>64</sup> See e.g., <https://corporatefinanceinstitute.com/resources/knowledge/trading-investing/implied-volatility-iv/>

<sup>65</sup> The reverse is true for when institutions “write” more options, thereby increasing supply c.p.

<sup>66</sup> If an option premium were traded above/below this fair value, the arbitrage-free assumption would be violated creating “free” profit opportunities in the market.

<sup>67</sup> For example, SABR (Hagan, Kumar, Lesniewski, Woodward, Managing smile risk, 2002), Heston (Heston, A closed-form solution for options with stochastic volatility with applications to bond and currency options, 1993), and Hull & White (Hull, White, One-factor interest-rate models and the valuation of interest-rate derivative securities, 1993)

*volatility surface: a practitioner’s guide, 2011*). The newly introduced concept of rough volatility represents a move away from these approaches and instead models the driver of volatility by a fBm with Hurst parameter H. As explored in the previous chapter “Rough Volatility”, this modeling choice has indeed been empirically validated by (*Gatheral et al., 2018*) and (*Bennedsen, et al., 2021*). More crucially for this chapter, rough volatility enables an option pricing approach that is able to mimic the term structure of ATM volatility skew (12) explored by (*Fukasawa, 2011*). The idea here is to find a model that is able to be calibrated to the observed implied volatility, matching it most closely in every observed and liquid traded contract. The parameters the calibrated model gives us back<sup>68</sup> can then be used to estimate the implied volatility for contracts that cannot be observed due to illiquidity or that do not even exist. Hence with such a model we are able to find prices for liquid and illiquid options alike at every point in time. One example of such an option pricing model under rough volatility is the rBergomi model by (*Bayer, Friz, Gatheral, Pricing under rough volatility, 2015*). The ability of the rBergomi model to price European options across all maturities and strikes renders it relevant for academia and professionals alike.

### The Bergomi Model

The foundation for the work by (*Bayer et al., 2015*) is drawn from two papers (co-) written by Lorenzo Bergomi, namely: (*Bergomi, Smile dynamics II., 2005*) and (*Bergomi, and Guyon, Stochastic volatility’s orderly smiles, 2012*). Bergomi and Guyon’s model is based on the idea of modeling the price evolution based on the observable forward variance curve  $\xi_t(u)$ <sup>69</sup>:

$$\frac{dS_t}{S_t} = \sqrt{\xi_t(t)} dZ_t , \quad (30)$$

with

$$d\xi_t(u) = \varepsilon\lambda(t, u, \xi_t) dW_t , \quad (31)$$

where  $Z_t$  and  $W_t$  are (anti-) correlated Brownian motions, driving the asset price and forward variance evolution respectively, see (*Bergomi, Guyon, 2012*) for details. They devised an

<sup>68</sup> The idea here is that the parameters correspond to unique features of the observed implied volatility for that option chain (or IV surface) in question. Applying these unique features again to any (hypothetical) option contract should give us back the respective fair price as of time of observation.

<sup>69</sup> The forward variance curve is the derivative of the variance swap curve with respect to time to maturity. It is the risk-neutral expectation of the instantaneous variance at some future date  $u$ , given the information we have today at time  $t$

asymptotic expansion of the implied volatility surface, s.t. the implied volatility  $\sigma_{BS}$  can be approximated for any value of log-strike  $k = \log K/S_0$  and expiration  $T$  by three factors, given by

$$\sigma_{BS}(k, T) = \widehat{\sigma_T} + \varepsilon \sqrt{\frac{w}{T}} \frac{1}{2w^2} C^{x\xi} k + O(\varepsilon^2). \quad (32)$$

Here,  $\sqrt{\frac{w}{T}} \frac{1}{2w^2} C^{x\xi} k$  translates to ATM skew<sup>70</sup>, where  $w$ <sup>71</sup> is total variance to expiration  $w = \int_0^T \xi_0(s) ds$ , and  $C^{x\xi}$  defines the autocorrelation of the price with the forward variance process.  $\varepsilon$  is simply a dimensionless expansion parameter. The Bergomi-Guyon expansion thus gives a one-to-one mapping between ATM level, skew and curvature and model dynamics written in forward variance curve form<sup>72</sup>.

For such dynamics, (Bergomi, 2005) proposed a model for stochastic changes in volatility, considering the forward variance curve:

$$\xi_t(u) = \xi_0(u) \mathcal{E} \left( \sum_{i=1}^n \eta_i \int_0^t e^{-\kappa_i(u-s)} dW_s^i \right), \quad u > t, \quad (33)$$

where  $\mathcal{E}$  is the stochastic exponential<sup>73</sup>. A neat feature here is that  $\xi_t(u)$  is a martingale<sup>74</sup> in  $t$ . Also take note that  $u > t$ , s.t.  $u$  indicates the future assumption corresponding to information available at  $t$  today. "s" is then simply one point in time throughout the time-interval  $[0, t]$ . With the spot dynamics, given by (24) and the forward variance curve (33), the correlation dynamics  $C^{x\xi}$  are then easily computable. All this together yields us an approximation of the implied volatility, which again is our main focus here.

---

<sup>70</sup> Means the symmetry of the smile around its lowest point. See <https://www.imperial.ac.uk/media/imperial-college/research-centres-and-groups/cfm-imperial-institute-of-quantitative-finance/events/distinguished-lectures/Gatheral-1st-Lecture.pdf>

<sup>71</sup> Corresponds to the variance-swap-curve. Indeed as (Bayer, et al., 2015) show, the variance swap curve can be estimated by the RFSV, as we show in Appendix C1.

<sup>72</sup> See <https://www.nsd.pku.edu.cn/attachments/89b43d3ae01449ed8ff6245feec7f4ba.pdf>

<sup>73</sup> The stochastic exponential is defined as  $\mathcal{E}(X)_t = \exp(X_t - X_0 - \frac{1}{2}[X]_t)$

<sup>74</sup> A martingale is defined as a stochastic process generating a sequence of random variables, for which the conditional expectation of the subsequent data-point is equal to the most recent data-point, no matter all previous data-points.  $\mathbb{E}[X_{t+1}|\mathcal{F}_t] = \mathbb{E}[X_t|\mathcal{F}_{t-1}] + \mathbb{E}[X_{t+1}|\mathcal{F}_t] = X_t + \mathbb{E}[X_{t+1}] = X_t$ .

Taking Bergomi's own model, the entire forward variance curve can be determined by  $n$  factors (Bergomi, 2005). For  $n = 1$ , we have:

$$\xi_t(u) = \xi_0(u) \exp \left( \eta e^{-\kappa(u-t)} Y_t - \frac{1}{2} \eta^2 e^{-2\kappa(u-t)} \mathbb{E}[Y_t^2] \right). \quad (34)$$

$\eta$  corresponds to the volatility of volatility,  $\kappa$  is a mean-reversion factor, one for each given factor, and the process  $Y_t$  is defined by  $dY_t = -\kappa Y_t dt + dW_t$ . (Bayer et al., 2015) report that to achieve a decent fit however, at least two factors are required, which equates to eight parameters needed to calibrate to option prices. At this point the model is already over-parameterized, as the authors report. Nevertheless, it manages to roughly achieve an ATM term structure of the form  $1/\sqrt{T}$ , close to the empirically observed one (12):  $\psi(\tau) \sim \tau^{-\alpha}$  with  $\alpha$  a constant.

### Pricing under the physical probability measure

In order to improve the fit, (Bayer et al, 2015) suggest replacing the exponential kernel in (33) with a power law kernel s.t. we would have:

$$\xi_t(u) = \xi_0(u) \mathcal{E} \left( \eta \int_0^t \frac{dW_s}{(u-s)^\gamma} \right). \quad (35)$$

Following this slight edit, the underlying variance process would be of the form:

$$\sigma_t = \sqrt{\xi_t(t)} = \sigma_0 \sqrt{\mathcal{E} \left( \eta \int_0^t \frac{dW_s}{(t-s)^\gamma} \right)}, \quad (36)$$

$$= \sigma_0 \exp \left\{ \frac{\eta}{2} V_t - \frac{\eta^2}{4} \mathbb{E}[V_t^2] \right\}, \quad (37)$$

with

$$V_t = \int_0^t \frac{dW_s}{(t-s)^\gamma}. \quad (38)$$

The last equation defines the Volterra-fBm with Hurst parameter  $H = \frac{1}{2} - \gamma$ , or rearranging  $\gamma = \frac{1}{2} - H$ . With these adaptations (Bayer et al, 2015) note the prominent similarity to the RFSV model (13).

As the idea of any option pricing methodology is to use implied, i.e., forward looking volatility, however available information is by nature backward-looking, (Bayer et al, 2015) distinguish between these two notions by the Physical measure  $\mathbb{P}$  for realized data and the Risk-Neutral measure  $\mathbb{Q}$ , which captures the forward-looking expectation. With this the authors start by introducing the model under  $\mathbb{P}$  and later translate it into the  $\mathbb{Q}$ -World. Starting from the simple concept of volatility increments introduced with (13) in the RFSV model  $\log \sigma_{t+\Delta} - \log \sigma_t = v(W_{t+\Delta}^H - W_t^H)$ , (Bayer et al, 2015) reformulate the fBm  $W_t^H$  in terms of Wiener Integrals as

$$W_t^H = C_H \left\{ \int_{-\infty}^t \frac{dW_s^{\mathbb{P}}}{(t-s)^{\gamma}} - \int_{-\infty}^0 \frac{dW_s^{\mathbb{P}}}{(-s)^{\gamma}} \right\}, \quad (39)$$

where

$$C_H = \sqrt{\frac{2 H \Gamma(3/2 - H)}{\Gamma(H + 1/2)\Gamma(2 - 2H)}}, \quad (40)$$

ensures the covariance between the increments evolves according to (1) ensuring the process is a fBm. As motivated in Chapter 2, (1) defaults to the standard covariance structure of the Brownian motion for  $H = \frac{1}{2}$ . Moreover, for  $H > 1/2$  the increments are positively correlated and hence the resulting time-series of volatility (driven by the fBm) would be trending. For  $H < 1/2$  the increments are negatively correlated, and the time-series would instead be mean-reverting. For  $H$  between 0.1 and 0.2, as observed, the time-series would exhibit (the observed and desired) jaggedness, i.e., erratic up-and-down movements in rapid succession. Some might think of this as noise, however Gatheral notes, noise likely does not play such a big role in the observed data for high quality variance estimates at least<sup>75</sup>. Given (39), and in terms of (13) (with  $v_t = \sigma_t^2$ ) this leads to

$$\begin{aligned} \log(v_u) - \log(v_t) &= 2vC_H \left\{ \int_{-\infty}^u \frac{dW_s^{\mathbb{P}}}{(u-s)^{\gamma}} - \int_{-\infty}^t \frac{dW_s^{\mathbb{P}}}{(t-s)^{\gamma}} \right\}, \\ &= 2vC_H \left\{ \int_t^u \frac{dW_s^{\mathbb{P}}}{(u-s)^{\gamma}} + \int_{-\infty}^t \left[ \frac{dW_s^{\mathbb{P}}}{(u-s)^{\gamma}} - \frac{dW_s^{\mathbb{P}}}{(t-s)^{\gamma}} \right] \right\}, \quad (41) \\ &= 2vC_H \{M_t(u) + Z_t(u)\}. \end{aligned}$$

---

<sup>75</sup> See Gatheral's presentation here: <https://www.youtube.com/watch?v=gW073Tnx7CE>

This presents the evolution of the increments of forward volatility under  $\mathbb{P}$ . Note how the evolution of the increments is subject to two distinct parts: While  $Z_t(u)$  is implicit by the set of information<sup>76</sup> that is available at time  $t$  (called  $\mathcal{F}_t$ ),  $M_t(u)$  is normally distributed according to:  $M_t(u) \sim N(0, \frac{(u-t)^{2H}}{2H})$ . For easier numerical computation, (Bayer et al, 2015) introduce

$$\widetilde{W}_t^{\mathbb{P}}(u) = \sqrt{2H} \int_t^u \frac{dW_s^{\mathbb{P}}}{(u-s)^{\gamma}}, \quad (42)$$

and

$$\eta = 2\nu C_H \frac{1}{\sqrt{2H}}. \quad (43)$$

Therefore<sup>77</sup>, this leads us to  $2\nu C_H M_t(u) = \eta \widetilde{W}_t^{\mathbb{P}}(u)$ . With which we get

$$\mathbb{E}^{\mathbb{P}}[v_u | \mathcal{F}_t] = v_t \exp \left\{ 2\nu C_H Z_t(u) + \frac{1}{2} \eta^2 \mathbb{E} |\widetilde{W}_t^{\mathbb{P}}(u)|^2 \right\}. \quad (44)$$

Rearranging, (44) takes the form of (47) shown below. It constitutes the forward-looking volatility process for the rough Bergomi model under the physical measure  $\mathbb{P}$  for a suitable drift  $\mu_u$  and the Brownian motions  $Z_u^{\mathbb{P}}$  and  $W_t^{\mathbb{P}}$  that are correlated with some constant  $\rho$ :

$$\frac{dS_u}{S_u} = \mu_u du + \sqrt{v_u} dZ_u^{\mathbb{P}}, \quad (45)$$

$$v_u = v_t \exp \left\{ 2\nu C_H Z_t(u) + \eta \widetilde{W}_t^{\mathbb{P}}(u) \right\}, \quad (46)$$

$$= \mathbb{E}^{\mathbb{P}}[v_u | \mathcal{F}_t] \mathcal{E} \left( \eta \widetilde{W}_t^{\mathbb{P}}(u) \right). \quad (47)$$

### Change of Measure, Introducing the risk-neutral measure

Given the fully developed model under the physical measure  $\mathbb{P}$ , the authors now perform a so-called change of measure, from the  $\mathbb{P}$  world to the  $\mathbb{Q}$ -risk-neutral world. This is needed as it is generally not possible to price derivatives according to some probability measure, without introducing arbitrage. In technical terms, the expected price process is not a martingale. That

<sup>76</sup> In mathematical jargon this is called the filtration, the available past information generated by a suitable Brownian motion.

<sup>77</sup> Also take note of (43) which will become important later when we need to estimate  $\eta$  from our estimates  $v_{est}$  obtained in the previous chapter.

means for example  $C_t = \mathbb{E}[e^{-r(u-t)}C_s | \mathcal{F}_t]$  does not hold in the  $\mathbb{P}$  world for a derivative price  $C_t$ , expectation  $\mathbb{E}$  under information  $\mathcal{F}_t$ ,  $r$  some discount rate, and  $u > t$ . Hence, we seek a different measure under which the derivative process *is* a martingale. However, not just any measure  $\mathbb{Q}$  works to our advantage. To ensure that the change of measure does correspond to the real world, the expectation with respect to (wrt) the right choice of measure  $\mathbb{Q}$  should give the real-world price of the derivative in question. To do this, (Bayer et al, 2015) rely on the well-known Girsanov change of measure<sup>78</sup>. While the physical measure  $\mathbb{P}$  describes the probability or expectation that a financial instrument assumes a certain value, pricing under  $\mathbb{Q}$  allows us to express derivative prices in “risk-neutral” form. A suitable (that is arbitrage-free) risk-neutral measure then needs to not only be risk neutral for the derivative  $C_t$  on the underlying  $S_t$ , but also for  $S_t$  itself. Given the price process under  $\mathbb{P}$  (45), Girsanov’s theorem basically tells us how the Brownian motion  $W_t^{\mathbb{P}}$  evolves under  $\mathbb{Q}$ , where the drift term  $\mu_u du$  cancels out and the resulting diffusion process<sup>79</sup> will be (almost surely) a martingale. Thus, we are able to compute the expectation in the risk-neutral world, pricing the derivative<sup>80</sup>. (Bayer et al, 2015) put the change of measure into effect by defining the Girsanov measure change for  $dZ_u^{\mathbb{P}}$  as

$$dZ_u^{\mathbb{Q}} = dZ_u^{\mathbb{P}} + \frac{\mu_u}{\sqrt{\nu_u}} du, \quad t \leq u \leq T, \quad (48)$$

resulting in a price process of the form

$$\frac{dS_u}{S_u} = \sqrt{\nu_u} dZ_u^{\mathbb{P}}, \quad (49)$$

effectively removing the drift term from the spot price process (45), as motivated above. The resulting martingale process can now be used to build an expectation. Assuming the information  $\mathcal{F}_t$  generated by  $W^{\mathbb{P}}$  coincides with  $W^{\mathbb{Q}}$ , (Bayer et al, 2015) write the realized variance behavior (47) for the  $\mathbb{Q}$  measure as

$$\nu_u = \mathbb{E}^{\mathbb{P}}[\nu_u | \mathcal{F}_t] \exp \left\{ \eta \sqrt{2H} \int_t^u \frac{dW_s^{\mathbb{P}}}{(u-s)^{\gamma}} - \frac{\eta^2}{2} (u-t)^{2H} \right\}, \quad (50)$$

---

<sup>78</sup> See for example: [https://en.wikipedia.org/wiki/Girsanov\\_theorem](https://en.wikipedia.org/wiki/Girsanov_theorem) and the original paper (*Girsanov, "On transforming a certain class of stochastic processes by absolutely continuous substitution of measures"*, 1960)

<sup>79</sup> Brownian motions or Ornstein-Uhlenbeck processes are examples of diffusion processes

<sup>80</sup> For further reading see for example (Burgess, Martingale Measures & Change of Measure Explained, 2014) or alternatively:<https://www.quora.com/How-does-one-explain-what-change-of-measure-is-in-Girsanovs-Theorem-How-is-Girsanovs-Theorem-used-and-what-applicability-does-this-all-have-to-Mathematical-Finance-and-Probability>

$$= \mathbb{E}^{\mathbb{P}}[v_u | \mathcal{F}_t] \mathcal{E}\left(\eta \tilde{W}_t^{\mathbb{Q}}(u)\right) \exp\left\{\eta \sqrt{2H} \int_t^u \frac{\lambda_s}{(u-s)^r} ds\right\}. \quad (51)$$

Now to build the change of measure for the volatility process  $W_t$ , the authors note that to avoid flat volatility smiles for the CBOE Volatility Index (VIX), the change of measure  $\lambda_s$  must be correlated positively with  $W_t^{\mathbb{Q}}$ , the Brownian motion driving the volatility process  $v_u$ . However, they proceed with a change of measure (52) as if  $\lambda(s)$  were a deterministic function of  $s$ , forgoing the VIX fit<sup>81</sup>:

$$dW_s^{\mathbb{P}} = dW_s^{\mathbb{Q}} + \lambda(s). \quad (52)$$

From (51) this leads to

$$v_u = \xi_t(u) \mathcal{E}\left(\eta \tilde{W}_t^{\mathbb{Q}}(u)\right), \quad (53)$$

where  $\xi_t(u)$  is the forward variance curve :

$$\xi_t(u) = \mathbb{E}^{\mathbb{P}}[v_u | \mathcal{F}_t] \exp\left\{\eta \sqrt{2H} \int_t^u \frac{1}{(u-s)^r} \lambda(s) ds\right\}, \quad (54)$$

$$= \mathbb{E}^{\mathbb{Q}}[v_u | \mathcal{F}_t]. \quad (55)$$

In other words, the forward variance curve is simply given by the variance expectation under  $\mathbb{Q}$ , where the forward variance process is a martingale and  $\lambda(s)$  is the price of risk, as we translate between risk-neutrality and the physical world  $\mathbb{P}$  (52). To summarize, the rough Bergomi (rBergomi) model<sup>82</sup> then takes the form

$$\frac{dS_t}{S_t} = \sqrt{v_t} dZ_t, \quad (56)$$

$$v_t = \xi_s(t) \mathcal{E}\left(\eta \tilde{W}_s^{\mathbb{Q}}(t)\right). \quad (57)$$

Thus, the variance process feeds directly into the price process. The empirically observed anti-correlation between variance and price moves (see Table 3), may then be modeled by anti-

<sup>81</sup> Upon further analysis (Bayer et al, 2015) conclude that either the RFSV-model may be mis-specified or that market prices of options on SPX are not consistent with the observed option premia on VIX. This problem has since been addressed by various authors, e.g., (*Jacquier, Martini, Muguruza, On VIX futures in the rough Bergomi model, 2017*), (*Gatheral, Jusselin, Rosenbaum, The quadratic rough Heston model and the joint S&P 500/VIX smile calibration problem, 2020*), and (*Horvath, Jacquier, Tankov, Volatility Options in Rough Volatility Models, 2020*).

<sup>82</sup> Compare this to the mathematical concept introduced by Black Scholes (24):  $dS_t = \mu S_t dt + \sigma S_t W_t$ . While in the case of the Black Scholes model, we have a deterministic (constant) value for the volatility  $\sigma$ , the rBergomi model allows for stochastic changes in the volatility process.

correlating the Brownian motion  $Z$  driving the price process  $\frac{dS_t}{S_t}$  from the diffusion process  $W_t$  driving the volatility,

$$dZ_t = \rho dW_t + \sqrt{1 - \rho^2} dW_t^\perp. \quad (58)$$

$\rho$  sets the correlation between changes in price and volatility. In essence, (58) generates a bivariate normal random variable  $dZ_t$  from the volatility driving Brownian motion  $dW_t$ , given  $\rho$ . An accompanying proof can be found in Appendix C1. The appeal of the rBergomi model stems from the relatively little number of parameters it takes to calibrate the model to option prices: They are  $H, \eta, \rho, \xi(t)$ . Also notice how the volatility process  $v_t$ , the resulting spot price process  $\frac{dS_t}{S_t}$ , and the anti-correlation  $dZ_t$  are all modeled via these parameters only. In essence, they allow us to model the spot price process in its entirety. Simulating the price for a given financial asset over many paths allows us to build an expectation for that spot price in the future. With such expectation we can also define how likely a given option on that financial asset is to provide any pay off, and thus a corresponding price can be found for that option. With knowing the price, we can also know the respective IVs, given formula (29). Using the rBergomi model we can then find the parameters via calibration techniques and use the same parameters to price options for which we like to know the price. The subject of finding that price, as well as calibration and fit will be explored in the following chapter in greater detail.

To fit observed IV we need:

- The forward variance curve  $\xi_t(u)$ , as seen from time  $t$
- The Hurst parameter  $H$ , which controls the decay of the term-structure of the ATM volatility skew explored in (12)
- The volatility of volatility  $\eta$
- And the parameter of anti-correlation  $\rho$  for the two diffusion processes

The latter two may be used in tandem to calibrate the volatility-level and -bias of the volatility smile. In fact, the straightforward interpretation each parameter carries even allows for relatively simple manual calibration to observed IVs. This is shown in Figure 19, where the authors themselves guessed the parameters to fit to observed bid and ask implied volatilities for available maturities. We see a remarkably good fit for the entire surface, where the simulated model IVs lie almost perfectly between the bid and ask bounds.

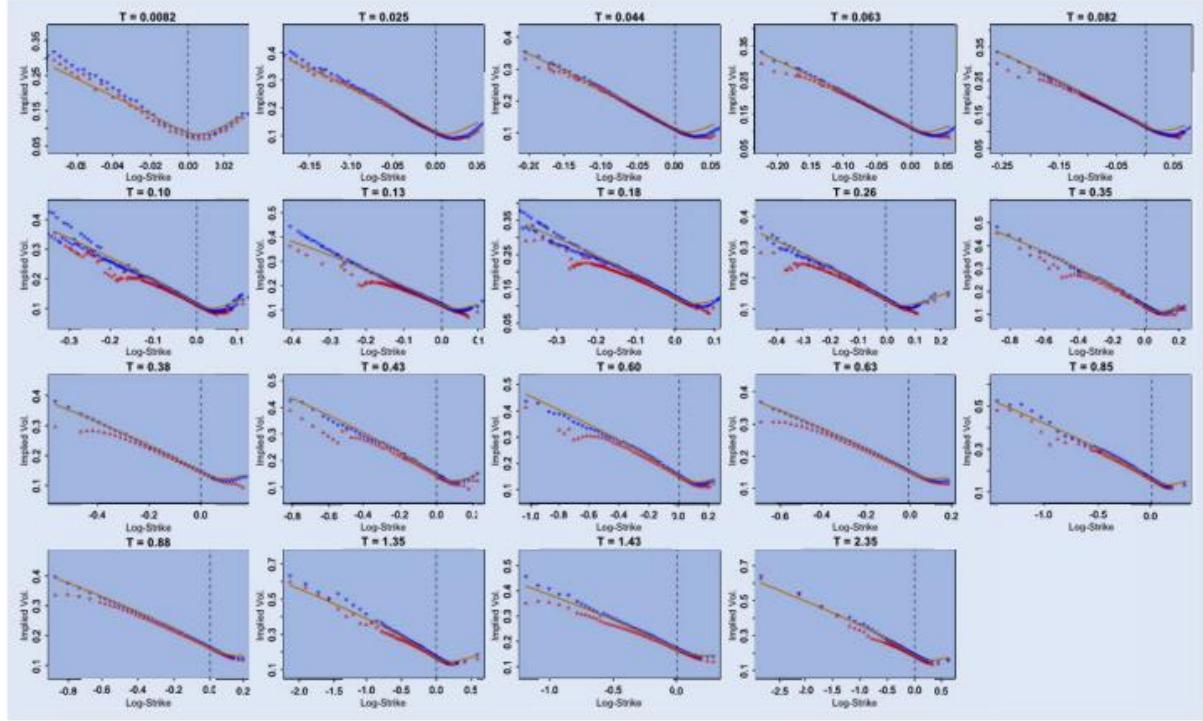


Figure 19: SPX smiles observed on 14.08.2013: blue and red points represent ask and bid IVs. The orange lines are from the rBergomi simulation, calibrated by (Bayer et al, 2015). See (Bayer et al, 2015, p 897)

Therefore, we now have a model that is able (by means of calibration) to extract the intrinsic features of implied volatilities from an option chain. As motivated above, this means given the observed data and our rBergomi model we now have the power to price *any* option, simply by knowing the set of model parameters. The independence of time  $t$  for the whole construct is by design, as noted in the previous chapter: The volatility surface, although fluctuating in level and orientation, features similar shapes. While the rBergomi model fits the observed implied volatility surface remarkably well, as observed in Figure 19, it comes at the expense of being non-Markovian<sup>83</sup>. While (Bayer et al, 2015) propose a simulation technique<sup>84</sup>, its slow calibration and simulation render it impractical from a practitioner's point of view. Further literature has addressed these shortcomings, of which two of the most prominent are considered in the next chapter and are subsequently used to calibrate and price observed options.

<sup>83</sup> That means a handy closed form solution such as for the Black Scholes model (25) does not exist. Therefore, other means become necessary to allow for practical application.

<sup>84</sup> Simulation relies on the idea to simulate all possible paths that the price may take from today until some expiration date  $T$  in the future. Given all possible simulated prices at this point  $T$ , we can then build an expectation and thereby attain the likelihood that a given option with strike price  $K$  is profitable or worthless.

# Pricing using the rBergomi Framework with Monte Carlo Simulation

While a time-efficient method calibrating the rBergomi model hasn't been available at the time of publication of their paper "Pricing under rough volatility", other authors have tried to come up with unique solutions to the non-Markovian calibration problem. Research has mainly gone two different ways:

- Improved simulation techniques, speeding up the process significantly
- Utilizing neural networks to learn the rBergomi parameter and implied volatility map

In the following two chapters this thesis will introduce the two distinct methods, discuss advantages and disadvantages as well as show how they can be used to price an observed option chain. Using the rBergomi model to price an option given an option chain, allows us to replicate the implied volatility structure, i.e., its distinct features, most closely.

- 1) Calibration: By calibrating to the option chain (i.e. minimizing the distance) to empirical data, we are able extract the calibrated model parameters  $(H, \eta, \rho, \xi_t(u))$ . These parameters *supposedly* carry all implied volatility information available at time of observation.
- 2) Pricing: Now with our true parameters, we are able to price any option contract, since we know the dynamics of the implied volatility. With these dynamics, the volatility behavior (57) of the underlying is controlled, which feeds into the price process (56). Thus, also enabling the computation of option prices at maturity T via the formula for an option's intrinsic value. Knowing the theoretical option prices, implied volatility can be computed in a straightforward manner.

## Turbo Charging and the Hybrid Scheme

Addressing the problem of long calibration times for the rough Bergomi model, Ryan McCrickerd and Mikko S. Pakkanen propose a more efficient simulation technique than employed by (Bayer et al, 2015). Specifically, they rely on a Volterra process to drive (rough) volatility and the resulting price process. To model the Volterra process efficiently, they utilize the hybrid scheme devised by (Bennedsen, et al, 2017) together with so-called variance-

reduction techniques. (McCrickerd, Pakkanen, Turbocharging Monte Carlo pricing for the rough Bergomi model, 2018) start by defining the rBergomi model by:

$$S_t = \mathcal{E} \left( \int_0^{\cdot} \sqrt{V_u} d \left( \rho W_u^1 + \sqrt{1 - \rho^2} W_u^2 \right) \right), \quad (59)$$

$$V_t = \xi_0(t) \exp \left( \eta W_t^\alpha - \frac{\eta^2}{2} t^{2\alpha+1} \right). \quad (60)$$

$W_t^\alpha$  is the Volterra process, following notation by (Bennedsen, et al, 2017), as compared to (38):

$$W_t^\alpha = \sqrt{2\alpha + 1} \int_0^t (t - u)^\alpha dW_u^1. \quad (61)$$

For  $\eta > 0$  and  $\rho \in [-1, 1]$ ,  $\alpha = H - 1/2$  and  $t$  time until expiration in years. While the process formulated in (59) and (60) looks different to the formulation by (Bayer et al, 2015), it features the same dynamics:  $\alpha$  controls the term-structure of volatility skew and level,  $\rho$  the skew itself and  $\eta$  the smile. Now as the integral in (61) is typically approximated by a Riemann sum, here this introduces computational problems in “rough” environments. Therefore, (McCrickerd, Pakkanen, 2018) draw from the hybrid scheme developed by (Bennedsen, et al, 2017) for the simulation of  $W_t^\alpha$ . The hybrid scheme is defined as being a unification of Wiener Integrals (see (39)) and a Riemann sum<sup>85</sup>, applied for processes like (62):

$$X_t = \int_{-\infty}^t g(t - s) \sigma_s dW_s, \quad (62)$$

with the kernel function  $g(\cdot)$ , stochastic volatility process  $\sigma$  and Brownian motion  $W$ . As the Riemann-sum is only able to *approximate* the kernel function  $g$ , the precision of the approximation becomes of central interest. (Bennedsen, et al, 2017) find that at near zero the approximation falls short of the exact volatility process, and hence they formulate the so-called hybrid scheme to address poor precision. Its main advantage constitutes providing a lower root-

---

<sup>85</sup> Note the similarity to (61). Also compare to the fOU process used in (14) which uses an exponential kernel instead of  $g(\cdot)$

mean-square error (RMSE) for the approximation of the kernel function over the standard Riemann-sum. With this, the hybrid scheme allows for a rather significant improvement over the Riemann-sum, reducing the error by figures between 50 and 100%. This is achieved while having the same computational complexity. They propose the hybrid scheme of the simple form of

$$X_t^n = \check{X}_t^n + \hat{X}_t^n , \quad (63)$$

where

$$\check{X}_t^n = \sum_{k=1}^{\kappa} L_g \left( \frac{k}{n} \right) \sigma_{t-\frac{k}{n}} \int_{t-\frac{k}{n}}^{t-\frac{k+1}{n}} (t-s)^\alpha dW_s , \quad (64)$$

$$\hat{X}_t^n = \sum_{k=\kappa+1}^{N_n} g \left( \frac{b_k}{n} \right) \sigma_{t-\frac{k}{n}} \left( W_{t-\frac{k+1}{n}} - W_{t-\frac{k}{n}} \right) . \quad (65)$$

For a process like (62), we are applying the hybrid scheme with  $g(t-s) = (t-s)^\alpha$ , where  $L_g$  and  $g$  from (64) and (65) are defined as by (Bennedsen, et al, 2017). Also see (66). For a base case with  $\kappa = 0$ , the hybrid scheme  $X_t^n$  recovers the standard Riemann-sum scheme, whereas for  $\kappa \geq 1$  the RMSE improves significantly. The hybrid scheme is using a (linear) combination of Wiener integrals (64) (as an “appropriate” power function at values close to zero) and a Riemann sum step function (65) elsewhere. In simple terms the hybrid scheme allows for rougher paths of the volatility process, while the Riemann-sum breaks down in highly rough environments.

(Bennedsen, et al, 2017) also apply the hybrid scheme to the rBergomi framework and its volatility process and note that the resulting IV smiles are tantamount to the exact simulation of the volatility process. They note that while the Riemann-sum is able to replicate the shape of the IV smile, it cannot capture its level. Moreover, for extremer (farther) ITM and OTM options, the Riemann-sum also falls short<sup>86</sup>. Contrary to this, the hybrid scheme manages to replicate implied volatilities for  $\kappa = 1, 2$  that are virtually identical to their benchmarks<sup>87</sup>.

---

<sup>86</sup> With pricing according to  $C(S_o, K, T) = \mathbb{E}[(S_T - K)^+]$ , the resulting option prices are so low that corresponding IVs would return zero for standard root-finding algorithms, using the logic (29).

<sup>87</sup> Specifically, they look at two different maturities, one for long and one for short maturity with  $\tau = 0.041$  and  $\tau = 1$ .

Given the insight into the applicability of the hybrid scheme, (McCrickerd, Pakkanen, 2018) implement its variant under  $\kappa = 1$ , to simulate  $W_t^\alpha$  in an efficient and precise manner,

$$W_{\frac{i}{n}}^\alpha \approx \tilde{W}_{\frac{i}{n}}^\alpha = \sqrt{2\alpha + 1} \int_{\frac{i-1}{n}}^{\frac{i}{n}} \left( \frac{i}{n} - s \right)^\alpha dW_u^1 + \sum_{k=2}^i \left( \frac{b_k}{n} \right)^\alpha \left( W_{\frac{i-(k-1)}{n}} - W_{\frac{i-k}{n}} \right), \quad (66)$$

with

$$b_k = \left( \frac{k^{\alpha+1} - (k-1)^{\alpha+1}}{\alpha+1} \right)^{\frac{1}{\alpha}}. \quad (67)$$

Figure 20 then shows the simulation of  $W_t^\alpha, V_t$ , and  $S_t$  using the hybrid scheme in action. For low roughness, with Hurst parameter close to 1/2 and thus  $\alpha \sim 0$ , we see a very regularized process, whereas the process exhibits high levels of roughness when  $H \rightarrow 0$ , example given for  $\alpha = -0.49$ . High roughness goes hand in hand with spikes in volatility and corresponding jumps in the price process, as can be observed on the left-hand side of the plot at  $t \sim 90$ . This shows that jump-like behavior can be achieved even for a continuous process. On the right-hand side then a very regular variance and price process can be observed.

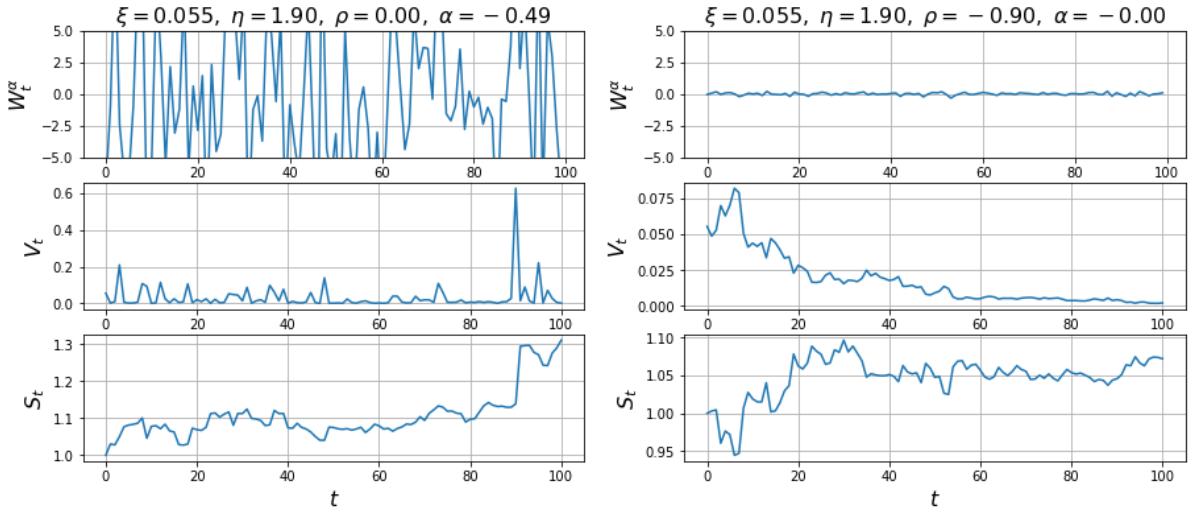


Figure 20: Simulation of  $W_t^\alpha, V_t$ , and  $S_t$  for different parameter combinations.

Given the set of parameters  $\alpha, \eta, \rho, \xi$ , a time of expiration  $T$  in years for a set of options (an option chain) and a corresponding set of log-strikes

$$k_t = \frac{\log K}{S_{t=0}}, \quad (68)$$

where  $K$  and  $S_{t=0}$  are the strike price and spot price at time  $t$  respectively, we can now define the model for practical application:

$$\begin{aligned}
Y_t^\alpha &= \sqrt{2\alpha + 1} \int_0^t (t-u)^\alpha W_u^1, \\
V_t &= \xi \exp\left(\eta Y_t^\alpha - \frac{\eta^2}{2} t^{2\alpha+1}\right), \\
B_u &= \rho W_u^1 + \sqrt{1-\rho^2} W_u^2, \\
S_t &= \exp\left(\int_0^t \sqrt{V_u} dB_u - \frac{1}{2} \int_0^t V_u du\right).
\end{aligned} \tag{69}$$

Here, starting from the Volterra process  $Y_t^\alpha$ , driven by the fBm  $W_u^1$ , we construct the volatility sample path  $V_t$ , with roughness given by the Volterra process. The (anti-) correlation  $\rho$  between volatility and price moves, modeled by  $B_u$ , is then fed with  $V_t$  into the spot price process  $S_t$ . Our result  $S_t$  represents the moves of an asset (e.g., SPX) from time  $t = 0$  up until  $t = T$ . However, this example only constitutes the simulation of one single path. Since we are interested in building an expectation on the price at expiration  $T$ , we require many such paths. Utilizing the GitHub repository provided by the authors here: [https://github.com/ryanmccrickerd/rough\\_bergomi](https://github.com/ryanmccrickerd/rough_bergomi), we can now explore how (McCrickerd, Pakkanen, 2018) price a “theoretical” option:

The rBergomi model implementation constructs  $N = 20.000$  sample paths for the Brownian motions  $W_u^1$  and  $W_u^2$ . Each with step size per year (length per year) of  $n = 366$ , s.t. we ensure at least 1 simulation step at any time until expiration, given by  $T$  (in years). The authors norm the spot price to  $S_{t=0} = 1$ , and predefine the range for the log-strike  $k$ . The end result then is an  $n \times N$  array of (simulated) spot prices from time  $t$  until maturity  $T$ . As we are dealing with European options that can only be exercised at maturity  $T$ , we are only interested in the last step of the vector  $n$ . As such, for the sequence  $t_1, t_2, \dots, t_{n=366}$ , we extract the vector:  $S_T = t_n \times N$ , giving us 20.000 simulated values for the last day at expiration. Now, with all possible realizations which the process  $S_t$  could attain at  $S_T$ , we want to compute the payoff that would result from option contracts with strike prices  $K$ , according to the formula:

$$C_N(K, t) = \frac{1}{N} \sum_{i=1}^N \max(S_T^i - K, 0). \quad (70)$$

Here, ignoring the summation for a moment, we compute the difference between one value of the array  $S_T$  and *every* strike price  $K$ . The “ $\max(\cdot)$ ”-function ensures that for negative values we would instead set the payoff equal to zero, as an OTM option is never exercised. This step is repeated  $N = 20.000$  times for every value of  $S_T^{i \dots N}$ . Note that we call the resulting  $K \times N$  array of payoffs  $C_N$ , the call price. This is because the expected payout an option gives (under the sample path  $i$ ) at expiration  $T$  should be exactly equal<sup>88</sup> to the price one pays for it today at  $t$ . Now, averaging all call prices  $C_N$  along the length of  $N$  (20.000 values) results in our expectation of what the fair call-price should be. Having computed the set of call prices (one for every strike price  $K$ ), we arrive at the implied volatility via:

$$\sigma_{BS}^N(K, t)^2 t = BS^{-1}(C_N(K, t)), \quad (71)$$

Which is just using the implied volatility formula introduced in (29), relying on the “`brentq`” method<sup>89</sup>.

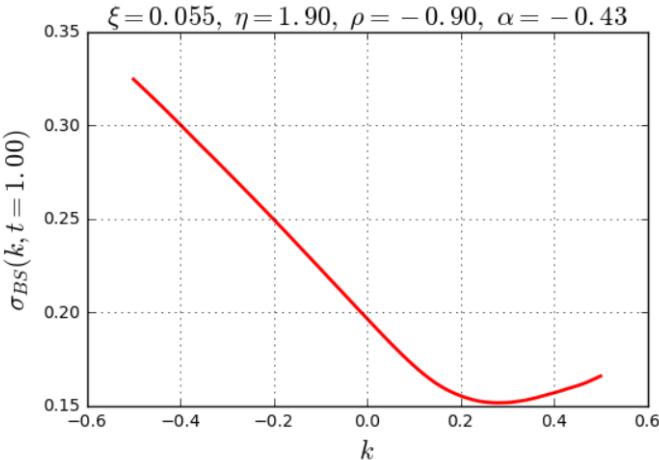


Figure 21: Implied volatility smile for arbitrary rBergomi parameters, and spot and expirations values.

In other words, it consumes a vector of rBergomi generated option prices  $C_N$ , the current observed spot price  $S_{t=0}$ <sup>90</sup>, a vector of observed strike prices  $K$ , and time to expiration  $T$ . Then for an arbitrary set of strike prices, rBergomi parameters, and a spot price and maturity, the latter two normed to 1, we display our calculations:

Figure 21, shows the implied volatility smile corresponding to the completely artificial set of option parameters (here:  $\alpha = -0.43, \eta = 1.90, \rho = -0.9, \xi = 0.055$ ). We see the model is not only able to replicate the skew but also the bias of the volatility smile (smirk); Properties

<sup>88</sup> This is of course ignoring the compound interest rate that discounts the payoff to present day value and dividend the asset  $S$  may pay between today and maturity.

<sup>89</sup> Further reading on the implementation can be found here:

<https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.brentq.html>

<sup>90</sup> The authors rely on a slight modification to the general Black Scholes model, namely that the model accepts the forward price  $F$  instead of the spot price  $S$ . Thus, the price compounding is not computed as part of the formula,

that are also observed for real option data, as motivated in the previous chapter. Hence, we can finally put a figure to the rBergomi model, showing us its real power of matching empirical properties of implied volatility. Thus, the rBergomi model allows us to price an option chain given the parameters. With these prices we can construct the volatility smile and surface, using the basic Black-Scholes framework. In essence, given option data, the rBergomi parameters control the shape or form of the volatility smile. The authors (McCrickerd, Pakkanen, 2018) report that their simulation technique manages to achieve a 20x reduction in computation. This is possible by variance reduction, antithetic sampling, and conditional Monte Carlo simulation as they note. With this a *relatively* quick simulation and pricing can be achieved.

### Pricing an observed Option Chain

Pricing an observed option chain seems redundant at first glance. However, knowing that not all contracts within such a chain are always liquid, as can be observed from Figure 2 or the option chain in Appendix A1, motivates this section. Here we will explore how to detect illiquid contracts and subsequently find accurate prices for those.

Now given the framework by (McCrickerd, Pakkanen, 2018), we implement relatively simple changes to accommodate our goal of pricing an observed option chain. The idea is to calibrate the model to an option chain of liquid options. The resulting set of parameters may then be used to price all (illiquid) options from the same option chain, following standard procedure by the various authors discussed in this work thus far. For this experiment we will use the option chain from Appendix A1.

We begin by recalling that (McCrickerd, Pakkanen, 2018) use a slight adaptation to the standard Black Scholes option pricing model:

$$C(F, K, \sigma_{BS}) = w * F * \Phi(wd_+) - w * K * \Phi(wd_-),$$

$$d_{\pm} = \frac{\log(F/K)}{\sqrt{\sigma_{BS}}} \pm \frac{\sqrt{\sigma_{BS}}}{2}. \quad (72)$$

---

as in (25), but is instead provided via the Forward Price  $F$ . Effectively reducing computational complexity. This will only become relevant later when using actual data.

It is here used to compute the implied volatilities for rBergomi generated theoretical option prices (71), with  $w = +/ - 1$  being a call/put switch. As we can see, the call option price  $C(F, K, \sigma_{BS})$  is not a function of the spot price  $S$ , but of the Forward price  $F$ <sup>91</sup>. This means, as the Black Scholes model does not compound the spot price, we need to compute the Forward price externally before feeding it into the formula:

$$F_T = (S_0 - D_T) \exp(r * \tau). \quad (73)$$

Where  $S_0$  is the spot price as observed today,  $D_T$  is the dividend which may be paid over the life of the Forward contract<sup>92</sup> (73) until maturity  $T$ , and  $r$  and  $\tau$  give the risk-free rate and the life until maturity respectively.  $D_T$  is then simply the sum of each incurred dividend's present value until maturity of the Forward contract:

$$D_T = d_1 * \exp(-r * t_{d_1}) + d_2 * \exp(-r * t_{d_2}) + \dots + d_n * \exp(-r * t_{d_n}). \quad (74)$$

Here  $d_1, d_2, \dots, d_n$  denote the singular dividend payments through life of the Forward contract. The corresponding variable  $t_{d_x}$  denotes the time until that dividend payment is due, facilitating the discounting back to today's value. As no 3<sup>rd</sup> party data vendors provide dividend estimations to the open public, here we have to include such estimation technique ourselves. To forecast dividends, we first estimate the dividend payout ratio for the past 365 days via:

$$\text{Dividend Payout Ratio}_t = \frac{\text{Dividends Paid}_t}{\text{Net Income}_t}. \quad (75)$$

Next, assuming the dividend payout ratio remains constant<sup>93</sup>, we pull expected earnings per share forecasts from yahoo finance<sup>94</sup> and apply the estimated payout ratio (75) to get expected dividends paid per share. Lastly, we discount the dividends back to today's value and then compound those until maturity as already explained in (73) and (74). At time of writing our data vendor had no earnings forecasts available for our asset under consideration in this section (AAPL). As such we assumed  $D_T = 0$  for all our results in this and the following chapter, dividends will however be incorporated for the empirical evaluation in Chapter 6. Now with

<sup>91</sup> The other symbols correspond to the Black Scholes formula introduced in the previous chapter, see (25) and (27), while the variable  $w$  can take value 1 or -1, to compute call or put option prices respectively.

<sup>92</sup> The term Forward contract means the price of the Forward  $F$  until expiration of the option. Here no actual Forward contracts are used.

<sup>93</sup> Which is a reasonable assumption for all assets under consideration, as these so called "dividend aristocrats" are known to pay a steady dividend rate.

<sup>94</sup> See <https://finance.yahoo.com/calendar/earnings/>

the formulae out of the way, we can discuss the procedure to calibrate and subsequently price observed options.

As the rBergomi model is able to generate price paths entirely from its set of model parameters (up to maturity of the observed option  $T$ ), the only two additional values we need to change in the model (70) of the previous section, are  $S_T$  and the set of strike prices  $K$ . To incorporate the expectation on the stock market price of the underlying at maturity date  $T$  we have to multiply the array  $S_T$  by our Forward price  $F_T$ . This will give us our vector of 20.000 potential prices that the underlying of the option may achieve at maturity (under today's information, however in the risk neural world  $\mathbb{Q}$ ). As we evaluate against a set of strike prices  $K$ , see (70) we need to incorporate these as well from the observed option chain. Also, in order to know the IV bounds that our rBergomi calibrated implied volatility smile may live in, we need to incorporate the bid and ask bounds for the option chain, akin to what (Bayer, et al., 2015) have done in Figure 19. We do this by extracting the bid and ask prices and compute the respective implied volatilities on them. Lastly, since not all options are liquid enough to be used for calibration<sup>95</sup>, we want to exclude any option contract from an option chain that a) is too illiquid and hence may provide unreasonable prices, and b) has not been traded recently. We identify illiquid options by a so-called inverse liquidity measure introduced by (Bayer, et al., 2018):

$$\text{inv\_liquidity}_i = \frac{\text{spread}_i}{\text{mid}_i} \leq 5\%, \quad (76)$$

with  $\text{spread}_i = \text{ask}_i - \text{bid}_i$ ; and

$$\text{mid}_i = \frac{\text{ask}_i + \text{bid}_i}{2}. \quad (77)$$

Here we want to exclude any option  $i$  that meets or exceeds the inverse liquidity bound (76) of 5%. Also, we want to exclude any contract that hasn't been traded recently:

$$\Delta\%_{\text{last\_trade}} = \frac{(t_{\text{obs}} - t_{\text{last\_trade}})}{\tau} \leq \max(5\%, 24\text{hours}). \quad (78)$$

---

<sup>95</sup> The idea here is that we want to find the *intrinsic* shape of the volatility smile for the latest available information. As such, only option contracts should be used that are most recent. After calibration we will know the shape and the parameters that generate this very shape. It can then be used to price any other (excluded) option.

Therefore, we propose function (78), which sets time since last trade in context to time to expiration  $\tau$ . We exclude any option contract that hasn't been traded in the past 24 hours or 5% of the time to expiration. This is necessary, as we are looking at many different option chains, with different time to expiration. Only when both functions (76) and (78) are not violated, we consider the option contract to be liquid enough to be used in calibration.

Et voila, our theoretical rBergomi framework is now ready to calibrate to observed options only incorporating the four variables  $T, F_T, D_T, K$ , the bid and ask bounds, as well as our two liquidity measures (76) and (78). For an initial guess of parameters (using the same as (McCrickerd, Pakkanen, 2018), here the fit for the option chain from Appendix A1 is shown:

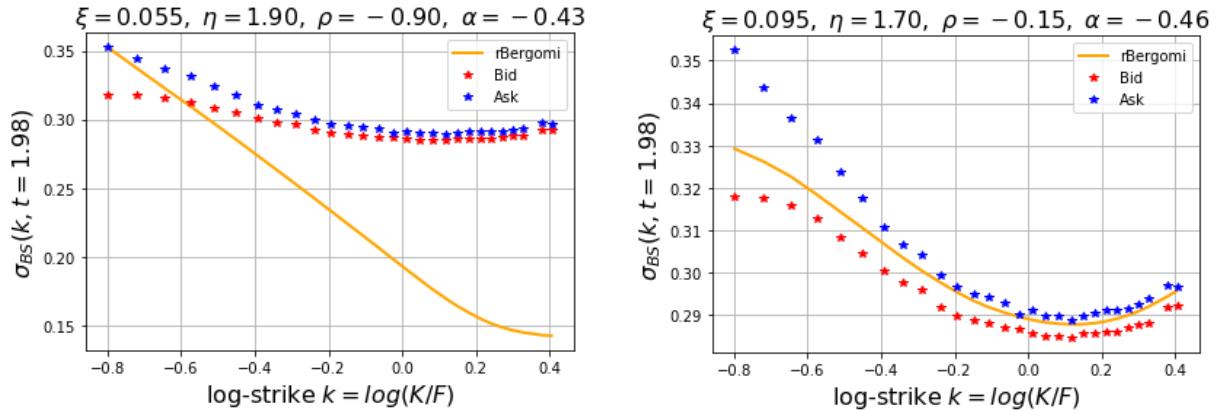


Figure 22: Showing before and after manual calibration of the rBergomi model to an observed option chain, given by the bid and ask bounds in red and blue respectively.

In Figure 22 we can observe that before calibration (left plot) the bid/ask and the rBergomi generated data lie quite far from each other. However, knowing the interpretation of each of the parameters as given in chapter 3, and observing that our generated curve is too low and does not exhibit enough curvature, we simply need to adjust  $\xi$  for the level and  $\eta, \rho$  for the smile dynamics to arrive at a much better fit (right plot).

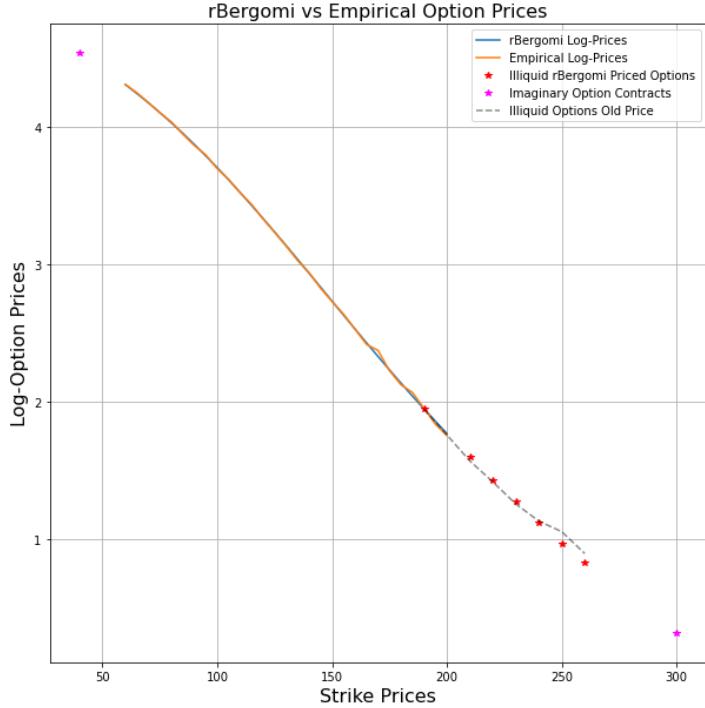


Figure 23: Comparison between rBergomi and empirical log option prices. Illiquid options are priced via calibrated rBergomi parameters (red stars), compared to the illiquid price (grey dashed line). In magenta color are imaginary option contracts, also priced using the calibrated model.

excluded option contracts. The grey dashed line shows the latest *illiquid* price these contracts have been traded for. We notice the contracts are quite close in price, only the last two (illiquid) options deviate somewhat from the calibrated rBergomi model. Looking at the option chain in Appendix A1, we see that the option contract with strike price equal to 250 is comparatively old with “Last Trade Date” older than 24 hours (hence why it’s been excluded). This is reflected in the plot as a very minor mismatch between calibrated prices and empirical (excluded) data. Demonstrating the power of the rBergomi model, we price totally imaginary option contracts shown in magenta color (at strikes of 40 and 300). These do not exist and are simply to show how financial institutions may go about tailoring option contracts to specific clients’ needs.

Lastly, we would like to propose a novel way to look at the rBergomi generated price expectations, which to our knowledge hasn’t been used in the literature thus far. From the rBergomi model, we simulate 20.000 price paths in order to take an expectation at the maturity date for all option contract that expire. Thus, we have 20.000 simulated potential spot price values at maturity  $T$  stemming from our rBergomi simulation. It comes only naturally to look at the implied distribution of these potential spot prices. This is depicted in Figure 24 below:

The pricing step is then simply to calculate the corresponding option price given the implied volatility fit via the Black Scholes model (25) or (26) depending on whether it’s a call or put option. We visualize the data by plotting the respective log-call prices, against each strike: In Figure 23 we can see how perfectly the rBergomi option prices match the empirical ones. The choice of using the log-call price is simply to exacerbate any difference between the two. The red stars show new prices as generated by the rBergomi model for the previously

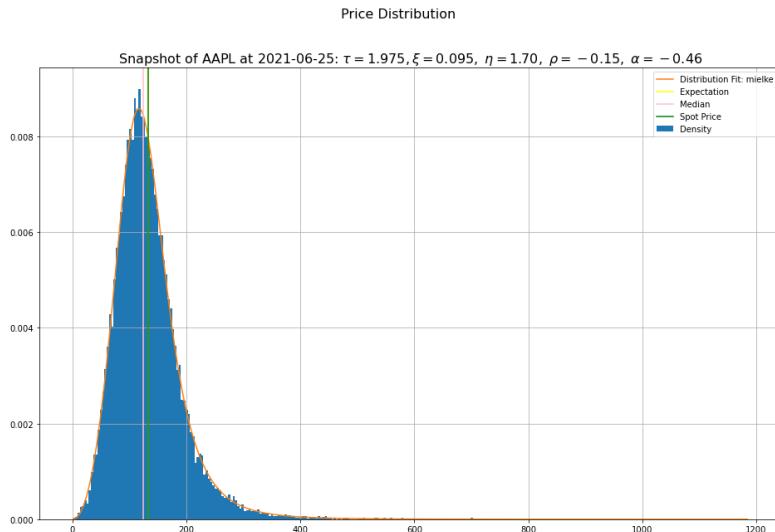


Figure 24: Implied price distribution of the rBergomi simulation given the calibrated parameters

Here we show, next to the current spot price, also the median as well as the arithmetic mean of the data. We see the distribution to be right skewed with a long tail. The expectation of a decreasing price can be explained by the positive implied volatility bias, as motivated in chapter 3. Now

that we have the risk-neutral expectation under  $\mathbb{Q}$  and also the distribution estimated<sup>96</sup>, we can easily calculate the probability of moneyness (in%) for each option<sup>97</sup>. This is done by using the CDF of the respective distribution, which depends on parameter combination and option chain.

However, the manual calibration that is necessary, as explored in Figure 22, is far from ideal. To arrive at such level of fit takes about 5-15 minutes, which renders the model impractical from a practitioner's point of view. Also, we only calibrated to one single option chain with one expiration. Usually at each time of observation, multiple expirations are available and hence the manual calibration task will become multitudes more difficult. Therefore, we seek a way to automate the necessary calibration. This is going to be explored in the following section.

### Automating the Calibration: Minimization Techniques

While we looked at one single option chain for AAPL from the 25.06.2021, many more expirations exist for that date<sup>98</sup>, these are shown in Table 4 below and visualized in Figure 10:

02.07.21	09.07.21	16.07.21	23.07.21	30.07.21	20.08.21	17.09.21	15.10.21
19.11.21	17.12.21	21.01.22	17.06.22	16.09.22	20.01.23	17.03.23	16.06.23

Table 4: Showing all available maturities (option chains) for AAPL, observed on 25.06.21

Moreover, we cannot just use the calibrated parameters from the option chain in the previous section to price all available option chains from Table 4. The fit for these parameters is shown

<sup>96</sup> The distribution is found by calculating the sum of squared errors (SSE) on each available distribution in the python SciPy package and taking the one with lowest SSE; here “mielke”. The SSE is defined as  $SSE = \sum_{i=1}^n (y_i - f(x))^2$ . However, the distribution may be different for each option chain and time of observation, as also the parameters change, as observed throughout this work.

<sup>97</sup> See Table 12 in Appendix D1

<sup>98</sup> Combining the IVs from them all, yields us the implied volatility surface, shown in Figure 10

in Figure 63 (Appendix D2), where the computation of IVs breaks at times, especially for those maturities that are closest to maturity. As such, we would like a quicker and more reliable way to calibrate to observed option chains. Now the literature has come up with various minimization techniques, of which we will use the Nelder-Mead<sup>99</sup> (NM) and the so-called Trust Region Reflective<sup>100</sup> (TRF) least squares algorithms to showcase two distinct techniques. Firstly, we will use both to calibrate to the same single option chain discussed previously, before we attempt at calibrating to all maturities from Table 4. To make automated calibration feasible, we will need a few functions. Namely, we require a function, which we will optimize, called the objective function, a function that determines the Subset of option chains, which we are going to optimize, and the optimization algorithm that does the optimization. Also here, we will only calibrate to cleaned data, as we do not want old, illiquid option contracts to skew our calibrated parameters. As our objective function we will use the MSE, computing the distance between rBergomi generated IVs and the mid-price implied volatilities (77). Since such a MC simulation-based calibration may be very computationally expensive (multiple hours of runtime are not uncommon), we want to have control over the number of option chains we are going to calibrate for. Therefore, we will use a function that controls the size of the optimization problem, by selecting and discarding certain maturities. The optimization functions are then the Nelder-Mead and TRF minimizers. To get an idea, we will begin by calibrating only to the single option chain discussed in the previous section:

<b>Minimizer</b>	<b>MSE</b>	<b>Runtime</b>	<b>Parameters (<math>\alpha, \eta, \rho, \xi</math>)</b>
Manual Calib.	4.7856384715e-06	15:00:00 Minutes (estimate)	-0.46 1.7 -0.15 0.095481
Nelder Mead	9.8524823041e-07	12:35:75 Minutes	-0.46006768 1.82672824 -0.14831442 0.09617495
Trust Region Reflective	1.0784500735e-06	18:41:20 Minutes	-0.45381377 1.70037475 -0.15073547 0.09606013

---

<sup>99</sup> Developed by J. A. Nelder, R. Mead A Simplex Method for Function Minimization, The Computer Journal, Volume 8, Issue 1, April 1965, Page 27

<sup>100</sup> Developed by Branch, Coleman, Li, A Subspace, Interior, and Conjugate Gradient Method for Large-Scale Bound-Constrained Minimization Problems, 1999), the TRF method solves small sub-problems within its trust-region. The shape of the full trust-region is determined by the distance to the bounds and the gradient's direction. This helps avoiding steps into bounds and explore the whole space of variables effectively.

Table 5: Comparing different minimization techniques for their accuracy and speed<sup>101</sup>

In Table 5 we compare our manual calibration to automated techniques. We can already see that simple calibration by hand within the bid and ask price bounds gives a very competitive result. The parameters we selected by hand are very close to what both minimizers achieved with comparable runtimes. However given the choice, letting the algorithm run is clearly advantageous to calibrating by hand. The Nelder-Mead method is certainly preferable to both other minimization techniques when calibrating to a single option chain. We can also visualize our result in Figure 25:

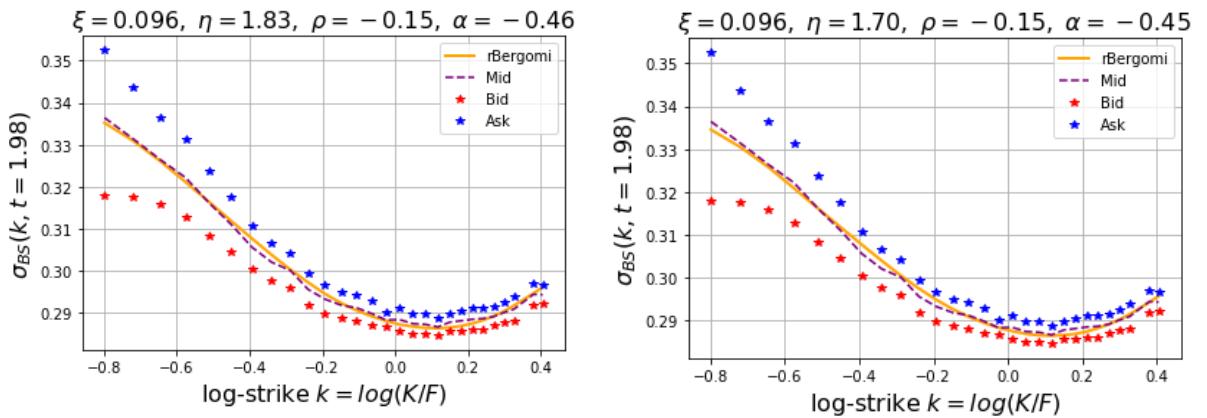


Figure 25: Comparing fit to mid-prices. Left: Nelder-Mead; Right: Trust Region Reflective

As expected from the very similar MSE, both fitted curves look virtually identical and are different only in the slightest, minuscule detail. Before we draw any conclusion on the best minimizer however, we want to see how they compare in a different, more demanding scenario: Calibrating the rBergomi model across the whole volatility surface.

As we want to keep runtimes within a reasonable interval, we restrict the calibration problem to a select range of maturities, which exhibit similar behavior to the rest of the (excluded) maturities. We choose the following option chains:

The Dates used for optimization are: July 2, 2021; July 16, 2021; August 20, 2021; September 16, 2022; June 16, 2023

Using the manual calibrated parameters again as an initial guess yields us the fit in Figure 63 in Appendix D2: We are reporting quite a large MSE of around 0.1690 for the selected optimization slice and a MSE of 0.0927 for the full surface, indicating much room for improvement. The results for calibration by minimization can be seen from Table 6:

---

<sup>101</sup> The optimization was done on a laptop with the 6 core 12 thread Intel Core i7 9750H and 16 GB DDR4 Ram

Minimizer	MSE (Optimization Slice)	MSE (Full Surface)	Runtime	Parameters ( $\alpha, \eta, \rho, \xi$ )
Nelder Mead	0.003932540442	0.003014108827	84:57:20 Minutes	-0.49999999 0.83515963 -0.93144967 0.06974231
Trust Region Reflective	0.158454507788	0.090654455106	16:51:31 Minutes	-0.41436948 1.70114322 -0.16172928 0.07606519

Table 6: Fit to the full implied volatility surface. Comparison between Nelder-Mead and Trust Region Reflective.

While the NM minimizer presents convincing numbers, the fit isn't great for all maturities shown. Especially those with longest time to expiration provide a subpar fit to observed values, see Figure 26. Moreover, we observe a difference in MSE and estimated parameters depending on the initial parameters given, see Appendix D3, where we provide a comparison when using a different set of initial parameters. Therefore, we conclude that the NM minimizer *may* provide good results (fit), but it is inconsistent and depends greatly on the initial parameters. Thus, it seems overwhelmed when dealing with problems of high dimensionality as it is unable to find global minima (just local minima) as observed here. The TRF minimizer on the other hand (Figure 27) manages a reduction in runtime for a reduction in accuracy. We can only influence the difficulty of the problem by setting the tolerance. However, for too small tolerances it seems no optimal solution can be found. Even after over six hours of runtime, we weren't able to attain any result from the function for a tolerance of 1E-10, at which point we stopped the minimization. The best compromise we managed to achieve between fit and runtime can be observed from Table 6, where we used a tolerance of 0.0009.

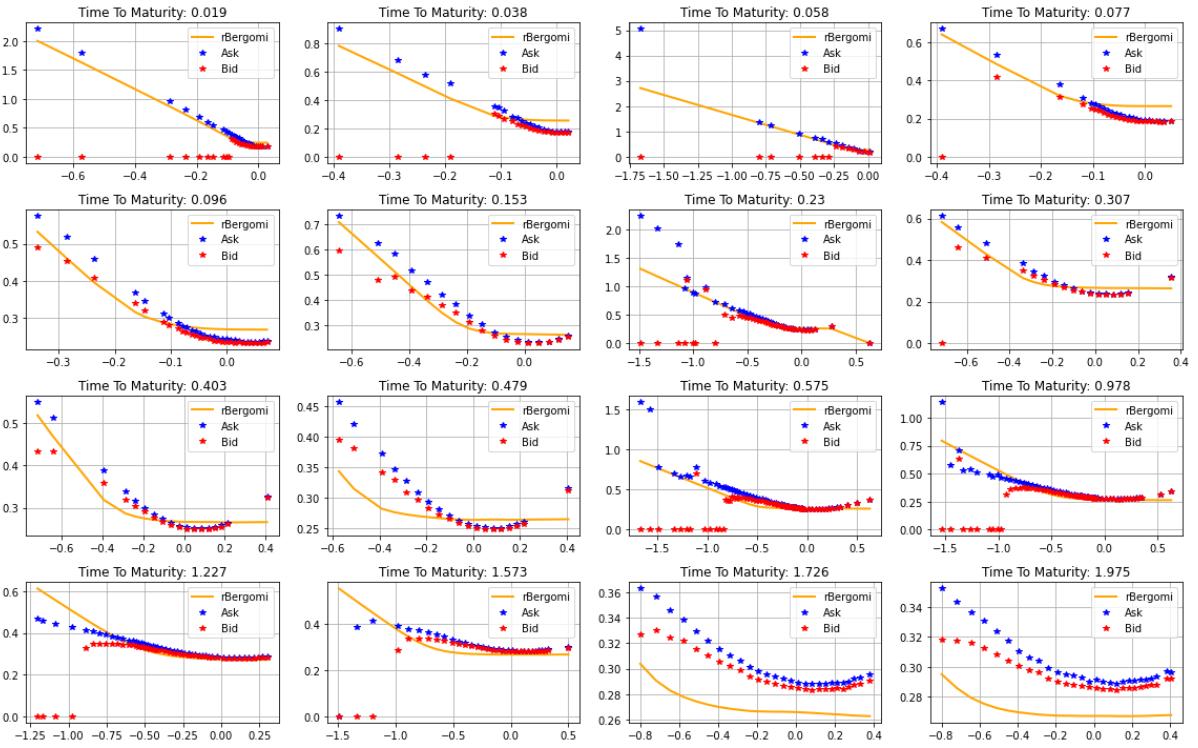


Figure 26: Showing the Nelder-Mead calibration (yellow) to bid and ask implied volatilities.

The calibration results we managed to achieve with TRF can be seen in Figure 27.

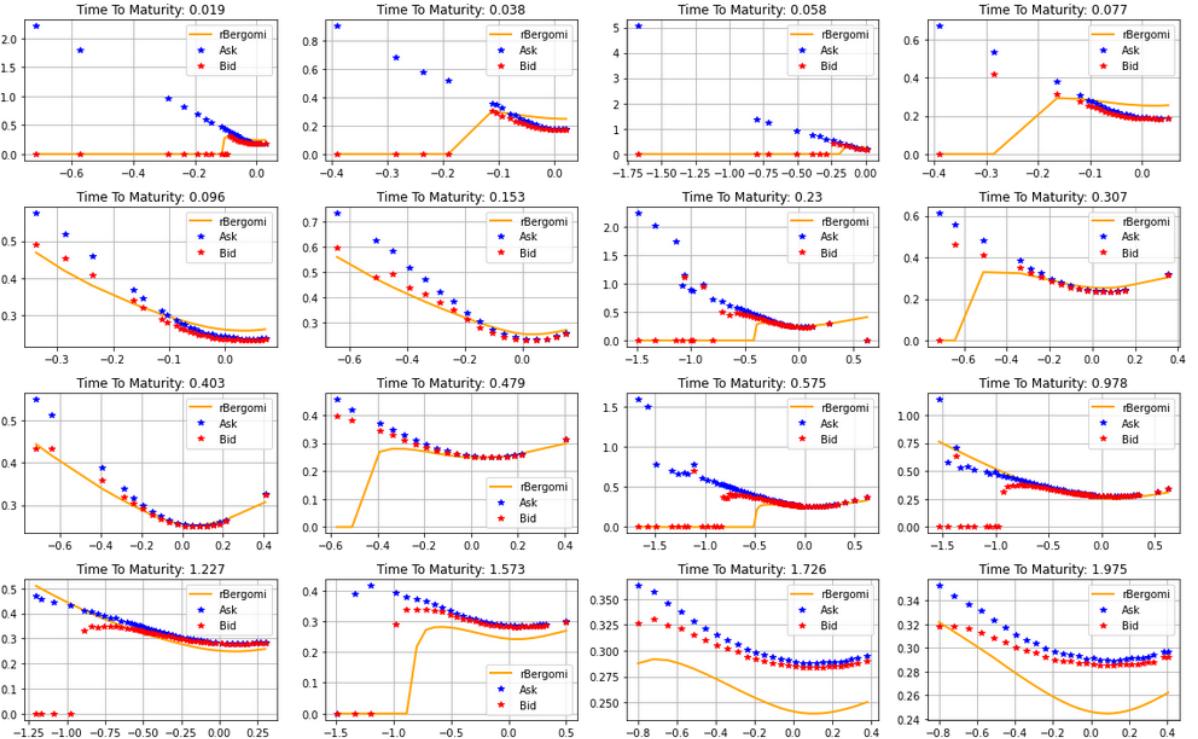


Figure 27: Showing the Trust Region Reflective calibration (yellow) to bid and ask implied volatilities.

Unfortunately, the fit is only decent for a few select expirations, while it deviates strongly or fails in IV computation for the remaining maturities. Especially far ITM (right hand side)

options fail to compute. It is also very close to the fit using the initial parameters. As such, we deem the TRF minimizer to be unsuitable for the calibration problem shown.

Taking a step back and observing the close to linear scaling of log-call prices in strike prices, as seen from our pricing attempt in Figure 23, we explore using option prices in the objective function instead of implied volatilities in Appendix D4. Unfortunately, we find that calibration to log-prices did not bring our hoped-for improvement in accuracy when calibrating to a full surface or even a single maturity date. In fact, it is unambiguously worse as calibration is significantly less accurate. While there is some improvement in speed, this did not outweigh the poor fit we observed.

Concluding the foray into calibration by minimization, we can see the task being far from straight forward. More work is required to understand the limitations of each minimizer and how its respective powers can be leveraged best in a scenario of high dimensionality and complexity such as here. We have seen that both minimizers are able to achieve decent results under optimal conditions (namely calibrating to one maturity at a time), however they depend greatly on the specifications of the minimization problem, the initial parameters given and the desired accuracy. From our data we see that the Nelder Mead method is far superior to the Trust Region Reflective calibration under MC simulation. However, many more such minimizers exist<sup>102</sup>. Additional insight is needed how such minimizers can be best applied to the calibration problem of rough volatility models. Unfortunately, the time it takes to calibrate using the turbo charged MC simulation (McCrickerd, Pakkanen, 2018), renders the model only applicable in research and academia. To make the model usable in high paced productive environments we require more speed still. The following chapter will provide an in-depth look into how such speedup may be achieved using the relatively novel approach of deep neural networks. We will discuss how such a network is constructed and what to look out for when dealing with volatility models. Subsequently, we will use such a network to calibrate to the problem at hand.

---

<sup>102</sup> For a comparison of some optimizers see for example (Alekseev, Navon, Steward, Comparison of advanced large-scale minimization algorithms for the solution of inverse ill-posed problems, 2011) and (Lyu, Lin, She, Zhang, A comparison of typical  $\ell_p$  minimization algorithms, 2013)

# Neural Network assisted Calibration

With the turbo charging approach by (McCrickerd, Pakkanen, 2018), we observed the problem shift from simulation to calibration: Now that we are able to simulate the rBergomi model efficiently, duration of calibration still renders the model prohibitive in time sensitive environments, such as market making, which highlights what (Bayer et al, 2019) term the “*need for speed*”.

A relatively recent approach represents calibration by deep learning techniques such as neural networks. Prominently introduced by (*Bayer, Horvath, Muguruza, Stemper, Tomas, On deep calibration of (rough) stochastic volatility models, 2019*) and (*Horvath, Muguruza, Tomas, Deep Learning Volatility, 2019*) but also by (*Liu, Grzelak, Oosterlee, The Seven-League Scheme: Deep Learning for Large Time Step Monte Carlo Simulations of Stochastic Differential Equations, 2022*), the many authors involved explore neural network based solutions with the goal of accurate and quick calibration of the rBergomi model. However, they note that their solution is agnostic to the specific model: Whether in rough volatility environments or considering more classical stochastic volatility models, neural networks may improve calibration for all such applications. Hence this approach should open up a new area of research, how to construct and calibrate volatility models using deep learning.

Before we begin, it is useful to understand the general structure and works of a neural network (NN), see (**Kinsley, Kukieta, Neural Networks from Scratch in Python, 2020**)<sup>103</sup> for a more detailed explanation on deep learning and neural networks or also (**Goodfellow, Bengio, Courville. Deep Learning. MIT Press, 2016**)<sup>104</sup>. A neural network is used to find a systematic connection between a given input and output. Especially for non-linear problems with multiple inputs and outputs a NN can be of advantage. It is comprised of a set of layers and a set of neurons within each layer (Figure 28). The minimum number of layers a network must have is restricted to three, with a single hidden layer and one input and output layer. As their names suggest, the input layer handles the input into the network, while the output layer gives out the end-result. The number of hidden layers should scale with the complexity of the problem and

---

<sup>103</sup> The authors made an accompanying youtube series available here:

[https://www.youtube.com/watch?v=Wo5dMEP\\_BbI&list=PLQVvaa0QuDcjD5BAw2DxE6OF2tius3V3](https://www.youtube.com/watch?v=Wo5dMEP_BbI&list=PLQVvaa0QuDcjD5BAw2DxE6OF2tius3V3)

<sup>104</sup> Another book of interest is (Bishop, Pattern Recognition and Machine Learning, Springer: New York, 2006)

the difficulty for the network to learn that problem. A neural network is called “*deep*” (DNN) when it has more than one such hidden layer.

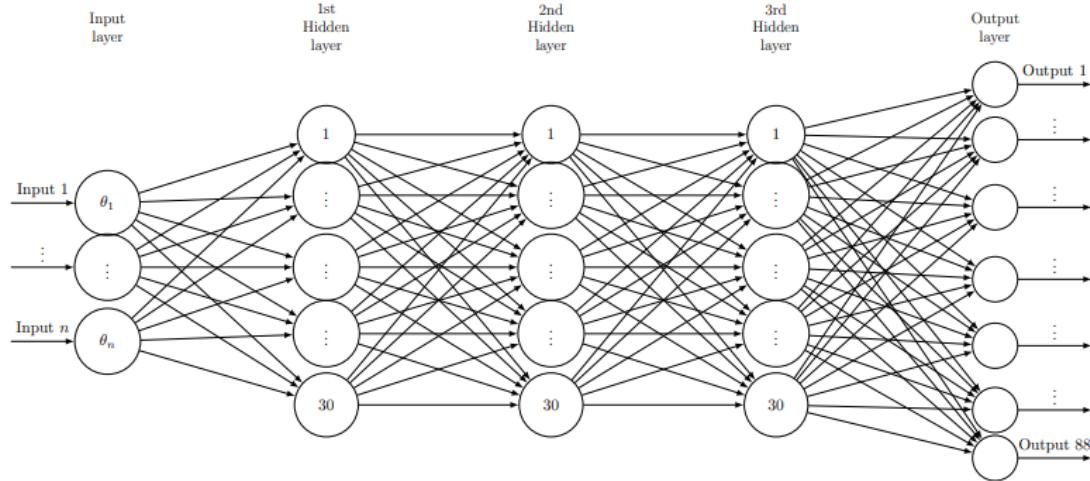


Figure 28: An example neural network as used in the work by (Horvath, et al., 2019). Taken from (Horvath, et al., 2019, p 15)

All neurons at each (hidden) layer are connected with every neuron from both the previous and next layer (Figure 29). Each neuron receives their (multiple) inputs from its prefixed neurons  $x_1 \dots x_n$  in the preceding layer.

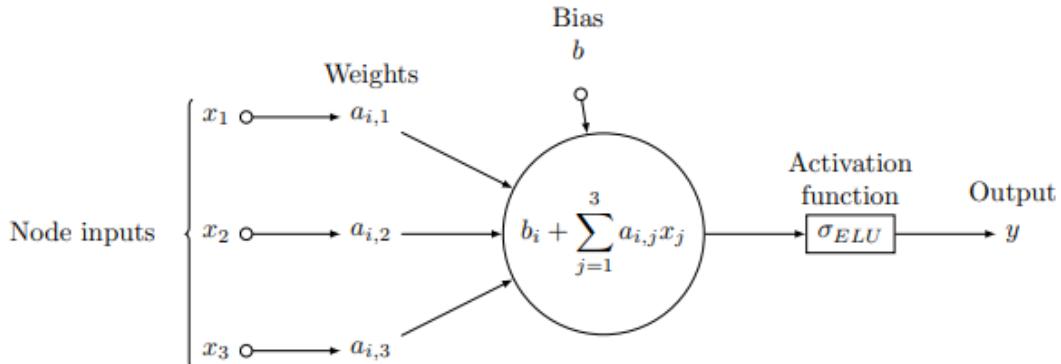


Figure 29: Basic construct of a neuron as part of a neural network. Taken from (Horvath, et al., 2019, p 11)

The neuron now assigns (through training) fixed corresponding weights  $a$  to each of its inputs  $x$ . Combining all these inputs in a weighted summation (plus a bias  $b$ ) provides the input to the activation function  $\sigma_{ELU}$ . This function then gives the output of the neuron, which will be consumed in similar fashion by its immediate posterior neurons of the next layer. These neurons then apply their own weights to this neuron’s output yet again<sup>105</sup>. The activation

---

<sup>105</sup> This is of course only the case when this neuron is not located in the output layer

function is the last step before the output. It is a transformation of the neuron's weighted input to account for the type of learning problem. It can be of linear (or any other) form to help the neural network attain a good fit to its training data<sup>106</sup>. The associated bias allows the neuron to shift its activation function, similar how a linear function  $y = ax + b$  depends on its intercept  $b$  to shift it up or down, the bias allows the activation function to attain the “true” values more closely and consistently. Arriving this way at an output of the *whole* NN, the objective function, or loss function, computes the distance between the training data and the neural network computed solution. The loss function then simply helps to minimize the distance between true and computed data. It needs to be carefully chosen to allow for a good fit to the (training) data.

With the structure of the NN built and set, we now need to train the network before being able to use it. Firstly, we decide on a suitable algorithm to use, such as gradient descent for example. Then in training the neural net is exposed to data, which should align both in quality and in input/output result with the data the NN is later going to be exposed to in production. For the gradient descent algorithm, which is here used, the training set is split into several batches, s.t. the  $i$ -th batch of batch-size  $m$  takes the form  $X_{i,m}^{batch} \subseteq X^{train}$ . The appropriate size of batch sizing is an import part of the overall training design. Large batch sizes correspond to an increase in computational complexity, as the batches need to be stored in memory and more gradients need to be computed at a given time. However larger batch sizes should reduce the variance of the gradient estimate, as noted by (Horvath, et al., 2019). On the other hand, small batch sizes lead to noisier gradient estimates, which may reduce parameter estimation accuracy. An epoch is the process of running the optimization algorithm of the network through all batches (Goodfellow, et al., 2016). For best effect, the neural net should be trained on that dataset multiple times, however training it too often on the same set of data may result in overfitting to the training set and lost generalizability when dealing with out of sample data. Once the network is trained, it should be able to instantaneously provide an output for any given input, without the need for any mathematical model. However, the inherent black-box design and the unknown works of the hidden layers in training come with its own set of issues. Only rigorous out of sample validation (which by design comes as the last step after design, implementation, and training) can shed light on whether the neural network works as intended

---

<sup>106</sup> The idea here is that a functional transformation takes  $x$  as an input and gives  $y$  as an output for any function  $y=f(x)$ . More reading can be done here: <https://stackoverflow.com/questions/2480650/what-is-the-role-of-the-bias-in-neural-networks> and [https://en.wikipedia.org/wiki/Activation\\_function](https://en.wikipedia.org/wiki/Activation_function)

and is not over- or underfitted to the training set. Therefore, finding the right network design and training data can be a long and tedious process.

Notice how we explicitly only mentioned calibration as the application for a neural network in rough volatility environments. An increasingly popular approach to pricing in quantitative fields is to develop deep learning algorithms that are entirely disconnect from any theoretical modeling framework. One obvious problem to this is of course that such advances cannot put a meaning to the hyperparameters<sup>107</sup> needed calibrating the model to the training set, nor justify the node and network design. This leaves the accuracy of such models in question whenever evaluating unseen data, which highlights a direct problem in heavily audited and regulated environments such as financial markets. Another, more model-focused design was proposed by (**Hernandez, Model calibration with neural networks, 2017**), developing the first NN-based approach to pricing options with the famous Hull and White model. In his model, option pricing and parameter calibration are done within the same step in the network. (Hernandez, 2017) proposes to learn the complete calibration routine, taking market data as input and returning calibrated model parameters. What (Bayer et al, 2019) call the *one-step* approach comes with several problems however: According to (Hernandez, 2017) himself, the main shortcoming is that his neural network does not give its user control over the mapping from pricing to calibrated parameters. Moreover, overfitting and so-called “black-box-solutions”, indicate the non-verifiability of NN hyperparameter-calibration and tractability of model accuracy as (Bayer et al, 2019) note. This is also precisely observed by (Hernandez, 2017). According to him the accuracy tended to differ between in sample and out of sample, possibly highlighting non-generalizability of his approach. Instead, (Bayer et al, 2019) take a different route: By firstly identifying the inherent bottlenecks of the calibration with standard pricing approaches<sup>108</sup>, the NN may only be applied where it is needed. Therefore, the authors tailor the neural network only to the problem at hand, while relying on more standard methods for the rest. Hence, the shortcomings of the approach pioneered by (Hernandez, 2017) can be avoided. This ensures that we retain control and the power to validate our network. To this end, (Bayer et al, 2019) develop what they call a *two-step* approach<sup>109</sup>:

---

<sup>107</sup> The term hyperparameter means the parameters needed to control and configure the learning in neural networks.

<sup>108</sup> As we have seen from our MC simulation exercise in the previous chapter, the simulation step requires extensive computational power, as every parameter combination needs to be tested requiring an iterating simulation of thousands of price paths over and over again.

<sup>109</sup> The two-step approach is in the spirit of (Hutchinson, Lo, Poggio, A nonparametric approach to pricing and hedging derivative securities via learning networks 1994), who used neural nets to learn the BS formula

1. The first step consists of learning the pricing map via a neural network, i.e., how the parameters of a given model and option prices (or mutatis mutandis implied volatilities) relate to one another. In simple terms, the neural network will take parameters as inputs and output corresponding implied volatilities.
2. The second step is then to calibrate the model to market data, using standard minimization techniques. This means, for given market (observed) implied volatilities the calibration algorithm will repeatedly feed parameters into the neural network (Step 1), until the NN output matches with the given market data most closely. The output will thus be the set of parameters that best reduce the distance between market data and neural network.

As such, the two-step method sits somewhere in between the traditional approach to pricing via simulation, and the direct one-step approach outlined above. Also compare to calibration by simulation as explored previously in this chapter. The neural network is effectively designed to replace the expensive simulation step, while the calibration step remains the same. This means, we build and train a network to approximate the pricing map (“offline”), and in a second step calibrate the model as approximated by the first step (“online”). (Bayer et al, 2019) note several benefits to the *two-step* approach:

- The training of the network can be done with synthetic and arbitrary data (e.g., by data generated using Monte Carlo simulation). This allows us to generate datasets as large or small as required. A side effect to this is that the training becomes more robust to changing environments, as its generally independent from empirical data and thus also independent from any current or past market regimes. As such, re-training of the network is not required.
- The main advantage over the *one-step* approach however is that we are able to comprehend the output and compare it to standard methods in accuracy and precision. As the neural network simply facilitates a speed-up to existing models, practitioners can apply their knowledge and leverage their years of expertise the same way as before, while also enjoying the advantages that new approaches using neural networks bring.

This approach in essence leaves out the expensive simulation step and thus calibration can be much faster. In simple terms, the neural network here is used as a compendium or dictionary, able to translate parameters directly into prices or implied volatilities, without having to resort to expensive simulation.

Recall that calibration is the optimization procedure that finds model parameters such that the implied volatility smile obtained by the model best approximates a given empirical implied volatility smile. In other words, calibration seeks to find parameters that find the smallest distance between observed and model generated data, leading to an optimal set of parameters, which provide the smallest distance. (Bayer et al, 2019) formalize the steps to calibration as follows: For a given volatility model that is tuned by a set of parameters  $\theta \in \Theta$  and an options contract that is defined by parameter  $\zeta \in Z$ , comprising of a maturity/strike-tuple  $\zeta = (\tau, K)$ , the pricing map, which results from the neural network in Step 1, is given by

$$(\theta, \zeta) \rightarrow \mathcal{P}(\theta, \zeta). \quad (79)$$

Mapping the set of model parameters and the set of option parameters to their corresponding (theoretical) option prices  $\mathcal{P}(\theta, \zeta)$ , computed by the model. However, as the model output can at best only approximate the true value, (Bayer et al, 2019) write  $\tilde{\mathcal{P}}(\theta, \zeta)_{\zeta \in Z'} \approx \mathcal{P}(\theta, \zeta)$ , to indicate that our pricing map is  $\tilde{\mathcal{P}}$  is an approximation of the true pricing map  $\mathcal{P}$ .  $Z'$  is the observable subset of all possible (infinite) option parameter tuples  $Z$ . The authors clarify that prices here and throughout any point in the neural network may be exchanged with their corresponding implied volatility, as practitioners find it often more informative dealing with IVs, rather than prices.

With this logic, calibration finds the set of parameters  $\theta$  that minimizes an error (distance)  $\delta$  between model prices  $\tilde{\mathcal{P}}(\theta, \zeta)_{\zeta \in Z'}$  and observed market prices  $\mathcal{P}(\zeta)_{\zeta \in Z'}$ .

$$\hat{\theta} = \operatorname{argmin}_{\theta \in \Theta} \delta \left( \tilde{\mathcal{P}}(\theta, \zeta)_{\zeta \in Z'}, \mathcal{P}(\zeta)_{\zeta \in Z'} \right). \quad (80)$$

The most practical way to implement the distance function  $\delta$  (the objective function) may be via computing the mean square error (MSE) between  $\mathcal{P}(\theta, \zeta)_{\zeta \in Z'}$  and  $\mathcal{P}(\zeta)_{\zeta \in Z'}$ . For the choice of a suitable minimizer, many options exist in the literature. Both (Bayer et al., 2019) and (Horvath, et al., 2019) propose calibration via the so-called Levenberg-Marquardt (LM) algorithm, a least squares method<sup>110</sup>. (Horvath, et al., 2019) actually show that calibration via

---

<sup>110</sup> Developed by (More, The Levenberg-Marquardt Algorithm: Implementation and Theory, 1977), is a robust and fast minimization technique based on least squares, usually best used for unconstrained problems.

the LM algorithm outperforms other methods such as BFGS, L-BFGS-B, and SLSQP<sup>111</sup> by a significant margin in average calibration time, which we were able to replicate in Figure 30.

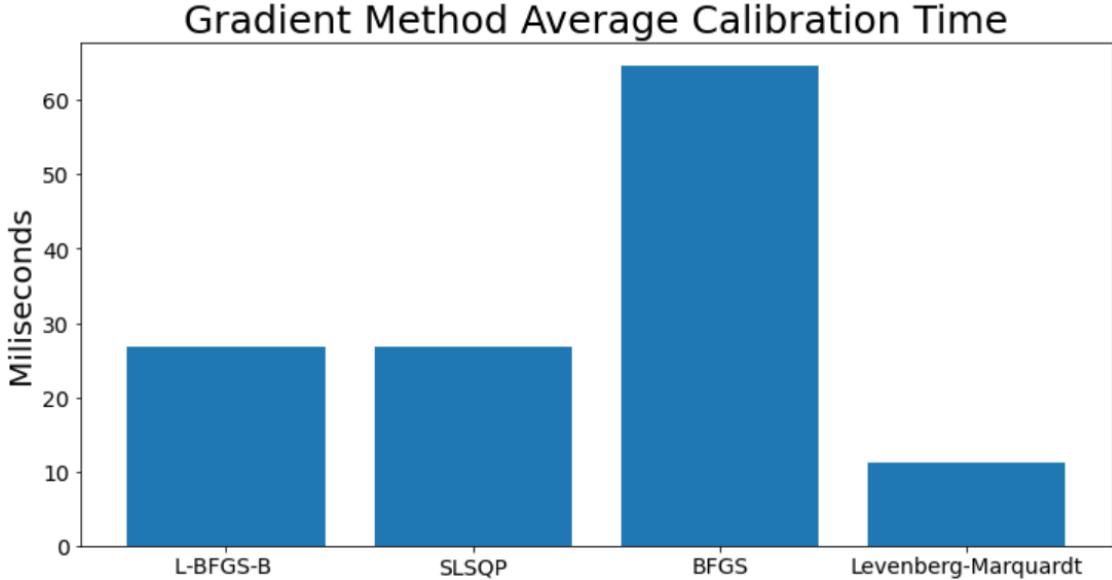


Figure 30: Verifying that Levenberg-Marquardt provides the greatest speed in average calibration time.

Having now understood the works of a neural network in the context of volatility models and the necessary calibration thereafter, (Bayer et al, 2019) emphasize two distinct ways of approximating the mapping (Step 1) between prices and parameters: **(I.) The point-based method** and **(II.) The grid-, or image-based method.**

The aforementioned *two-step approach* for point-based learning (I.) looks like the following:

- i. Learn the implied volatility map<sup>112</sup> and find the optimal weights  $\hat{\omega}$ , s.t.  
 $\tilde{P}(\theta, \zeta) = \tilde{\sigma}^{\mathcal{M}(\theta)}(\zeta)$ <sup>113</sup>. This corresponds to the optimization problem defined as:

$$\hat{\omega} = \operatorname{argmin}_{\omega \in \mathbb{R}^n} \sum_{i=1}^{N_{Train}} \eta_i \left( \tilde{F}(\omega, \theta_i, \tau_i, k_i) - \tilde{\sigma}^{\mathcal{M}}(\theta_i, \tau_i, k_i) \right)^2, \quad (81)$$

where  $\tilde{F}(\omega, \theta_i, \tau_i, k_i)$  denotes the neural network as a function of the NN weights  $\omega$  (which we are trying to find or learn), the model parameters  $\theta_i$  and the option specific

<sup>111</sup> For an overview on minimization techniques see <https://docs.scipy.org/doc/scipy/reference/optimize.html>. However, the LM minimizer does not allow for bounds, nor is it able to work when the residuals (e.g. distance between observed and model IVs) are less than the model parameters (four for the rBergomi model). This is precisely the problem we encountered trying to use the LM method for the turbocharged MC calibration discussed in the previous section.

<sup>112</sup> Which actually is the price map via the BS transformation in (29)

<sup>113</sup> This means model given implied volatility  $\tilde{P}(\theta, \zeta)$  (for some set of parameters  $\theta$  and maturity, strike tuple  $\zeta$ ) is equal to training set implied volatility  $\tilde{\sigma}^{\mathcal{M}(\theta)}$  for the same tuple  $\zeta$ , that was generated with the same parameters  $\theta$ .

parameters  $\tau_i, k_i$  for any maturity  $i$ . The artificial data which we train the deep neural network (DNN)  $\tilde{F}(\cdot)$  on is given by  $\tilde{\sigma}^{\mathcal{M}}(\theta_i, \tau_i, k_i)$ . It is the mapping of the model parameter set  $\mathcal{M}(\theta)$  in combination with the option parameters  $\zeta = (\tau, K)$  to their respective implied volatilities  $\tilde{\sigma}^{\mathcal{M}}$ .  $\eta_i$  is another vector of weights. In essence, (i) is another minimization problem, configuring and teaching the DNN how IVs and model/option parameter combinations relate to one another based on the artificial data in the training set  $\tilde{\sigma}^{\mathcal{M}}(\theta_i, \tau_i, k_i)$ . The DNN will automatically, by repeated training, find the optimal weights  $\hat{\omega}$ , which minimize the distance between (external) training set data and (internal) DNN evaluated data.

- ii. Once step i has been completed, we can finally shift our focus to “Production”: Find the optimal set of model parameters  $\hat{\theta}$ , which minimize the distance between market observed IVs and neural network generated implied volas from (i):

$$\hat{\theta} = \operatorname{argmin}_{\theta \in \Theta} \sum_{j=1}^m \beta_j \left( \tilde{F}(\hat{\omega}, \theta_i, T_i, k_i) - \sigma_{BS}^{MKT}(\tau_j, k_j) \right)^2, \quad (82)$$

with some user given weights  $\beta_j$  (for example liquidity focused weighting). Here we replaced the predefined artificial training data  $\tilde{\sigma}^{\mathcal{M}}(\theta_i, \tau_i, k_i)$  by actual market data  $\sigma_{BS}^{MKT}(\tau_j, k_j)$  and the deep neural network  $\tilde{F}(\hat{\omega}, \theta_i, \tau_i, k_i)$  has been configured with the appropriate weights (achieved through training)  $\hat{\omega}$  such that it can find implied volatilities given the option parameters  $\zeta = (\tau, K)$ . The standard calibration algorithm in (ii) then ensures that we obtain the parameters  $\hat{\theta}$ , which minimize the distance between DNN generated and observed IVs.

While step (ii) can be solved by standard calibration algorithms that are already well established in the literature<sup>114</sup>, step (i) provides novel challenges. Especially the data used for training, as well as the architecture of the network are key issues, remarked by (Bayer et al, 2019). As we only want to restrict ourselves to realistic scenarios when training, it makes sense to choose possible parameter combinations with this in mind (i.e., it’s not very useful to train the network on maturities that are too far in future). The authors propose to use uniform distributions on fixed parameter intervals, however more tailored training sets may be used,

---

<sup>114</sup> The reduction in dimensionality (discussed above) helps to minimize the room for pitfalls and errors, which we observed when attempting to calibrate via the MC simulation.

which take historical model parameter distributions into account. With this the simulation of model/option parameter combinations according to our wishes can be used to compute the respective sets of option prices or IVs, e.g., by Monte Carlo simulation.

Now for the grid-, or image-based approach (II), the authors propose to store the pricing or IV map as a grid of pixels in image form, which improves performance. Starting by defining a fixed maturity/strike grid  $\Delta = \{\tau_j, k_i\}_{i=1, j=1}^{n, m}$  a similar two step approach can be applied

- i. Approximate the pricing map  $\tilde{F}(\theta) = \{\sigma_{BS}^M(\tau_i, k_i)\}_{i=1, j=1}^{n, m}$  similar in form to (79) in the point wise method. The objective function then is:

$$\hat{\omega} = \operatorname{argmin}_{w \in \mathbb{R}^n} \sum_{i=1}^{\mathcal{N}_{Train}^{Reduced}} \sum_{j=1}^L \eta_j \left( \tilde{F}(\theta_i, \omega)_j - \tilde{\sigma}^M(\theta_i, \tau_i, k_i) \right)^2, \quad (83)$$

where  $\mathcal{N}_{Train} = \mathcal{N}_{Train}^{Reduced} \times L$  and  $L = \text{strikes} \times \text{maturities} = nm$ . Notice how the neural network  $\tilde{F}$  is now only dependent on the set of model parameters  $\theta$ , this is because the option parameters  $\zeta = (\tau, K)$  are fixed by design and thus no longer needed in the learning stage (i).  $\eta_i$  is again a vector of weights, which may be used to liquidity weigh the training set, as suggested by the authors.

- ii. Calibrate the trained neural network from (i) to the observed market data, i.e., find the set of parameters  $\hat{\theta}$ , that match or minimize the distance to the empirical implied volatilities  $\sigma_{BS}^{MKT}$ .

$$\hat{\theta} = \operatorname{argmin}_{\theta \in \Theta} \sum_{j=1}^L \beta_j \left( \tilde{F}(\theta)_i - \sigma_{BS}^{MKT}(\tau_i, k_j) \right)^2. \quad (84)$$

Where  $\tilde{F}(\theta) = F(\theta, \hat{\omega}(\Delta))$  and  $\beta_j$  some user given weights. Notice also that the optimal weights  $\hat{\omega}$  depend on  $\Delta$  implicitly, which influences the neural network. Once again, we are now using the neural network to iteratively pass in parameters  $\theta$  and observe which match the  $\sigma_{BS}^{MKT}$  IVs the closest.

While the point-based approach should be highly accurate, the learning phase may take a long time as every possible combination of strike  $K$  and time to expiration  $\tau$  has to be mapped to its corresponding model parameters. The grid-based approach however utilizes a pre-defined grid

of points in maturity/strike space and maps these points to the corresponding IVs. As the model is then only able to directly price at the pre-defined points, (Bayer et al, 2019) suggest interpolating between these points using cubic splines<sup>115</sup>, which happens in a separate step outside the actual neural network, while in the point-wise method the interpolation between sampling points (in option-parameter space  $\zeta = (\tau, K)$ , as well as in model-parameter space  $\Theta$ ) is handled by the neural network  $\tilde{F}(\theta)$ . A downside of course is that (next to added computational cost) the spline-interpolation may result in reporting wrong IVs in edge cases (as additional room for approximation errors is introduced) and that the neural network is only able to calibrate to contracts that fall within the grid, all other points will be disregarded. While at first this second approach seems more complex, the authors note several benefits to this option and the two-step approach in general:

- The grid-based approach is particularly designed for applicability and efficiency in everyday calibration practice, as overall complexity for the NN is reduced.
- With the grid-based approach the calibration problem can be tackled by a remarkably small NN. (Bayer et al, 2019) report that with only 30 neurons over 3 layers the network is well suited to perform the pricing task.
- With such a small network, the computational complexity is comparatively manageable, s.t. building, training, and application can be performed on everyday computer hardware, rendering it practically feasible for daily use in the financial industry without requiring hardware upgrades.

The main differences between the point- and image-based methods indicated by (Bayer et al, 2019) are summarized below:

- As already commented above, the greatest difference between point- and image-based learning is the difference in interpolation that happens either in the neural network itself (for the point-wise learning) or outside the network. The interpolation within the neural net happens within model parameter space, s.t.  $\theta \rightarrow \sigma^M(\theta, \tau, k)$ , which, according to (Bayer et al, 2019), is less well understood as interpolation in market IV space  $(\tau, k) \rightarrow \sigma^M(\theta, \tau, k)$ . Hence the authors choose interpolation outside the network over interpolation within the network.

---

<sup>115</sup> Splines are a kind of polynomial function. Such functions are able to mimic the shape of empirical IVs. However, also the so-called Chebyshev or SVI interpolation may be used to approximate the implied volatilities, as remarked by (Bayer et al, 2019).

- Relying on the image-based approach may then be seen as a way to efficiently reduce the dimension of the learning problem, as the dimensionality of the task is shifted from the input to the output of the network. The behavior of volatility smiles at each given expiration (i.e., the volatility surface) is well understood, thus interpolation outside of the network can be seen as reasonable.
- One advantage of the point-wise approach is that it is easier to train the network on the structure of real financial data, by adjusting the sampling distribution. This means as we are more interested in liquid options and as such the distribution of training data can be adjusted accordingly, whereas the grid-wise learning has to rely on a careful choice of the vector of weights  $\eta_j$
- For the same size of training set, the grid-based approach leads to a variance reduction, as only the model parameters need to be sampled, while strike and maturity is fixed.
- However, if generalizability is of primary concern, the point-wise approach may be preferable, as it is more generally applicable as (Bayer et al, 2019) remark.

While generation of data and training are both quite expensive, they only need to be performed once when building the network. For implementation of the grid-based approach, both (Bayer et al, 2019) and (Horvath, et al., 2019) agree on the same structure with three hidden layers<sup>116</sup>, featuring 30 neurons each. The input of model parameters is given by dimension  $n^{117}$ , while they restrict the output to a fixed grid of eight maturities and eleven strikes. With this, the network parameters to calibrate through training amount to:

$$(n + 1) \times 30 + 3 \times (1 + 30) \times 30 + (30 + 1) \times 88 = 30n + 5548. \quad (85)$$

Whereas for Black Scholes or lognormal SABR models the pricing map can be learned to decent accuracy with only a single hidden layer (Bayer et al, 2019), the non-linear more complex nature of rough volatility models necessitates deeper networks for accurate pricing, as mentioned by (Bayer et al, 2019). Lastly, as activation function they choose  $\sigma_{Elu} = \alpha(e^x - 1)$ . The neural network is trained using the “Adam minibatch” scheme introduced by (Kingman, and Ba, A Method for Stochastic Optimization, 2015), which is a gradient decent algorithm. For the calibration the already discussed Levenberg-Marquardt algorithm is used.

---

<sup>116</sup> (Bayer et al, 2019) report that adding additional hidden layers beyond four did not reduce errors in a consistent manner.

<sup>117</sup> n=4 for the rBergomi model with a fixed forward variance curve. With a floating forward variance curve, the number of inputs scales with 3+ number of discrete time-points of the forward variance curve.

With the neural network set and done, the authors (Bayer et al, 2019) and (Horvath, et al., 2019), achieved a speed improvement over standard Monte Carlo simulation to the order of between 21.000 to 35.000 times. (Horvath, et al., 2019) happen to provide code on their GitHub repository here: <https://github.com/amuguruza/NN-StochVol-Calibrations>, with which we are able to use their “dummy” neural network. (Horvath, et al., 2019) here use a grid-based approach, however without interpolation. As such the neural network is only able to assess volatility input data that lies on these grids exactly. (Horvath, et al., 2019) validate their NN again with artificial data; this time from their test-set (distinct from their training-set), which lies on these grids as well. Therefore, even without interpolation, pricing can be done alongside the grid lines. To assess the performance of the NN, the authors feed the true parameters  $\bar{\theta}$  into the neural net  $\tilde{F}(\bar{\theta})$  and compare the output against artificial IV data  $\sigma_{BS}^{\tilde{F}(\bar{\theta})}(\tau, k)$ :

$$\tilde{F}(\bar{\theta}) \approx \sigma_{BS}^{\tilde{F}(\bar{\theta})}(\tau, k). \quad (86)$$

This is shown in Figure 31, where we see the neural network matches the validation data to a degree, almost indistinguishable by eye.

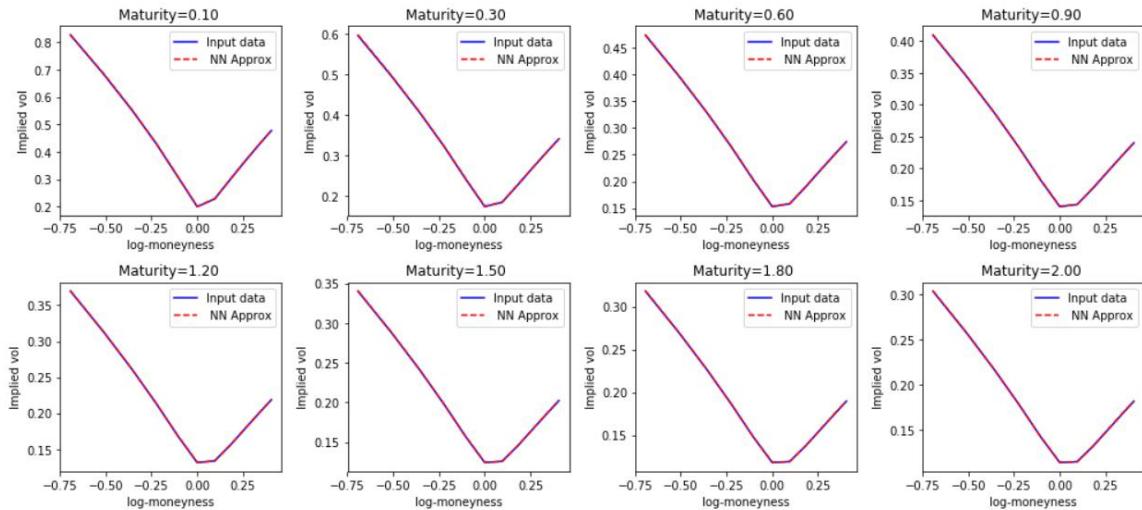


Figure 31: Implied Volatility for different expirations (i.e., different option chains) both stemming from artificial generated test-set data and neural network approximations, using the data and NN design employed by (Horvath, et al., 2019).

As the estimations in Figure 31 are indistinguishable from the true data, Figure 32 highlights the relative errors<sup>118</sup> in strike/maturity space, via the function:

---

<sup>118</sup> This are the deviations between implied volatility from validation data vs NN approximated IVs, given the true parameters  $\bar{\theta}$

$$E_{\mathcal{R}} \left( \sigma_{BS}^{\tilde{F}(\bar{\theta})} \right) = \frac{\left| \sigma_{BS}^{\tilde{F}(\bar{\theta})}(\tau, k) - \sigma_{BS}^{MKT}(\tau, k) \right|}{\sigma_{BS}^{MKT}(\tau, k)} . \quad (87)$$

While we report slightly higher errors across all three evaluation matrices, compared to what (Horvath, et al., 2019) show, the conclusions the authors make are here valid the same.

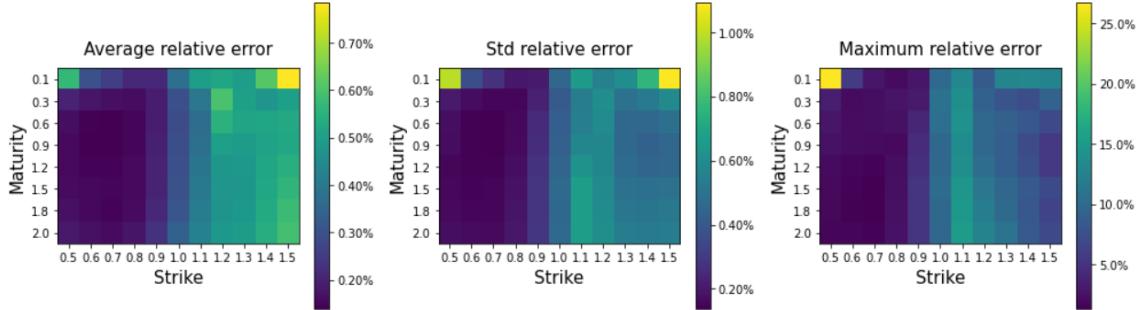


Figure 32: Strike/Maturity relative error (computed as a percentage) between validation set and NN estimation.

(Horvath, et al., 2019) state, as easily observable, that errors of the neural network are more pronounced for short to maturity far OTM and ITM option contracts. These errors also align with the errors from MC simulation, as they remark. However, as we are dealing here *exclusively* with artificial data, these observations clearly present best-case scenarios for the neural network. Nonetheless they exemplify the power of the rBergomi framework with neural network assisted calibration. Unfortunately, the neural network employed by (Horvath, et al., 2019) is too restrictive to price actual observed market data. This has to do with **a**) that no interpolation is allowed between grids, and **b**) that the training data has no generalizability. As such, while their NN framework looks promising, it may only be used as a starting point to develop a more sophisticated neural network.

With Figure 31 and Figure 32 highlighting the performance of the neural network (Step1), given true parameters  $\bar{\theta}$ , we need to look at estimates for the parameters to assess the accuracy of the whole *two-step approach*. Using the NN in Step 1 as discussed, and the LM optimizer function in Step 2, we want to assess the parameter accuracy between training and validation set via the parameter relative error, defined as

$$E_{\mathcal{R}}(\hat{\theta}) = \frac{|\hat{\theta} - \bar{\theta}|}{|\bar{\theta}|} . \quad (88)$$

Where  $\hat{\theta}$  are the calibrated model parameters (resulting from the *two-step approach*) and  $\bar{\theta}$  presents the true model parameters associated with the generated artificial data that's been split into training and validation sets. Figure 33 then shows the relative error by parameter.

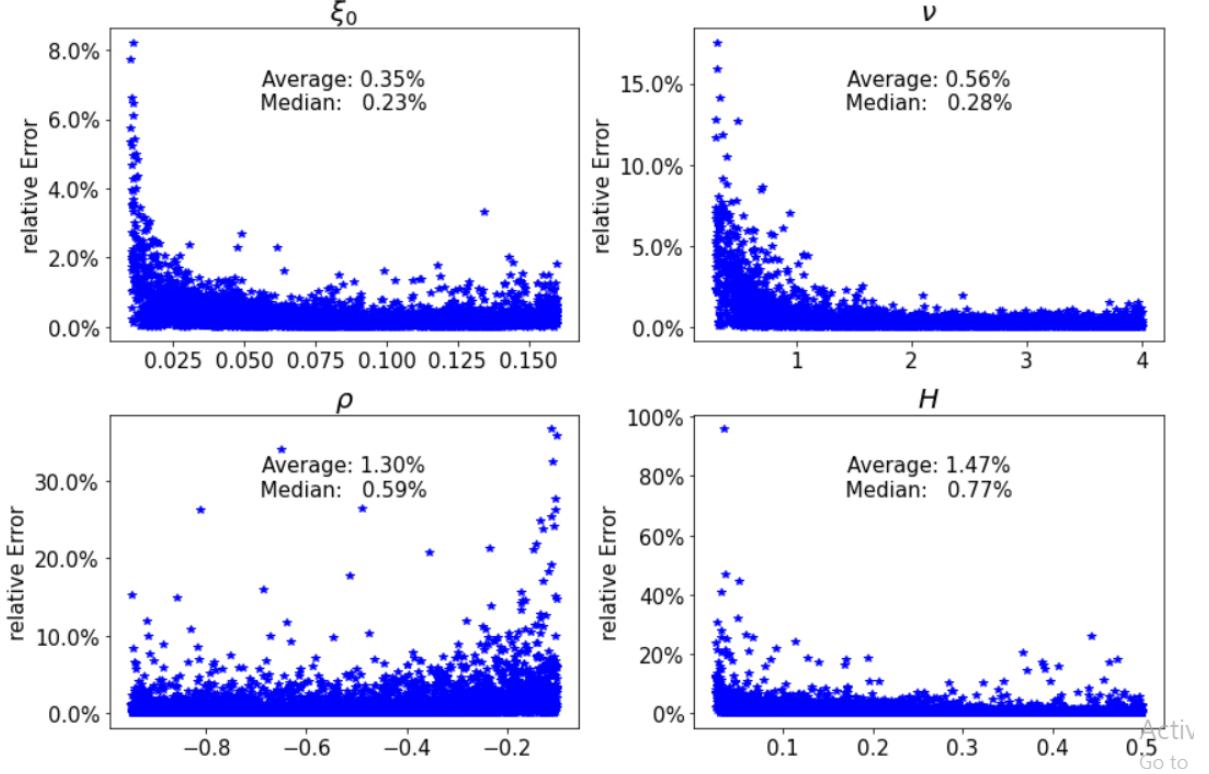


Figure 33: Relative parameter errors after calibration via LM in the rBergomi model.

Notice how the largest *relative* errors are concentrated around near zero for all parameters. This is to be expected. Imagine an *absolute* error of small but fixed value, say  $e^{-10}$  in the numerator (88). The *relative* error will increase in percentage terms the closer the true value (the denominator) gets to zero. Excluding these extremes, the error distribution seems rather even across all levels.

Moreover, we can look at the performance of the NN (Step 1) using the parameters from (Step 2)  $\tilde{F}(\hat{\theta})$  to fit to the market data  $\sigma_{BS}^{MKT}$  (here synthetic data) via the root mean square error (RMSE):

$$RMSE(\hat{\theta}) = \sqrt{\sum_{i=1}^n \sum_{j=1}^m \left( \tilde{F}(\hat{\theta})_{ij} - \sigma_{BS}^{MKT}(\tau_i, k_j) \right)^2}. \quad (89)$$

The performance of both, the parameter accuracy, and the RMSE is shown in Figure 34 across all quantiles. Strikingly, the RMSE in the 99.5% quantile is below 1%, and below 3.5% for the maximum RMSE, suggesting extraordinary accuracy of the neural network

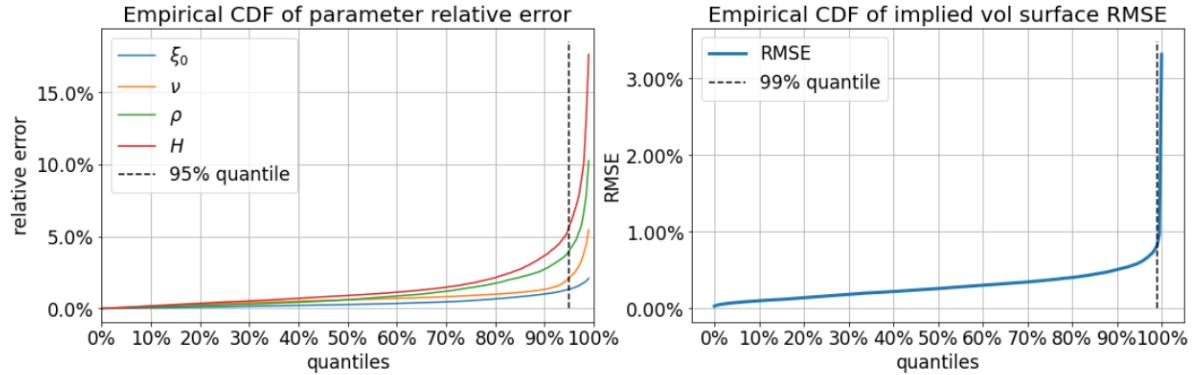


Figure 34: Showing the parameter relative errors for the rBergomi model, as well as the RMSE between approximated IVs and the true (synthetic) data

Interestingly, (Horvath, et al., 2019) also look into the evolution of model parameters under  $\mathbb{Q}$  shown in Figure 35. These estimates stem from their neural network calibrated to SPX data between 01.01.2010 through 18.03.2019. As we can see, also for implied volatility data the parameters vary quite significantly for a large asset such as the SPX. Comparing the data to what we show in Chapter 2 (Figure 9), one might say that  $\mathbb{P}$  provides a rather noisy model parameter estimate whereas under  $\mathbb{Q}$  the parameters seem to be confined to a more rigid interval. Nevertheless, also here parameters vary quite starkly through time.

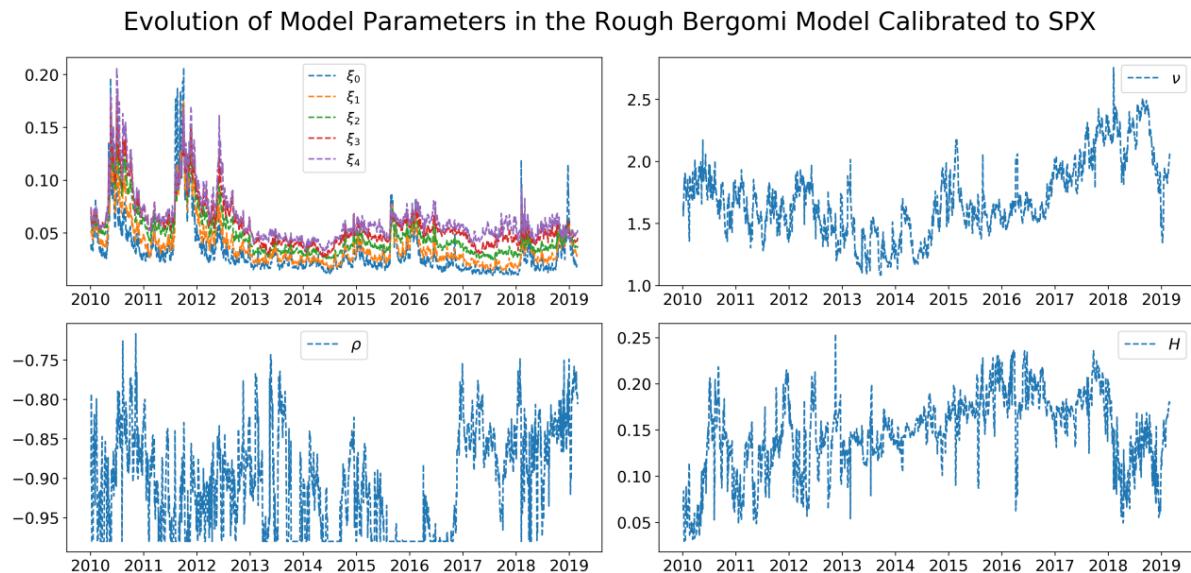


Figure 35: Evolution of model parameters for SPX under  $\mathbb{Q}$  using the rBergomi framework taken from (Horvath, et al., 2019, p 25)

We can thus say, while the construction of a neural network may be far from trivial and demands a lot of thought and diligence, its ability to virtually instantaneously compute outputs to complex problems render it very interesting from a practitioner's point of view. We have seen the power a neural network possesses and how it can be used to calibrate and price options in a theoretical environment, where it performs exceptionally well. However, in order to be able to make concrete statements we need to evaluate it in real world scenarios. The next section of this chapter is dedicated to such an endeavor.

### **Image based Implementation**

After having discussed the neural network approach to pricing in detail, we want to explore how such a network performs when exposed to actual data. Recall that the steps to use a grid based neural network for pricing are the following:

- 1) Train a neural network such that it can find model parameters corresponding to a given set of implied volatilities (Step 1)
- 2) Calibrate the neural network to observed option implied volatility data using standard calibration techniques. Thus, minimizing the difference between observed and neural network implied volatilities. This calibration results in a set of optimal (true) parameters, which the neural network uses to match the observed implied volatilities (Step 2).
- 3) Using these parameters, we are then able to find (price) any option contract, whether it exists or not using the MC simulation by (McCrickerd, Pakkanen, 2018), introduced above (Pricing Step).

Unfortunately, the network employed by (Horvath, et al., 2019) is not directly applicable to market data, as its training set is too restrictive and interpolation between gridlines is missing. (Rømer, Historical analysis of rough volatility models to the SPX market, 2020) has since addressed this and built a neural network based on the work of (Horvath, et al., 2019), using the image-based implementation of the two-step approach. (Rømer, 2020) developed his neural network for the purpose of calibrating to and pricing SPX options, which should exhibit lower volatility as compared to our other assets under consideration. As one of his findings he provides estimates for parameters through time, which can be found in Figure 67 and Figure 68 in Appendix E. As already indicated by (Horvath, et al., 2019, p 25), also (Rømer, 2020) finds that the parameters vary significantly through time.

To make his computations feasible, Rømer provides an extensive GitHub repository including code in Python, MATLAB, and R-Studio. Not only does he provide code on the rBergomi model, but also for (rough) Heston<sup>119</sup> and for the extended rough Bergomi model<sup>120</sup>: [https://github.com/sigurdroemer/rough\\_volatility](https://github.com/sigurdroemer/rough_volatility).

To incorporate the rBergomi model, (Rømer, 2020) chooses the following parameterization:

$$\begin{aligned} dS_t &= S_t(r(t) - q(t))dt + S_t\sqrt{V_t}dW_{2,t}, \\ V_t &= \xi_0(t)\mathcal{E}\left(\eta\sqrt{2H}\int_0^t(t-s)^{H-\frac{1}{2}}dW_{1,s}\right), \\ \xi_t(u) &= \mathbb{E}(V_u|\mathcal{F}_t), \\ dW_{1,t} dW_{2,t} &= \rho dt. \end{aligned} \tag{90}$$

Where  $dS_t$  gives the dynamics of the underlying in the risk neutral  $\mathbb{Q}$  world,  $V_t$  is the variance process,  $\xi_t(u)$  denotes the forward variance at time of observation  $t$ , and the (anti-) correlation between variance and spot price Brownian motions  $dW_{1,t}, dW_{2,t}$  is given by  $\rho$ . Lastly,  $r(t)$  and  $q(t)$  are the risk-free rate and the dividend yield respectively, and the remaining rBergomi model parameters are  $H \in [0, \frac{1}{2}]$  and  $\eta > 0$ . Generating MC price paths for the training set, (Rømer, 2020) relies on the hybrid scheme by (Bennedsen, et al, 2017) discussed above, and uses the estimator of option prices by (McCrickerd, Pakkanen, 2018), with 50.000 price paths. Importantly, (91) shows that it is possible to train the network on a spot price at  $t = 0$  of 1 and recover the spot price and log-strike by rescaling the spot price with the forward price in the pricing step.

$$F_{0,T} \exp(-y_r(0,T)T) \mathbb{E}(\max\{\beta \cdot (\tilde{S}_T - \tilde{K}), 0\}). \tag{91}$$

---

<sup>119</sup> See for different parameterizations of the rough Heston model the following papers  
El Euch, Gatheral, Rosenbaum, Roughening Heston. Risk, May 2019, pp. 84-89.

El Euch, Rosenbaum, Perfect hedging in rough Heston models. Ann. Appl. Probab., 2018, 28(6), 3813-3856.

El Euch, Rosenbaum, The characteristic function of rough Heston models. Math. Financ., 2019, 29(1), 3-38.

<sup>120</sup> The extended rough Bergomi model decouples explosion rates for curvature and skew of the implied volatility surface, s.t. the volatility process is now driven by two separate factors. (Rømer, 2020) decides on this model after finding that the rBergomi model (on average) allows for too little skew at longer dated expirations and also not enough curvature for short maturities. However, also the extension of the rBergomi model did not provide an adequate solution to the issue observed.

Here  $F_{0,T}$  denotes the Forward price observed at time  $t = 0$  with maturity at  $T$ ,  $y_r(t, T) = \frac{1}{T-t} \int_t^T r(s) ds$  accumulates the interest payment over the same time horizon and  $\tilde{S}_0 = 1$  with  $d\tilde{S}_t = \sqrt{V_t} dW_{2,t}$ . Lastly,  $\tilde{K} = \frac{K}{F_{0,T}}$  gives the adjusted strike and  $\beta$  is either 1 or -1 for call, put respectively.

Besides the advantages and disadvantages discussed above, (Rømer, 2020) notes that constructing a NN with a fixed maturity strike combination  $\Delta = \{\tau_j, k_i\}_{i=1, j=1}^{n, m}$  (a Cartesian product) comes with its own set of issues.

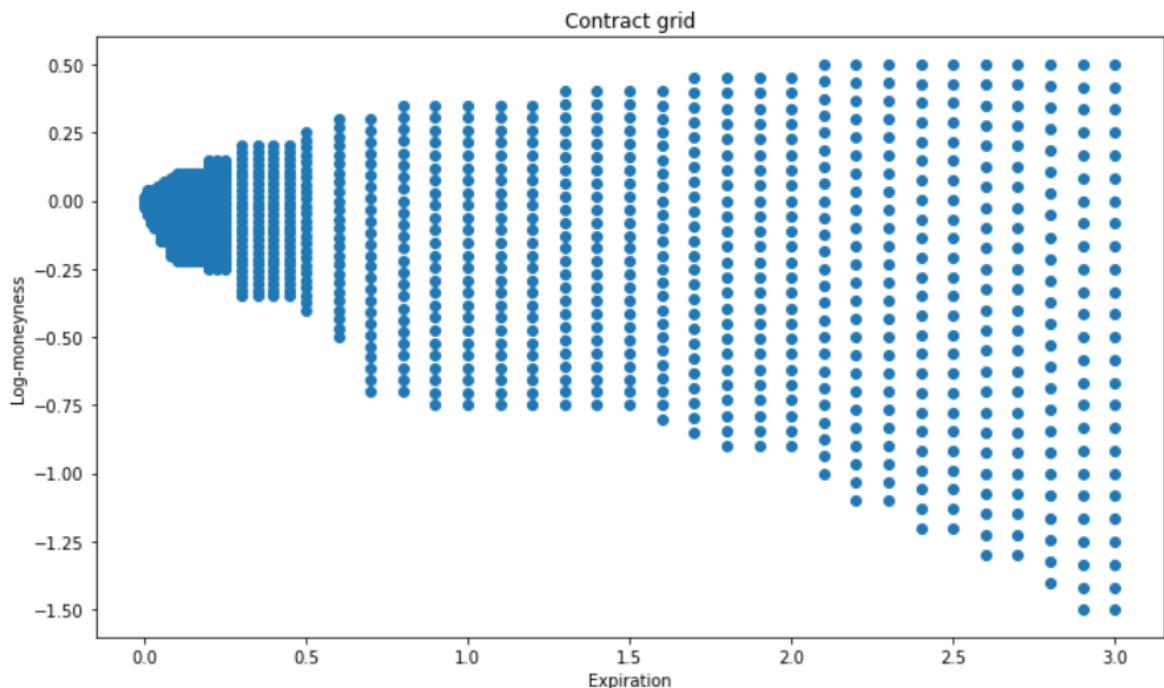


Figure 36: Showing the strike/maturity grid used in the deep neural network by (Rømer, 2020)

Building a synthetic training set using MC simulation and including far ITM or OTM contracts may lead to pricing errors within the training set. This is because such options may not be accurately estimated when not enough price paths exist for those contracts. As a remedy, the author restricts the NN to certain bounds, shown in Figure 36. We can see how the grid becomes narrower around ATM contracts the closer we get to maturity. All contracts that lie outside of this grid are excluded, and contracts are interpolated by natural cubic splines between the blue dots when lying within the grid bounds. As suggested by (McGhee, An Artificial Neural Network Representation of the SABR Stochastic Volatility Model, 2018), (Rømer, 2020) splits the entire IV surface into six different groups and trains them separately. While this helps to maintain a network of reasonable size, it also helps to improve approximation as maturities of short and longer term behave quite differently, as he remarks. The DNN by (Rømer, 2020) uses

the following architecture for each of the six sets of expirations: The four rBergomi parameters  $\theta = (H, \eta, \rho, \xi_{t=1\dots27})$  are consumed by the input layer with 30 neurons<sup>121</sup>. The following three hidden layers are constructed with 200 neurons each, while the output layer returns between 150 to 500 implied volatilities  $\sigma_{BS}$ , depending on the set of expiration<sup>122</sup>. The activation function  $\sigma_{Elu} = \alpha(e^x - 1)$  is the same as used by (Horvath, et al., 2019). The subsequent calibration is done using the L-BFGS-B method, see Figure 30.

Once the network is trained and the weights are loaded (step one), it is ready to be used to calibrate to options and return the set of estimated parameters  $\hat{\theta}$  (step two), following the below procedure:

1. Evaluate the observed option data (including all available option chains) with respect to the gridlines in Figure 36 and discard any data that lies outside this pre-defined grid.
2. Define a *best guess* of parameters  $\theta$  (a starting point) for the calibration (minimization) of distance function  $\delta$ , see (84).
3. Lastly, the observed option data (strike and maturity) that falls within the grid of the neural network in point 1., is loaded into the objective function (84) of the calibration.
4. The calibration (step two) returns an (approximated) fit to the observed implied volatility surface, achieved for a (theoretically<sup>123</sup>) best set of parameters  $\hat{\theta}$ .

As Rømer notes, the pricing algorithm of his code is only scripted in MATLAB, using the option pricing estimation by (McCrickerd, Pakkanen, 2018). However, since we have the needed code already available in Python from the original authors, we can safely rely on our Python environment. Thus, we are in theory able to price the full option implied volatility surface using the parameters  $\hat{\theta}$  resulting from Rømer’s DNN, akin to what we showed in the previous section of this chapter.

---

<sup>121</sup> One for each of the first three parameters, and 27 for the forward variance  $\xi$ , at different times

<sup>122</sup> With the neural network of such size, Rømer reports one week of training time for an Intel Xeon Platinum 8175 3.1 GHz server CPU (96 logical cores) with 384 GB RAM. As such, the neural network may not be as easily trainable as initially suggested by (Bayer et al, 2019) after all.

<sup>123</sup> We use the term theoretically here to exemplify the point that the “best” set of parameters the calibration can find is directly influenced by how “good” (generalizable) the training data is.

## Calibrating to an observed Option Chain

As the pricing is pretty straight forward as we have shown in the previous chapter using the turbo charged MC simulation, here we restrict our focus to the calibration problem. By means of the neural network devised by Rømer, we will calibrate to the option chain shown in Appendix A1, together with all other maturities observed at that date (25-06-2021) for the same symbol AAPL, see Table 4. In order to use the neural network with our data, only simple changes are necessary:

As previously, we extract the spot price at time of observation, compute the mid-price, the log-strike (68) and the time to expiration. We also use the same methodology w.r.t. cleaning the data as introduced in (76) and (78). However, before we continue we need to adjust the log-strike (68) by the forward price (73) as shown above where  $\tilde{K} = \frac{K}{F_{0,T}}$ , to account for the adjusted strike in (91). With these changes we feed Rømer's neural network with the array of (adjusted) log strikes and the corresponding array of expirations in years. As the last step before calibration, the neural network will discard all contracts that do not lie within the grid (compare Figure 36).

Original number of observations: 492  
After filtering: 341

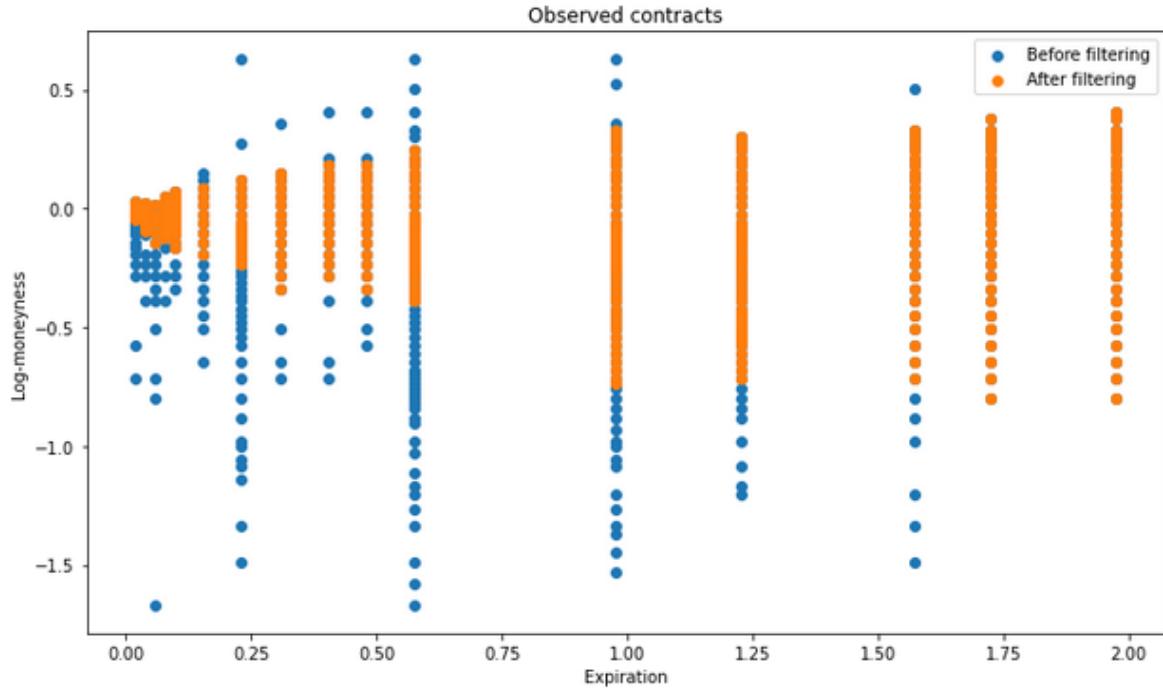


Figure 37: Filtering option contracts. Original data in blue, orange indicates option contracts that the NN can be calibrated to

Figure 37 then shows the grid exposed to the data. We can observe that of 492, only 341 contracts remain in the dataset, which are a “*reasonable*” distance from at-the-moneyness. To get started, we calibrate only to two single option chains independently (via L-BFGS-B), showcasing best case scenario in Figure 38. We can observe that for both option chains farthest from maturity (0.019 and 1.975) the neural network is able to provide an extraordinary fit to the observed data. This situation changes however when we calibrate the neural network to the full data available across all option chains.

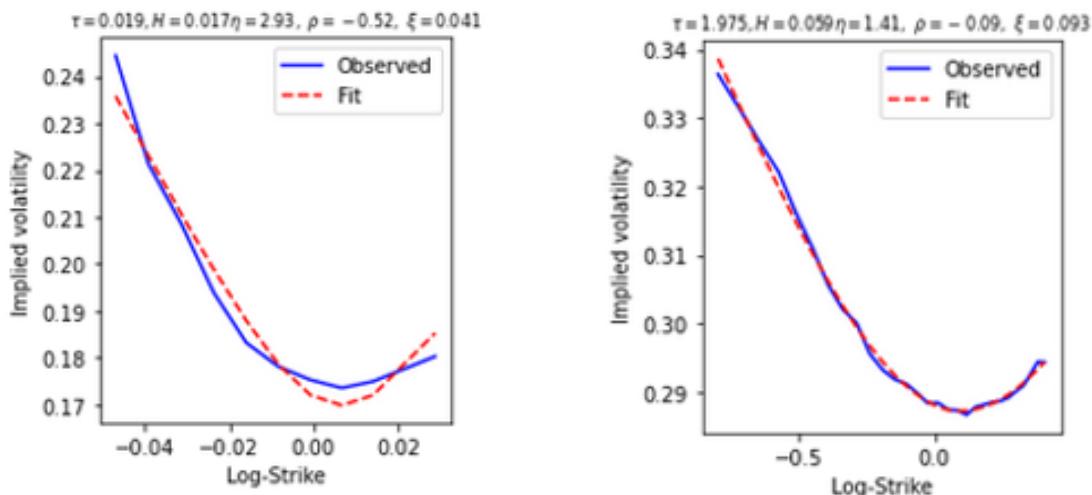


Figure 38: Option contracts for AAPL on 25-06-2021 for shortest (0.019) and longest (1.975) maturity in years

Now with the calibration task more difficult, we can see also the neural network is struggling to fit to the observed data in Figure 39. Especially for close-to- and long-to-expiration contracts we see the level of the “volatility smirk” to be the culprit. For maturities between both extremes the fit however is quite decent. The resulting MSE (Table 7) then paints the complete picture:

Calibration via	MSE (Optimization Slice)	MSE (Full Surface)	Runtime	Parameters ( $\alpha, \eta, \rho, \xi$ )
<b>Nelder Mead</b> (Monte Carlo)	0.003932540442	0.003014108827	84:57:20 Minutes	-0.49999999 0.83515963 -0.93144967 0.06974231
<b>Trust Region Reflective</b> (Monte Carlo)	0.158454507788	0.090654455106	16:51:31 Minutes	-0.41436948 1.70114322 -0.16172928 0.07606519
<b>L-BFGS-B</b> (Grid-based Neural Network)	-	0.216495265787	00:02:64 Minutes	-0.499999 3.42118087 -0.999999000 0.0992848866

Table 7: Neural Network fit vs traditional MC calibration

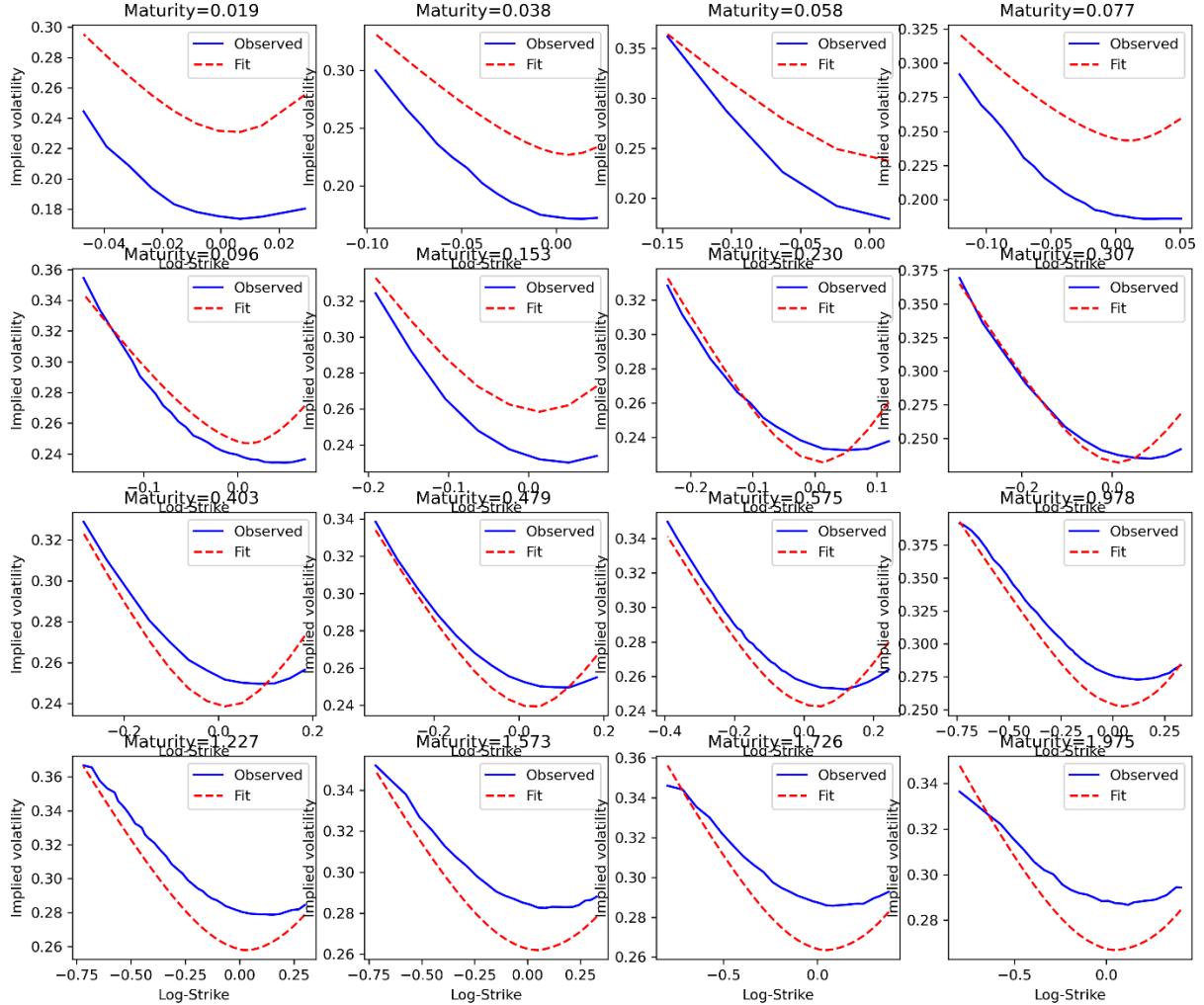


Figure 39: Neural network calibration to all available 16 option chains for AAPL, observed on 25-06-2021

While the MSE is the highest we have witnessed thus far, showing a subpar fit compared to more traditional MC simulation, at this point we also want to recall that the training data and the neural network by (Rømer, 2020) overall were built with pricing SPX options in mind, which should exhibit lower levels of volatility than AAPL. Nevertheless, we can see that the curvature and skew are well replicated by the fit. Only the height of the IV curves seems off.

As we remember the forward variance  $\xi_t(u)$  controls the level of the smile for a given expiration. Up until now we have dealt with the forward variance as if it was a constant number. However in practice, forward variance can be derived from variance swap curves (see Figure 62 in Appendix C1), which (usually) aren't flat and thus also the forward variance curve should be changing through time. The rBergomi model accounts for that by allowing forward variances to vary. Rømer however decides to fix the forward variance curve over 27 time-steps to a single number, which according to him prevents an overparameterized model. This bears the result we observe here. Rømer summarizes this as a structural problem in that the rBergomi

model fails to produce enough curvature at short maturities and not enough skew for long expirations. Without the ability to fit each expiration by its own forward variance, naturally the level for each maturity cannot be controlled and is therefore “off” by some degree. With this being said, the speed at which the DNN is able to calibrate is remarkable. After only 2.64 seconds the neural network had calibrated to all available option chains. With such rapid calibration, it may be a feasible endeavor to calibrate to the entire surface by each option chain independently without sacrificing too much speed and hopefully improving fit.

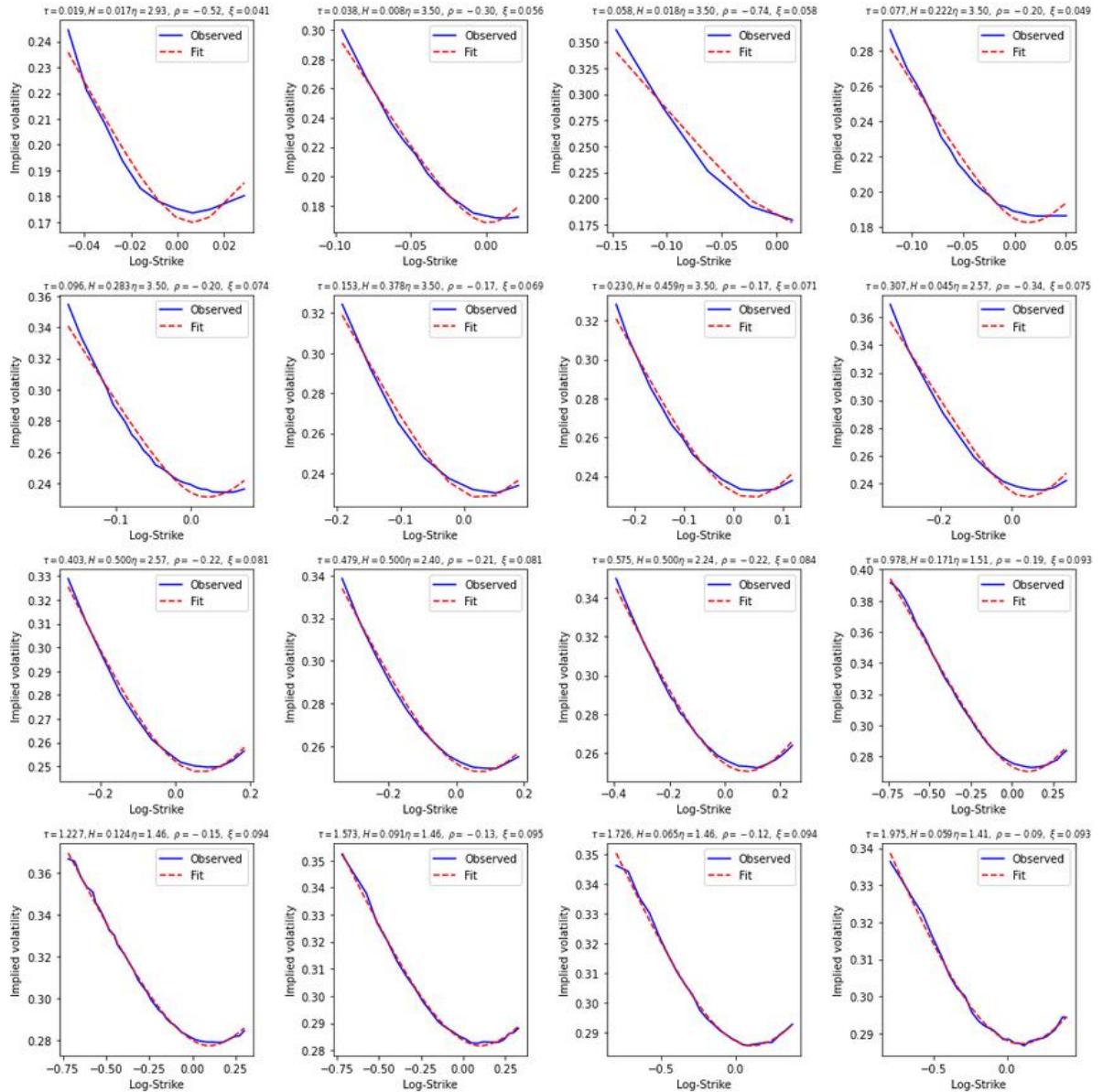


Figure 40: Neural Network calibration to every smile independently. Parameters are reported for each chain separately

Indeed, as can be observed from Figure 40, with only slight changes to the neural network, we managed to calibrate to every option chain individually and obtained an extraordinary fit as the network managed to replicate the observed mid-implied volatilities almost perfectly.

Calibration via	MSE (Full Surface)	Runtime
Nelder Mead (MC)	0.003014108827	84:57:20 Minutes
Trust Region Reflective (MC)	0.090654455106	16:51:31 Minutes
L-BFGS-B (Neural Network)	0.003218926366	00:09:58 Minutes

Table 8: Neural Network maturity-wise calibration vs calibration by MC simulation

Table 8 then highlights the performance in comparison to the other calibration techniques, shown in the previous chapter. While its MSE is surprisingly 0.0002 points higher than what we achieved with the Nelder Mead calibration<sup>124</sup>, the neural network is faster by a factor of over 880 times<sup>125</sup>, which might just quench the “thirst for speed” of practitioners. Managing such level of fit shows how the rBergomi may be used in a productive environment. While the Nelder-Mead calibration showed it is potentially able to achieve extraordinary fits as well, its long calibration time necessitated by slow MC simulation render it only a second choice after novel neural network approaches.

Having identified a flat forward variance curve as the main culprit for a lackluster fit we attempt at improving calibration by estimating what the forward variance might be. In the Appendix E2 we experimented with extracting the  $\xi_t(u)$  estimates from our independent calibration result shown in Figure 40. Therefore, we had to spread the 16  $\xi_t(u)$  estimates across a forward variance length of 27 time points, fixed by the neural net<sup>126</sup>. Unfortunately, the fit did not improve *at all* with either averaging the other parameters  $H, \eta, \rho$  across the 16 maturities, nor with simply taking our parameters from the first calibration exercise with fixed forward variance. Inspired by (Horvath, et al., 2019) who have incorporated a piece-wise forward variance curve to be calibrated as part of their (dummy) neural network, we attempted at calibrating to the entire forward variance curve. To do this we re-engineered Rømer’s neural network and split the calibration according to the following three-step approach: Firstly,

---

<sup>124</sup> Note that both methods here are not directly comparable as the neural network applies additional filtering as shown, while the Nelder-Mead calibration does not.

<sup>125</sup> Also, the code used here was not optimized for speed.

<sup>126</sup> Curiously Rømer mis-specifies these time points as his model only defines 24 such points, however demands 27 points to be given as input. His error stems from a simple coding mistake where the `numpy.arange()` function does not include the endpoint. See file “NeuralNetworkPricing.py” in line 58 through 62

calibrate to all observed maturities (i.e., the entire volatility surface) with a fixed forward variance curve. Next take the estimated parameters  $(H, \eta, \rho)$  as fixed and let the forward variance curve be floating, i.e., freely calibrated. With 27 calibration points the Neural Network took roughly one full minute for the second calibration step alone. Lastly joining all 30 parameter estimations together gives us our end result, observable from Figure 41 and summarized in Table 9:

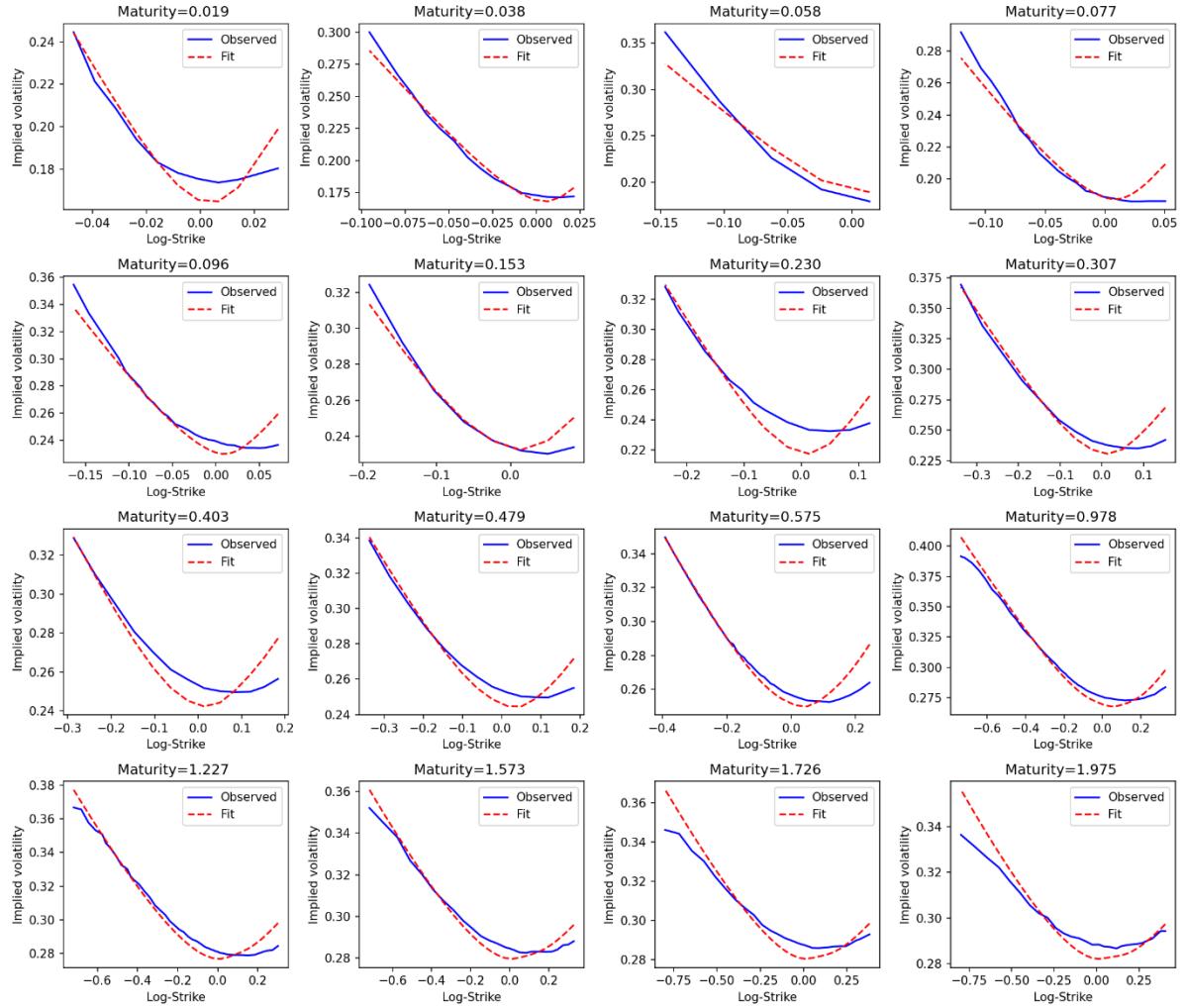


Figure 41: Neural Network fit with floating Forward Variance Curve to entire implied volatility surface

Neural Network Calibration	MSE (Full Surface)	Runtime
Calibration to whole surface	0.216495265787	00:02:64 Minutes
Calibration to individual maturities	0.003218926366	00:09:58 Minutes
Calibration with floating forward variance	0.017133495479	01:03:66 Minutes

Table 9: Comparing MSE of Neural Network with different calibration techniques

Across the entirety of the surface, we observe a good level of fit, somewhere in-between the standard calibration result and with calibration to each maturity individually. There seems to

be no systemic difference in fit between short nor long maturities. For only a few select expirations we see that the fit deviates somewhat from the observed data, most notably the model produces too much curvature and not enough volatility bias. However, this may simply be due to the interpolation errors or not granular enough training data. Overall, the fit is still best when calibrating to each maturity individually, however for evaluation of the power of the rBergomi model or when we require these parameters estimated in the risk neural ( $\mathbb{Q}$ ) world a respectable degree of accuracy can be achieved in a relatively short time when letting the forward variance be floating.

With this, we can say that at least for AAPL on the 25.06.2021 the rBergomi model is able to fit (with exceptions) to the entire volatility surface using the neural network. We can also disregard Rømer's critique of producing too little curvature at the extreme endpoints of the surface. His observation was merely induced by his decision to keep the forward variance curve fixed through time, which is not only at odds with the rBergomi model specifications but also with empirical data. The forward variance curve we estimated under  $\mathbb{Q}$  is shown in Figure 42. It tends to increase in time to maturity with quite large discrepancies between the endpoints. This alone exemplifies why the forward variance shouldn't be fixed as in Rømer's work.

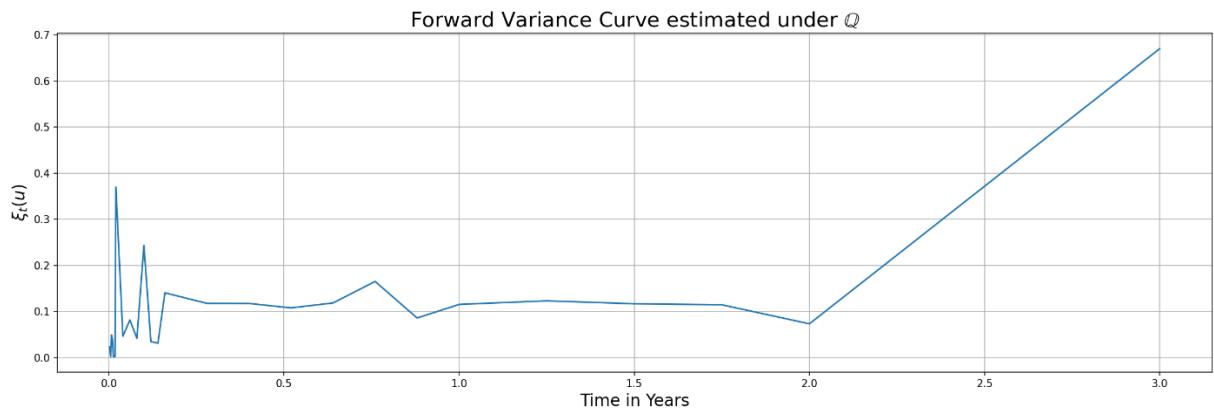


Figure 42: Forward Variance Curve estimation under  $\mathbb{Q}$  through time

Lastly, we also explored in Appendix E3 how both MC and NN approaches map parameters to corresponding IVs. This is important s.t. we may assess whether both methods were properly implemented or are prone to mapping to erroneous IVs. We can report that the MC simulation and neural network coincide, if we allow for at least 50.000 Monte Carlo simulation paths<sup>127</sup> for each maturity, as can be observed from Figure 72.

---

<sup>127</sup> In effect that means the 20.000 paths we used prior when exploring the MC method were too few. However, this is only relevant in comparison to the neural net. In itself the MC method is still correctly specified as

We can thus conclude this chapter with the preliminary finding that the rBergomi model is in fact able to fit to an entire implied volatility surface as we have seen from the Nelder Mead calibration example in the previous chapter and also utilizing the novel approach of neural networks. While the latter method is still in its infancy, we have seen the incredible power it possesses when calibrating to both single maturities and entire surfaces. In productive environments, where speed and accuracy are of upmost importance, we believe that the individual calibration to each single expiration is the best application of the neural network. Should it become necessary to price an option contract between these observed maturities, the literature has already come up with various interpolation techniques, be it by cubic splines, Chebyshev interpolation or stochastic implied volatility (SVI)<sup>128</sup>. For academia or non-option specific applications calibrating to the entire surface may be worthwhile when speed is not a critical factor as calibration to the entire forward variance curve is computationally expensive, even for neural networks. Letting all parameters (not only the forward variance) calibrate freely is still the most accurate method of matching implied volatilities. Horvath and Rømer have shown how parameters fluctuate through time, and also we have found how parameters change. We believe this does not constitute a misspecification of the rBergomi model but exemplifies that each expiration carries an expectation of a future volatility regime, influencing the roughness (Hurst parameter), the correlation between log returns and log volatility ( $\rho$ ), and also the volatility of volatility ( $v$ ). Therefore, these parameters will vary between different time points and as such they may be different for every expiration. The rBergomi model has the power to extract these expectations and help understand what the market is assuming about the future.

However, it is not ideal that different points in time have different parameters from an academic standpoint: Instead, we seek a model that is able to explain the difference in volatility regimes through time, i.e., a model that is independent of time. While the model may be used to extract market expectations relating to different volatility regimes at different points in time, it is not able to explain where this difference comes from. Nevertheless, the fit we managed to achieve across different asset classes is exceptionally good when comparing to standard models such as Heston.

---

calibration and pricing were done with the same number of paths. Problems may only arise when using the parameters for other models and estimations, e.g., volatility and variance swap curve estimations.

<sup>128</sup> See (Gatheral, Jacquier, Arbitrage-free SVI volatility surfaces, 2013)

# Empirical Evaluation

In this chapter we explore,

- a) The ability of the rBergomi model to fit to several entire implied volatility surfaces
- b) How persistent the rBergomi model parameters are through time
- c) How parameters estimated by time series analysis ( $\mathbb{P}$ ) and estimated by option chain calibration ( $\mathbb{Q}$ ) compare.

The previous two chapters examined the different calibration techniques s.t. we gain an understanding how the rBergomi may be evaluated in a practical context. Here we apply this knowledge to more data points to see whether our preliminary conclusions hold up under more scrutiny.

## Methodology

From our previous exercises we learned that the Nelder-Mead calibration and the neural network coincide when using the same parameters, see Appendix E3. As such we see no reason to calibrate by the slower Monte Carlo simulation, as the neural network is able to fit an entire surface quite well when allowing the forward variance to be calibrated freely. However, the constrained training of the neural network forces us to use the MC simulation for point b) and c), as we observe positive correlation in Table 3, which the NN is not able to calibrate to, and the constrained bounds of the NN make evaluation trickier. For the general procedure here, we follow the steps outlined in Chapter 4 and 5: Starting by cleaning the data via functions (76) and (78), we compute the dividend payout ratio (75), the total dividends that occur until maturity (74), and extract the forward price (73) for all expirations. Next, we will filter the data according to the bounds of the neural network (Figure 36) and apply the forward price to the resulting construct of the NN via (91). As the neural network returns implied volatilities (within its pre-defined grid) for an input of 3 rBergomi parameters and 27 forward variances for a total of 30 inputs, it calibrates to the observed implied volatilities via the L-BFGS-B minimization technique. We will use the 3-step calibration approach as explored earlier: After calibration to the whole surface with the NN and a fixed forward variance, we extract all parameters but  $\xi_t(u)$ , next we supply these parameters as fixed to the calibration yet again to calibrate to the entire forward variance curve. Combining both sets of parameters will then constitute our estimation of the true rBergomi parameters.

Company / Asset Name	Stock Symbol
Apple	AAPL
AT&T	T
Gamestop	GME
Microsoft	MSFT
Nio	NIO
S&P 500	SPX
Tesla	TSLA

In point a) and c) we will evaluate the fit on the trading day of 5<sup>th</sup> of August 2021. This is, s.t. we may gain an understanding how parameters estimated under the physical probability measure ( $\mathbb{P}$ ) and the risk neutral measure ( $\mathbb{Q}$ ) compare.

### Assessment of Fit to the Implied Volatility Surface

In this section we are going to evaluate whether or not the rBergomi model with a floating forward variance curve is able to fit to the entire implied volatility surface for the above assets, observed on the 5<sup>th</sup> of August 2021.

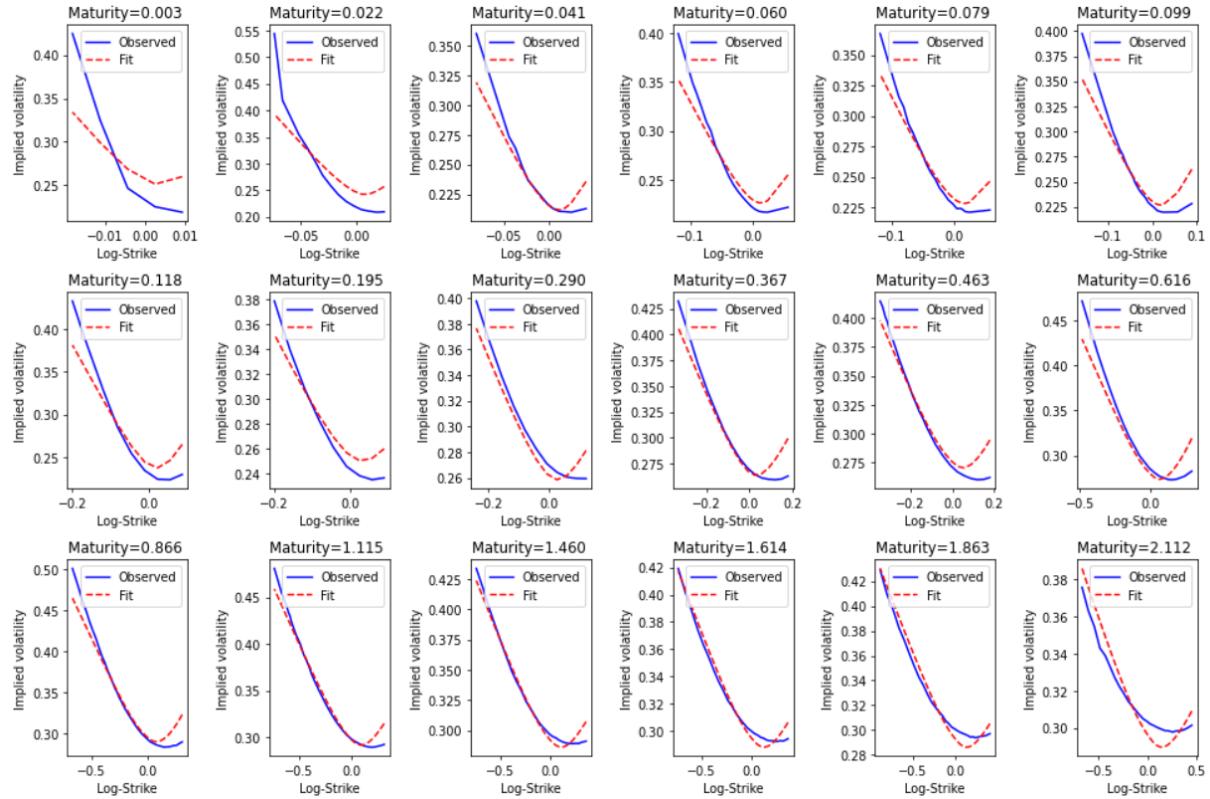


Figure 43: rBergomi fit to the AAPL volatility surface as observed on the 05.08.2021

We chose this date as we observe a relatively low incidence of volatility in the market, and the option chain data for that day is also of relatively decent quality. Here we will collect the visual representations of the surfaces for each respective asset, before evaluating the fit at the end of this section.

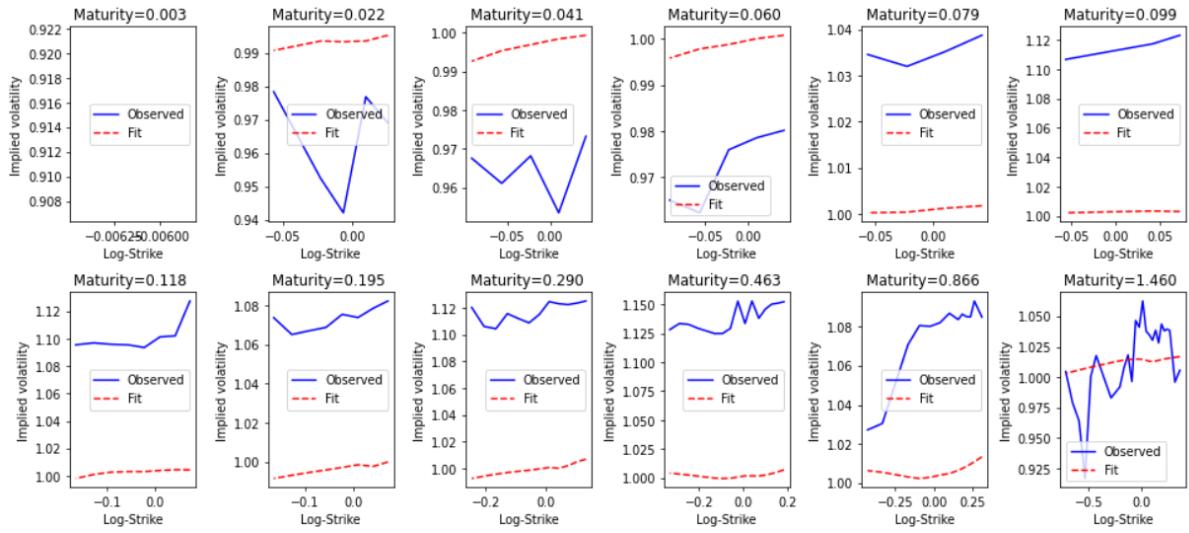


Figure 44: rBergomi fit to the GME volatility surface as observed on the 05.08.2021

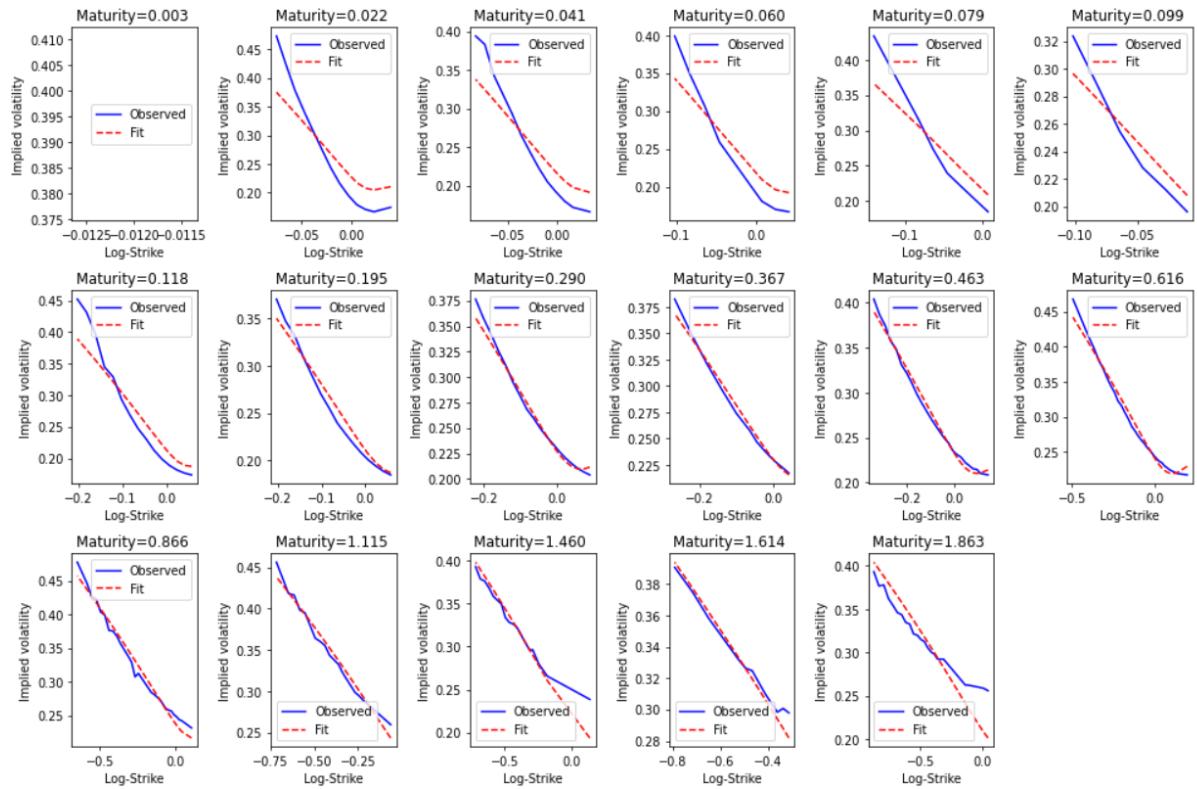


Figure 45: rBergomi fit to the MSFT volatility surface as observed on the 05.08.2021

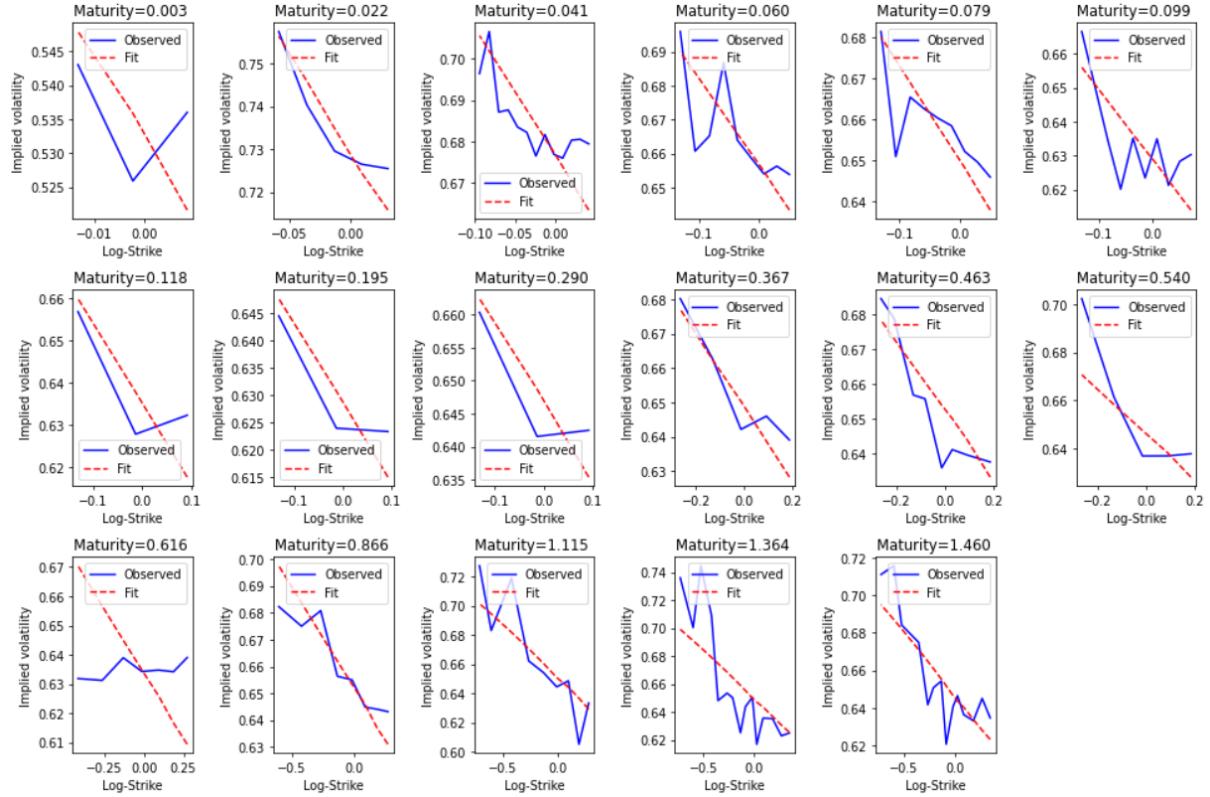
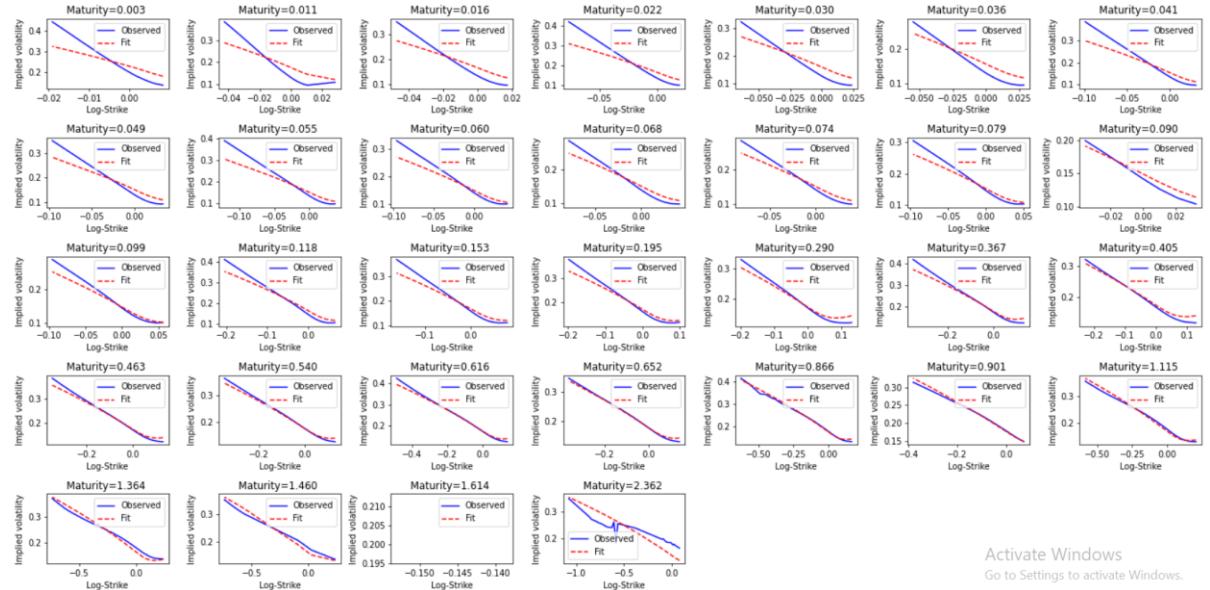
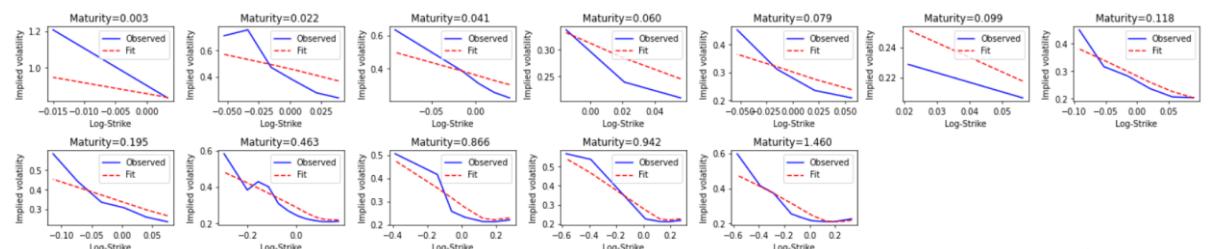


Figure 46: rBergomi fit to the NIO volatility surface as observed on the 05.08.2021



Activate Windows  
Go to Settings to activate Windows.

Figure 47: rBergomi fit to the SPX volatility surface as observed on the 05.08.2021



Activate Windows

Figure 48: rBergomi fit to the T volatility surface as observed on the 05.08.2021

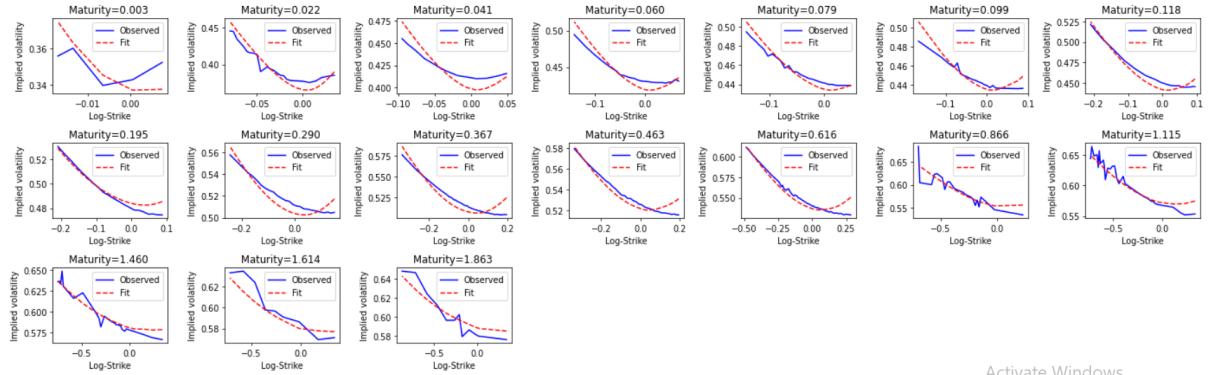


Figure 49: rBergomi fit to the TSLA volatility surface as observed on the 05.08.2021

Activate Windows

From Figure 43 to Figure 49 we see a very diverse picture of the rBergomi model when calibrating to different assets, each with their own volatility profile. Unfortunately, the data quality for GME, NIO, and T was so poor that we had to allow a bid and ask spread of 50% relative to mid prices (76) to retain any usable IVs for calibration. While for GME and NIO this may be explained by extreme levels of volatility of the underlying, this explanation surely does not hold up for the latter asset (T), which is known to be usually trading flat. While the data is close to being unusable for GME, we observe that the forward variance  $\xi_t(u)$  of the calibration fit is not able to go beyond 1.0. This highlights one of the constraints when using neural networks as they are only able to assess inputs within its trained bounds. Had the network allowed for greater values of  $\xi_t(u)$ , the fit could surely be improved. This is something that one may explore by MC simulation further. Obviously, data quality is a great concern with both assets, NIO, and T, signified by a lackluster fit<sup>129</sup>. Going down the “volatility stack” we see a *decent enough* fit for TSLA, if we ignore the IV spikes. This gives us hope that high volatility equity (with improved data quality) can be priced using the rBergomi model. While also TSLA obviously suffers from some data quality issues, they are far less severe than what we observed for the former three assets just discussed. MSFT then calibrates even better, if we ignore the longest maturity. For AAPL and SPX we see the fit being the best. However, both are not without flaws either. Especially short-dated expirations seem to suffer from too little curvature in the far ITM end. Judging the general performance of the rough Bergomi model across all assets we can make a few concluding remarks: When data quality is not of great concern, the model manages to achieve a good enough fit for most maturities when using fixed rBergomi parameters with a floating forward variance curve. For some expirations we see too

<sup>129</sup> Nevertheless, calibrating to each maturity individually, managed to achieve a decent calibration result, highlighting the versatility of both the rBergomi model and the neural network by Rømer.

much curvature in the OTM options. Moreover, we see that oftentimes the model is not able to produce enough volatility bias. As critiqued by (Rømer, 2020) we make the same observations here that the rBergomi model produces too little curvature when ITM option contracts become close to expiration. This is most prominent for AAPL, MSFT, and SPX data as we see from the respective figures. However the same cannot be said for curvature in long maturities, if anything we observe too much curvature for increasingly OTM options. The caveats that we must mention here are that these results only hold for the data under consideration. Improved data quality may shed some additional light on the ability of the rBergomi model to fit to observed option data. Also, we must remind the reader that this neural network is constrained in its training set, s.t. some parameter combinations may be unknown to the network, thus this exercise here should be representative of the worst-case scenario for the rBergomi model. Greater in-depth analysis, using MC simulation, may shed additional light on the issue.

### Persistence of rBergomi Parameters

In this section we investigate whether the rBergomi parameters hold any predictive value. We assess this quality by following one maturity for SPX options through time until maturity on the 17.09.2021.

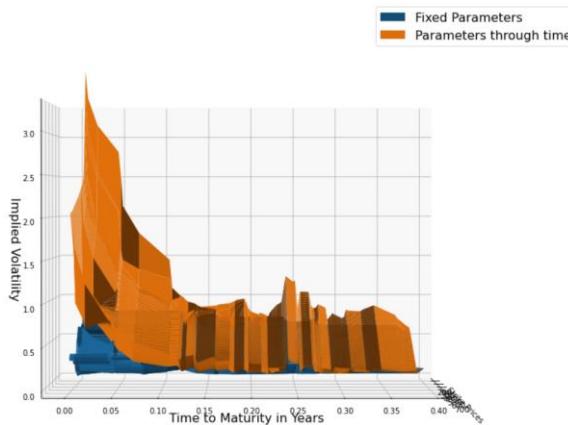


Figure 50: Tracking maturity 'September 17, 2021' for SPX through time. Orange: true volatility term structure. Blue: Estimated term structure

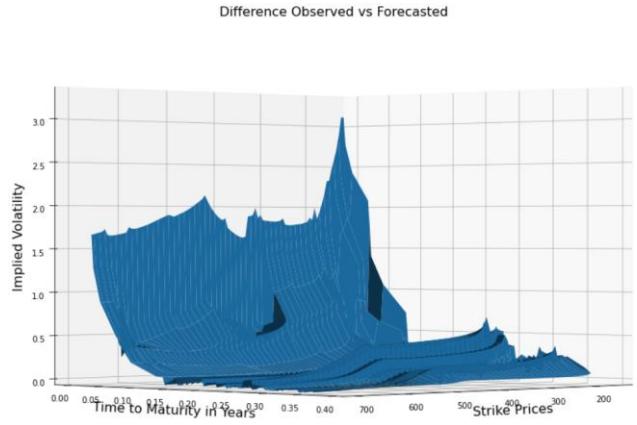


Figure 51: Tracking maturity 'September 17, 2021' for SPX through time. Difference between observed and predicted IVs

Starting from the 29.04.2021, we estimate the fixed rBergomi parameters on this day with help of the Neural Network. We also estimate the forward variance curve  $\xi_t(u)$  under  $\mathbb{Q}$ , via the steps outlined in the previous chapter<sup>130</sup>, such as to incorporate variance expectations of the future<sup>131</sup>. We will then only change  $\tau$ , the time to expiration, at every time step the closer we

<sup>130</sup> Namely, we observed the whole volatility surface at this point in time and extracted the corresponding forward variance curve.

<sup>131</sup> In order to make this endeavor practical, we use the forward variance curve and estimated under  $\mathbb{Q}$  and interpolate it over 1000 time-steps with the python scipy package. This much increased granularity (over Romer's

move to maturity. This constitutes our prediction for the true observed implied volatilities through time. For observed data, unfortunately the option data quality was too poor to allow for accurately depicting the observations graphically. As such, we used the neural network to estimate the parameters at every step, which can be highly accurate as shown in Chapter 5. We then kept the strike range fixed and used the parameters to estimate the implied volatilities for every time step within the fixed strike range. The result can be observed from Figure 50 and Figure 51. We can immediately see that the parameters of the rBergomi model here fail to capture the term structure of volatility for close to expiration IVs. While the prediction is not far off from the true data for the first couple of months, beginning with  $\tau \sim 0.13$  the observed IVs start to deviate strongly from the estimated IVs. An interesting observation is that OTM options exhibit far less error in prediction than ITM options for the same time step. This could potentially be a saving grace for institutions that prefer to write OTM call options over ITM puts, as they are thus able to assess their risk for the foreseeable future more accurately. However once again we need to mention the subpar quality in data. Not only did we have to use the rBergomi model to estimate the observed IVs, we also had to use linear interpolation for the surface, highlighting some potential erroneous reported data by yahoo finance. In fact we noticed large discrepancies between yahoo reported IVs and calculated ones that could not simply be explained by *old* quotes or trades. This makes us less confident in the reported data. Further analyses on different maturities may confirm our observations or find other interesting facts hidden in the data. Also observing an entire volatility surface through time may be feasible with better quality data.

### Historical Time Series Implied Parameters

Here we explore what contribution, if any, time series estimated parameters bring in the rBergomi model. We recall Table 3 from Chapter 1, in which we estimated H, nu, and rho. Here we need to translate H into alpha via  $\alpha = H - 0.5$  and nu into rho via (43), see Table 10.

	<b>alpha_est</b>	<b>eta_est</b>	<b>rho_est</b>
AAPL	-0.389427	0.933545	-0.127195
GME	-0.407374	0.993994	0.128399
MSFT	-0.353681	0.873635	-0.091008
NIO	-0.460772	0.950223	0.153142
SPY	-0.329716	0.974130	-0.120013
T	-0.389353	0.920411	-0.089833
TSLA	-0.410002	0.961631	-0.009374

Table 10: Time Series estimated Parameters for alpha, eta, and rho

---

27 time-steps) then allows us to find the predicted forward variance more accurately throughout our forecasting period.

Unfortunately, we do not have corresponding estimates for the forward variance curve  $\xi_t(u)$ . In theory we could derive them from the Variance Swap curve, which the rBergomi model is able to estimate see Figure 62, however we do not have data to verify these estimations, as such we will simply focus on the decay and smile dynamics and calibrate to the forward variance curve (the best possible case). We are specifically interested in whether the estimations for parameters under the physical probability measure<sup>132</sup> ( $\mathbb{P}$ ) carry any intrinsic information that are of value in option pricing or in general volatility estimations of the underlying. As already mentioned, we will rely on the MC estimation method by (McCrickerd, Pakkanen, 2018)

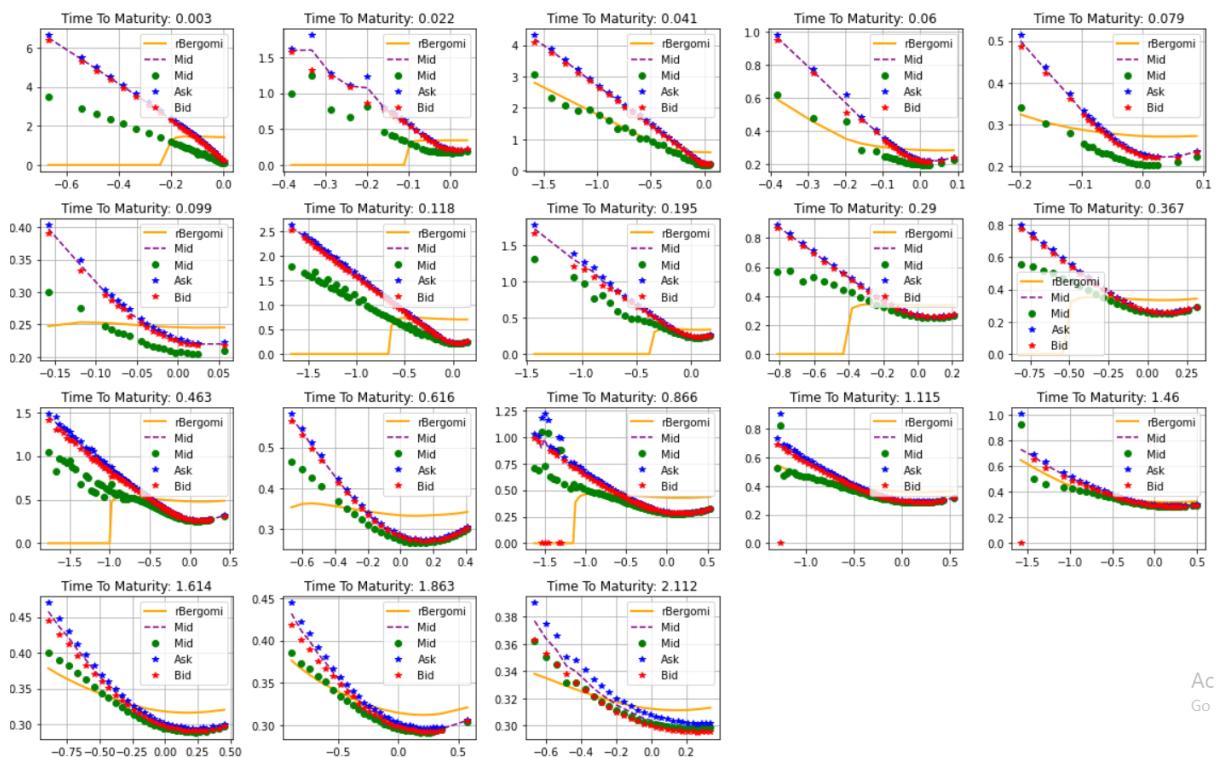


Figure 52: rBergomi fit to the AAPL volatility surface with time series implied parameters as observed on the 05.08.2021

<sup>132</sup> Which correspond to time series analysis estimation.

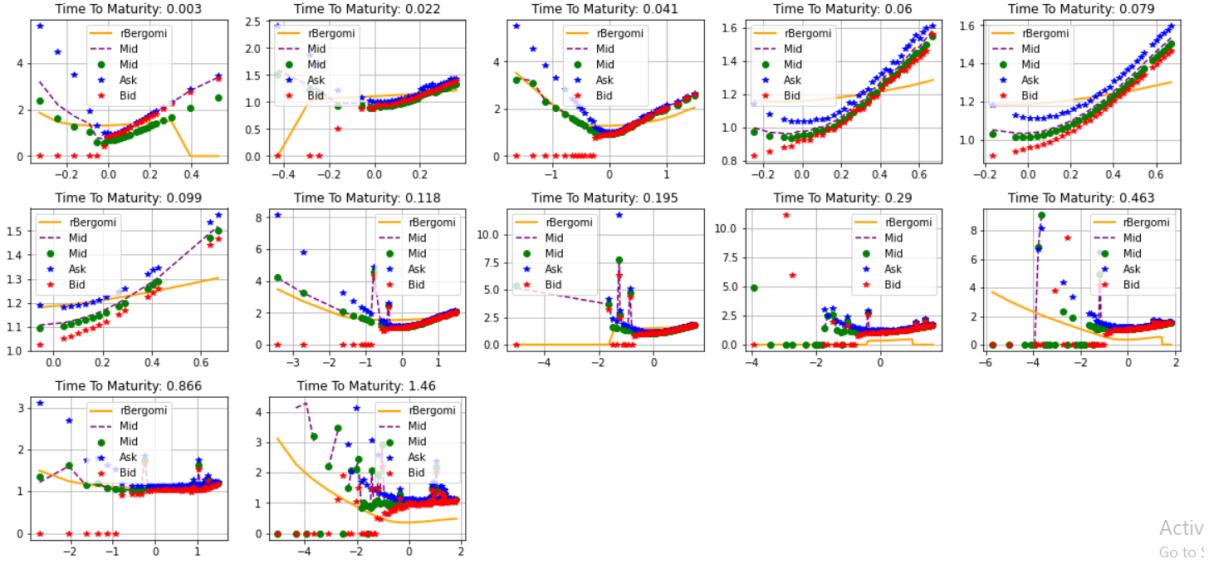


Figure 53: rBergomi fit to the GME volatility surface with time series implied parameters as observed on the 05.08.2021

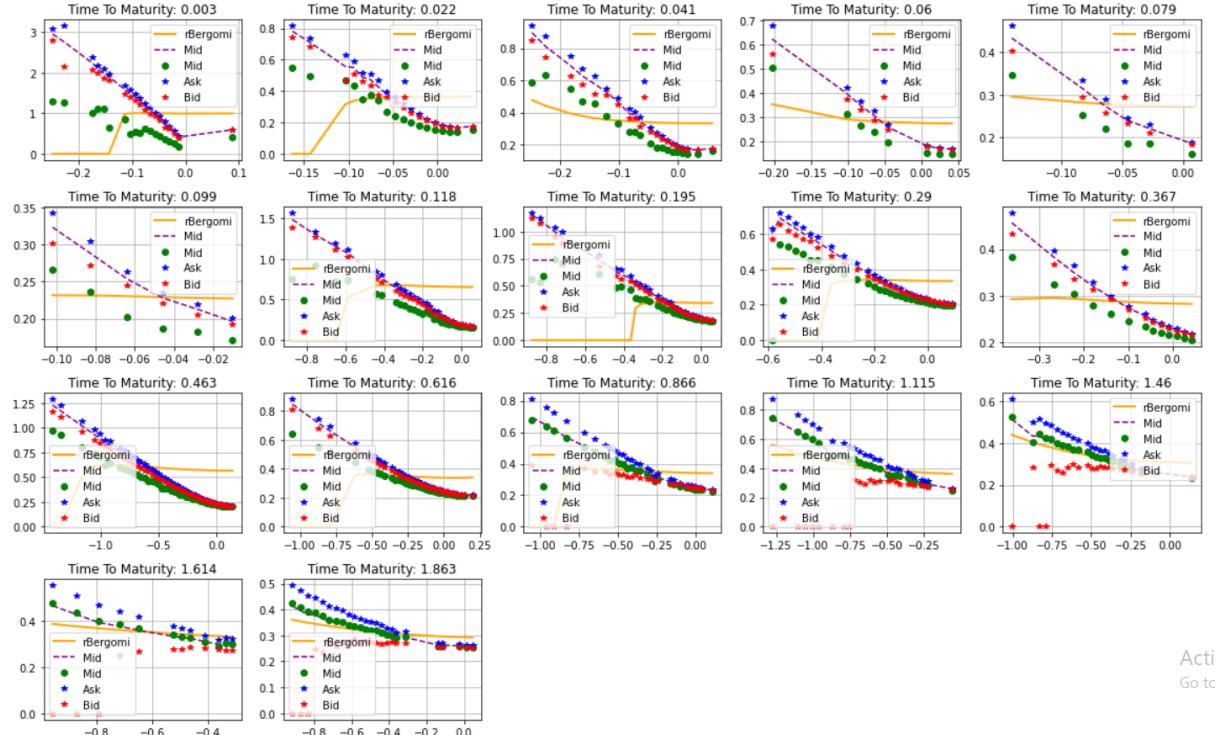


Figure 54: rBergomi fit to the MSFT volatility surface with time series implied parameters as observed on the 05.08.2021

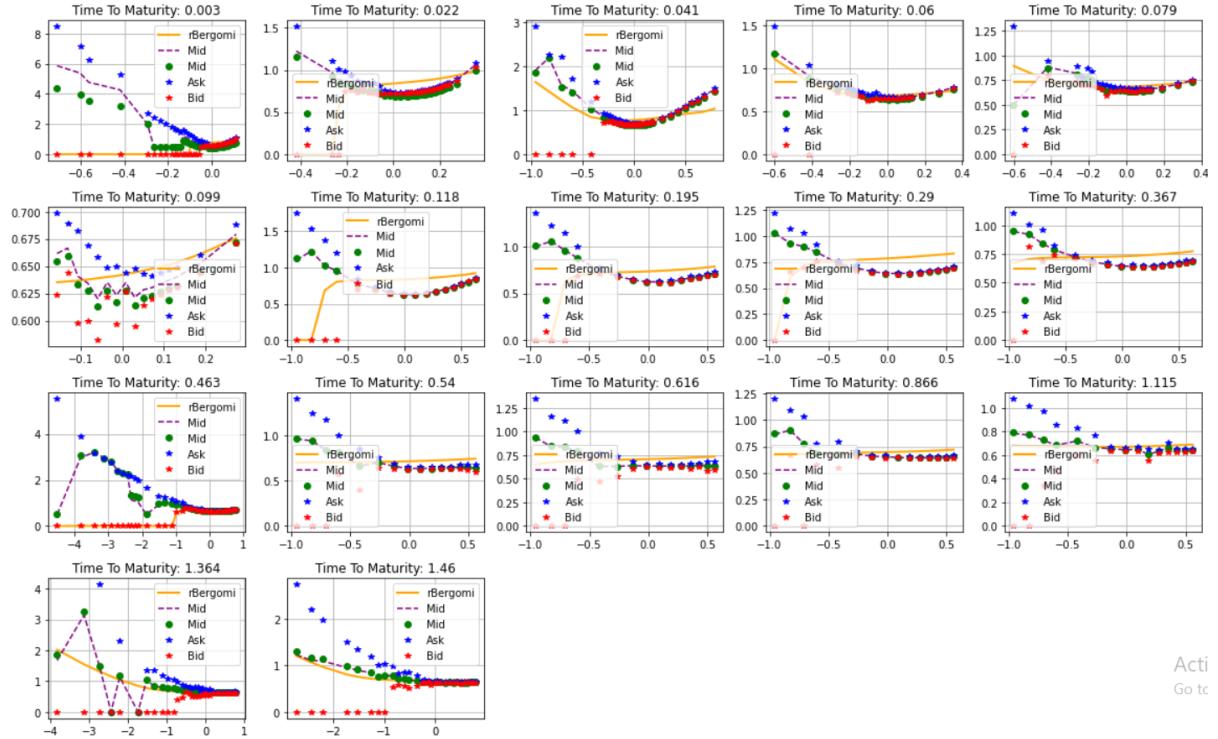


Figure 55: rBergomi fit to the NIO volatility surface with time series implied parameters as observed on the 05.08.2021

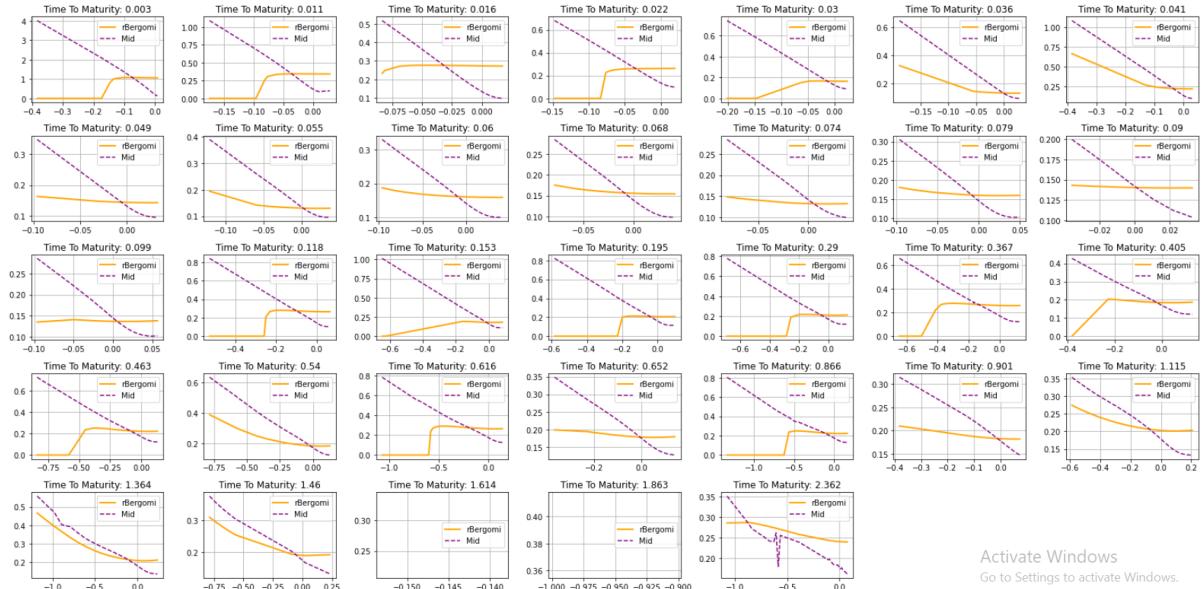


Figure 56: rBergomi fit to the SPX volatility surface with time series implied parameters as observed on the 05.08.2021

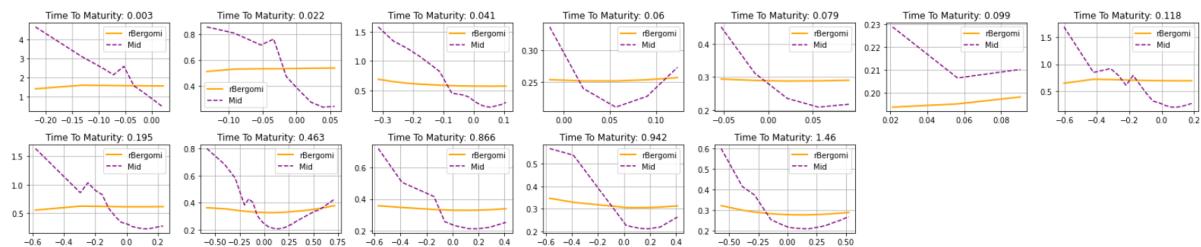


Figure 57: rBergomi fit to the T volatility surface with time series implied parameters as observed on the 05.08.2021

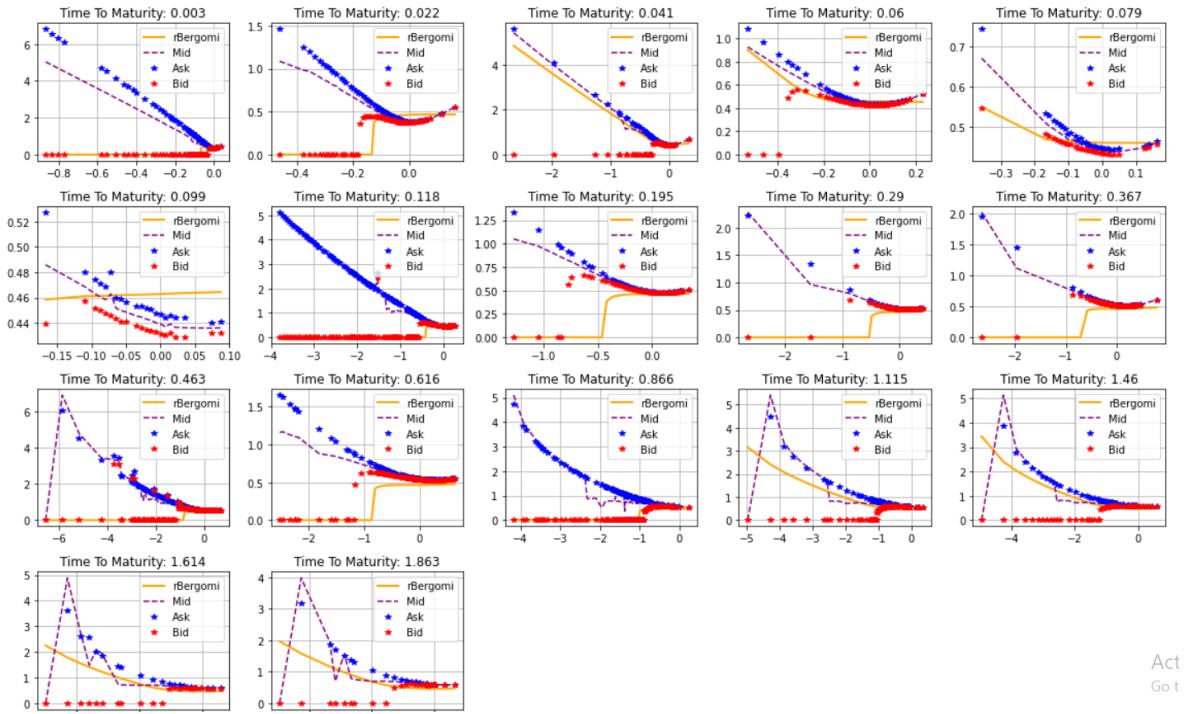


Figure 58: rBergomi fit to the TSLA volatility surface with time series implied parameters as observed on the 05.08.2021

As we can see the fit is “all over the place” and generally a “mixed bag”. Starting from the underlyings with the worst data quality, GME, NIO and T, we observe more data points than what the neural network used for calibration. This is because we do not need to apply the additional filtering as required by the network. However we can observe that the data is simply too poor in quality to help us derive any clear meaning from the fit. Nevertheless, we observe that for *some* expirations the fit to mid prices is quite close for GME and NIO, while for T the fit is subpar across any maturity, implying a flat surface. Moving on to TSLA, we observe that the best fit was achieved for maturity 0.041 which translates to exactly 15 days from day of observation. Interestingly on that exact day, the 20.08.2021, Tesla held their so called “AI Day”, a big company event where new technology and path of the company are unveiled. This was a big volatility event as the stock rose nearly 4% over the following trading day. Therefore, we can say that the parameters estimated through time series analysis ( $\mathbb{P}$ ) on the stock price observed on the 05.08.2021 carried expectations directly related to the company event that happened 15 days later. For MSFT on the other hand the fit is not great at any expiration. It improves somewhat only for the longest maturities, while for all others the quality of fit is unusable. Also for AAPL and SPX the fit is only improving with options for the longest time to maturity, options with shorter duration are not able to achieve any meaningful fit. Singling out the maturity of 0.041, for our most volatile assets, GME, NIO, and TSLA (and ignoring the

poor data quality for a moment) the fit can be described as surprisingly good. However for other (less volatile) assets, the fit at this maturity is poor. This curiosity may be pure coincidence, but it is an interesting observation into how parameters between the  $\mathbb{P}$  and  $\mathbb{Q}$  world may relate. One could argue that the value of more stable assets, such as AAPL, SPX, and MSFT depends less on short term developments but more on the general future directions of the companies, hence the volatility expectations embedded in the time series of realized variance relate to implied volatility far into the future. For smaller much more speculation driven underlyings such as GME, NIO, and TSLA on the other hand, the short-term expectations predominate and realized variance relates to implied volatility from options that are close to today's date. With their comparatively high valuations, which these latter companies carry, they rely on short term investor sentiment. I.e., if such a company underperforms in the short-term, higher valuations lead to increased volatility in the short term as well. These hypotheses are however pure speculation at this point. More data needs to be analyzed in order to draw any conclusions. Nevertheless, the research into relation between the  $\mathbb{P}$  and  $\mathbb{Q}$  world is an interesting one. It helps us understand how expectations today relate to asset prices tomorrow and potentially also helps us understand how volatility expectations are going to drive stock prices from today into the future.

## **Quality of Data**

All our observations throughout this work rely mainly on the quality of data used. While stock price data can be seen as a public good at this point, quality option data is difficult to come by. From the observations, which we manually gathered throughout the period between 29.04.2021 up until 01.10.2021 we saw days where the quality of data was not of great concern, however for most observation dates we experienced either complete lack of bid, ask, and corresponding volatility data, large discrepancies between reported IVs and mid-price implied IVs, or large spreads in both, time since last trade and liquidity spreads. When data quality was good, we noticed that the reported IVs by yahoo coincided with calculations that assumed a discount rate of zero and also no dividend expectations. This is clearly at odds with financial theory, as the Forward Price shouldn't be equal to the observed spot price, due to arbitrage opportunities. All these observations give us pause when analyzing the data. As such we cannot assume our conclusions here to be absolute, especially for non-standard quantitative investigations as we performed in the latter two sections of this chapter. Nevertheless, this thesis highlights topics which the rBergomi model can be used for and how to interpret the corresponding results.

# Conclusion

In this work and with the code found in our [GitHub](#) repository we have shown how the rBergomi model may be used in both a practical context as well as in academia. We hope this thesis motivates others to pick up the exploratory work we have done here and follow up with more in-depth analysis on the various topics we have touched. While rough volatility models are a relatively recent addition to the literature on option pricing and understanding volatility, we confirmed the discovery by (Gatheral, et al, 2018) that “volatility is rough”. Moreover, we found that the estimated parameters vary through time, sometimes quite significantly so. Subsequently, Gatheral devised the rough fractional stochastic volatility (RFSV) model on this observation which gives theoretical context to his findings and is also able to predict stock price volatility as well as the variance swap curve. With simple open, high, low, close (OHLC) data obtained from yahoo finance and an ample volatility estimator we confirmed Gatheral’s findings and estimated the RFSV parameters ourselves from time series analysis under  $\mathbb{P}$ .

Even though the rBergomi model may not be the most recent rough volatility model, as we discussed in our extensive literature review in Chapter 1, here we showed the power it wields. With its comparatively little number of parameters which all carry an explicit meaning, a single maturity may be manually calibrated with relative ease. Monte Carlo simulation is then feasible by utilizing one of the many minimizers that exist in order to assist with calibration. Once we know the “true” parameters of a single expiration or even the whole implied volatility surface, pricing any option whether in existence or purely hypothetical is possible and straight forward as we have shown. The recent additions in the literature regarding neural networks may be the last missing step that finally make such models applicable in a practical context. We show here in-depth how such a model may be built, what issues and potential problems could arise and how such a model may be used in practical and theoretical context. With its incredible speed and the possibility to fit to any expiration or even the whole volatility surface within seconds, financial institutions may finally be able to take advantage of this model in everyday business, which was hindered in the past due to the model’s non-Markovian nature. The challenges we had to overcome were mainly data quality related, as we used public data from yahoo-finance in this work. Without a reliable data-source we had to develop several cleaning algorithms, build a dividend and forward price forecasting engine, as well as interpolate the data ourselves where necessary. In order to fit to the data, we used both the neural network by (**Rømer, 2020**), as well as on the turbo charging approach by (McCrickerd, Pakkanen, 2018). Subsequently we

adapted both methods to be used with real data and improved them such that they can be used to their best extent. For the turbo charging, we discussed automated calibration via two distinct minimizers and how to apply them to an option chain and the whole surface. We found that unambiguously the Nelder-Mead calibration beats the Trust Region Reflective (TRF) method in both accuracy and speed, however further work may shed additional light on the subject of which minimizer can achieve best results for the rBergomi model. With the speed gain facilitated by the neural network we had the capability to go further in depth. To this end we re-engineered Romer's network, s.t. it allows for calibrating each maturity individually, as well as allowing for calibration with a floating forward variance curve. Therewith we were able to not only achieve a much-improved fit, but also to extract the forward variance expectations under the risk neutral probability measure  $\mathbb{Q}$ . The forward variance curve may be used to both gain an insight about future volatility expectations, but also price the volatility swap curve, which the rough fractional stochastic volatility (RFSV) model is able to predict. The fact that MC simulation and NN aligned in IV calibration and output makes us confident that we specified both approaches correctly. With these capabilities in our toolbox, we went ahead and calibrated to multiple assets with help of the neural network for which we manually gathered option data between 29.04.2021 and 01.10.2021. We found that for most assets the fit the neural network achieves with a single set of rBergomi parameters and a floating forward variance curve can be described as good, thereby validating the rBergomi model to be able to fit to an entire implied volatility surface with a single set of parameters. However, the fit from a constrained neural network will always lack some accuracy, as some parameter combinations may be simply unknown to the network. Specifically, we observed too little volatility curvature in ITM short to maturity options, but too much curvature for OTM contracts and too little volatility bias. Whether these observations hold for the rBergomi model in general or simply constitute minor misspecifications in the deep neural network needs to be explored further. Nevertheless, these small imperfections can be remedied by calibrating to each maturity individually where we managed to achieve a close to perfect fit with also less calibration time as explored in Chapter 5. Moreover, we looked at how predictive the rBergomi parameters are when tracking one maturity through time. By keeping the parameters fixed, we found that the prediction failed to capture the term structure of volatility well, even with incorporation of the estimated forward variance curve. Specifically close to maturity skew in volatility was not replicated. Even so, up until close to expiration the prediction was not *far* from the observed data. Interestingly, we found that OTM options exhibited less prediction error than ITM options. Lastly, we also looked at how the parameters estimated from realized volatility data

under  $\mathbb{P}$  relate to implied volatility in the market. We found that parameters estimated under  $\mathbb{P}$  may link to IVs at different points in time depending on the volatility of the underlying asset. I.e., for more stable assets we observed a better fit for far in the future contracts, while for highly volatile assets the fit was best for short to maturity option contracts. However, given the quality of data, we must caution against taking the latter two of our exploratory results as fact. Better and more data may bring further clarification on the subject. Nonetheless, our observations throughout this work all indicate that the rBergomi model parameters vary under the  $\mathbb{P}$  as well as under the  $\mathbb{Q}$  measure. This aligns with data shown by (Horvath, et. al, 2019) as well as (Rømer, 2020). Thus, one set of parameters may never fit an entire implied volatility surface perfectly, as each maturity carries expectations of a distinct volatility regime at that point in the future.

After having now seen the power of the rBergomi model, for future work it would make sense to go back and analyze the fractional behavior of volatility in greater detail. How do the parameters shape the volatility paths, what conclusions can be made about how parameters from  $\mathbb{P}$  and  $\mathbb{Q}$  relate to one another and can we uncover market expectations that these parameters correspond to? Also do the parameters in  $\mathbb{P}$  and  $\mathbb{Q}$  fluctuate in similar fashion?

This thesis has followed the topic of rough volatility from its discovery all the way till its latest iterations via the novel field of neural networks. We showed how roughness can be estimated using time series analysis, how options work and how they are priced, why option pricing is important and why the rBergomi model is a great addition to that subject. Using real data, we explored how MC simulation can find prices and corresponding implied volatilities, how we can calibrate using the same technique or even novel neural nets. This thesis looked at how institutions may go about finding totally hypothetical prices in the market and its corresponding IVs. Via the neural network we calibrated to a full implied volatility surface within seconds, both by each maturity individually, as well as the entire surface with a fixed set of parameters thereby confirming the rBergomi models' ability to fit to full implied volatility surfaces. Furthermore, we explored the predictive quality of a fixed set of parameters as well as how parameters estimated on realized volatility data relate to future implied volatility expectations. Now additional work is required to confirm or refute our findings with more and better data. The code that makes this feasible is publicly available from our [GitHub](#).

# References

## Papers

1. Alekseev, A. K., Navon, I. M., Steward, J. L., Comparison of advanced large-scale minimization algorithms for the solution of inverse ill-posed problems, Pages 63-87. 2011.
2. Alòs, E., León, J. A., Vives, J., On the short-time behavior of the implied volatility for jump-difusion models with stochastic volatility, 2007, Finance and Stochastics, 11 (4), 571–589.
3. Andersen, T., Bollerslev, T., Diebold, F., Ebens, H., The distribution of realized stock return volatility, 2001, \*Journal of Financial Economics\* \*\*\*61\*\*\*(1) 43-76 .
4. Antoine Jacquier, Claude Martini, Aitor Muguruza, On VIX futures in the rough Bergomi model, Quantitative Finance, Volume 18, 2018 - Issue 1, 45-61, 2017.
5. Asmussen, S., Stochastic simulation with a view towards stochastic processes, 1998, University of Aarhus. Centre for Mathematical Physics and Stochastics.
6. Bali, Turan G., Hovakimian, Armen, Volatility Spreads and Expected Stock Returns, 2009.
7. Bayer, C., Friz P., Gatheral, J., Pricing under rough volatility, 2015, Quantitative Finance, Volume 16, 2016 - Issue 6, Pages 887-904.
8. Bayer, C., Stemper, B., Deep calibration of rough stochastic volatility models, 2018
9. Bennedsen, M., A Rough Multi-Factor Model of Electricity Spot Prices, 2015, SSRN, 36 Pages.
10. Bennedsen, M., Lunde, A., Pakkanen, M. S., Decoupling the short and long-term behavior of stochastic volatility, 2021
11. Bennedsen M., Lunde, A., Pakkanen M. S., Hybrid scheme for Brownian semistationary processes, 2017, Finance Stoch, 21:931–965.
12. Bergomi, L., Smile dynamics II. Risk, October 2005, pp. 67–73.
13. Bergomi, L. and Guyon, J., Stochastic volatility's orderly smiles. Risk, May 2012, pp. 60–66.
14. Biagini, F., Hu, Y., Øksendal, B., Zhang, T., Stochastic calculus for fractional Brownian motion and applications, 2008, Probability and its Applications, Springer-Verlag, London.
15. Bishop, C.M., Pattern Recognition and Machine Learning, Springer: New York, 2006

16. Black, F., & Scholes, M. (1973). The Pricing of Options and Corporate Liabilities. *Journal of Political Economy*, 81 (3), 637-654.
17. (Bollerslev, T., Mikkelsen, H., Modeling and pricing long memory in stock market volatility, *Journal of Econometrics*, Volume 73, Issue 1, 1996, Pages 151-184
18. Branch, M. A., Coleman, T. F., Li, Y., A Subspace, Interior, and Conjugate Gradient Method for Large-Scale Bound-Constrained Minimization Problems, *SIAM Journal on Scientific Computing*, Vol. 21, Number 1, pp 1-23, 1999.
19. Burgess, Nicholas, Martingale Measures & Change of Measure Explained, 16, 2014.
20. Cesa, M., Mannix, R., Deep hedging strays when volatility gets rough – study, 2021, accessible here: <https://www.risk.net/derivatives/7808281/deep-hedging-strays-when-volatility-gets-rough-study> (19.12.2021, 18:51)
21. Cheridito, P, Kawaguchi H., and Maejima, M., Fractional Ornstein-Uhlenbeck processes. *Electron. J. Probab.*, 2003, 8(3), 1-14
22. Coeurjolly, J.-F., Simulation and identification of the fractional Brownian motion: a bibliographical and comparative study, *Journal of Statistical Software* 5, 1, 2000
23. Comte, F. and Renault, E., Long memory in continuous-time stochastic volatility models. *Math. Finance*, 1998, 8(4), 291–323.
24. Dandapani, A., Jusselin, P., Rosenbaum, M., From quadratic Hawkes processes to super-Heston rough volatility models with Zumbach effect, 2019, *Quantitative Finance*, Volume 21, 2021 - Issue 8.
25. Devasabai, K., Rough volatility's steampunk vision of future finance, 2021, accessible here: <https://www.risk.net/our-take/7816441/rough-volatilitys-steampunk-vision-of-future-finance> (19.12.2021, 18:52)
26. Dikshita Wadhawan, Harjit Singh, Estimating and Forecasting Volatility Using Arima Model: A Study on NSE, India, *Indian Journal of Finance*, volume 13, issue 5, p. 37 – 51,2019, Doi: 10.17010/ijf/2019/v13i5/144184
27. El Euch, O., Fukasawa, M. and Rosenbaum, M., The microstructural foundations of leverage effect and rough volatility, preprint arXiv:1609.05177, 2016.
28. El Euch, O., Gatheral, J. and Rosenbaum, M., Roughening Heston. *Risk*, May 2019, pp. 84-89.
29. El Euch, O. and Rosenbaum, M., Perfect hedging in rough Heston models. *Ann. Appl. Probab.*, 2018, 28(6), 3813-3856.
30. El Euch, O. and Rosenbaum, M., The characteristic function of rough Heston models. *Math. Financ.*, 2019, 29(1), 3-38.

31. Fukasawa, M., Asymptotic analysis for stochastic volatility: Martingale expansion. *Finance Stoch.*, 2011, 15(4), 635–654.
32. Fukasawa, M., Horvath, B., Tankov, P., Hedging under rough volatility, 2021, arXiv preprint arXiv:2105.04073
33. Fukasawa, M., Gatheral, J., A rough SABR formula, 2021, Available at SSRN 3844278
34. Garcin, M., Forecasting with fractional Brownian motion: a financial perspective, 2021, arXiv preprint arXiv:2105.09140
35. Garnier J., Sølna, K., Correction to Black--Scholes Formula Due to Fractional Stochastic Volatility, 2015, *SIAM J. Finan. Math.*, 8(1), 560–588, 29 pages
36. Gatheral, The Volatility Surface: A Practitioner’s Guide. Vol. 357, 2006 (John Wiley & Sons: Hoboken, NJ).
37. Gatheral, Jaisson, Rosenbaum, Volatility is Rough, 2018
38. Gatheral, Jusselin, P., & Rosenbaum, M. (2020). The quadratic rough Heston model and the joint S&P 500/VIX smile calibration problem. arXiv preprint arXiv:2001.01789, 2020)
39. Gatheral, J., Jacquier, A., Arbitrage-free SVI volatility surfaces, 2013
40. Gatheral, J. and Oomen, R.C., Zero-intelligence realized variance estimation. *Finance Stoch.*, 2010, 14(2), 249–283
41. Girsanov, I. V., On transforming a certain class of stochastic processes by absolutely continuous substitution of measures, *Theory of Probability and Its Applications*. 5 (3): 285–301, 1960.
42. I. Goodfellow, Y. Bengio and A. Courville. Deep Learning. MIT Press, 2016.
43. Hagan, P. S., Kumar, D., Lesniewski, A. S., Woodward, D. E., Managing smile risk, 2002, *The Best of Wilmott*, 1 , 249–296.
44. Henderson, V., Hobson, D., Howison, S., & Kluge, T., A comparison of q-optimal option prices in a stochastic volatility model with correlation, 2004, Oxford Financial Research Centre.
45. Hernandez, A., Model calibration with neural networks, 2017, *Risk*.
46. Heston, S. L., A closed-form solution for options with stochastic volatility with applications to bond and currency options, 1993, *The Review of Financial Studies*, 6 (2), 327–343.
47. Horvath, Jacquier, Tankov, Volatility Options in Rough Volatility Models, *SIAM J. Finan. Math.*, 11(2), 437–469, 2020.

48. Horvath, B., Muguruza, A., Tomas, M., Deep Learning Volatility, 2019, Available at SSRN 3322085.
49. Horvath, B., Teichmann, J., Zuric, Z., Deep Hedging under Rough Volatility, 2021, Swiss Finance Institute Research Paper No. 21-88.
50. Hull, J., White, A., One-factor interest-rate models and the valuation of interest-rate derivative securities, 1993, Journal of Financial and Quantitative Analysis, 28 (2), 235–254.
51. Hurst, H. E., The Nile Flood of 1913, 1913, Nature Publishing Group, Meteorological Office, Survey Department, Giza (Mudiria), Egypt
52. Hutchinson, J. M., Lo, A. W., Poggio T., A nonparametric approach to pricing and hedging derivative securities via learning networks, 1994 The Journal of Finance 49.3, 851-889.
53. Fleming, J., The quality of market volatility forecasts implied by S&P 100 index option prices, Journal of Empirical Finance, 1998, Volume 5, Issue 4, Pages 317-345.
54. D.P. Kingman and J. Ba, Adam: A Method for Stochastic Optimization. Conference paper, 3rd International Conference for Learning Representations, 2015.
55. Kinsley, Kukieta, 2020, Neural Networks from Scratch in Python, 2020.
56. Livieri, G., Mouti, S., Pallavicini A., Rosenbaum, M., Rough volatility: Evidence from option prices, 2017, ISE Transactions, Operations Engineering & Analytics, Volume 50, 2018 - Issue 9, Pages 767-776.
57. Liu, Grzelak, Oosterlee, The Seven-League Scheme: Deep Learning for Large Time Step Monte Carlo Simulations of Stochastic Differential Equations, Risks 2022
58. Lyu, Q., Lin, Z., She, Y., Zhang, C., A comparison of typical  $\ell_p$  minimization algorithms, Neurocomputing, ISSN 0925-2312, Volume 119, Pages 413-424, 2013
59. Mandelbrot, Benoit, Encyclopedia of Quantitative Finance, 2010
60. Mandelbrot, B., Fractals and Scaling in Finance, 1997, Springer. ISBN 978-1-4757-2763-0.
61. Mandelbrot, B., The fractal geometry of nature 1983, Macmillan. ISBN 978-0-7167-1186-5.
62. Mandelbrot, B., The Variation of Some Other Speculative Prices, 1967, The Journal of Business, Vol. 40, No. 4, pp. 393-413 (21 pages). Published By: The University of Chicago Press
63. Mandelbrot, B.; Hudson, R. L., The (Mis)Behaviour of Markets: A Fractal View of Risk, Ruin and Reward, 2008.

64. Mandelbrot, B., Van Ness, J.W., Fractional Brownian motions, fractional noises and applications. *SIAM Rev.*, 1968, 10(4), 422– 437.
65. Marcin Fałdziński, Magdalena Osińska, Volatility Estimators in Econometric Analysis of Risk Transfer on Capital Markets, 2016, *Dynamic Econometric Models*, p. 1234-3862.
66. McGhee, W.A., An Artificial Neural Network Representation of the SABR Stochastic Volatility Model, 2018.
67. Mishura, Y. S., Stochastic calculus for fractional Brownian motion and related processes, *Lecture Notes in Mathematics*, Vol. 1929, Springer-Verlag, Berlin, 2008
68. More, J. J., "The Levenberg-Marquardt Algorithm: Implementation and Theory," *Numerical Analysis*, ed. G. A. Watson, *Lecture Notes in Mathematics* 630, Springer Verlag, pp. 105-116, 1977
69. Muzy, J.F., Delour J., Bacry, E., Modelling fluctuations of financial time series: from cascade process to stochastic volatility model, 2000, *The European Physical Journal B - Condensed Matter and Complex Systems* volume 17, pages 537–548
70. Nelder, J. A., Mead R., A Simplex Method for Function Minimization, *The Computer Journal*, Volume 8, Issue 1, April 1965, Page 27
71. Nourdin, I., Selected Aspects of Fractional Brownian Motion, 2012
72. Nualart, D., *The Malliavin calculus and related topics* Probability and its Applications, New York, Probability and its Applications, second edn., Springer-Verlag, Berlin, 2006
73. Pasquariello, Paolo. "Financial market dislocations." *The Review of Financial Studies* 27.6: 1868-1914, 2014
74. Robert, C.Y. and Rosenbaum, M., A new approach for the dynamics of ultra-high-frequency data: The model with uncertainty zones. *J. Financ. Econom.*, 2011, 9(2), 344–366.
75. Robert, C.Y. and Rosenbaum, M., Volatility and covariation estimation when microstructure noise and trading times are endogenous. *Math. Finance*, 2012, 22(1), 133–164
76. Rosenbaum, M., First order p-variations and Besov spaces. *Stat. Probab. Lett.*, 2009, 79(1), 55–62.
77. Rosenbaum, M., A new microstructure noise index. *Quant. Finance*, 2011, 11(6), 883–899.

78. Rosenbaum, M., Zhang, J., Deep calibration of the quadratic rough Heston model, 2021, arXiv preprint arXiv:2107.01611.
79. Ruas, Joao Pedro, José Dias Curto, and Joao Pedro Vidal Nunes. "The Implied Volatility Bias: A No-Arbitrage Approach for Short-Dated Options." (2012).
80. Rømer, Sigurd Emil, Historical analysis of rough volatility models to the SPX market, 2020, Available at SSRN 3678235
81. Shevchenko, G., Fractional Brownian Motion in a Nutshell, 2014
82. Takaishi, T., Rough volatility of Bitcoin, 2019.
83. Zeron, M., Ruiz, I., Tensoring volatility calibration, Calibration of the rough Bergomi volatility model via Chebyshev Tensors, 2020.
84. Zhu, C., Byrd R. H., Nocedal. J., L-BFGS-B: Algorithm 778: L-BFGS-B, FORTRAN routines for large scale bound constrained optimization, 1997, ACM Transactions on Mathematical Software, 23: 4, pp. 550-560.
85. Zumbach, G., Time reversal invariance in finance, 2009, Quantitative Finance, 9(5):505–515.

## Online Sources

1. <https://sites.google.com/site/roughvol/home>  
(Date and time of access: 04.08.2021; 12:31pm)
2. <https://realized.oxford-man.ox.ac.uk/data/download>  
(Date and time of access: 04.08.2021; 12:31pm)
3. <https://realized.oxford-man.ox.ac.uk/documentation/econometric-methods>  
(Date and time of access: 04.08.2021; 12:31pm)
4. [https://www.tpq.io/p/rough\\_volatility\\_with\\_python.html](https://www.tpq.io/p/rough_volatility_with_python.html)  
(Date and time of access: 04.08.2021; 12:31pm)
5. <https://finance.yahoo.com>  
(Date and time of access: 04.08.2021; 12:31pm)
6. [https://www.youtube.com/playlist?list=PLAXSVuGaw0KxTEN\\_cy-RCuEzzRdnF\\_xtx](https://www.youtube.com/playlist?list=PLAXSVuGaw0KxTEN_cy-RCuEzzRdnF_xtx)  
(Date and time of access: 04.08.2021; 12:31pm)
7. <https://www.reddit.com/r/wallstreetbets/>  
(Date and time of access: 04.08.2021; 12:31pm)
8. <https://fred.stlouisfed.org/series/DFF>  
(Date and time of access: 04.08.2021; 12:31pm)
9. <https://www.youtube.com/watch?v=gW073Tnx7CE>  
(Date and time of access: 04.08.2021; 12:31pm)
10. [https://en.wikipedia.org/wiki/Girsanov\\_theorem](https://en.wikipedia.org/wiki/Girsanov_theorem)  
(Date and time of access: 04.08.2021; 12:31pm)
11. <https://www.quora.com/How-does-one-explain-what-change-of-measure-is-in-Girsanovs-Theorem-How-is-Girsanovs-Theorem-used-and-what-applicability-does-this-all-have-to-Mathematical-Finance-and-Probability>  
(Date and time of access: 04.08.2021; 12:31pm)
12. <https://corporatefinanceinstitute.com/resources/knowledge/trading-investing/implied-volatility-iv/>  
(Date and time of access: 04.08.2021; 12:31pm)
13. <https://corporatefinanceinstitute.com/resources/knowledge/trading-investing/implied-volatility-iv/>  
(Date and time of access: 04.08.2021; 12:31pm)
14. [https://github.com/ryanmccrickerd/rough\\_bergomi](https://github.com/ryanmccrickerd/rough_bergomi)

(Date and time of access: 04.08.2021; 12:31pm)

15. <https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.brentq.html>

(Date and time of access: 04.08.2021; 12:31pm)

16. <https://github.com/amuguruza/NN-StochVol-Calibrations>

(Date and time of access: 04.08.2021; 12:31pm)

17. <https://docs.scipy.org/doc/scipy/reference/optimize.html>

(Date and time of access: 04.08.2021; 12:31pm)

18. [https://www.youtube.com/watch?v=Wo5dMEP\\_BbI&list=PLQVvaa0QuDcjD5BAw2DxE6OF2tius3V3](https://www.youtube.com/watch?v=Wo5dMEP_BbI&list=PLQVvaa0QuDcjD5BAw2DxE6OF2tius3V3)

(Date and time of access: 04.08.2021; 12:31pm)

19. [https://en.wikipedia.org/wiki/Activation\\_function](https://en.wikipedia.org/wiki/Activation_function)

(Date and time of access: 04.08.2021; 12:31pm)

20. <https://stackoverflow.com/questions/2480650/what-is-the-role-of-the-bias-in-neural-networks>

(Date and time of access: 04.08.2021; 12:31pm)

21. <https://pypi.org/project/fbm/>

(Date and time of access: 03.10.2021; 16:54 pm)

22. <https://finance.yahoo.com/calendar/earnings/>

(Date and time of access: 03.10.2021; 16:54 pm)

23. Jean-Philippe Bouchaud, A step closer to the perfect volatility model, 2020, url:

<https://www.risk.net/comment/7717176/a-step-closer-to-the-perfect-volatility-model>

(Date and time of access: 12.12.2021; 00:21)

# Appendix

## Appendix A:

### Appendix A1:

	Contract Name	Last Trade Date	Strike	Last Price	Bid	Ask	Change	Volume	Open Interest	IV
0	AAPL230616C00060000	2021-06-25 3:23PM EDT	60.0	74.13	73.95	74.35	-0.77	8	2218	36.96%
1	AAPL230616C00065000	2021-06-25 11:45AM EDT	65.0	69.80	69.30	69.70	-0.60	11	1616	35.74%
2	AAPL230616C00070000	2021-06-25 12:26PM EDT	70.0	65.00	64.75	65.15	-0.25	6	440	34.73%
3	AAPL230616C00075000	2021-06-25 3:01PM EDT	75.0	60.34	60.30	60.75	-0.36	7	462	34.01%
4	AAPL230616C00080000	2021-06-24 1:34PM EDT	80.0	56.49	55.95	56.40	0.00	3	798	33.12%
5	AAPL230616C00085000	2021-06-25 3:13PM EDT	85.0	51.70	51.75	52.20	-0.55	4	403	32.38%
6	AAPL230616C00090000	2021-06-25 3:43PM EDT	90.0	47.60	47.70	48.10	-0.32	3	586	31.60%
7	AAPL230616C00095000	2021-06-24 10:23AM EDT	95.0	44.40	43.85	44.25	-0.30	5	225	31.11%
8	AAPL230616C00100000	2021-06-25 3:29PM EDT	100.0	40.20	40.25	40.65	-0.40	29	8033	30.81%
9	AAPL230616C00105000	2021-06-25 10:56AM EDT	105.0	37.15	36.70	37.10	-0.45	102	603	30.27%
10	AAPL230616C00110000	2021-06-25 3:57PM EDT	110.0	33.71	33.45	33.85	-0.29	13	1878	29.97%
11	AAPL230616C00115000	2021-06-25 1:37PM EDT	115.0	30.95	30.45	30.85	0.23	11	861	29.77%
12	AAPL230616C00120000	2021-06-25 3:16PM EDT	120.0	27.85	27.70	28.10	-0.55	125	5236	29.67%
13	AAPL230616C00125000	2021-06-25 3:38PM EDT	125.0	25.30	25.10	25.50	-0.35	35	2525	29.50%
14	AAPL230616C00130000	2021-06-25 3:51PM EDT	130.0	22.90	22.75	23.00	-0.45	123	5039	29.22%
15	AAPL230616C00135000	2021-06-25 3:46PM EDT	135.0	20.60	20.55	20.95	-0.35	96	4735	29.30%
16	AAPL230616C00140000	2021-06-25 2:49PM EDT	140.0	18.85	18.55	18.90	-0.15	53	2735	29.16%
17	AAPL230616C00145000	2021-06-25 1:35PM EDT	145.0	16.85	16.75	17.10	-0.69	19	1065	29.13%
18	AAPL230616C00150000	2021-06-25 2:55PM EDT	150.0	15.30	15.10	15.40	-0.15	20	7232	29.02%
19	AAPL230616C00155000	2021-06-23 1:16PM EDT	155.0	13.95	13.70	14.00	-0.65	1	434	29.13%
20	AAPL230616C00160000	2021-06-25 3:56PM EDT	160.0	12.50	12.35	12.70	-0.20	36	4525	29.19%
21	AAPL230616C00165000	2021-06-23 10:22AM EDT	165.0	11.20	11.15	11.50	-0.55	2	2074	29.23%
22	AAPL230616C00170000	2021-06-25 2:09PM EDT	170.0	10.72	10.05	10.40	0.42	11	3291	29.25%
23	AAPL230616C00175000	2021-06-25 1:42PM EDT	175.0	9.25	9.10	9.40	-0.22	16	2130	29.26%
24	AAPL230616C00180000	2021-06-25 2:44PM EDT	180.0	8.35	8.25	8.55	-0.10	91	11194	29.36%
25	AAPL230616C00185000	2021-06-24 12:47PM EDT	185.0	7.90	7.45	7.80	0.00	59	919	29.48%
26	AAPL230616C00190000	2021-06-25 12:01PM EDT	190.0	7.00	6.80	7.15	-0.09	49	904	29.65%
27	AAPL230616C00195000	2021-06-25 11:16AM EDT	195.0	6.27	6.25	6.55	-0.43	8	41253	29.80%
28	AAPL230616C00200000	2021-06-25 12:05PM EDT	200.0	5.77	5.65	5.90	-0.13	17	2476	29.76%
29	AAPL230616C00210000	2021-06-25 3:42PM EDT	210.0	4.80	4.70	5.05	-0.20	9	498	30.19%
30	AAPL230616C00220000	2021-06-25 2:21PM EDT	220.0	4.12	3.95	4.30	-0.23	3	229	30.51%
31	AAPL230616C00230000	2021-06-25 2:21PM EDT	230.0	3.51	3.35	3.65	-0.09	3	1745	30.76%
32	AAPL230616C00240000	2021-06-25 1:11PM EDT	240.0	3.10	2.91	3.20	0.00	1	1548	31.21%
33	AAPL230616C00250000	2021-06-24 10:21AM EDT	250.0	2.87	2.54	2.75	0.00	12	2182	31.46%
34	AAPL230616C00260000	2021-06-25 3:26PM EDT	260.0	2.45	2.30	2.47	0.02	181	4147	31.99%

Example option chain with 1.975 years to expiration for AAPL observed on 25-06-2021. It lists all information yahoo finance has available. From Bid/Ask the mid-price (77) , spread and liquidity (76) can be computed. The yahoo reported IVs coincide with mid-price implied volatilities under a risk-free rate of 0%. Many more option chains exist as observed on 25-06-2021. Each option chain has a unique date of expiration and thus also unique IV smile characteristics.

## Appendix A2:

Python 3.8 dependencies used to facilitate required programming and computation:

```
from datetime import datetime, timedelta
from ipywidgets import interact, interactive, fixed, interact_manual
from keras.layers import Activation
from keras import backend as K
from keras.utils.generic_utils import get_custom_objects
from matplotlib import cm
from matplotlib import pyplot as plt
from rbergomi import rBergomi
from utils import bsinv
from yahoo_fin.options import get_options_chain, get_expiration_dates (***)
```

```
import csv
import gzip
import ipywidgets as widgets
import keras
import matplotlib.pyplot as plt
import matplotlib.ticker as mtick
import numpy as np
import os
import pandas as pd
import pandas_datareader.data as web
import pickle
import pytz
import scipy
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
import time as time
%matplotlib inline
```

*(\*\*\*) Script had to be adapted, as yahoo changed their backend and thus the library does not work under the original version. Please see below for changes made to the package code:*

**ORIGINAL CODE:**

```
def get_options_chain(ticker, date = None, raw = True):  
  
    """Extracts call / put option tables for input ticker and expiration  
date. If  
    no date is input, the default result will be the earliest expiring  
option chain from the current date.  
  
    @param: ticker  
    @param: date"""  
  
    site = build_options_url(ticker, date)  
  
    tables = pd.read_html(site)  
  
    if len(tables) == 1:  
        calls = tables[0].copy()  
        puts = pd.DataFrame(columns = calls.columns)  
    else:  
        calls = tables[0].copy()  
        puts = tables[1].copy()  
  
    if not raw:  
        calls["% Change"] = calls["% Change"].str.strip("%").map(force_float)  
        calls["% Change"] = calls["% Change"].map(lambda num: num / 100 if  
isinstance(num, float) else 0)  
        calls["Volume"] = calls["Volume"].str.replace("-",  
"0").map(force_float)  
        calls["Open Interest"] = calls["Open Interest"].str.replace("-",  
"0").map(force_float)  
  
        puts["% Change"] = puts["% Change"].str.strip("%").map(force_float)  
        puts["% Change"] = puts["% Change"].map(lambda num: num / 100 if  
isinstance(num, float) else 0)  
        puts["Volume"] = puts["Volume"].str.replace("-",  
"0").map(force_float)  
        puts["Open Interest"] = puts["Open Interest"].str.replace("-",  
"0").map(force_float)  
  
    return {"calls": calls, "puts": puts}
```

**NEW CODE:**

```
def get_options_chain(ticker, date = None, raw = True):  
  
    """Extracts call / put option tables for input ticker and expiration  
date. If  
no date is input, the default result will be the earliest expiring  
option chain from the current date.  
  
    @param: ticker  
    @param: date"""  
  
    site = build_options_url(ticker, date)  
  
    # New Code as "tables = pd.read_html(site)" leads to an HTTP error  
    session = HTMLSession()  
    resp = session.get(site)  
    html = resp.html.raw_html.decode()  
  
    tables = pd.read_html(html)  
  
    if len(tables) == 1:  
        calls = tables[0].copy()  
        puts = pd.DataFrame(columns = calls.columns)  
    else:  
        calls = tables[0].copy()  
        puts = tables[1].copy()  
  
    if not raw:  
        calls["% Change"] = calls["% Change"].str.strip("%").map(force_float)  
        calls["% Change"] = calls["% Change"].map(lambda num: num / 100 if  
isinstance(num, float) else 0)  
        calls["Volume"] = calls["Volume"].str.replace("-",  
"0").map(force_float)  
        calls["Open Interest"] = calls["Open Interest"].str.replace("-",  
"0").map(force_float)  
  
        puts["% Change"] = puts["% Change"].str.strip("%").map(force_float)  
        puts["% Change"] = puts["% Change"].map(lambda num: num / 100 if  
isinstance(num, float) else 0)  
        puts["Volume"] = puts["Volume"].str.replace("-",  
"0").map(force_float)  
        puts["Open Interest"] = puts["Open Interest"].str.replace("-",  
"0").map(force_float)  
  
    return {"calls": calls, "puts": puts}
```

## Appendix B:

### Appendix B1

Following (Shevchenko, 2014) and starting from the same assumption that  $s_1 < t_1 < s_2 < t_2$  and thus the two intervals  $[W_{t_1}^H - W_{s_1}^H], [W_{t_2}^H - W_{s_2}^H]$  in (2) do not intersect. Therefore, (2) can be rewritten as  $\frac{(f(a_1) - f(a_2)) - (f(b_1) - f(b_2))}{2}$ , with  $a_1 = t_2 - s_1$ ,  $a_2 = t_2 - t_1$ ,  $b_1 = s_2 - s_1$ ,  $b_2 = s_2 - t_1$ , and  $f(x) = x^{2H}$ . Since  $a_1 - a_2 = b_2 - b_1 = t_1 - s_1$ :

$$\mathbb{E}[(W_{t_1}^H - W_{s_1}^H), (W_{t_2}^H - W_{s_2}^H)] > 0, \quad H \in (0, 1/2)$$

and

$$\mathbb{E}[(W_{t_1}^H - W_{s_1}^H), (W_{t_2}^H - W_{s_2}^H)] < 0, \quad H \in (1/2, 1)$$

### Appendix B2:

	<b>names</b>	<b>h_est</b>	<b>nu_est</b>		<b>names</b>	<b>h_est</b>	<b>nu_est</b>
<b>0</b>	SPX2.rk	0.129279	0.324802	<b>11</b>	AEX.rk	0.141698	0.291701
<b>1</b>	FTSE2.rk	0.140566	0.267038	<b>12</b>	SSMI.rk	0.176528	0.220953
<b>2</b>	N2252.rk	0.110684	0.326482	<b>13</b>	IBEX2.rk	0.123928	0.283392
<b>3</b>	GDAXI2.rk	0.145953	0.277026	<b>14</b>	NSEI.rk	0.108156	0.321499
<b>4</b>	RUT2.rk	NaN	NaN	<b>15</b>	MXX.rk	0.089456	0.323159
<b>5</b>	AORD2.rk	0.081706	0.358325	<b>16</b>	BVSP.rk	0.107941	0.311997
<b>6</b>	DJI2.rk	0.127363	0.318882	<b>17</b>	GSPTSE.rk	NaN	NaN
<b>7</b>	IXIC2.rk	NaN	NaN	<b>18</b>	STOXX50E.rk	0.116263	0.338917
<b>8</b>	FCHI2.rk	0.127367	0.292810	<b>19</b>	FTSTI.rk	0.127094	0.228910
<b>9</b>	HSI2.rk	0.098814	0.281513	<b>20</b>	FTSEMIB.rk	0.132047	0.295236
<b>10</b>	KS11.rk	0.118414	0.280221				

Table 11: Estimates for  $H$  and  $v$  with  $q=2$  for various indices in the Oxford-Man Institute of Quantitative Finance dataset. The name extension ".rk" signifies realized kernel estimates for realized variance.

### Appendix B3:

Here are the distributions for kernel, GKYZ, and log-return variance estimates.

Kernel Variance Estimate Increments Distributions

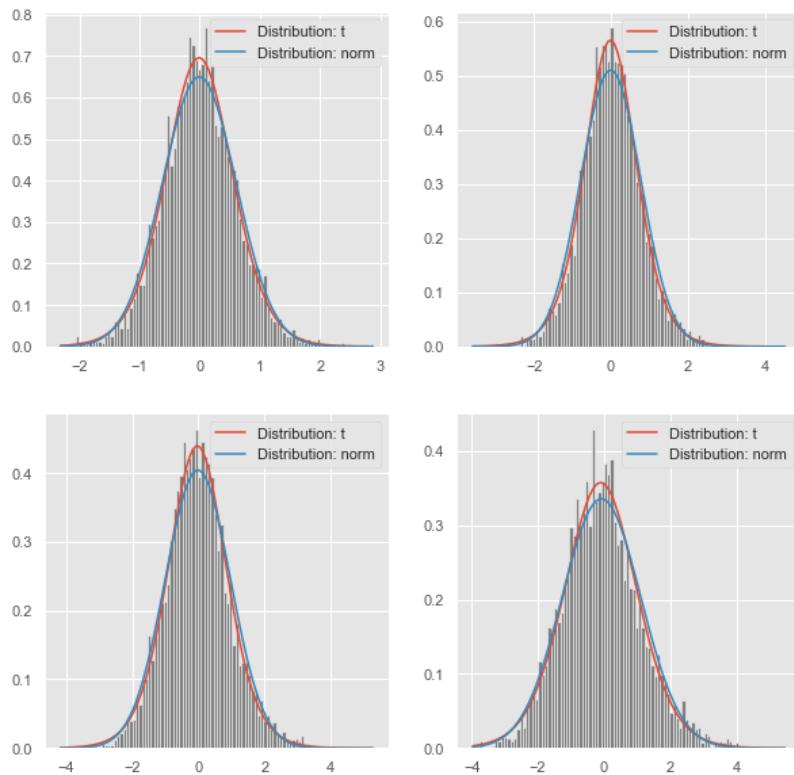


Figure 59: Kernel Variance Increment Distribution by lag

GKYZ Variance Estimate Increments Distributions

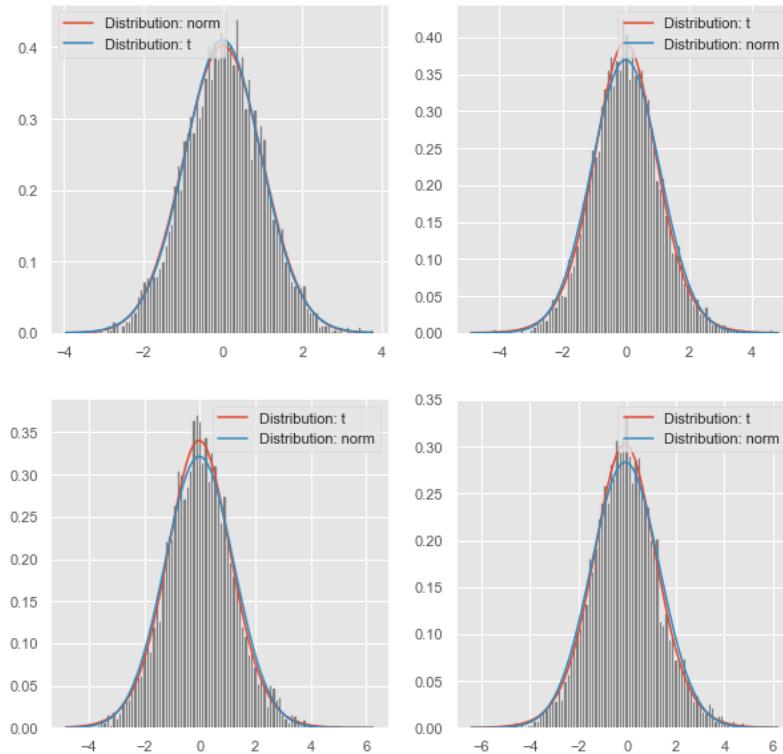


Figure 60: GKYZ Variance Increment Distribution by lag

Log Return Variance Estimate Increments Distributions

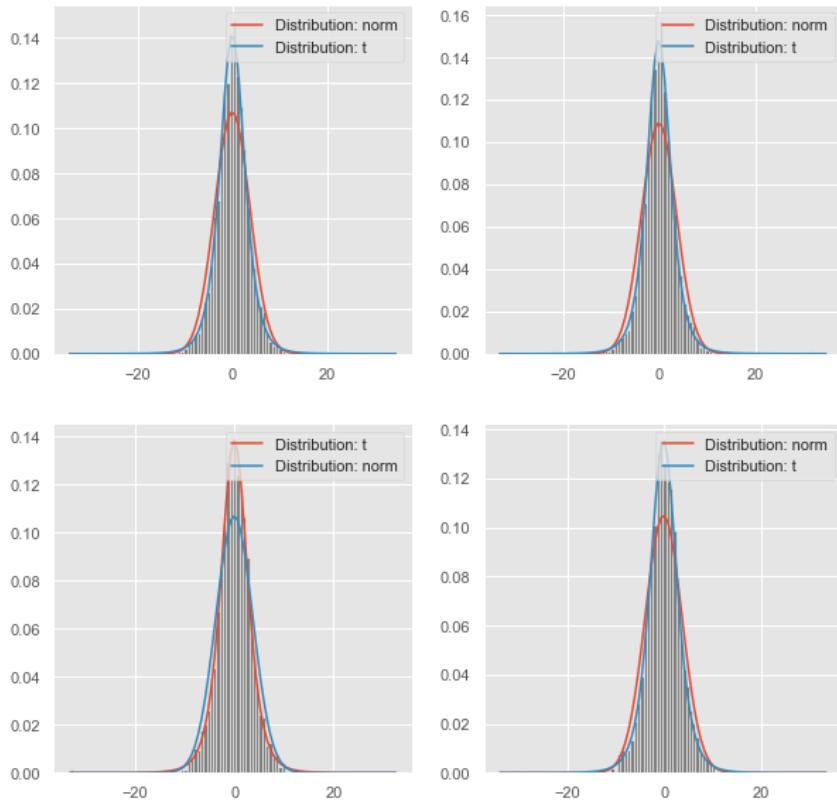


Figure 61: Log Return Variance Increment Distribution by lag

## Appendix C:

### Appendix C1

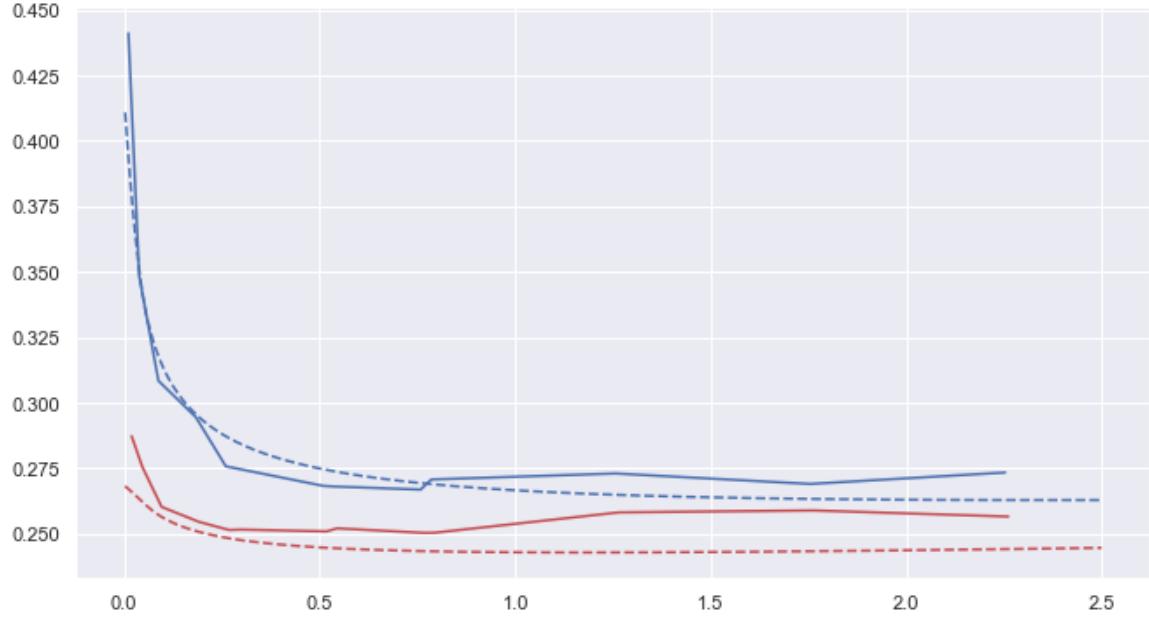


Figure 62: Shows the forecasted Variance Swap Curve (dashed line) vs the actual observed Variance Swap Curve(continuous) for the dates of 15.11.2008 (blue) and 12.11.2008 (red)

### Appendix C2

With  $Z_1$  and  $Z_2$  as two independent standard normal random variables, we define

$$X = Z_1$$

$$Y = \rho Z_1 + \sqrt{1 - \rho^2} Z_2$$

As  $Z_1$  and  $Z_2$  are jointly normal, they have the joint PDF

$$\begin{aligned} f_{Z_1 Z_2}(z_1, z_2) &= f_{Z_1}(z_1)f_{Z_2}(z_2) \\ &= \frac{1}{2\pi} \exp \left\{ -\frac{1}{2} [z_1^2 + z_2^2] \right\} \end{aligned}$$

We need to show  $aX + bY$  is normal for all  $a, b \in \mathbb{R}$ , thus:

$$\begin{aligned} aX + bY &= aZ_1 + b(\rho Z_1 + \sqrt{1 - \rho^2} Z_2) \\ &= (a + b\rho)Z_1 + b\sqrt{1 - \rho^2} Z_2 \end{aligned}$$

As this is simply a linear combination of  $Z_1$  and  $Z_2$ , we know that this quantity is normal.

## Appendix D:

### Appendix D1:

Strike Price	Prob. of Moneyness	Strike Price	Prob. of Moneyness
60.0	93.76	100.0	70.94
65.0	91.98	105.0	66.89
70.0	89.89	110.0	62.70
75.0	87.49	115.0	58.43
80.0	84.78	120.0	54.13
85.0	81.74	125.0	49.88
90.0	78.41	130.0	45.73
95.0	74.80	135.0	41.73
100.0	70.94	140.0	37.93
105.0	66.89	145.0	34.34
110.0	62.70	150.0	31.00
115.0	58.43	155.0	27.92
120.0	54.13	160.0	25.09
125.0	49.88	165.0	22.52
130.0	45.73	170.0	20.19
135.0	41.73	175.0	18.09
140.0	37.93	180.0	16.21
145.0	34.34	185.0	14.52
150.0	31.00	190.0	13.02
60.0	93.76	195.0	11.68
65.0	91.98	200.0	10.48
70.0	89.89	210.0	8.47
75.0	87.49	220.0	6.88
80.0	84.78	230.0	5.62
85.0	81.74	240.0	4.61
90.0	78.41	250.0	3.81
95.0	74.80	260.0	3.16

Table 12: Showing probability of moneyness in percent for each available strike price.

## Appendix D2:

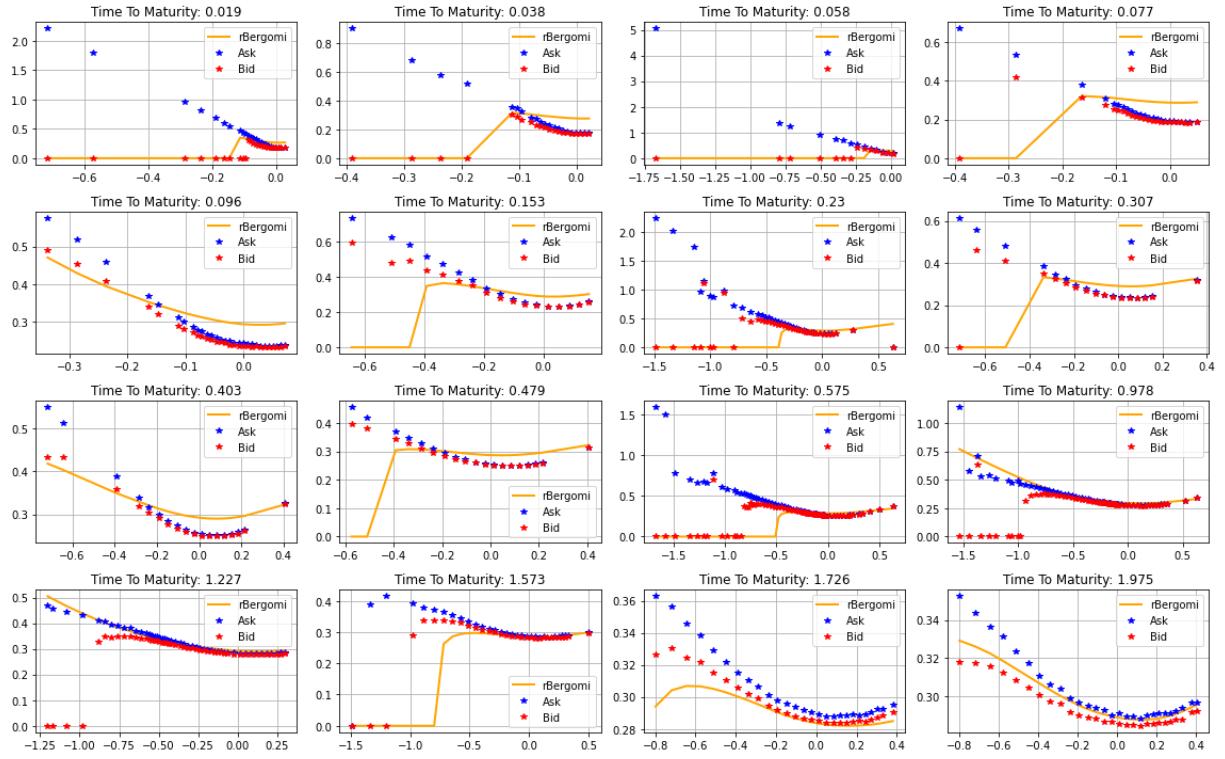


Figure 63: Using manual calibrated parameters (yellow) to fit to the full volatility surface. IVs that show values of zero reflect market prices for which the Black Scholes option pricing model is not able to compute corresponding values.

### Appendix D3:

With the Nelder-Mead minimizer and for initial params of: (-0.5, 0.83515963, -0.93144967, 0.06974231) we report an MSE of 0.035737996 for estimated parameters of (-0.396038, 3.60854902, 0.04045549, 0.0871459) with the following fit:

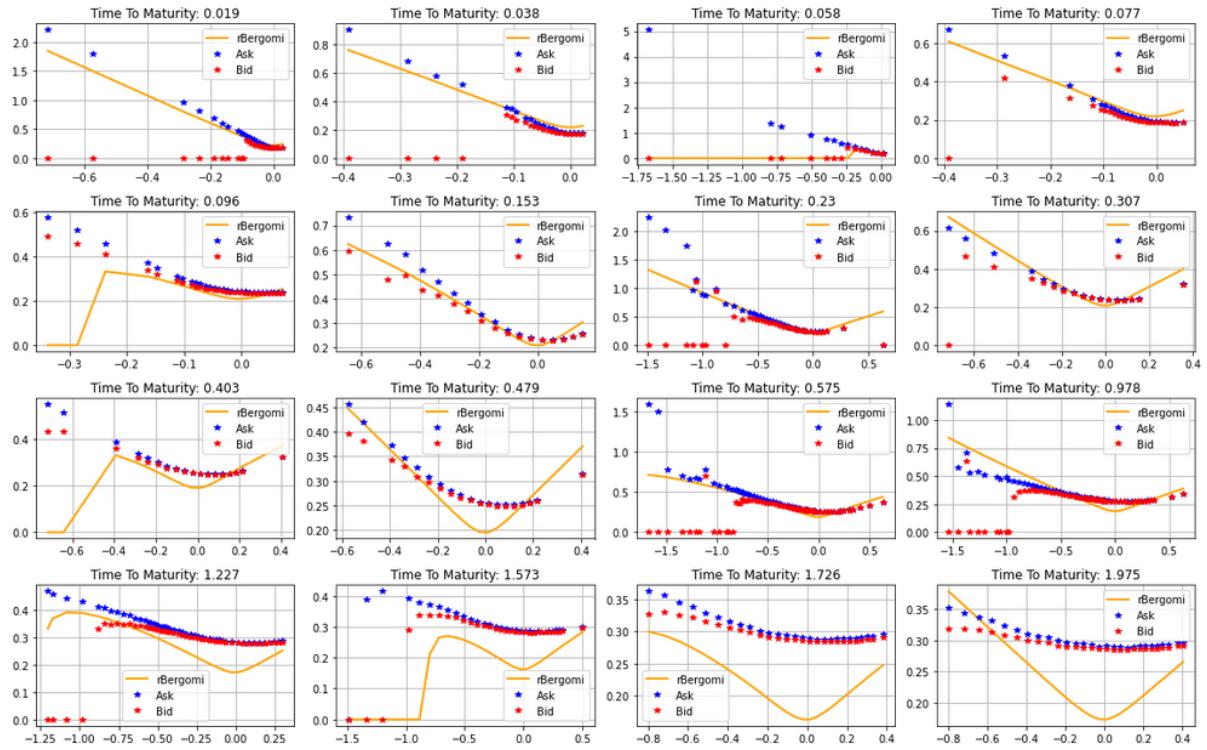


Figure 64: Nelder Mead calibration result for different starting parameters

## Appendix D4:

In order to reduce dimensionality of the calibration problem, calibration to observed log prices of options, instead of their implied volatilities, may be a way a worthwhile endeavor. As computation of IVs is generally more delicate and may result in breakage of computation, this approach may prove fruitful. Currently, the calibration problem has not only to learn how parameters map to option prices, via the rBergomi model, but also how these generated option prices relate to implied volatilities through the Black Scholes model. In order to reduce computational complexity and reduce the problem for the calibration meaningfully, calibrating to linear log prices seems reasonable. Unfortunately, calibration to both a single option chain

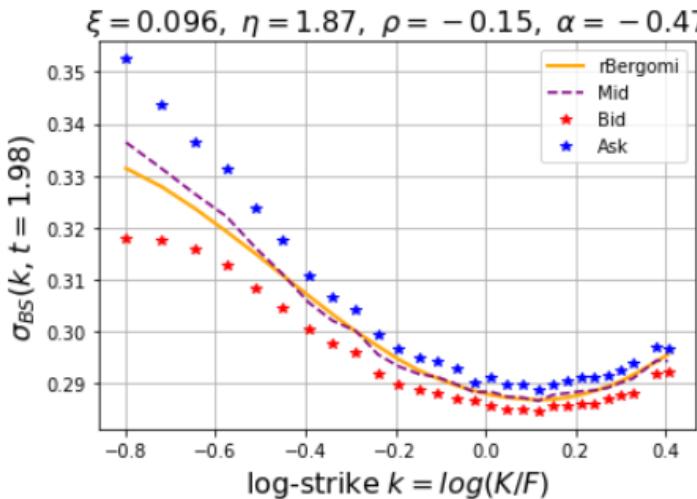


Figure 65: Showing fit from calibration to mid-prices with the Nelder-Mead minimizer

as well as to the full surface via the optimization slice method resulted in less accuracy as can be observed from Figure 65 and Figure 66. The respective summaries in Table 13 and Table 14 show the complete picture. For a single expiration, shown in Figure 65, the resulting fit was both (marginally) worse and took longer to calibrate (Table 14). We can however observe that

the estimated parameters are not greatly different. In fact, the smile dynamics as well as the level seem well replicated, though not as well as with the more traditional methods explored.

Minimizer	MSE	Runtime	Parameters ( $\alpha, \eta, \rho, \xi$ )
Nelder Mead	9.8524823041e-07	12:35:75 Minutes	-0.46006768 1.82672824 -0.14831442 0.09617495
Nelder Mead (Calib. Prices)	0.00207738879732	17:27:21 Minutes	-0.46566168 1.86640719 -0.1520598 0.09554467

Table 13: Nelder Mead Minimization Single Expiration: Calibration by implied volatilities vs option prices

For the full surface, the calibration to the optimization slice was done in less time compared to the more traditional method of calibrating to implied volatilities. However, this did not outweigh the significantly reduced fit compared to the standard Nelder-Mead minimization

(Table 14). Also observe that this time the estimated smile dynamics (controlled by  $\eta$  and  $\rho$ ) are quite distinct.

Minimizer	MSE (Optimization Slice)	MSE (Full Surface)	Runtime	Parameters ( $\alpha, \eta, \rho, \xi$ )
Nelder Mead	0.003932540442	0.003014108827	84:57:20 Minutes	-0.49999999 0.83515963 -0.93144967 0.06974231
Nelder Mead (Calib. Prices)	0.056179448439	0.186212929910	49:15.76 Minutes	-0.49999909 3.14690818 -0.1521208 0.106366

Table 14: Nelder Mead Minimization Full Surface: Calibration by implied volatilities vs option prices

The visualization in Figure 66 then shows what we were trying to avoid: Breakage of computation for multiple expirations highlighted by IVs of zero:

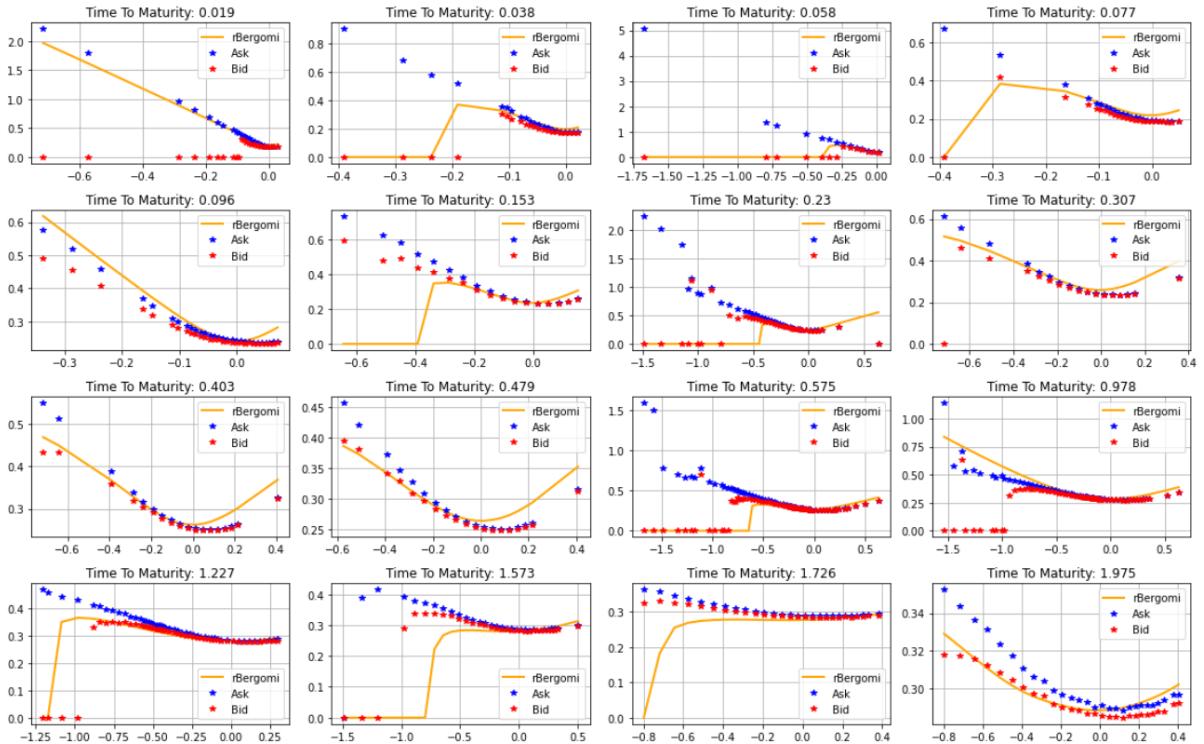


Figure 66: Showing the calibration by prices to the whole surface by optimization slice. Minimizer used: Nelder-Mead

## Appendix E:

### Appendix E1:

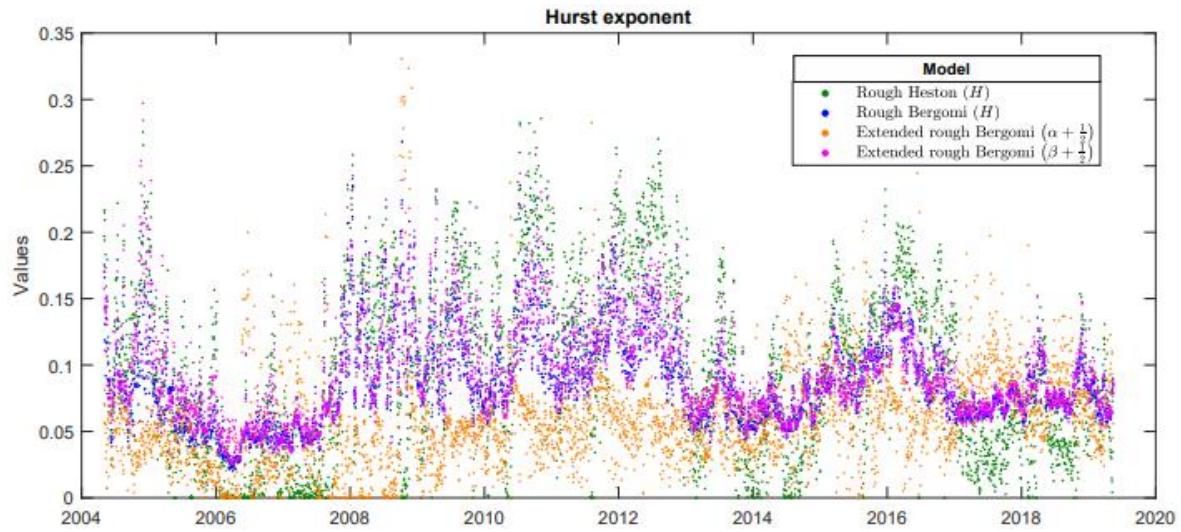


Figure 67: The Hurst parameter through time as estimated by (Rømer, 2020). Taken from (Rømer, 2020, p18)

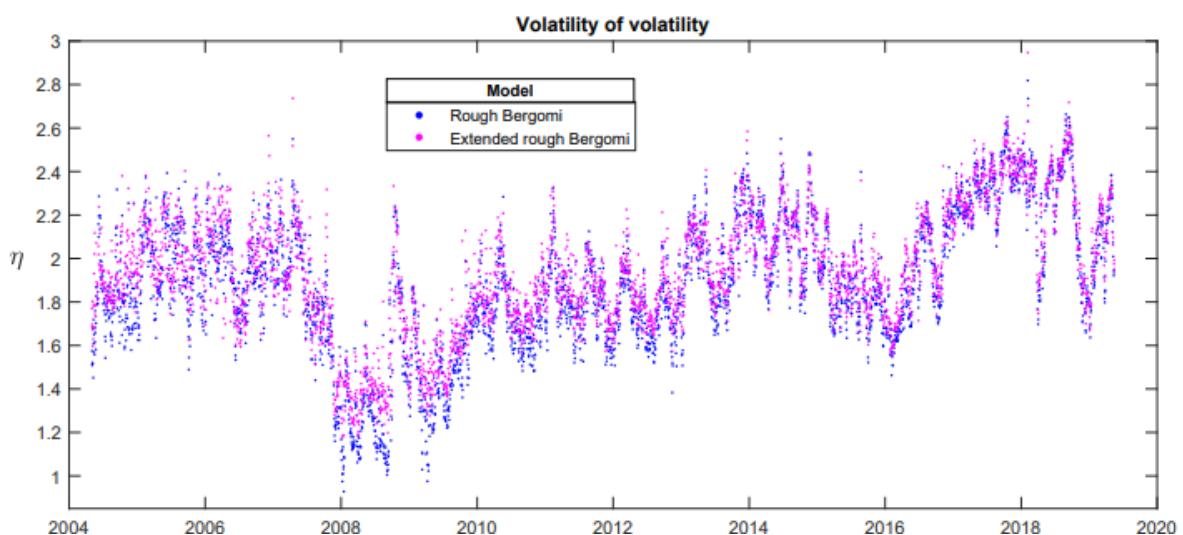
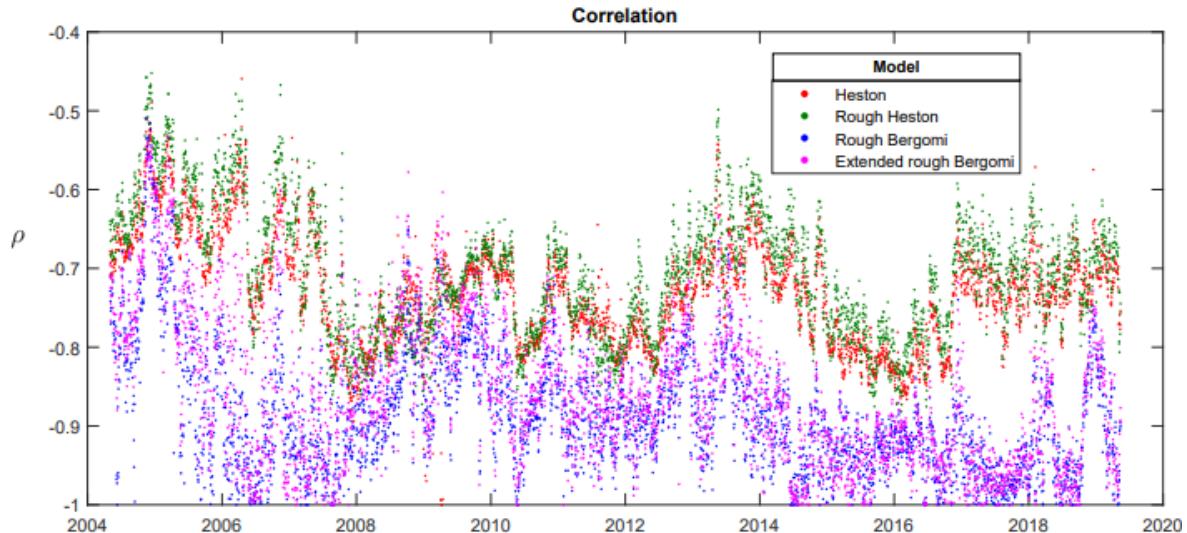


Figure 68: Correlation and Volatility of Volatility through time as estimated by (Rømer, 2020). Taken from (Rømer, 2020, p19)

### Appendix E3:

Having extracted the parameters used for calibrating each maturity individually (see Figure 40), we can visualize the change in parameters through time for this one volatility surface in Figure 69

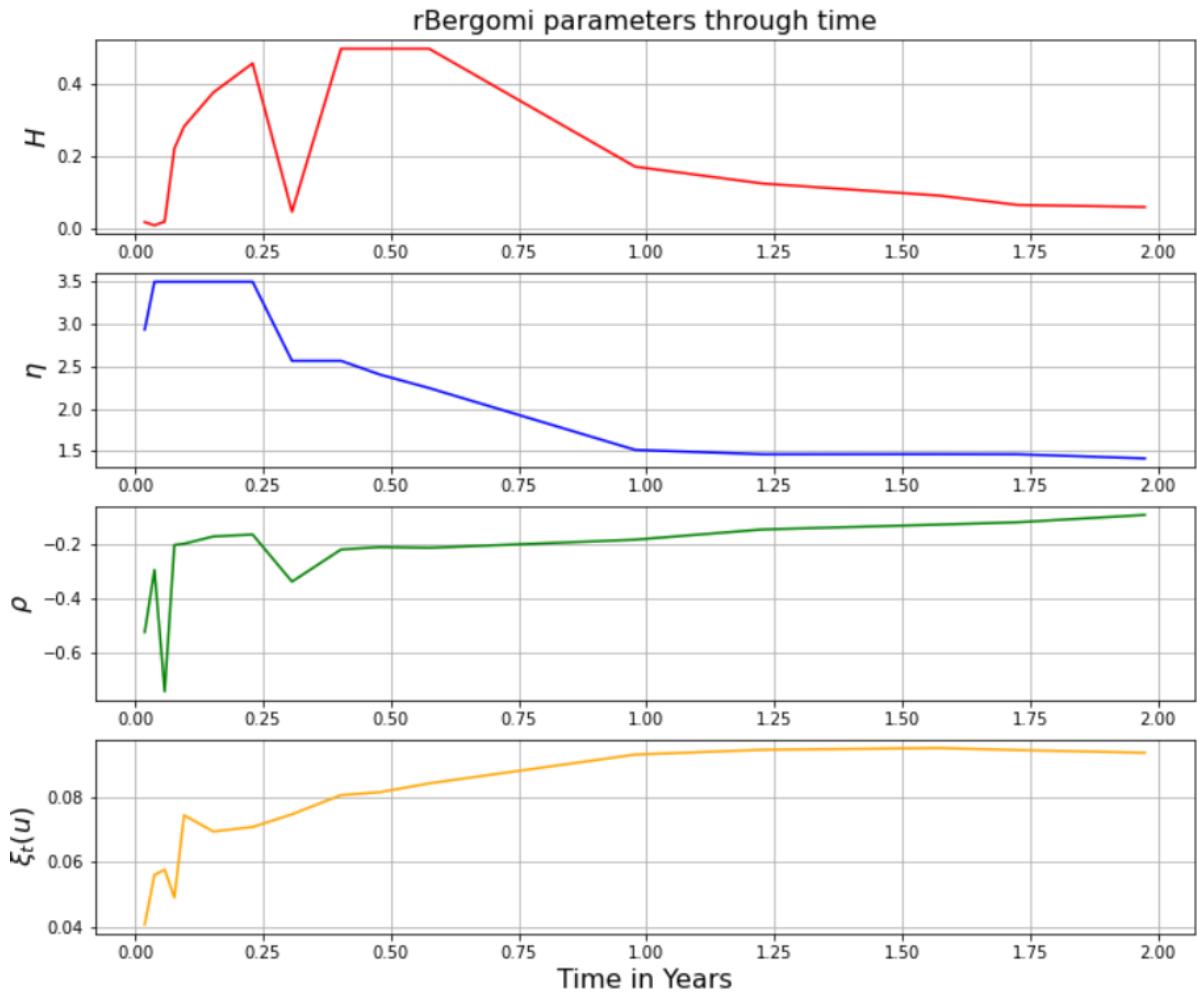


Figure 69: rBergomi parameters through time as extracted from our individual calibration exercise

Here we see how roughness is greatest for long dated maturities but also for shortest time to maturity. Also  $\eta$  decays through time, while the correlation  $\rho$  stays relatively flat when looking at medium to long expirations. Interestingly the forward variance estimation spikes and then increases steadily with maturity. Now we attempt to match our observed IVs by averaging the first three parameters and using the floating forward variance. The result is shown in Figure 70. We observe the fit becomes gradually worse until for long dated expirations the fit becomes so poor that we cannot speak of calibration in any true meaning of the word.

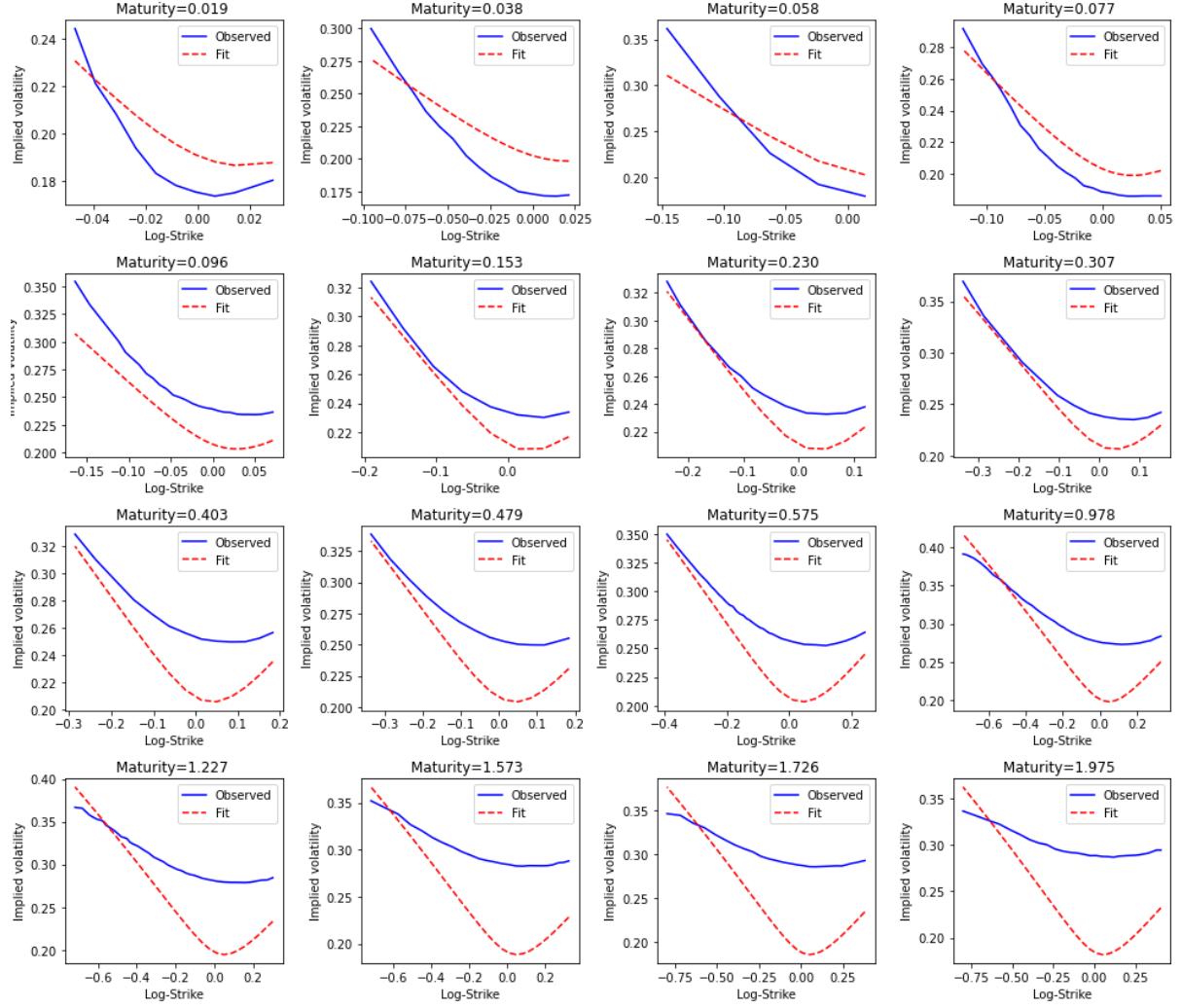


Figure 70: Calibration with floating forward variance and meand parameters as extracted from individual calibration exercise

For our next attempt, we use only the forward variance estimate from our individual calibration example (Figure 40) but use the estimates for  $H, \eta, \rho$  stemming from the initial calibration exercise, where we calibrated to the entire surface as a whole. In Figure 71 we observe the fit improving compared to Figure 70 however it is still worse than our initial fit in Figure 39. Table 15 shows the evaluation of the fit with help of the MSE. The data then confirms our observations: Both calibration exercises did not yield any improvement in accuracy. Therefore we can say that the parameters estimated via the individual calibration (Figure 69) are only usable for each respective maturity. While they surely help to understand how the parameters behave through time, they are unfit to gain any insight into what the true parameters are that calibrate an entire surface.

Calibration via Neural Network	MSE (Full Surface)
Whole Surface (fixed forward variance)	0.216495265787
Individual Maturities (fixed forward variance)	0.003218926366
Parameters from individual maturities (floating forward variance)	0.701304732814
Parameters from whole surface (floating forward variance)	0.409074390479

Table 15: MSE for different calibration exercises with the neural network by Rømer

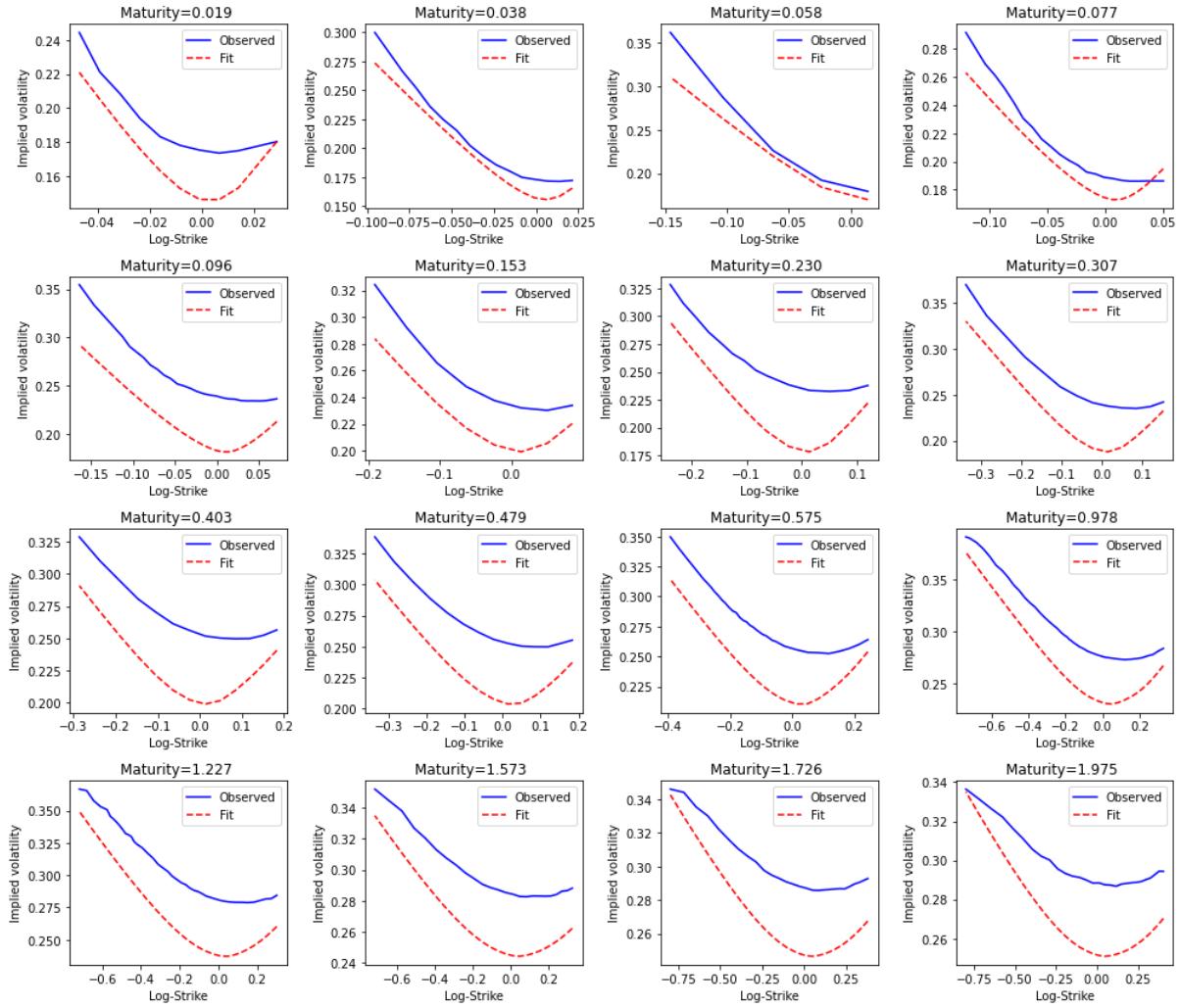


Figure 71: Calibration with floating forward variance and parameters as extracted from calibration to whole surface

### Appendix E3:

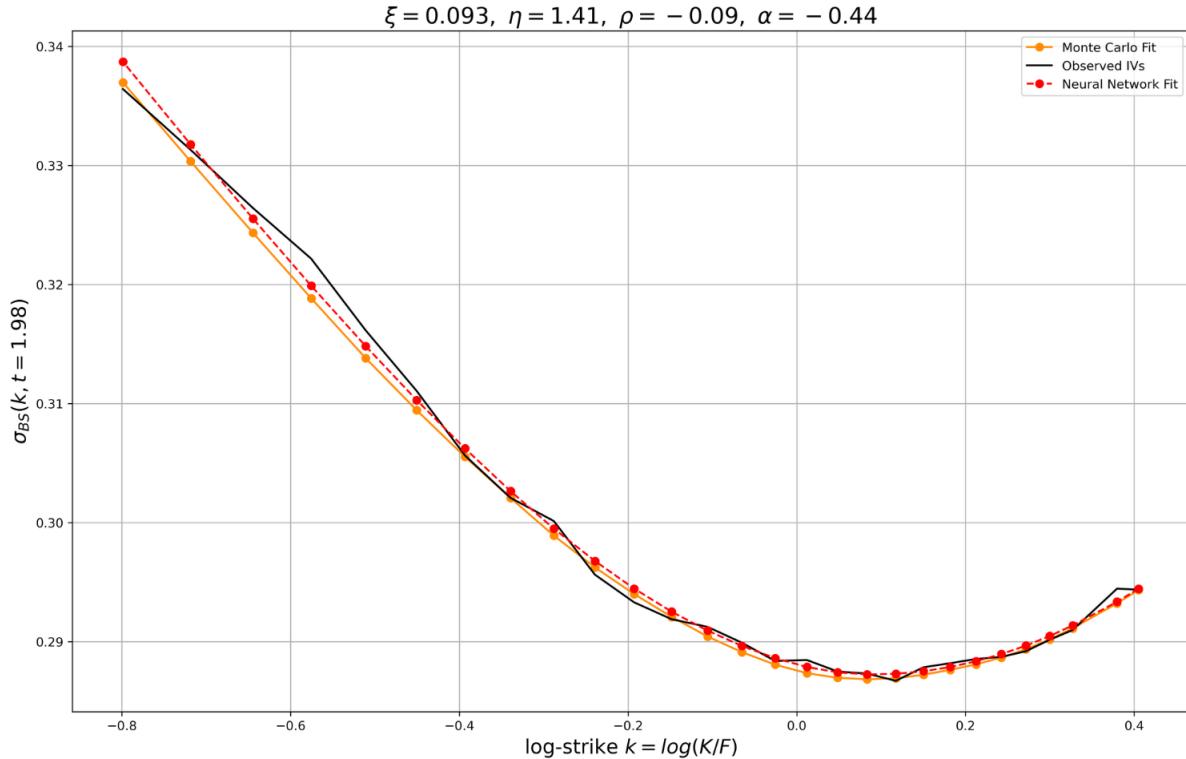


Figure 72: Neural Network VS Monte Carlo Simulation: Accuracy in mapping parameters to IVs

As the Neural Network is only able to assess fit within its constrained bounds, we propose the following procedure comparing both methods for a single expiration: First we let the NN calibrate to the observed maturity. With the resulting parameters we then evaluate the fit for both the NN and the MC simulation. We can see the MC simulation with 50.000 price paths matching the calibration by the neural network almost exactly<sup>133</sup>. Below that number we found significant deviations in simulated IVs for OTM and ITM contracts, stemming from misestimation of the smile dynamics controlled by  $\eta$  and  $\rho$ . Above 50.000 paths we only found a minuscule difference in resulting implied volatilities. As such we may say that with 50.000 paths the Neural Network and Monte Carlo simulation are able to achieve comparable fits for a given set of parameters and that both methods are viable alternatives in terms of their accuracy for a single expiration<sup>134</sup>.

<sup>133</sup> Slight differences may be due to interpolation errors of the neural network

<sup>134</sup> Of course, this also means that we can safely price options with the MC method, when calibrating with the faster neural network.