

**Imperial College
London**

IMPERIAL COLLEGE LONDON

DEPARTMENT OF MATHEMATICS


**The Rough Bergomi Model: From
Motivation to Implementation**

Author: CALLUM ROUGH (CID:01333836)

A thesis submitted for the degree of
MSc in Mathematics and Finance, 2022-2023

Declaration

The work contained in this thesis is my own work unless otherwise stated.

Signature and date:  Callum Kim Ming Rough, 04/09/2023

Acknowledgements

I would like to thank my thesis supervisor Dr. Ioannis Gasteratos for all of his time and support. His supervision was invaluable throughout the entirety of this project, and his insights helped to shape many of the ideas proposed throughout this thesis. Furthermore, I would like to thank my course tutor Dr. Antoine Jacquier for his support throughout the entirety of my master's course.

I would also like to thank Thibault Rivière and Herold Takodjou, from JP Morgan Chase, for all of their support, time, and encouragement. Thibault and Herold both provided me with the education and insights necessary to develop my skills and knowledge within the world of exotic equity derivatives and to gain an understanding of the practical implementation of the topics covered within this thesis.

Finally, I would like to thank my friends and family for their continued support for all of my endeavours.

Abstract

Current volatility models fail to both simultaneously capture the true shape of the implied volatility surface, and deliver realistic dynamics for the volatility surface. The Rough Bergomi model, a popular rough volatility model, attempts to fulfil both of these criteria and is able to fit a wide range of volatility surfaces with just 3 parameters, outperforming most conventional Brownian motion-based stochastic volatility models. Rough volatility models, first popularised by Gatheral et al. [1], replace the traditional Brownian motion within an assets variance process, with a fractional Brownian motion. The development of rough volatility models, their simulation, and their calibration has become popular as of late within quantitative finance. In this work, we present a thorough investigation, implementation, and calibration of a Rough Bergomi model. We begin with a summary of the motivation behind rough volatility models, before covering the pathway towards the specification of Bayer et al.'s [2] Rough Bergomi model. We subsequently delve into the requisite theory of fractional calculus and fractional Brownian motion, the process driving rough volatility models. Finally, we present calibration methodologies for the Rough Bergomi model, including a novel grids-based framework, and subsequently the results of these calibration schemes to the SPX index.

Contents

1	Introduction	7
2	Preliminaries	8
2.1	Brownian motion	8
2.2	Volatility Modelling	8
2.2.1	Implied Volatility	9
2.2.2	Features of the Implied Vol Surface	9
2.3	Useful Products	10
2.3.1	Cliquet Options	10
2.3.2	Variance Swaps	10
3	Motivation	12
3.1	The Black-Scholes Model	12
3.2	Local Volatility	12
3.3	Other Stochastic Volatility Models	13
3.3.1	Heston Model	13
3.4	Term Structure of the ATM Forward Skew	14
4	Fractional Brownian motion	16
4.1	Definition and Elementary Properties	16
4.1.1	Self-Similarity	17
4.1.2	Stationary Increments	17
4.1.3	Dependence of Increments	18
4.1.4	Additional Properties	19
4.2	Hölder Continuity	20
4.2.1	Continuity of Fractional Brownian Motion	20
4.3	Hurst Index	21
4.4	Integral Representations	21
4.4.1	Riemann-Liouville operators	23
4.4.2	The Mandelbrot-Van Ness Representation	23
4.4.3	Canonical Representation on \mathbb{R}_+	25
4.4.4	Molchan-Golosov representation	25
4.5	Type 2 Fractional Brownian motion	26
4.5.1	Volterra Gaussian processes	26
4.5.2	The Riemann-Liouville processes	27
5	Additional Motivation: Assessing the Roughness of Volatility	29
5.1	Smoothness of a process	29
5.1.1	Besov Spaces	29
5.1.2	Regularity Measures	30
5.2	Realised Volatility and Variance	31
5.3	Analysis of the volatility process	32
5.3.1	Framework of the Investigation	32
5.3.2	Empirical Calculations	32

6	The Rough Fractional Stochastic Volatility Model	36
6.1	The Fractional Stochastic Volatility Model	36
6.2	RFSV Model	37
6.2.1	Specification of the RFSV model	37
6.3	Comparison with the FSV Model	38
7	The Rough Bergomi Model	40
7.1	The Path to the Rough Bergomi Model	40
7.1.1	Change of Measure	41
7.2	The Rough Bergomi Model	42
7.2.1	The Bergomi Model	43
7.2.2	1-factor model	43
8	Simulation Techniques	45
8.1	Preliminaries	45
8.1.1	Monte Carlo Simulations	45
8.1.2	Dependence Structure	45
8.2	Simulating the Riemann-Liouville process	46
8.2.1	Cholesky Decomposition Method	47
8.2.2	Hybrid Scheme Method	48
8.2.3	Rough Donsker Scheme	50
8.2.4	Comparing Simulation Methods for the Riemann-Liouville process	51
8.2.5	Choice of Riemann-Liouville process simulation technique	52
8.3	Simulating the Rough Bergomi Model	52
8.3.1	Pricing Multiple Options	53
8.3.2	Parallelisation	53
8.3.3	Variance reduction - "Turbocharging" Monte Carlo	54
8.4	Market Information	55
8.4.1	Forward Curve	55
8.4.2	Variance Swap Curve	56
8.4.3	Discount Curve	57
8.4.4	Implied volatility surface	57
9	Parameters of the Rough Bergomi Model	58
9.1	Exploring the Effect of the Rough Bergomi Parameters	58
9.1.1	The effect of H	58
9.1.2	The effects of ρ and η	62
10	Calibration of the Rough Bergomi Model	65
10.1	Existing Methods	65
10.1.1	Variance Swap Calibration	65
10.2	Calibration Setup & Cost function	66
10.3	High Variance Issues	67
10.4	Minimisation Algorithms	68
10.5	SLSQP Minimisation	68
10.6	Sequential Refined Grids Calibration Method	70
11	Calibration results	74
11.1	SLSQP Calibration	74
11.2	Sequential Refined Grids Calibration	79
11.3	Comparison of SLSQP and SRFG methods	83
A	Additional Figures	86
A.1	Effect of Rough Bergomi Parameters	86
A.2	Cost Function Heat Maps	88
	Bibliography	91

List of Figures

3.1	Plot of the term structure of ATM skew for the S&P 500 Index, as of June 20, 2013. The black dots are non-parametric estimates of the S&P ATM volatility skews, the red curve is a power-law fit $\psi(\tau) = A\tau^{-0.4}$ [1, Figure 1.2 Page 5]	14
4.1	Paths of fBm for varying value of H [3, Figure 1 Page 5].	22
5.1	Plot of the Daily Realised Volatility of the S&P 500 Index, between January 3rd 2000, to the June 27th 2018	33
5.2	Plot of estimates of $\log m(q, \Delta)$ against $\log \Delta$ for various q , with accompanying linear regression lines, for SPX data	33
5.3	Plot of ζ_q against q , with accompanying linear regression line, for SPX data	34
9.1	Implied volatility smiles generated by the Rough Bergomi Model (Hybrid scheme method), with varying values of H , and fixed $\rho = -0.9$, $\eta = 1.9$, $\xi = 0.055$	59
9.2	Term structures of Rough Bergomi (Hybrid scheme method) ATM forward skew, with varying values of H , and fixed $\rho = -0.9$, $\eta = 1.9$, ξ constructed from SPX Data from the 30th May 2022 (Note: The shortest time to maturity considered here is 5 business days).	60
9.3	Plots of the approximate ATM forward skew as a function of H , for varying τ . Each plot has the value of H that maximises $\psi(\tau)$. These values were generated using equation (9.1.1), with $\rho = -0.9$, $\eta = 1.9$, $\xi = 0.04$ (flat forward variance curve) . . .	61
9.4	Plot of $H^*(\tau)$ as a function of τ , considering both $H^* \in (0, \frac{1}{2})$ and $H^* \in (0, 1)$. These values were generated using equation (9.1.1), with $\rho = -0.9$, $\eta = 1.9$, $\xi = 0.04$ (flat forward variance Curve)	62
9.5	Implied Volatility Smiles generated by the Rough Bergomi Model (Hybrid scheme method), with varying values of ρ or η . Note: Smiles for a wider range of maturities are included in Appendix A.1, they have been omitted here as a single maturity is sufficient to show the desired effects.	63
9.6	Plot of D_H against H	64
10.1	Cost function surfaces for fit to the SPX implied volatility surface on 30th May 2022, using grid G_1 , for varying parameter values. The Hybrid scheme method was used to simulate the Rough Bergomi model which was utilised for these cost function evaluations.	69
10.2	Visual representation of the SRFG Calibration Method of H, ρ, η on the parameter domain \mathcal{D} represented by a parameter cube. Here we have utilised 2 calibration cycles, with 2 grids per cycle.	73
11.1	Plots of volatility smiles for varying maturities, for both market implied volatilities, and SLSQP calibrated Rough Bergomi model delivered implied volatilities. The Rough Bergomi model was simulated with the following parameters obtained from the SLSQP calibration to the SPX volatility surface on the 30th May 2022: $H = 0.0996$, $\rho = -0.848$, $\eta = 1.991$	77
11.2	Term structures of ATM forward volatility and skew from SLSQP calibrated Rough Bergomi Model, with $H = 0.0996$, $\rho = -0.848$, $\eta = 1.991$. Market data from the SPX volatility surface on the 30th May 2022 was used to plot market estimates for ATM forward volatility and skew, and for calibration of the Rough Bergomi model.	78

11.3	Plots of volatility smiles for varying maturities, for both market implied volatilities, and SRFG calibrated Rough Bergomi model delivered implied volatilities. The Rough Bergomi model was simulated with the following parameters obtained from the SRFG calibration to the SPX volatility surface on the 30th May 2022: $H = 0.0946, \rho = -0.896, \eta = 1.944$	81
11.4	Term structures of ATM Forward volatility and skew from SRFG calibrated Rough Bergomi Model, with Rough Bergomi Model, with $H = 0.0946, \rho = -0.896, \eta = 1.944$. Market data from the SPX volatility surface on the 30th May 2022 was used to plot market estimates for ATM forward volatility and skew, and the market data was used for calibration.	82
A.1	Implied Volatility Smiles generated by the Rough Bergomi Model (Hybrid scheme Method), with varying values of ρ	86
A.2	Implied Volatility Smiles generated by the Rough Bergomi Model (Hybrid scheme Method), with varying values of η	87
A.3	Heat maps of cost function values for fit to the SPX implied volatility surface on 30th May 2022, using grid G_1 , for varying parameter values.	88

List of Tables

5.1	Hurst Parameter Estimates for Stock Indices	34
10.1	Table representing the choice of options in grid G_1 . Cells in green with an \mathbf{x} in them indicate options we will consider in G_1 , for the purpose of calibrating our model	66
10.2	Table of standard deviation of 20 Monte Carlo pricings of a call option with maturity of 5 Years and strike of 100% of the forward, for varying H . Each pricing utilised 200,000 paths, and the standard deviation is expressed as a % of the mean price for each batch of 20 pricings. Parameters used : $\rho = -0.7, \eta = 2$. SPX data from 30th May 2022 was utilised for these simulations.	67
10.3	Table of standard deviation of 20 evaluations of cost function $c(H, \rho, \eta, G_1)$, for varying H . Each pricing within a cost function evaluation utilised 200,000 paths, and the standard deviation is expressed as a % of the mean cost function value for each batch of 20 evaluations. The SPX volatility surface on the 30th May 2022 was utilised for the market volatilities within the cost function, and appropriate market data was also taken from this index on the specified date. Parameters used : $\rho = -0.9, \eta = 1.9$	68
11.1	Calibration results from SLSQP calibration of Rough Bergomi Model to SPX Implied volatility surface on the 30th May 2022, utilising options grid G_1	74
11.2	Table of absolute error of the SLSQP calibrated Rough Bergomi model delivered implied volatility estimates, compared to the market implied volatility values. Strikes are expressed as % of the forward price. In this simulation, the Rough Bergomi utilised the calibrated parameters, $H = 0.0996, \rho = -0.847, \eta = 1.991$, and 200,000 simulation paths were used. Market data and the implied volatility surface were taken from the SPX index on the 30th May 2022.	75
11.3	Table of mean absolute error and standard deviation of absolute error for each maturity considered in Table 11.2.	75
11.4	Table of mean absolute error and standard deviation of absolute error for each strike considered in Table 11.2.	76
11.5	Calibration results from SRFG calibration of Rough Bergomi Model to SPX Implied volatility surface on the 30th May 2022, utilising options grid G_1	79
11.6	Table of absolute error of the SRFG calibrated Rough Bergomi model delivered implied volatility estimates, compared to the market implied volatility values. Strikes are expressed as % of the forward price. In this simulation, the Rough Bergomi utilised the calibrated parameters, $H = 0.0996, \rho = -0.848, \eta = 1.991$, and 200,000 simulation paths were used. Market data and the implied volatility surface were taken for the SPX index on the 30th May 2022.	80
11.7	Table of Mean absolute error and standard deviation of absolute error for each maturity considered in Table 11.6.	80
11.8	Table of Mean absolute error and standard deviation of absolute error for each strike considered in Table 11.6.	81

Chapter 1

Introduction

In 2014 Gatheral et al. [1] published an analysis of the roughness of realised variance, and subsequently, they proposed a Rough Fractional Stochastic Volatility model. Building upon this, in 2015 Bayer et al. [2] proposed the Rough Bergomi model, a specific case of Gatheral et al.'s model. In this thesis, we shall explore the Rough Bergomi model, and present an implementation and calibration of the model to the SPX volatility surface.

Here, we shall briefly provide an overview of the various chapters within this work. We begin with Chapter 2, which details some of the relevant background theory and terminology used throughout the paper. Chapter 3 then proceeds to describe some of the relevant motivations for the development of rough volatility models, a class of volatility models of which the Rough Bergomi model is a part of. Chapter 4 introduces fractional Brownian motion and the theory surrounding it. This chapter is particularly important as fractional Brownian motion is the process driving rough volatility models. With this knowledge of fractional Brownian motion at hand, we proceed to Chapter 5, which details an investigation into the roughness of the realised variance process of several equity indices. With all of the motivation and theory covered, we then turn to the specification of the particular models we are going to explore. Chapter 6 briefly covers Gatheral et al.'s [1] Rough Fractional Stochastic volatility model, which can be viewed as both an extension of Comte and Renault's [4] model and a precursor to Bayer et al.'s Rough Bergomi model [2]. In Chapter 7 we introduce the main focus of this work, namely the Rough Bergomi model. We detail the pathway towards the development of the Rough Bergomi model and detail the specification of the model. With this model at hand, we explore relevant simulation techniques in Chapter 8, and in Chapter 9 we investigate the effects of the various parameters of the model. In Chapter 10 we detail 2 calibration methodologies for the Rough Bergomi model, including a novel grid based calibration method. We then proceed to calibrate the Rough Bergomi model to the SPX volatility surface, using both methodologies considered, and present the results in Chapter 11.

Chapter 2

Preliminaries

In this chapter, we shall briefly introduce certain concepts that will be used within this paper. This is not meant to be an exhaustive list of required theory, however, it is intended to serve as an introduction or reminder of key areas that we shall build upon.

2.1 Brownian motion

A 'standard Brownian motion', often called a Wiener process, is arguably the most important Stochastic process within mathematical finance. Brownian motion is used to model randomness or noise within a system, and is the process driving many of the stochastic differential equations (SDEs) we shall see within this paper. As it is such an important process, we begin by recalling the following definition:

Definition 2.1.1 (Brownian motion). The Brownian motion $(W_t)_{t \in [0, \infty)}$ is the continuous time stochastic process that satisfies the following:

- 1 $W_0 = 0$,
- 2 W_t is almost surely continuous,
- 3 W_t has stationary and independent increments,
- 4 The increment $W_t - W_s \sim \mathcal{N}(0, t - s)$ for all $0 \leq s \leq t$.

In addition to this definition, we state (without proof) the following two well-known facts of Brownian motion :

- 1 The quadratic variation of a Brownian motion, $[W_t, W_t] = t$,
- 2 W_t is a continuous Martingale.

2.2 Volatility Modelling

In 1973, Black and Scholes [5] published the Black-Scholes model, the associated Black-Scholes equation, and the closed-form Black-Scholes formulae. Their work revolutionised the derivatives pricing landscape of the global financial market. Derivatives exploded in popularity, and as a result derivatives pricing became a key focus of financial institutions worldwide.

When pricing derivatives, we often utilise a pricing approach based upon martingale theory, by modelling certain processes as martingales. One of the key assumptions of derivatives pricing, is that the discounted price-process of an asset is a martingale under the risk-neutral measure. With this in mind, we often model the logarithm of an asset's price as a continuous semi-martingale [1, Section 1.1 Page 1]. Given an asset S_t , the price process of the asset can be modelled as follows

$$dS_t = \mu_t S_t dt + \sigma_t S_t dW_t \tag{2.2.1}$$

where μ_t denotes the drift process, σ_t the volatility process, and W_t a one-dimensional Brownian motion.

The specification of the volatility process is central to determining the dynamics of the associated model. The volatility process not only affects the prices generated at inception, but also underpins the dynamics of the model as time progresses.

2.2.1 Implied Volatility

Option prices are often discussed with regard to their implied volatility. The implied volatility, or Black-Scholes implied volatility, is the unique value of the volatility parameter such that the Black-Scholes pricing formula is equal to the given price of a specific option [6, Section 1 Page 1]. The Black-Scholes formula for a European Call option is defined as follows:

$$C^{BS}(\sigma) := C^{BS}(S_t, t, K, T, \sigma) := S_t \mathcal{N}(d_+) - K e^{-r(T-t)} \mathcal{N}(d_-), \quad (2.2.2)$$

where

$$d_+ = \frac{1}{\sigma \sqrt{T-t}} \left[\ln \left(\frac{S_t}{K} \right) + \left(r + \frac{\sigma^2}{2} \right) (T-t) \right]$$

$$d_- = d_+ - \sigma \sqrt{T-t},$$

S_t is the price of the asset at time t , K is the options strike, T is the maturity of the option, t is the current time of pricing, and σ is the volatility parameter [6, Section 1.3 Page 3].

Now, given an observed call option price $C(K, T)$, the implied volatility for strike K and maturity T is defined as the value $\sigma^{BS}(K, T)$ that solves [6, Equation 1.6 Page 3]

$$C(K, T) = C^{BS}(K, T, \sigma^{BS}(K, T)).$$

This solution will be unique as the function C^{BS} is strictly increasing in σ .

Implied volatilities are used to calculate the volatility surface of a certain asset within a specific market. To construct such a surface, implied volatilities are calculated from a range of liquid options traded in the market. These implied volatilities can then be plotted on a grid of maturity against strike (or moneyness). These points can then be used to construct a surface, which can be used to calibrate various volatility models, with certain models requiring a smooth arbitrage-free surface for calibration, and others making do with a bare grid of points in space. The surface is used for calibration, as for a model to generate prices consistent with those in the market, its model-generated implied volatility surface must match the implied volatility surface of the market.

2.2.2 Features of the Implied Vol Surface

Certain features of the shape of the implied volatility surface are of particular interest within the field of volatility modelling. A feature of the implied volatility surface that can be particularly difficult for certain models to replicate is the term structure of the at-the-money (ATM) volatility skew. To explore this feature, we first need to define the following:

Definition 2.2.1 (Volatility Skew). The volatility skew is the partial derivative of the implied volatility with respect to strike

$$\phi(k, T) := \frac{\partial}{\partial k} \sigma_{BS}(k, T).$$

Definition 2.2.2 (Forward). Let S_t be the price of an asset at time t , whose dynamics follow (2.2.1) under the risk-neutral measure \mathbb{Q} . Then the forward at time t with maturity T , $F(t, T)$, can be expressed as

$$F(t, T) = \mathbb{E}^{\mathbb{Q}} \left[S_t \times \exp \left(\int_t^T \mu_s ds \right) \middle| \mathcal{F}_t \right] = S_t \times \mathbb{E}^{\mathbb{Q}} \left[\exp \left(\int_t^T \mu_s ds \right) \middle| \mathcal{F}_t \right].$$

Furthermore, in the case that μ_t is a deterministic drift term, we can remove the integral from the expectation, obtaining

$$F(t, T) = S_t \times \exp \left(\int_t^T \mu_s ds \right)$$

Definition 2.2.3 (At-The-Money (Forward) Volatility Skew). The ATM volatility skew, a function of $\tau := T - t$, is the partial derivative of the implied volatility with respect to strike, evaluated at log-moneyness equal to zero, that is $\ln(M) := \ln\left(\frac{F(t,T)}{K}\right) = 0$. The ATM (forward) volatility skew is defined as:

$$\psi(\tau) := \left. \frac{\partial}{\partial k} \sigma_{BS}(k, \tau) \right|_{\ln(M)=0}.$$

With this at hand, we can plot the term structure of the ATM volatility skew. The term structure of the ATM volatility skew is obtained by plotting the ATM forward skew against time to expiry τ . These plots show how the gradient of the surface, with respect to log-moneyness, varies with time to maturity.

Capturing the shape of the implied volatility surface at inception is important in order for model-generated vanilla prices to be consistent with the market. Furthermore, it is important for the model to deliver realistic implied volatility surfaces over time. In particular, when dealing with exotic equity derivatives, many products, such as cliquets and autocallables are sensitive to the dynamics of the implied volatility surface, thus proper dynamics of the model delivered implied volatility surface are integral to ensure proper pricing and hedging.

2.3 Useful Products

As alluded to above, certain exotic equity derivatives require specific features from a volatility model. As we shall soon see, the pricing of these products plays a large role in the motivation behind the development of rough volatility models. Moreover, for the purpose of the Rough Bergomi model, certain products are used to construct the forward variance curve which is taken as an input for the model. With this in mind, we shall briefly introduce some of these products and their associated payoffs.

2.3.1 Cliquet Options

A cliquet options is an exotic option which consists of a series of consecutive forward starting options. The first option within the cliquet is active at the inception of the product, and expires at a pre-determined maturity. Upon the expiry of the first option, the holder of the cliquet, then obtains a second option, with strike equal to the current price of the underlying. At the expiry of the second option, a third options contract is entered into, again with the strike being equal to the current asset value at the time of the second options expiration. The exact number of options and their payoffs vary depending on the exact specification of the cliquet product. In general, cliquet products can be thought of as a series of ATM options, which are pre-purchased, and which become active in turn.

The payoff structure of cliquet options makes them very sensitive to the dynamics of the implied volatility surface. As the holder of the option receives an option at a future time, the value of this option will be linked to the future implied volatility surface. Furthermore, the ATM forward skew is of particular importance, as the forward-starting options within a cliquet are typically ATM options. As a result of this dependence on the forward skew and forward volatility, two market models, which are both consistent with vanilla options prices at inception, can price cliquet style product very differently, if they produce different dynamics for the implied volatility surface.

2.3.2 Variance Swaps

A variance swap (VS) is a product that allows a market participant to directly expose themselves to the volatility of an asset. A VS involves two participants, with one side paying out the realised variance of an underlying, and the other paying out a predetermined strike, quoted at the inception of the swap. The strike of a variance swap is typically chosen to be the 'fair-strike' of the swap, which is that strike such that the initial value of the VS is zero. A VS is settled at the maturity of the swap, with one side receiving $(R \text{Var}_{0,T} - K_{\text{var}})$, and the other $(K_{\text{var}} - R \text{Var}_{0,T})$. Although variance swaps are products of realised variance, the fair strike of a variance swap is usually quoted in terms of volatility.

In the context of a VS, the realised variance of an asset is usually the annualised realised variance.

Definition 2.3.1 (Annualised Realised Variance). Consider a underlying asset with price process $(S_t)_{t \geq 0}$, observed at the close of trading each day for n trading days, over a period $[0, T]$, with $0 = t_0 < \dots, < t_n = T$. Then, the annualised realised variance over the n trading days is

$$R \text{Var}_{0,T} := \frac{d}{n} \sum_{i=1}^n \left[\log \left(\frac{S_{t_i}}{S_{t_{i-1}}} \right) \right]^2, \quad (2.3.1)$$

where d is the number of trading days per year, usually taken to be fixed as $d = 252$.

Now, denote by V_t^T the price of a variance swap with maturity T , viewed at time t . Then under \mathbb{Q} we have that

$$V_t^T = \mathbb{E}_t^{\mathbb{Q}} [R \text{Var}_{0,T}].$$

Variance Swaps are useful as the fair strikes of variance swaps within a market allow us to determine a forward variance curve. We shall expand on this idea later when we study the Bergomi and Rough Bergomi models, as the forward variance curve plays a key role in the dynamics of these models.

Chapter 3

Motivation

In this chapter we will explore some of the motivation behind the recent development of rough volatility models. We will primarily delve into some of the shortcomings of traditional volatility models, before also addressing certain features of the implied volatility surface that are problematic to fit.

Exotic derivative instruments are becoming increasingly popular financial instrument, and thus the pricing and risk management of such instruments has grown in importance within banks and other institutions. Exotic derivatives are hedged by using vanilla options as hedging instruments, in doing so the hedger lowers their exposure to realised volatility. However, the hedger acquires an exposure to the dynamics of the implied volatilities, as the value of their hedging instruments now depends upon these implied volatilities. Volatility models for exotic derivatives are therefore built with the aim of properly simulating the implied volatilities of vanilla options in a market, rather than to model the actual dynamics of realised volatility [7, Section 1.3 Page 18]

In order for a volatility model to properly simulate these implied volatilities, we need the model to both fit the market implied volatility surface at inception, and to predict realistic dynamics for the surface over time. We will begin by examining commonly used volatility models and assessing their strengths and limitations with regard to these criteria.

3.1 The Black-Scholes Model

In the Black-Scholes model, the underlying asset follows a Geometric Brownian motion. That is, given a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ supporting a one-dimensional Brownian motion $(W_t)_{t \geq 0}$. The asset price process $(S_t)_{t \geq 0}$ is the unique strong solution to the following SDE:

$$dS_t = rS_t dt + \sigma S_t dW_t, \quad S_0 > 0, \quad (3.1.1)$$

where $r > 0$ is a constant risk-free interest rate and $\sigma > 0$ is a constant instantaneous volatility. While the Black-Scholes model is elegant, it does not perform well in practice. Crucially, by modelling the volatility as a deterministic constant the Black-Scholes model delivers a flat implied volatility surface. The volatility surface delivered by the Black-Scholes model fails to capture any of the volatility smile or volatility skew that we see in market implied volatility surfaces, as the volatility of the Black-Scholes model varies neither with time, nor moneyness. As a result of this, option prices generated by the Black-Scholes model are inconsistent with most market-observed prices [6, Section 1 Page 1]. Though the Black-Scholes model is still widely used in certain areas of derivatives pricing, most notably to calculate the implied volatility of a particular option's market price, it has little to no place within the modelling of exotic products. Volatility is not deterministic, it is itself a stochastic process and should be modelled as such.

3.2 Local Volatility

The simplest and most widely used class of stochastic volatility model are local volatility models [7, Section 2.1 Page 25]. In a local volatility model, the local volatility $\sigma(S_t, t)$ is a deterministic function of the t and S_t . Local volatility is a 'stochastic volatility model', as the volatility depends on the spot price of the asset, which is itself stochastic. Local volatility was introduced as an

extension of the Black-Scholes model, which can easily be calibrated to the entire implied volatility surface at inception [7, Section 2.1 Page 25].

In a local volatility model, we have the following SDE for S_t :

$$dS_t = (r - q)S_t dt + \sigma(t, S_t)S_t dW_t \quad (3.2.1)$$

, where we now choose to include the effect of the borrow rate q . The dynamics of the asset within a local volatility model are synonymous with the Black-Scholes model, with the adjustment that $\sigma(t, s)$ replaces σ . Consequently, the pricing equation of a vanilla option is identical to that of the Black-Scholes model, with the aforementioned adjustment to the volatility term. Thus, we have :

$$\frac{dp}{dt} + (r - q)S \frac{dP}{dS} + \frac{\sigma(t, S)^2}{2} S^2 \frac{d^2P}{dS^2} = rP \quad [7, Equation 2.3 Page 26]. \quad (3.2.2)$$

From (3.2.2), given a choice of local volatility function $\sigma(t, S)$, we can price a vanilla option by setting $P(t = T, S)$ equal to the terminal payoff of the option, and then solving (3.2.2) backwards from T to t , to obtain $P(t, S)$ [7, Section 2.1.1 Page 26].

The local volatility function is chosen to match the implied volatility surface generated by the market prices of vanilla options of the desired underlying. The specification of the local volatility function is obtained through the use of market vanilla options prices and the Dupire Formulae, formulated by Bruno Dupire.

Theorem 3.2.1. *Let S_t follow (3.2.1), and $C(K, T)$ denote the price of a European Call option with strike K and maturity T , then*

$$\sigma(t, S)^2 = 2 \frac{\frac{dC}{dT} + qC + (r - q)K \frac{dC}{dK}}{K^2 \frac{d^2C}{dK^2}} \Bigg|_{K=S, T=t} \quad [7, Equation 2.3 Page 26]. \quad (3.2.3)$$

Proof. A proof of 3.2.1 can be found in Section 2.2.1 of Lorenzo Bergomi's book 'Stochastic Volatility modelling' [7, Section 2.2.1 Page 27]. \square

The characteristic feature of these models is that they have the ability to perfectly match any market-implied volatility surface provided that it is free of arbitrage. Unfortunately, the simplicity of the model comes at a price, with the model producing dynamics of the implied volatility surface that are highly unrealistic. The dynamics of the implied volatility surface generated by a local volatility model are entirely determined by the implied volatility surface it is fit to at inception. The model delivers implied volatility surfaces that change shape substantially as time passes, notably lacking a persistent skew. The surfaces generated by local volatility models do not fulfil a stylised fact of implied volatility surfaces, that although the level and orientation of the implied volatility surface may vary over time, the general shape itself is rather time invariant [1, Section 1.3 Page 4]. While this model is still widely used, especially for products that are not sensitive to the dynamics of the implied volatility surface, it is seldom used for products that are sensitive to these dynamics.

3.3 Other Stochastic Volatility Models

As volatility is not deterministic and is in fact stochastic, it is natural to attempt to model the volatility process of an asset, as a stochastic process. Stochastic volatility models attempt to capture the stochastic nature of volatility, by modelling the volatility process as a continuous Brownian semi-martingale [1, Section 1.1 Page 2]. Many different variations of stochastic volatility models have been developed and utilised, including the Heston Model, the Hull and White Model, and the Bergomi model.

3.3.1 Heston Model

The Heston model in particular is popular due to its tractability, given that the model presents a semi closed-form solution for the price of European call options. The Heston model, first published by Steven Heston in 1993 [8], relaxes the assumption of constant volatility, present in the Black-Scholes model, and instead utilises an instantaneous stochastic variance process. The asset and variance processes in the Heston model adhere to the following SDEs,

$$\begin{aligned}
dS_t &= rS_t dt + S_t \sqrt{V_t} dW_t^S, & S_0 = s \geq 0, \\
dV_t &= \kappa(\theta - V_t)dt + \nu \sqrt{V_t} dW_t^V, & V_0 = v_0 \geq 0, \\
d\langle W^S, W^V \rangle_t &= \rho dt,
\end{aligned}$$

where S_t represents the price of the asset at time t , V_t is the instantaneous variance at time t , r is the risk-free rate of return, θ the long term average variance, κ the speed of mean version, and ν the volatility of the variance process. Furthermore, W_t^V and W_t^S are two Brownian motions under the risk-neutral measure \mathbb{Q} , correlated with instantaneous correlation ρ [9, Chapter 2 Page 681]. The Heston model explicitly models the variance process as a mean-reverting process, which is consistent with stylised facts of realised variance of financial assets.

The Heston model is able to capture a wide range of smile and skew patterns, with its 4 parameters [10, Section 1 Page 3], however, the model is unable to fit a wide range of volatility surfaces with a degree of fit comparative to local volatility models.

More generally, stochastic volatility models are praised for their ability to deliver implied volatility surfaces with more realistic dynamics, in particular, delivering surfaces that retain their general shape as time passes. However, they often struggle to fully fit the market-implied volatility surface at inception. As a consequence of this, the prices for vanillas generated will not be consistent with the market, and thus hedges and subsequently prices of exotic options will not be modelled correctly. Moreover, stochastic volatility models are often unable to capture the steepness of the term structure of the ATM forward skew, for short expiry's [1, Section 1.3 Page 4].

3.4 Term Structure of the ATM Forward Skew

When assessing the goodness of fit of a stochastic volatility model to the market-implied volatility surface, we often focus on the fit of model-generated volatility smiles or volatility skews. The term structure of ATM skew is a particular feature that has proven to be intimately related to the underlying volatility dynamics [2, Section 1 Page 3], while also being notoriously hard to fit.

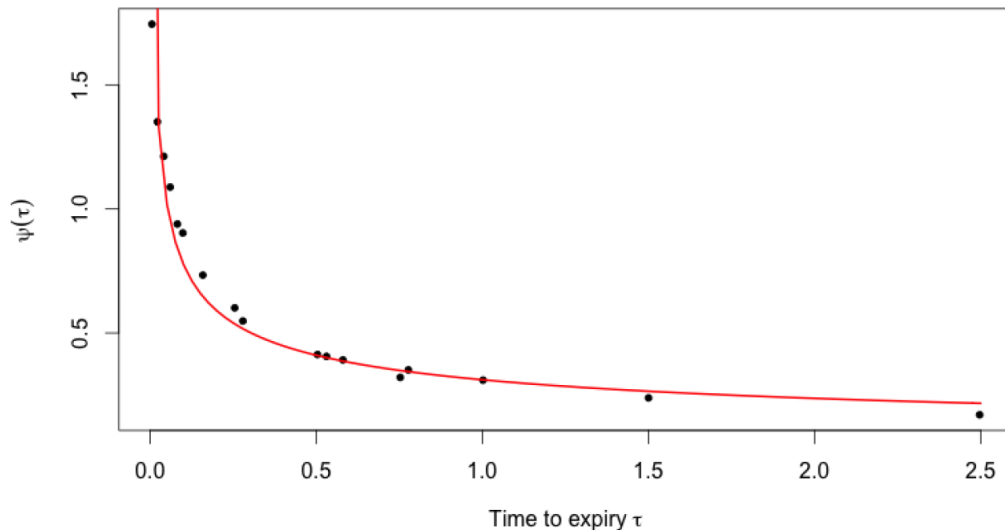


Figure 3.1: Plot of the term structure of ATM skew for the S&P 500 Index, as of June 20, 2013. The black dots are non-parametric estimates of the S&P ATM volatility skews, the red curve is a power-law fit $\psi(\tau) = A\tau^{-0.4}$ [1, Figure 1.2 Page 5]

Figure 3.1 from the paper 'Volatility is Rough' [1, Figure 1.2 Page 5] shows a typical shape of the term structure of the ATM skew, for equity volatility surfaces. We can observe that a power-law function of time to expiry τ provides a good approximation of the term structure, and in particular is able to capture the 'explosion' of skew as time to expiry approaches 0. Unfortunately, traditional stochastic volatility models, with a Brownian motion driving their variance processes, have as of

yet been largely unable to capture this rapid increase in ATM Skew as time to expiry approaches 0. Traditional stochastic volatility models usually generate a term structure that is "constant for small τ , and behaves as a sum of decaying exponentials for larger τ " [1, Section 1.3 Page 5]. The term structures generated by these models are unlike those we see within the market.

Interestingly, an analysis by Fukasawa [11, Section 3.3 Page 647], showed that a stochastic volatility model driven by a fractional Brownian motion with Hurst parameter H , generates an ATM forward volatility skew of the form $\psi(\tau) \sim \tau^{H-\frac{1}{2}}$, for small τ . Thus, a term structure generated by such a model for very small H could potentially fit the empirical term structure well, as it replicates the power-law behaviour we see. This also provides evidence to dispute the belief that the explosion in skew observed as $\tau \rightarrow 0$ requires a model with the presence of jumps.

From our analysis, we can see that neither local volatility nor traditional stochastic volatility models are able to simultaneously deliver realistic dynamics for the implied volatility surface, while also providing a satisfactory fit to the shape of the surface. Models such as Local-Stochastic Volatility have been proposed as a way to remedy this issue, these usually consist of a stochastic volatility model with a local volatility add-on, allowing the model to fit the surface perfectly while delivering appropriate dynamics. However, another approach is to search outside the realm of traditional Brownian motion-based stochastic volatility models. In particular recent work from Jim Gatheral and other prominent members in the field of quantitative finance have suggested that perhaps a traditional Brownian motion is not the correct driver for the volatility process. With this in mind, perhaps a model driven by a fractional Brownian motion could be the solution. This idea was the main motivation for the development of rough volatility models.

We conclude for now this overview of the motivation for rough volatility models. We will, in Chapter 5, present a final motivating factor. However, we now turn our attention to fractional Brownian motion and the theory surrounding it.

Chapter 4

Fractional Brownian motion

Fractional Brownian motion is the process driving fractional stochastic volatility models, a class of models that includes rough volatility models. In this chapter, we will delve into the theory of fractional Brownian motion. We begin with the definition of a fractional Brownian motion and then explore various characteristics and representations of fractional Brownian motion and some related processes.

4.1 Definition and Elementary Properties

Fractional Brownian motion was first introduced by Kolmogorov in 1940, and serves as a generalisation of the standard Brownian motion. It is widely used within financial mathematics (see [1],[2]), physics (see [12]), and network research (see [13]) among other areas, primarily as a model for both short-range and long-range dependence, or for systems involving Gaussian noise, where a standard Brownian motion will not suffice.

When considering fractional Brownian motion (fBm) we shall adopt the convention laid out by Picard in [14] and refer to both type 1 and type 2 fractional Brownian motions in turn. However, unless explicitly stated otherwise, any reference to 'fractional Brownian motion' refers to a type 1 fBm, as this is the prevailing 'standard' type.

Definition 4.1.1 (Gaussian process). A process $X = \{X_t, t \in \mathbb{R}_{\geq 0}\}$ is said to be Gaussian if for any $t_1, \dots, t_n \in \mathbb{R}_{\geq 0}$, we have that X_{t_1}, \dots, X_{t_n} has a multivariate Normal distribution. X_t is called a centred Gaussian process if $E(X_t) = 0$ for all $t \geq 0$.

Definition 4.1.2 (Fractional Brownian motion). Let (Ω, \mathcal{F}, P) be a complete probability space. The (type 1) fBm with Hurst parameter/index $H \in (0, 1)$ is the centred Gaussian process $B^H = \{B_t^H, t \in \mathbb{R}\}$ on (Ω, \mathcal{F}, P) , having the following properties [15, Section 1.2 Page 7]

(i) $B_0^H = 0$,

(ii) $E(B_t^H) = 0, t \in \mathbb{R}$,

(iii) $E(B_t^H B_s^H) = \frac{1}{2} (|t|^{2H} + |s|^{2H} - |t - s|^{2H}), s, t \in \mathbb{R}$.

Remark 4.1.3. In order to specify the distribution of a Gaussian process, it is sufficient to specify its mean and covariance function, this fact will be used to prove some of the properties of a type 1 fBm. Furthermore, as type 1 fBm is a Gaussian process, we have that, for each fixed H , the distribution of B^H is uniquely determined by the definition of type 1 fBm. However, this does not guarantee the existence of the fBm. In order to show the fBm exists we need to show that the covariance function is non-negative definite [3, Section 2.2 Page 2]

Remark 4.1.4. In the case of $H = \frac{1}{2}$, the covariance function is $E(B_t^{\frac{1}{2}} B_s^{\frac{1}{2}}) = t \wedge s$, therefore $B^{\frac{1}{2}} = W$ is a "standard Brownian motion", otherwise known as a Wiener process. Thus, we can see that fBm is simply a generalisation of the widely used Wiener process, where the increments of the process are permitted to no longer be independent of each other. Moreover, we must note

that for $H \neq \frac{1}{2}$, B^H is neither a martingale nor a Markov process, and it is not a semimartingale (we shall return to this fact later).

From (iii) of definition 4.1.2, we can obtain the following representation for the covariance structure of the increments of fBm,

$$\mathbb{E} [(B_{t_1}^H - B_{s_1}^H) (B_{t_2}^H - B_{s_2}^H)] = \frac{1}{2} (|t_1 - s_2|^{2H} + |t_2 - s_1|^{2H} - |t_2 - t_1|^{2H} - |s_2 - s_1|^{2H}). \quad (4.1.1)$$

We will proceed to show some elementary properties of the (type 1) fBm.

4.1.1 Self-Similarity

Definition 4.1.5 (Self-similarity). Let $b \geq 0$. Then, a stochastic process $X = \{X_t, t \in \mathbb{R}_{\geq 0}\}$ is called b -self-similar if for any $a \geq 0$

$$\{X_{at}, t \in \mathbb{R}_{\geq 0}\} \stackrel{d}{=} \{a^b X_t, t \in \mathbb{R}_{\geq 0}\}$$

with d referring to finite-dimensional distributions [15, Section 1.2.5 Page 7]. We can represent this definition concisely by writing

$$X_{at} \simeq a^b X_t.$$

To put it in less formal words, a self-similar stochastic process is one that has a probability distribution that is invariant under suitable scaling in time and space.

Proposition 4.1.6. *Type 1 fBm is H -self-similar.*

Proof. Consider for fixed $a \geq 0$, the process $Z_t = B_{at}^H$. Now using the definition of type 1 fBm, we have that

$$\begin{aligned} \text{Cov}(Z_t) &= \mathbb{E} (B_{at}^H B_{as}^H) = \frac{1}{2} (|at|^{2H} + |as|^{2H} - |a(t-s)|^{2H}) \\ &= \frac{a^{2H}}{2} (|t|^{2H} + |s|^{2H} - |t-s|^{2H}) \\ &= \text{Cov}(a^H B_t^H). \end{aligned}$$

Then as $Z_t = B_{at}^H$ and $a^H B_t^H$ have the same covariance structure and expectation, they are equal in distribution. Thus we have that B_t^H is H -self-similar. \square

4.1.2 Stationary Increments

Definition 4.1.7. A stochastic process $X = \{X_t, t \in \mathbb{R}_{\geq 0}\}$ is said to have stationary increments if for any $s, t \in \mathbb{R}$

$$(X_{t+s} - X_s) \stackrel{d}{=} X_t.$$

We think of a stochastic process with stationary increments as processes where the probability distributions of its increments are invariant under a shift in time.

Proposition 4.1.8. *Type 1 fBm has stationary increments.*

Proof. From the definition of type 1 fBm we have that

$$\begin{aligned} \mathbb{E} \left((B_t^H - B_s^H)^2 \right) &= \mathbb{E} \left((B_t^H)^2 \right) + \mathbb{E} \left((B_s^H)^2 \right) - 2\mathbb{E} (B_s^H B_t^H) = t^{2H} + s^{2H} - (t^{2H} + s^{2H} - |t-s|^{2H}) \\ &= |t-s|^{2H} \end{aligned}$$

so $B_t^H - B_s^H \sim N(0, |t-s|^{2H})$. Therefore, B^H has stationary increments. \square

Remark 4.1.9. B^H is not itself stationary, though it does have stationary increments.

4.1.3 Dependence of Increments

In this subsection will will show some properties of the dependence of the increments of fBm, and also exhibit when the fBm has long-range dependence (long-memory).

Consider the following discrete-time Gaussian process $X_n = B_n^H - B_{n-1}^H, n \in \mathbb{N}$, known as a fractional Gaussian noise, a generalisation of a traditional white noise corresponding to the case where $H = \frac{1}{2}$. Here X_k is a sequence of $N(0, 1)$ random variables that are not necessarily independent of one another.

If we consider $\rho_n = \mathbb{E}(X_{k+n}X_k)$, we can observe that it depends only on n , and not k . We have that

$$\begin{aligned}\rho_n &= E(X_{k+n}X_k) = E((W_{k+n}^H - W_{k+n-1}^H)(W_k^H - W_{k-1}^H)) \\ &= R_H(k+n, k) + R_H(k+n-1, k-1) - R_H(k+n, k-1) - R_H(k+n-1, k) \\ &= \frac{1}{2} [(n+1)^{2H} - n^{2H} - (n^{2H} - (n-1)^{2H})].\end{aligned}$$

Remark 4.1.10. Consider the function $g(n) = n^{2H}$, which is convex when $H \in (\frac{1}{2}, 1)$, and concave when $H \in (0, \frac{1}{2})$. Then, we can see that $\mathbb{E}(X_{k+n}X_k) > 0$ for $H \in (\frac{1}{2}, 1)$. In this case we see that X_{k+n} and X_k will be positively correlated, and we say that the process displays persistence. Conversely, for $H \in (0, \frac{1}{2})$ we see that $\mathbb{E}(X_{k+n}X_k) < 0$, and thus X_{k+n} and X_k are negatively correlated, and we say that the process displays anti-persistence.

Now using Taylor's theorem, with a small $\epsilon = \frac{1}{n}$ we can re-write ρ_n as follows

$$\begin{aligned}\rho_n &= \frac{1}{2}n^{2H} \left[\left(1 + \frac{1}{n}\right)^{2H} + \left(1 - \frac{1}{n}\right)^{2H} - 1 - 1 \right] \\ &= \frac{1}{2}n^{2H} \left[2H\frac{1}{n} + \frac{1}{2} \cdot 2H(2H-1)\frac{1}{n^2} - 2H\frac{1}{n} + \frac{1}{2} \cdot 2H(2H-1)\frac{1}{n^2} \right] \\ &\sim H(2H-1)n^{2H-2} \quad (n \rightarrow \infty) \\ &= 2\alpha H n^{2\alpha-1}.\end{aligned}$$

This is one way to derive the autocovariance function of the fractional Gaussian noise. Let us now consider an alternative derivation, presented by Mishura [15, Section 1.3 Page 8].

Consider $H \in (0, \frac{1}{2}) \cup (\frac{1}{2}, 1)$ and $t_1 < t_2 < t_3 < t_4$, then, it can be shown from (4.1.1) for $\alpha = H - 1/2$ that

$$E(B_{t_4}^H - B_{t_3}^H)(B_{t_2}^H - B_{t_1}^H) = 2\alpha H \int_{t_1}^{t_2} \int_{t_3}^{t_4} (u-v)^{2\alpha-1} dudv \quad [15, Section 1.3 Page 8].$$

From this, we can again see that the increments of fBm are positively correlated for $H \in (\frac{1}{2}, 1)$ and negatively correlated for $H \in (0, \frac{1}{2})$.

Then, taking $t_1 = 0, t_2 = 1, t_3 = n$, and $t_4 = n+1$, for any $n \in \mathbb{Z} \setminus \{0\}$, we have that the autocovariance function is given by

$$\begin{aligned}\rho_n &:= E[B_1^H (B_{n+1}^H - B_n^H)] = 2\alpha H \int_0^1 \int_n^{n+1} (u-v)^{2\alpha-1} dudv \\ &\sim 2\alpha H n^{2\alpha-1}, \quad n \rightarrow \infty \quad [15, Section 1.3 Page 8].\end{aligned}$$

With this expression for ρ_n at hand, we can see the long-range dependence properties of fBm.

- If $H \in (\frac{1}{2}, 1)$, then $\sum_{n=1}^{\infty} |\rho_n| \sim \sum_{n \in \mathbb{Z} \setminus \{0\}} |n|^{2\alpha-1} = \infty$. This is what is known as long-range dependence (long-memory).
- If $H \in (0, \frac{1}{2})$, then $\sum_{n \in \mathbb{Z}} |\rho_n| \sim \sum_{n \in \mathbb{Z} \setminus \{0\}} |n|^{2\alpha-1} < \infty$. In this case, the fBm does not exhibit long-range dependence.

4.1.4 Additional Properties

As alluded to earlier, a key property of fBm is that it is not a semimartingale. This is a particularly important property, as it prevents us from utilising regular Itô calculus when handling fBm.

To prove this fact, we first recall the definition of a semimartingale.

Definition 4.1.11 (Semimartingale). A process $X = X_t, t \geq 0$ with associated filtration \mathcal{F}_t is called a semimartingale if it admits the following representation:

$$X_t = X_0 + M_t + Y_t,$$

where X_0 is \mathcal{F}_0 -measurable, M is a \mathcal{F}_t -local martingale with $M_0 = 0$, and Y_t is a process of locally bounded variation.

Lemma 4.1.12. *A semimartingale X_t has bounded quadratic variation.*

With this at hand, we can address the non-semimartingale property of fBm.

Theorem 4.1.13. *Let B_t^H be a fBm with Hurst index $H \in (0, 1)$. Then, for $H \neq \frac{1}{2}$, B_t^H is not a semimartingale.*

Proof. Consider the interval $[0, T]$, for $T > 0$, and let B_t^H be a fBm with Hurst index H defined on said interval. Furthermore, let $\pi_n(T) := \{t_k^n = \frac{kT}{n} | 0 \leq k \leq n\}$, $n \in \mathbb{N}$ be a sequence of equidistant partitions of $[0, T]$.

For $p \geq 1$ the p -variation of B_t^H is

$$V_p(B) = \lim_{n \rightarrow \infty} v_p(B^H, \pi_n(T)),$$

where

$$v_p(B^H, \pi_n(T)) := \sum_{k=0}^{n-1} |B_{t_{k+1}^n}^H - B_{t_k^n}^H|^p.$$

Now, we present the next two formulae without proof. Using the self-similarity of B_t^H , we have that

$$v_p(B^H, \pi_n(T)) \stackrel{d}{=} \sum_{k=0}^{n-1} \frac{1}{n} T^{pH-1} \frac{|B_{t_{k+1}^n}^H - B_{t_k^n}^H|^p}{n} \quad [16, \text{Section 2.2 Page 3}].$$

And from this we have that

$$V_p(B) = \begin{cases} +\infty & \text{if } H < \frac{1}{p} \\ T \cdot \mathbb{E}_{\mathbb{P}}(|B_1^H|)^{\frac{1}{p}} & \text{if } H = \frac{1}{p} \\ 0 & \text{if } H > \frac{1}{p} \end{cases} \quad [16, \text{Section 2.2 Page 3}]. \quad (4.1.2)$$

The proof of the previous two relations requires the use of the Ergodic Theorem, and an understanding of Ergodic theory. For this reason, we have omitted these interim steps in the proof. A full proof can be found in Jost [16, Section 2.2 Page 3].

Now, we shall proceed using the fact that if $p < q$, then $V_q \leq V_p$.

We first consider the case where $H < \frac{1}{2}$. Choosing $p > 2$, from (4.1.2), we see the p -variation of B^H is infinite when $H < \frac{1}{p}$. This implies, in particular, that the quadratic ($p=2$) variation of B^H must be infinite. Hence the quadratic variation of B^H is unbounded, and thus by Lemma 4.1.12 B^H is not a semimartingale.

Now, turning to the case that $H > \frac{1}{2}$, we begin by assuming that the quadratic variation of B^H is finite. Then, taking $p \in (\frac{1}{H}, 2)$ so that $H > \frac{1}{p}$, and using (4.1.2), we see that the p -variation of B^H is 0. Therefore for any $p \geq 2$, the p -variation must be 0, and thus B^H has a quadratic variation of 0.

As B^H has a quadratic variation of 0, it must be a process of finite variation. However, if we now take $p \in (1, \frac{1}{H})$ such that $H < \frac{1}{p}$, we see that the p -variation of B^H is infinite, and thus B_t^H cannot have finite quadratic variation, thus we have a contradiction. Therefore our assumption

was false, and B^H must have unbounded quadratic variation. Then, again using Lemma 4.1.12 we have that B^H is not a semimartingale. □

4.2 Hölder Continuity

Traditional Brownian motion and fBm are part of a group of processes which are said to have 'rough paths'. These processes have paths that are continuous, but nowhere differentiable. When dealing with such processes of varying levels of continuity and smoothness, a crucial tool we can use is the notion of Hölder continuity. Hölder continuity is a property that a process or function can possess if it fulfils the Hölder condition.

Definition 4.2.1 (Hölder Condition). Let $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ be a function such that for real constants $C \geq 0$, $\alpha \in (0, 1)$, we have

$$|f(x) - f(y)| \leq C|x - y|^\alpha \quad (\text{Hölder Condition})$$

for all x and y in the domain of f . Then, f satisfies the Hölder condition with exponent α , and we say that f is α -Hölder continuous [17, Section 15.1 Page 143]. We note that when $\alpha = 1$ this is equivalent to the Lipschitz condition.

With this notion in mind, we can proceed to define Hölder spaces.

Definition 4.2.2 (Hölder Space). Let Ω be an open set in \mathbb{R}^n and $0 < \alpha < 1$. A function $f : \Omega \rightarrow \mathbb{R}^n$ is uniformly Hölder continuous with exponent α in Ω if

$$[f]_{\alpha, \Omega} = \sup_{\substack{x, y \in \Omega \\ x \neq y}} \frac{|f(x) - f(y)|}{|x - y|^\alpha}$$

is finite [18, Section 2.1 Page 283]. A function $f : \Omega \rightarrow \mathbb{R}$ is locally uniformly Hölder continuous with exponent α in Ω if $[f]_{\alpha, \Omega'}$ is finite for every $\Omega' \Subset \Omega$. We denote by $C^{0, \alpha}(\Omega)$ the space of locally uniformly Hölder continuous functions with exponent α in Ω [18, Section 2.1 Page 283], which we call a Hölder space. If Ω is bounded, we denote by $C^{0, \alpha}(\bar{\Omega})$ the space of uniformly Hölder continuous functions with exponent α in Ω . We will often use the shorthand notation C^α to refer to $C^{0, \alpha}(\bar{\Omega})$.

With these definitions at hand, we can investigate the continuity of fBm.

4.2.1 Continuity of Fractional Brownian Motion

Of the methods to derive the Hölder continuity of type-1 fBm, the most popular method is to utilise the following theorem.

Theorem 4.2.3 (Kolmogorov-Chentsov Continuity Theorem). *Let $\{X_t, t \geq 0\}$ be a Stochastic process. If for X_t there exist such $K > 0, p > 0, \beta > 0$ such that for all $t \geq 0, s \geq 0$*

$$\mathbb{E}[|X_t - X_s|^p] \leq K|t - s|^{1+\beta}$$

Then the process X has a continuous modification. That is a process $\{\tilde{X}_t, t \geq 0\}$ with the same index set, state space and probability space as X , such that $\tilde{X} \in C[0, \infty)$ and for all $t \geq 0, \mathbb{P}(X_t = \tilde{X}_t) = 1$. Furthermore, for any $\alpha \in (0, \beta/p)$ and $T > 0$ the process \tilde{X} is α -Hölder continuous on $[0, T]$ [19, Section 2.8 Page 53].

With this theorem in mind, we can determine the continuity of type 1 fBm.

Proposition 4.2.4. *The type 1 fBm B^H has a continuous modification. Furthermore, this modification is locally α -Hölder continuous for all $\alpha \in (0, H)$.*

Proof. For a type 1 fBm, we have that

$$E |B_t^H - B_s^H|^p = |t - s|^{pH} E [|B_1^H|^p]$$

thus taking a fixed $p := \frac{1+R}{H}$ for a fixed constant R , $K = E [|B_1^H|^p]$, and $\beta = pH - 1 = R$, we fulfil the Kolmogorov-Chentsov Continuity Theorem. Therefore B^H has an α -Hölder continuous modification for all

$$\alpha < \frac{\beta}{p} = H \frac{R}{1+R},$$

and as R is arbitrary we can obtain a modification for Hölder exponent $\alpha < H$. \square

As type 1 fBm has a modification which is α -Hölder continuous for $\alpha \in (0, H)$, we shall always choose to use this modification, and this will be the default from now onward.

Lemma 4.2.5. *Let $H \in (0, 1)$. Then the type 1 fBm B_t^H has sample paths that are almost surely Hölder continuous up to order $H - \epsilon$, for arbitrarily small ϵ .*

4.3 Hurst Index

We can now consolidate the information we have gained as to the effect of the Hurst index on fBm. The Hurst index controls the following:

- 1 **Dependence Structure** - The Hurst index controls the dependence structure of the increments of fBm. For $H < \frac{1}{2}$ we have that the increments of B^H are negatively correlated, and conversely for $H > \frac{1}{2}$ we have that the increments are positively correlated. Finally, in the case that $H = \frac{1}{2}$, B^H is a "standard Brownian motion" and the increments are independent.
- 2 **Memory** For $H > \frac{1}{2}$, B^H has long-range dependence (long-memory).
- 3 **Smoothness** The paths of B^H are almost surely α -Hölder continuous for $\alpha < H$.

From Figure 4.1 we can see a visual demonstration of how H affects the smoothness of the paths of fBm.

4.4 Integral Representations

It is often useful to be able to represent B^H as a linear function of a Wiener process. In order to do this we search for appropriate kernel functions $k_H(t, s)$ such that the Wiener-Itô integral

$$B_t^H = \int k_H(t, s) dW_s \tag{4.4.1}$$

is a fBm with Hurst parameter H .

In this section, we will show some representations of type 1 fBm, in terms of integrals of a standard Brownian motion, or in terms of a Gaussian measure. We must first recall the Gamma function which will be utilised in the representations to come.

Definition 4.4.1 (Gamma Function). For the purposes of this paper, we only need to consider the Gamma function on \mathbb{R} . The gamma function, $\Gamma(z) : \mathbb{R} \rightarrow \mathbb{R}$, is defined as follows

$$\Gamma(z) = \int_0^\infty e^{-t} t^{z-1} dt \quad [20, \text{Equation 5.2.1}] \tag{4.4.2}$$

Lemma 4.4.2. *For $z \in \mathbb{R}$,*

$$\Gamma(z + 1) = z\Gamma(z). \tag{4.4.3}$$

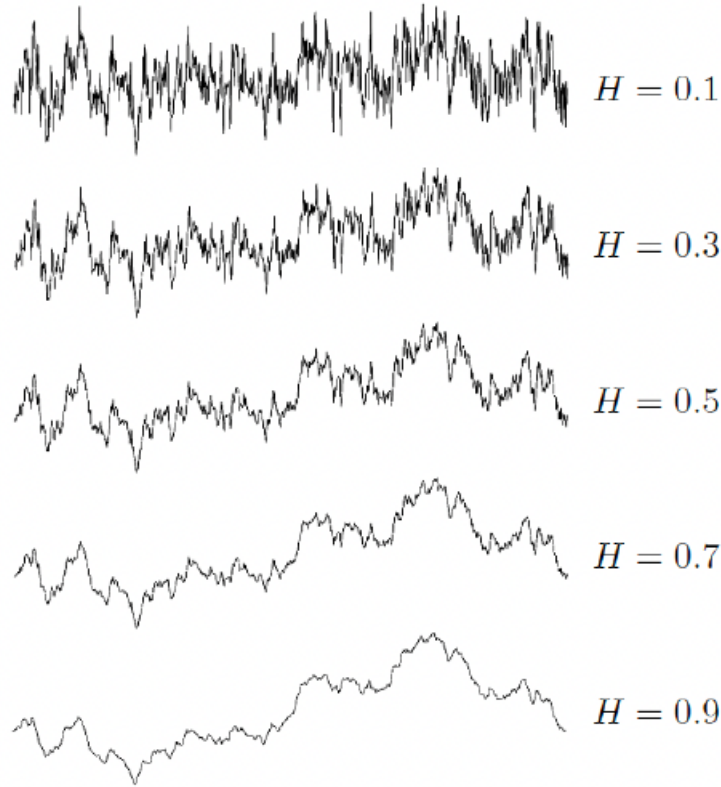


Figure 4.1: Paths of fBm for varying value of H [3, Figure 1 Page 5].

Proof. Starting from the definition (4.4.2) and then using integration by parts we have

$$\begin{aligned}
 \Gamma(z+1) &= \int_0^\infty e^{-t} t^z dt \\
 &= -e^{-t} t^z \Big|_{t=0}^{t=\infty} + z \int_0^\infty e^{-t} t^{z-1} dt \\
 &= z\Gamma(z).
 \end{aligned}$$

□

Lemma 4.4.3.

$$\Gamma(n+1) = n!, \quad n = 0, 1, 2, \dots \quad (4.4.4)$$

Proof. It is clear from (4.4.2) that $\Gamma(1) = 1$. Using this and (4.4.3) we can use mathematical induction to obtain the desired expression.

□

Before we begin exploring various integral representations of fBm, we will introduce the idea of a canonical representation. An integral representation of the form (4.4.1), is denoted as canonical if $\mathcal{F}_t(B^H) = \mathcal{F}_t(W)$. Therefore, a canonical representation is one in which the final generated process and the process driving the integral generate the same filtration, and can be said to be deduced from one another. We can also think of a canonical representation as one where B_t^H depends on past values W_s for $s \leq t$, or equivalently that the infinitesimal increment dB_t^H depends on past increments dW_s for $s \leq t$ [14, Section 1 Page 3]. We often seek canonical representations, and they are often found by restricting t to a subset of \mathbb{R} . Moreover, we often choose to restrict t in this way so that in order to produce our process B^H on an interval $[0, T]$, we only require values of W on the same interval.

4.4.1 Riemann-Liouville operators

The Riemann-Liouville operators I_{\pm}^{α} are important tools for the stochastic calculus of fBm. These operators can be defined for any real index α .

Operators with finite horizon

Definition 4.4.4. For $\tau \in \mathbb{R}$ and $\alpha > 0$, the Riemann-Liouville operators are defined as

$$I_{\tau+}^{\alpha} f(t) = \frac{1}{\Gamma(\alpha)} \int_{\tau}^t (t-s)^{\alpha-1} f(s) ds, \quad I_{\tau-}^{\alpha} f(t) = \frac{1}{\Gamma(\alpha)} \int_t^{\tau} (s-t)^{\alpha-1} f(s) ds,$$

respectively for $t > \tau$ and $t < \tau$ [14, Section 2.2.1 Page 10].

Remark 4.4.5. These Riemann-Liouville operators are fractional integral operators, where α represents what order of integral we are taking. Fractional integral operators are fractional-order generalisations of regular integral operators. In the case that $\alpha = 1$ we are simply calculating the anti-derivative, with a constant factor in front, in this case we simply have a standard integral operator. Fractional integral operators are one way of defining what $\frac{1}{2}$ an integral is, i.e what is inbetween a function and its anti-derivative. Furthermore, if we take $\alpha = 2$, we are in effect integrating our function twice. The Riemann-Liouville operators are part of a broader class of generalised fractional operators, for a more in-depth introduction to them see [21, Definition 1.1 Page 3].

Operators with infinite horizon

In the case of integral representations of fractional Brownian motion, we often utilise an infinite time horizon.

The infinite time horizon Riemann-Liouville operators I_{\pm}^{α} are defined by letting $\tau \rightarrow \mp\infty$ in $I_{\tau\pm}^{\alpha}$. However, we are often more interested in the modified operators, defined by

$$\tilde{I}_{\pm}^{\alpha} f(t) = I_{\pm}^{\alpha} f(t) - I_{\pm}^{\alpha} f(0) = \lim_{\tau \rightarrow \mp\infty} (I_{\tau\pm}^{\alpha} f(t) - I_{\tau\pm}^{\alpha} f(0))$$

when the limit exists [14, Section 2.2.2 Page 14].

4.4.2 The Mandelbrot-Van Ness Representation

Recall that the two-sided Wiener process is the Gaussian process with independent increments satisfying $E(W_t) = 0$ and $E(W_t W_s) = s \wedge t$, $s, t \in \mathbb{R}$.

In 1968 Mandelbrot and Van Ness [22] derived an expression for the two-sided fBm, in terms of the two-sided Wiener process.

Theorem 4.4.6 (Mandelbrot & Van Ness, 1968). *Let $W = \{W_t, t \in \mathbb{R}\}$ be the two-sided Wiener process, and let $k_H(t, u) := (t-u)_{+}^{\alpha} - (-u)_{+}^{\alpha}$, where $\alpha = H - \frac{1}{2}$. Then the process $\bar{B}^H = \{\bar{B}_t^H, t \in \mathbb{R}\}$ defined by*

$$\bar{B}_t^H := C_H \int_{\mathbb{R}} k_H(t, u) dW_u, \quad H \in \left(0, \frac{1}{2}\right) \cup \left(\frac{1}{2}, 1\right)$$

$$\text{where } C_H = \left(\int_{\mathbb{R}_+} ((1+s)^{\alpha} - s^{\alpha})^2 ds + \frac{1}{2H} \right)^{-\frac{1}{2}} = \frac{(2H \sin(\pi H) \Gamma(2H))^{1/2}}{\Gamma(H + 1/2)},$$

has a continuous modification which is a normalized two-sided fBm [15, Theorem 1.3.1 Page 9].

Proof. This shortened proof is based on the proof by Mishura [15, Theorem 1.3.1 Page 9].

The two-sided Wiener process is a Gaussian process, and therefore \bar{B}^H being an integral of the kernel function with respect to a Gaussian process, is also a Gaussian process. Furthermore, we can see that $\bar{B}_0^H = 0$ and $\mathbb{E}(\bar{B}_t^H) = 0$. Therefore, \bar{B}^H is a centred Gaussian process.

Now, to show that \bar{B}^H has stationary increments, we rely on the fact that the Wiener process has stationary and independent increments. Taking a constant $s > 0$, we calculate an increment of our process \bar{B}^H

$$\begin{aligned}
(\bar{B}_t^H - \bar{B}_s^H) &= \left(C_H \int_{\mathbb{R}} ((t-u)_+^\alpha - (s-u)_+^\alpha) dW_u \right) \\
&= \left(C_H \int_{\mathbb{R}} ((t-s-v)_+^\alpha - (-v)_+^\alpha) dW_{v+s} \right) \\
&\stackrel{d}{=} \left(C_H \int_{\mathbb{R}} ((t-s-v)_+^\alpha - (-v)_+^\alpha) dW_v \right) \\
&= (\bar{B}_{t-s}^H).
\end{aligned}$$

Therefore, \bar{B}^H has stationary increments. Now it remains to show that \bar{B}^H is a type 1 fBm, by showing that it possesses the defining covariance structure from Definition 4.1.2.

Take $t \in \mathbb{R} \setminus \{0\}$, then by substituting $y := \frac{s}{t}$ for $t > 0$ and $y := 1 - \frac{s}{t}$ for $t < 0$ that

$$\begin{aligned}
E(Z_t^H)^2 &= (C_H)^2 \int_{\mathbb{R}} ((t-s)_+^\alpha - (-s)_+^\alpha)^2 ds \\
&= (C_H)^2 |t|^{2H} \int_{\mathbb{R}} ((1-y)_+^\alpha - (-y)_+^\alpha)^2 dy \\
&= (C_H)^2 |t|^{2H} \left(\int_{-\infty}^0 ((1-y)^\alpha - (-y)^\alpha)^2 dy + \int_0^1 (1-y)^{2\alpha} dy \right) \\
&= |t|^{2H}
\end{aligned}$$

Finally, combining the above facts we can deduce that

$$\begin{aligned}
E(\bar{B}_s^H, \bar{B}_t^H) &= \frac{1}{2} \left(\mathbb{E}_{\mathbb{P}} (\bar{B}_s^H)^2 + \mathbb{E}_{\mathbb{P}} (\bar{B}_t^H)^2 - \mathbb{E}_{\mathbb{P}} (\bar{B}_s^H - \bar{B}_t^H)^2 \right) \\
&= \frac{1}{2} (|s|^{2H} + |t|^{2H} - |t-s|^{2H}), \quad s, t \in \mathbb{R}.
\end{aligned}$$

And therefore \bar{B}^H is a type 1 fBm. □

Remark 4.4.7. The Mandelbrot-Van Ness representation is popular, due to the simplicity of the Kernel function. Unfortunately, the natural filtrations of the Wiener process driving the representation, and the resulting type 1 fBm do not coincide. Therefore, it is a non-canonical representation, reducing its usefulness in certain applications [16, Section 3.5.2 Page 8].

Remark 4.4.8. The kernel function $k_H(t, s)$ used in the Mandelbrot-Van Ness representation is not a proper convolution kernel, as it has an additional term preventing it from being written in the form $k_H(t-s)$. The search for an integral representation of fBm with a convolution kernel will lead us to type 2 fBm.

The Mandelbrot-Van Ness representation can be generalised to the following representation

$$B_t^H = \kappa \int_{-\infty}^{+\infty} \left((t-s)_+^{H-1/2} - (-s)_+^{H-1/2} \right) dW_s$$

for κ a positive parameter depending only on the value of H . Given a fixed H , κ is merely a constant scaling the process. It is easy to show that, like above the resulting process is a fBm. As we work through the literature, we will see many different constant (given a fixed H) terms utilised in various expressions.

We shall now, as done by Mandelbrot and Van-Ness in [22], consider the case of

$$\kappa = \kappa(H) = 1/\Gamma(H + 1/2).$$

This choice was made to ensure that the variance of the process at time $t = 1$ is equal to the following

$$\text{Var}(B_1^H) = \nu(H) = -2 \frac{\cos(\pi H)}{\pi} \Gamma(-2H), \quad \nu(1/2) = 1 \quad [14, \text{Section 4.1 Page 23}].$$

This choice of κ also allows us to present the following result, using the infinite horizon modified Riemann-Liouville operators

Lemma 4.4.9. *The Mandelbrot-Van Ness representation of fBm, with $\kappa = 1/\Gamma(H + 1/2)$ can be written as*

$$B^H = \tilde{I}_+^{H-1/2} W \quad [14, \text{Theorem 4.1 Page 23}]. \quad (4.4.5)$$

Proof. A full proof can be found on Page 23 of 'Representation formulae for the Fractional Brownian motion' by Jean Picard [14, Theorem 4.1 Page 23]. It has been omitted here for brevity. \square

A useful property of expression (4.4.5) is that it can be restricted to \mathbb{R}_- , therefore in order to know the path of B^H on \mathbb{R}_- , we only need to know the path of W on \mathbb{R}_- . While this is useful theoretically, for most purposes we are mainly concerned with positive time indexing, thus we would hope we have a similar relation on \mathbb{R}_+ . Unfortunately, (4.4.5) cannot be restricted to \mathbb{R}_+ [14, Section 4.1 Page 23]. If we desire knowledge of the paths of B^H on \mathbb{R}_+ , we need to know W on the entire real line \mathbb{R} . An attempt to circumvent this issue can be to reverse the time index ($t \rightarrow -t$) of all the processes. This reversal, allows us to obtain the following backward formulae on \mathbb{R}_+

$$B_t^H = \frac{1}{\Gamma(H + 1/2)} \int_0^\infty \left(s^{H-1/2} - (s-t)_+^{H-1/2} \right) dW_s. \quad [14, \text{Equation 54 Page 24}]$$

However, this does not solve our issues completely. If we want to know the value of B^H at a single time $t > 0$, we require knowledge of the path of W on the whole of \mathbb{R}_+ . This is very impractical, and thus we search for a representation of fBm where we would only require the path of W on $[0, t]$ in order to obtain the value of B_t^H . This brings us to our next representation formulae.

4.4.3 Canonical Representation on \mathbb{R}_+

Theorem 4.4.10 (Canonical Representation of Fractional Brownian motion on \mathbb{R}_+). *Define*

$$\varphi^{J,H}(u) = (H - J) \int_1^u (v^{H+J-1} - 1) (v-1)^{H-J-1} dv + (u-1)^{H-J}$$

for $0 < J, H < 1$ and $u > 1$, and let

$$K_{0+}^{J,H}(t, s) = \frac{1}{\Gamma(H - J + 1)} \varphi^{J,H} \left(\frac{t}{s} \right) s^{H-J},$$

Then on \mathbb{R}_+ , the fBm B^H can be represented by the following Itô integral

$$B_t^H = \int_0^t K_{0+}^{1/2,H}(t, s) dW_s \quad (4.4.6)$$

Proof. The proof is lengthy and has been left out for brevity. It can be found on Page 25 of 'Representation formulae for the fractional Brownian motion' by Jean Picard [14, Theorem 4.3 Page 25]. \square

Lemma 4.4.11. *The representation of fBm given by (4.4.6) is canonical, with B^H and W generating the same natural filtration.*

4.4.4 Molchan-Golosov representation

The third and final representation of type 1 fBm is the Molchan-Golosov representation, which is also often called the Volterra-type representation, not to be confused with the type 2 fBm we shall explore next. An advantage of this representation is that the kernel function used has compact support and that the representation is canonical (on \mathbb{R}_+).

The following theorem was presented by Molchan and Golosov in 1969 [23], however this representation is based on a translated version by Jost [16, Section 3.1 Page 5].

Theorem 4.4.12 (Molchan & Golosov). *Let $t \in [0, \infty)$, and $H \in (0, 1)$. Then define X_t^H as*

$$(X_t^H) = \left(\int_0^t z_H(t, s) dW_s \right).$$

Where, for $H > \frac{1}{2}$,

$$z_H(t, s) := \frac{C(H)}{\Gamma(H - \frac{1}{2})} s^{\frac{1}{2}-H} \int_s^t u^{H-\frac{1}{2}} (u-s)^{H-\frac{3}{2}} du, 0 < s < t < \infty.$$

And for $H \leq \frac{1}{2}$,

$$z_H(t, s) := \frac{C(H)}{\Gamma(H + \frac{1}{2})} \left(\frac{t}{s}\right)^{H-\frac{1}{2}} (t-s)^{H-\frac{1}{2}} - \frac{C(H)}{\Gamma(H - \frac{1}{2})} s^{\frac{1}{2}-H} \int_s^t u^{H-\frac{3}{2}} (u-s)^{H-\frac{1}{2}} du, 0 < s < t < \infty.$$

In both cases

$$C(H) := \left(\frac{2H\Gamma(H + \frac{1}{2})\Gamma(\frac{3}{2} - H)}{\Gamma(2 - 2H)} \right)^{\frac{1}{2}}.$$

Then X_t^H is a type 1 fBm with Hurst parameter H .

Proof. The proof is lengthy, and can be found on page 6 of 'Integral Transformations of Volterra Gaussian processes' by Jost [16, Theorem 3.1 Page 6]. \square

The Molchan-Golosov representation has been widely used since its inception, and many subsequent results in the study of fBm have been based on this representation, including the notable Lévy characterisation of fBm.

4.5 Type 2 Fractional Brownian motion

Riemann-Liouville processes are often referred to as a type 2 fBm. They are constructed from a type 1 fBm by applying Riemann-Liouville fractional operators, in contrast to the weighted fractional operator used in the Mandelbrot-Van Ness representation of type 1 fBm.

Type 2 fBm is also often referred to as Volterra-type fBm, due to it being a Volterra Gaussian process. Before we delve into type 2 fBm, we will first define Volterra Gaussian processes.

4.5.1 Volterra Gaussian processes

Volterra Gaussian processes are a generalisation of a standard Wiener process. Furthermore, a Volterra Gaussian process can also be thought of as a generalisation of the Molchan and Golosov representation of type 1 fBm, where we allow arbitrary Volterra kernel functions. While fBm is the unique Gaussian process that provides a model for both self-similarity and long-range dependence [16, Section 1 Page 1], Volterra processes allow for more flexibility when modelling only self-similarity is required.

Definition 4.5.1 (Volterra Kernel). Let $z_X \in L_{loc}^n([0, \infty)^n)$.

Then we call z_X a Volterra Kernel on $[0, \infty)^n$ if it satisfies the causality condition, which is as follows

$$z_X(x_1, x_2, \dots, x_n) = 0 \text{ if any of } x_1, x_2, \dots, x_n < 0$$

Furthermore, in the context of fBm, we frequently use convolution kernels of the form $z_X(t-s)$, in this in order for z_X to be a Volterra kernel we must also have that

$$z_X(x_1, x_2) = 0, \text{ if } x_1 < x_2.$$

[16, Definition 6.1 Page 17]

With that in mind, we can continue to define a Volterra Gaussian process.

Definition 4.5.2 (Volterra Gaussian process). Let $(W_t)_{t \in [0, \infty)}$ be a standard Brownian motion. Then, a Gaussian process $(X_t)_{t \in [0, \infty)}$ is called Volterra, if there exists a Volterra kernel $z_X \in L^2_{\text{loc}}([0, \infty)^2)$, such that

$$X_t = \int_0^t z_X(t, s) dW_s$$

[16, Definition 6.1 Page 17]

Theorem 4.5.3. *Let X_t be a Volterra Gaussian process. Then the following hold*

- 1 X_t is a centred Gaussian.
- 2 X_t has the following covariance function

$$R^X(s, t) = \int_0^{s \wedge t} z_X(t, u) z_X(s, u) du, s, t \in [0, \infty) \quad [16, Section 6.2 Page 16].$$

Notable Volterra Gaussian processes include the Molchan-Golosov representation of type 1 fBm (see 4.4.4), and the Riemann-Liouville processes (see 4.5.2), also known as type 2 fBm. To avoid confusion, any reference to 'Volterra-type' fBm will refer to the Riemann-Liouville processes, which we shall now explore.

4.5.2 The Riemann-Liouville processes

Definition 4.5.4 (Riemann-Liouville processes). Let $t \in \mathbb{R}_+$, then define

$$X_t^H = \frac{1}{\Gamma(H - 1/2)} \int_0^t (t - s)^{H-1/2} dW_s. \quad (4.5.1)$$

Then X_t^H is a Riemann-Liouville process with Hurst Index H . Furthermore, the paths of X_t are almost surely α -Hölder continuous for $\alpha < H$ [14, Section 5.1 Page 34].

Remark 4.5.5. From the definition, we can see for a fixed H that the Riemann-Liouville process and the type 1 fBm, are of the same Hölder continuity.

These Riemann-Liouville processes are very similar to type 1 fBm, hence why they are often referred to as type 2 fBm. We have already established that type 1 and type 2 fBm share the same smoothness for a given H , if we want to further quantify the similarity of the paths of these two processes, we can utilise the following result.

Theorem 4.5.6. *Let $0 < H < 1$, then both type 1 and type 2 fBm exist, and we can jointly realise both processes (X^H, B^H) on \mathbb{R}_+ , where B^H is a type 1 fBm, and X^H is a type 2 fBm. Then $X^H - B^H$ is C^∞ on \mathbb{R}_+ . Furthermore, for $T > 0, S > 0$ and $1 \leq p < \infty$, we have that*

$$\left\| \sup_{0 \leq t \leq T} |(X_{S+t}^H - X_S^H) - (B_{S+t}^H - B_S^H)| \right\|_p \leq C_p S^{H-1} T \quad (4.5.2)$$

(where $\|\cdot\|_p$ denotes the $L^p(\Omega)$ -norm of the underlying probability space) [14, Theorem 5.1 Page 35].

Proof. A full proof can be found on Page 35 of Picard [14, Theorem 5.1 Page 35]. □

Remark 4.5.7. Inequality (4.5.2) indicates that the process of the increments of type 2 fBm, $X_t^{S,H} = X_{S+t}^H - X_S^H$, is close to that of a type 1 fBm when S is sufficiently large.

While type 1 and type 2 fBm are quite similar, there are notable differences in their behavior.

Differences between type 1 and type 2 Fractional Brownian motion

Consider a Riemann-Liouville process of the form

$$V(t) = \int_0^t (2H(t-s)^{H-1/2})dW_s \quad (4.5.3)$$

For this choice of process V_t , we have that $\text{Var}(V_t) = t^{2H}$, which coincides with that of type 1 fBm.

Remark 4.5.8. Here we again reiterate that the choice of constant factor in front of the integral in the definition of fBm processes, simply scales the process, changing its variance. What is important are the terms within the integral, even with different constant factors the paths of the process still have the same properties. Practitioners often choose the constant term to ensure a specific expressing for the variance of the process.

While the variance of this type 2 fBm and type 1 fBm match, we unfortunately get a rather different covariance structure for type 2 fBm.

Let $0 < s < t$, then we have that

$$\mathbb{E}[V(t)V(s)] = 2Hs^{2H} \int_0^1 (1-u)^{H-1/2} \left(\frac{t}{s} - u\right)^{H-1/2} du = s^{2H}G\left(\frac{t}{s}\right), \quad (4.5.4)$$

where, for $x \geq 1$

$$G(x) = \frac{2H}{H+1/2} x^{1/2-H} F_1(1, 1/2-H, H+3/2, x), \quad [2, \text{Equation 4.1 Page 15}],$$

with ${}_2F_1$ denoting the Gaussian hypergeometric function (see [16, Section 3.3.1 Page 7]).

Remark 4.5.9. Substituting $v = u = t$ into equation (4.5.4), we obtain $\text{Var}(V_t) = t^{2H}$.

Another major difference between the two types of fBm, is that the Riemann-Liouville process no longer has stationary increments. However, an interesting paper by Lim [24] shows that under various conditions that the increments of the Riemann-Liouville process can be stationary. Thus, the Riemann-Liouville does not totally lack the property of stationary increments. We encourage the reader to consult the Lim paper [24] if interested, however it is out of the scope of this investigation.

Even with these differences, the Riemann-Liouville processes still allow us to capture the roughness and long-term dependence of type 1 fBm. In fact, the Riemann-Liouville processes are arguably more important than type 1 fBm within mathematical finance, as they are more frequently used for the purpose of modelling. Though the standard type 1 fBm possesses useful properties such as stationary increments and self-similarity, it is often not used for modelling purposes as it does not represent a casual time-invariant system. On the other hand, the type 2 fBm can be viewed a system driven by a white noise process [24, Section 1 Page 1302].

Therefore, practitioners often choose to utilise Riemann-Liouville processes, and their paths are often considered similar enough to type 1 fBm. As a result of this, these Riemann-Liouville processes will be the processes that we shall use within our construction of a Rough Volatility model.

Now that we have introduced fBm, and covered some of the theory regarding Hölder continuity and the Hurst parameter, we can proceed in the next chapter to introduce the final motivating factor for the development of rough volatility models. This motivating factor will be introduced in the form of an investigation into the roughness of the variance process of several equity indices.

Chapter 5

Additional Motivation: Assessing the Roughness of Volatility

Now that we have covered the requisite theory on fBm, we can delve into an additional motivating phenomenon towards the development of rough volatility models. In this chapter, we will display an investigation into the roughness of the realised volatility processes of various equity indexes. The conclusion from this investigation will serve as our final motivating factor for the development of rough volatility models.

5.1 Smoothness of a process

How does one determine the smoothness of a continuous time process? The literature concerning the smoothness of process as of recent has supported the notion of utilising Besov spaces, as a 'natural framework to study the smoothness of the sample paths of a continuous time random process' [25, Section 1 Page 55].

5.1.1 Besov Spaces

A Besov space is a complete function space, that can be viewed as containing functions/processes of a certain smoothness. In order to define a Besov space, we first let Δ_h^n be the operator defined by $\Delta_h^1 f(x) = f(x+h) - f(x)$ and $\Delta_h^n f(x) = \Delta_h^1 (\Delta_h^{n-1}) f(x)$ [25, Appendix A.1 Page 61].

Now we must also associate a modulus of smoothness/continuity. A modulus of smoothness is a function used to quantify the uniform continuity of a function. We shall utilise the n th order L^p modulus of smoothness of f on $[0, 1]$, defined as

$$\omega_n(f, t)_p = \sup_{|h| \leq t} \|\Delta_h^n f\|_{L^p(\Omega_{h,n})},$$

where $\Omega_{h,n} = \{x \in [0, 1]; x + kh \in [0, 1], k = 0, \dots, n\}$ [25, Appendix A.1 Page 61].

Furthermore, let us define the following sequence space l^p :

Definition 5.1.1 (l^p Sequence Space). Let $0 < p < \infty$, then l^p is the space consisting of all sequences $(x_n)_{n \in \mathbb{N}}$ such that

$$\sum_n |x_n|^p < \infty.$$

We can now proceed to define a Besov Space.

Definition 5.1.2 (Besov Space). Let $s > 0$, $0 < p$, and $q < +\infty$, then the the Besov space $\mathcal{B}_{p,q}^s([0, 1])$ consists of those functions $f \in L^p[0, 1]$ such that

$$\left\{ 2^{sj} \omega_n(f, 2^{-j})_p \right\}_{j \geq 0} \in l^q,$$

where $n \in \mathbb{N}$ such that $s < n$.

- (i) Additionally, if $p, q \geq 1$ then the Besov space is a Banach space when equipped with the norm

$$\|f\|_{\mathcal{B}_{p,q}^s([0,1])} = \|f\|_{L^p} + \left\| \left\{ 2^{sj} \omega_n(f, 2^{-j})_p \right\}_{j \geq 0} \right\|_{qq}.$$

- (ii) For p or q less than 1 and $s > \max\{1/p - 1, 0\}$, the Besov space $\mathcal{B}_{p,q}^s([0,1])$ is only a quasi-Banach space [25, Appendix A.1 Page 61].
- (iii) In the case that $q = +\infty$, $s > 0$, and $1 \leq p \leq +\infty$ the Besov Space $\mathcal{B}_{p,\infty}^s([0,1])$ consists of those functions $f \in L^p[0,1]$ such that

$$\sup_{j \geq 0} \left\{ 2^{sj} \omega_n(f, 2^{-j})_p \right\}_{j \geq 0} < \infty,$$

where $n \in \mathbb{N}$ such that $s < n$ [26, Appendix Page 899].

While Besov spaces are useful we are mainly concerned with Hölder continuity. In order to link Besov Spaces to Hölder-Spaces and Hölder continuity, we shall use the following Theorem.

Theorem 5.1.3 (Besov to Hölder Embedding). *Let f belong to $\mathcal{B}_{p,\infty}^s$ for any $p > 0$. Then f belongs to the Hölder space C^α and is α -Hölder continuous for $s < \alpha$ [1, Section 2.1 Page 7].*

5.1.2 Regularity Measures

Now that we have defined a Besov (smoothness) space, we wish to find quantities which allow us to derive Besov smoothness proprieties for the paths of various continuous time processes. We make use of the novel regularity measure proposed by Rosenbaum in his 2011 paper, titled 'A new microstructure noise index' [26, Equation 3 Page 884].

For a given $p \in \mathbb{R}$, we consider the following function

$$q \rightarrow S_q^p$$

with

$$S_q^p = \frac{1}{p} \left\{ 1 + \log_2 \left(\frac{V_{q+1}^p}{V_q^p} \right) \right\} \quad [26, \text{Equation 3 Page 884}]$$

where

$$V_q^p = \sum_{k=0}^{T2^{N-q}-1} |Y_{(k+1)2^{q-N}} - Y_{k2^{q-N}}|^p \quad [26, \text{Equation 3 Page 884}].$$

Rosenbaum showed that S_q^p can be considered a regularity measure of a process, with the associated sampling frequency of 2^{N-q} . The interpretation of $S^p - q$ as a regularity measure, laid out by Rosenbaum in [26], is as follows. Let $T = 1$, and use the Besov smoothness space $\mathcal{B}_{p,\infty}^s([0,1])$. Furthermore, assume a process Y_t belong to $\mathcal{B}_{p,\infty}^s([0,1])$ and does not belong to $\mathcal{B}_{p,\infty}^{s+\varepsilon}([0,1])$, for any $\varepsilon > 0$. Then under the assumptions laid out in [25], we can consider the following asymptotic approximation

$$V_q^p \simeq c2^{(q-N)(ps-1)}, [26, \text{Equation 3 Page 884}]$$

with $c > 0$. Utilising this approximation, we have that

$$\begin{aligned} S_q^p &\simeq \frac{1}{p} \left\{ 1 + \log_2 \left(\frac{c2^{(q+1-N)(ps-1)}}{c2^{(q-N)(ps-1)}} \right) \right\} \\ S_q^p &\simeq \frac{1}{p} \left\{ 1 + \log_2 (c2^{ps-1}) \right\} \\ S_q^p &\simeq s \end{aligned}$$

From, above we can see that S_q^p is the value of s such that Y_t belongs to the Besov smoothness space $\mathcal{B}_{p,\infty}^s([0,1])$, and does not belong to a smoother space. Therefore in the circumstances

above, S_q^p is approximately a measure of the smoothness of Y_t , providing an upper bound in terms of smoothness and associated Besov smoothness space for the process.

One can utilise the index S_q^p as follows: take a process Y_t , calculate S_q^p , then $s = S_q^p$ is the maximum value of s such that Y_t is a member of the Besov smoothness space, $\mathcal{B}_{p,\infty}^s([0, 1])$.

Proposition 5.1.4. *Let Y_t be a continuous semimartingale. Then for any p , S_q^p is asymptotically equal to $\frac{1}{2}$ [26, Section 2.2 Page 885].*

Proof. See Jacod 2008 for a proof [27]. □

Proposition 5.1.5. *Let T be fixed, $N \rightarrow \infty$, and $\frac{1}{2} < H < 1$. Then for any $p > 0$, S_q^p for a fBm with Hurst Parameter H , converges almost surely to H .*

5.2 Realised Volatility and Variance

Consider a simple stochastic volatility model,

$$dS_t = \mu_t S_t dt + \sigma_t S_t dW_t \quad (5.2.1)$$

Unfortunately, the instantaneous volatility process σ_t is not directly observable within the market, thus in order to analyse its properties, and in particular its roughness, we must estimate it somehow. A natural choice is to examine the realised variance and realised volatility of a price process.

Definition 5.2.1 (Realised Variance & Volatility). Let π^n be the time partition defined as the sequence $\pi_1, \pi_2 \dots \pi_n$, such that $\pi_i < \pi_{i+1}$ for all i , and $t = \pi_1$, $\pi_n = t + \delta$. Then, the realised variance of a price process S over $[t, t + \delta]$ sampled along the time partition π^n is defined as

$$RVar_{t,t+\delta}(\pi^n) = \sum_{\pi^n \cap [t,t+\delta]} \left(\log S_{t_{i+1}^n} - \log S_{t_i^n} \right)^2,$$

and subsequently, the realised volatility is defined as

$$RV_{t,t+\delta}(\pi^n) = \sqrt{\sum_{\pi^n \cap [t,t+\delta]} \left(\log S_{t_{i+1}^n} - \log S_{t_i^n} \right)^2}.$$

Furthermore, let us define the integrated variance of a price-process,

Definition 5.2.2 (Integrated Variance). Let S follow (5.2.1). Then the integrated variance is defined as [28, Equation 3 Page 254]

$$IVar_{0,t} = \sigma_t^{2,*} = \int_0^t \sigma_s^2 ds.$$

We can easily calculate the realised variance and volatility of an asset's price process. As we wish to utilise these quantities as estimates for the instantaneous variance σ_t^2 and instantaneous volatility σ_t , we shall utilise the following results from Barndorff-Nielsen and Shepard [28].

Proposition 5.2.3. *Let S follow (5.2.1), then as the size of the mesh of the time partition π^n approaches 0*

1 $RVar_{t,t+\delta}(\pi^n)$ converges in probability to $IVar_{t,t+\delta}$ [28, Section 2.1 p257]

2 $RVar_{t,t+\delta}(\pi^n)$ is a consistent estimator of σ_t^2 as $\delta \rightarrow 0$ [28, Equation 7 Page 255]

With these at hand, we can proceed to our investigation into the roughness of the realised variance process of several equity indices.

5.3 Analysis of the volatility process

5.3.1 Framework of the Investigation

We present an investigation of the roughness of the volatility process, based on the work of Gatheral, Jaisson, and Rosenbaum, in their 2014 paper titled "Volatility is rough" [1, Section 2 Page 6-13].

In order to develop a framework for assessing the roughness of the volatility process, we shall begin working under the pretense that we have direct discrete observations of the volatility process. These observations of σ_t are taken on a time grid with mesh Δ on $[0, T]$, such that we have:

$$\sigma_0, \sigma_\Delta, \dots, \sigma_{k\Delta}, \dots,$$

with $k \in \{0, [T/\Delta]\}$.

We wish to calculate, for some $q \geq 0$, the following quantity:

$$\mathbb{E}[|\log(\sigma_\Delta) - \log(\sigma_0)|^q]. \quad (5.3.1)$$

Now assuming the increments of log-volatility process are stationary and that the law of large numbers holds, we can approximate (5.3.1), via the following:

$$m(q, \Delta) := \frac{1}{N} \sum_{k=1}^N |\log(\sigma_{k\Delta}) - \log(\sigma_{(k-1)\Delta})|^q \quad [1, \text{Section 2.1 Page 7}]. \quad (5.3.2)$$

Building on Rosenbaum's 2009 smoothness index, detailed above in Section 5.1.2, Gatheral, Jaisson and Rosenbaum explored the following smoothness estimate, based on $m(q, \Delta)$.

Working under the assumption that for some $s_q > 0$ and $b_q > 0$, as $\Delta \rightarrow 0$,

$$N^{qS_q} m(q, \Delta) \rightarrow b_q \quad [1, \text{Equation 2.1 Page 7}]. \quad (5.3.3)$$

Under suitable technical conditions, laid out in Rosenbaum's 2009 paper 'A new microstructure noise Index' [26], if (5.3.3) holds, we have that σ_t belongs to Besov smoothness space $B_{q,\infty}^{s_q}$, but does not belong to Besov smoothness space $B_{q,\infty}^{s_q+\epsilon}$ for any $\epsilon > 0$. Furthermore, sample paths of σ_t will belong to $B_{q,\infty}^{s_q}$ almost surely [1, Section 2.1 Page 7]. We view s_q as the smoothness of the volatility process, with regard to the l_q norm, and crucially note that if $\sigma_t \in B_{q,\infty}^{s_q}$, for $q > 0$, then σ_t is α holder continuous for $\alpha < s_q$ [1, Section 2.1 Page 7].

5.3.2 Empirical Calculations

As previously mentioned, unfortunately we are not able to directly observe the volatility process σ_t , thus we must find a proxy for spot volatility σ_t , to be able to compute approximate values for σ_t . Inspired by 'Volatility is Rough', we chose to proxy daily integrated variances by using daily realized variance estimates from the Oxford-Man Institute of Quantitative Finance Realized Library [29]. Unfortunately this data set was discontinued, however we found the data through an archive, containing values from January 3rd 2000, to the June 27th 2018, corresponding to 4714 trading days. As these pre-computed integrated variance estimates are calculated for the entire trading day, we take note of the bias pointed out in 'Volatility is Rough', where the authors warn of an upwards bias in the smoothness of the volatility process, as integration is a regularizing operation [1, Section 2.1 Page 8]. Figure 5.1 shows a plot of the realised volatility, calculated from the realised variance estimates, for the SPX Index.

We proceed to estimate the smoothness of the volatility process σ_t , by estimating $m(q, \Delta)$ with a proxy for σ_t , for several different values of Δ . From this point onward, We will utilise the the notation σ_t and $m(q, \Delta)$, with the understanding that we are utilising the appropriate proxies. We subsequently shall regress $\log m(q, \Delta)$ against $\log \Delta$, and plot these regressions for various values of q .

We performed this investigation for 30 Stock Indices within the Oxford-Man data set. Figure 5.2 shows a plot of the scaling of $\log m(q, \Delta)$ against $\log \Delta$ for various q , for the SPX Index. From this plot we can see that the points are well clustered around their respective linear regression lines, with a improved fit to the regression line being seen for lower values of q . If we consider the regression lines a good fit for each q , then for all q the points can be viewed as having an approximately linear relationship. Under stationary assumptions, this linear relationship implied the following scaling property:

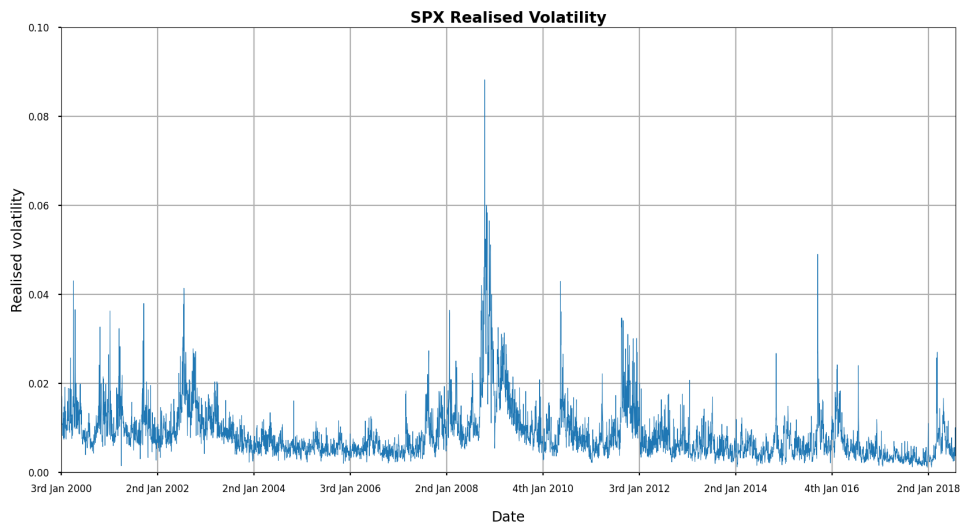


Figure 5.1: Plot of the Daily Realised Volatility of the S&P 500 Index, between January 3rd 2000, to the June 27th 2018

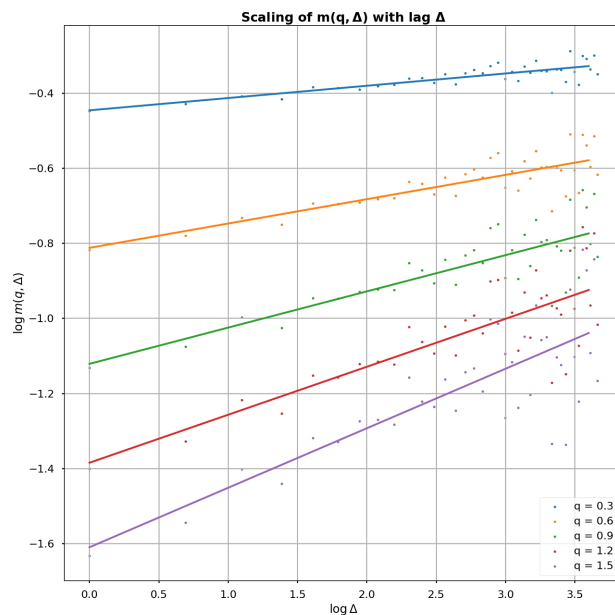


Figure 5.2: Plot of estimates of $\log m(q, \Delta)$ against $\log \Delta$ for various q , with accompanying linear regression lines, for SPX data

$$\mathbb{E}[|\log(\sigma_\Delta) - \log(\sigma_0)|^q] = K_q \Delta^{\zeta_q} \quad [1, \text{Section 2.2 Page 9}],$$

with $\zeta_q = \frac{\partial \log m(q, \Delta)}{\partial \log(\Delta)} Y_q$, where Y_q is the regression line associated with q . This suggests that our smoothness index $S_q \approx \frac{\zeta_q}{q}$. Therefore, a plot of ζ_q against q should reveal an estimate of the smoothness index of our volatility process. Figure 5.3 shows such a plot, for the same SPX

data. The gradient of this line represent the smoothness of the underlying volatility process. If the volatility process was driven by a standard Brownian motion, we would expect to see a value of $S_q = \frac{1}{2}$, however this is not what we observe in practice. Ultimately, we obtain a estimate of $S_q = 0.105$ for the SPX Index, this value also corresponds to the Hurst parameter H such that the fBm with Hurst Parameter H , will have sample paths of H -Holder continuity. This process was repeated for all 30 indices, and the resulting estimates for H are shown in Table 5.1.

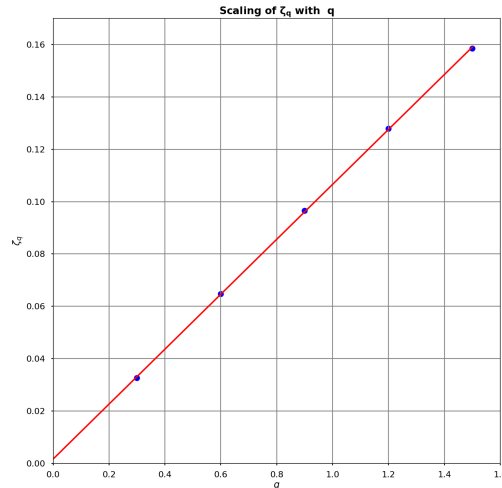


Figure 5.3: Plot of ζ_q against q , with accompanying linear regression line, for SPX data

Index	H Estimate	Index	H Estimate
AEX	0.116	KS11	0.101
AORD	0.072	KSE	0.105
BFX	0.124	MXX	0.066
BSEN	0.094	N225	0.098
BVLG	0.094	NSEI	0.092
BVSP	0.128	OMXC20	0.097
DJI	0.100	OMXHPI	0.084
FCHI	0.118	OMXSPI	0.102
FTMIB	0.127	OSEAX	0.096
FTSE	0.094	RUT	0.089
GDAXI	0.110	SMSI	0.114
GSPTSE	0.077	SPX	0.105
HSI	0.082	SSEC	0.113
IBEX	0.097	SSMI	0.140
IXIC	0.119	STOXX50E	0.113

Table 5.1: Hurst Parameter Estimates for Stock Indices

From Table 5.1, we can see that the estimate for H for each index is significantly below 0.5, suggesting that the variance processes of each of these indexes are rougher than a traditional Brownian motion. With this in mind, it seems natural to attempt to model the variance process of each asset as a rough process, driven by a process such as a fBm with $H < \frac{1}{2}$. As a result of this investigation, and further discoveries during their paper titled 'Volatility is Rough', Gather et al. [1] proposed the Rough Fractional Stochastic volatility model, which we shall explore in the next chapter.

Comparing our results to those of Gather et al. [1, Table B.1 Page 34], we see that our H estimates are more or less in line, while noting that each of our estimates is slightly lower for each index considered. However, we must recall that our investigation was performed on data from the

Oxford-Man data set from January 3rd 2000, to the June 27th 2018, whereas Gatheral et al. [1] used only up to March 31st 2014. Our estimates can thus be viewed as an updated version of the estimates from Gather et al. [1].

Remark 5.3.1. We stress that the aim of this investigation was to demonstrate that equity indexes have variance processes that are likely rougher than a traditional Brownian motion, rather than a thorough investigation to determine the exact Hurst parameter H to be used in any future rough volatility models. As mentioned in Section 5.3.2, these estimates are likely to be upwardly biased. Nevertheless, while we don't stress too heavily the exact value of each estimate for H , we shall utilise them somewhat as starting values for calibration attempts for the Rough Bergomi model (see Chapter 10), unless another suitable H is provided, for example from a previous calibration.

Chapter 6

The Rough Fractional Stochastic Volatility Model

In this chapter, we will explore the Rough Fractional Stochastic Volatility (RFSV) model, proposed by Gatheral, Jaisson and Rosenbaum [1, Section 3.1 Page 14]. As a precursor to the RFSV model, we will explore the FSV model of Comte and Renault [4], which the RFSV model was based upon.

6.1 The Fractional Stochastic Volatility Model

The Fractional Stochastic Volatility model (FSV) was proposed by Comte and Renault in 1998 [4]. The idea of the model was to replace the standard Brownian motion found in most stochastic volatility models, with a fBm (with $H > \frac{1}{2}$), in order to better capture the long-memory property of volatility. The long-memory of volatility was at the time seen as a stylised fact of volatility, however, as we shall later see, this notion may not be entirely accurate. Nevertheless, we shall briefly explore the FSV, before moving on to the more recent Rough Fractional Stochastic Volatility Model (RFSV).

The FSV model considers the following fractional stochastic differential equation [4, Section 2.1 Page 295]

$$dX_t = -\kappa X_t dt + \sigma dB_t^\alpha \quad x_0 = 0, \kappa > 0, \alpha := H - \frac{1}{2}, 0 < \alpha < \frac{1}{2} \quad (6.1.1)$$

Remark 6.1.1. The FSV utilises $H > \frac{1}{2}$, in order to capture the long-memory property of volatility. Recall that for $H > \frac{1}{2}$, the increments of B^H are positively correlated, and thus in order to obtain a mean-reverting volatility process, a mean reversion term must also be included in the model. We will later see that models utilising B^H with $H < \frac{1}{2}$, have this mean-reverting feature 'built-in' to the model, without needing explicit mean-reversion terms.

The solution to (6.1.1), derived by Comte and Renault [4, Section 2.1 Page 295], is as follows

$$X_t = \int_0^t e^{-\kappa(t-s)} \sigma \, dB^\alpha(s)$$

Remark 6.1.2. We must be careful as integration with respect to B^H is not always well-defined. Integration with respect to B^H for $H \neq \frac{1}{2}$ is defined only in the Wiener L^2 sense, and only for the integration of deterministic processes [4, Section 2.1 Page 295].

Furthermore, we can express this solution in terms of a Volterra-Gaussian process, with a suitable choice of kernel function, as follows

$$X_t = \int_0^t z(t-s) dW_s \quad [4, Section 2.2 Page 295] \quad (6.1.2)$$

where

$$\begin{aligned} z(x) &= \frac{\sigma}{\Gamma(1+\alpha)} \frac{d}{dx} \int_0^x e^{-\kappa u} (x-u)^\alpha du \\ &= \frac{\sigma}{\Gamma(1+\alpha)} \left(x^\alpha - \kappa e^{-\kappa x} \int_0^x e^{\kappa u} u^\alpha du \right). \end{aligned}$$

This allows us to express the solution as an integral with respect to a standard Wiener process, resulting in a substantially easier to work with form.

Now within their FSV model, Comte and Renault worked under the assumption that the logarithm of volatility was the solution to (6.1.1), i.e that $\ln \sigma(t) = X_t$.

Under this assumption, they obtain the following expression for the volatility process [4, Section 2.7 Page 296]

$$\tilde{\sigma}(t) = \exp \left(\int_{-\infty}^t e^{-\kappa(t-s)\gamma} dB^\alpha(s) \right).$$

Comte and Renault chose to use a 'stationary' version of X_t , by choosing to integrate from $-\infty$ to t , in order to produce an asymptotically stationary volatility process.

This FSV model inspired the RFSV model proposed by Gatheral, Jaisson and Rosenbaum in their paper 'Volatility is Rough' [1], however, they chose to utilise B^H with $H < \frac{1}{2}$. We shall now explore the RFSV model.

6.2 RFSV Model

In their paper 'Volatility is Rough' [1], Gatheral, Jaisson and Rosenbaum found, through an empirical analysis of realised volatility, that the increments of log-volatility were approximately normally distributed [1, Section 3.1 Page 14]. Furthermore, they discovered that the increments displayed a scaling property with a constant smoothness parameter [1, Section 3.1 Page 14]. These two results led them to propose their novel RFSV model, an adaptation of the aforementioned FSV.

6.2.1 Specification of the RFSV model

Gatheral, Jaissons and Rosenbaum's RFSV model is a FSV model, with a choice of Hurst Parameter $H < \frac{1}{2}$, chosen to be consistent with their roughness estimates of realised volatility [1, Section 2 Page 6-13].

Gatheral, Jaissons and Rosenbaum initially propose a simple model, which assumes the following underlying dynamics of the increments of log-volatility

$$\log \sigma_{t+\Delta} - \log \sigma_t = \nu (B_{t+\Delta}^H - B_t^H) \quad [1, \text{Equation 3.1 Page 14}] \quad (6.2.1)$$

where B^H is fBm with $H < \frac{1}{2}$ and $\nu > 0$. This permits the following equivalent expression for σ_t

$$\sigma_t = \sigma \exp(\nu B_t^H) \quad [1, \text{Equation 3.2 Page 14}] \quad (6.2.2)$$

for some constant $\sigma > 0$.

Unfortunately, as noted by Gatheral, Jaisson and Rosenbaum, this model is not-stationary [1, Section 3.1 Page 14]. A reminder of the definition of a stationary stochastic process is presented below.

Definition 6.2.1 (Stationary process). Let (X_t) be a stochastic process, with $F_X(x_{t_1}, x_{t_2}, \dots, x_{t_n})$ representing the cumulative distribution function of the unconditional joint distribution of (X_t) at times t_1, t_2, \dots, t_n . Then, we call (X_t) stationary if the following holds:

$$F_X(x_{t_1}, x_{t_2}, \dots, x_{t_n}) = F_X(x_{t_1+s}, x_{t_2+s}, \dots, x_{t_n+s}).$$

Therefore F_X is invariant under a shift in time.

Stationarity in a model is desirable as it helps to ensure the model is still valid over large time periods, and also makes the model more tractable and easy to deal with. In an attempt to find such a stationary model, Gatheral, Jaisson and Rosenbaum chose to model log-volatility as a fractional Ornstein-Uhlenbeck (fOU) process, with a long reversion time scale [1, Section 3.1 Page 14].

Definition 6.2.2 (Fractional Ornstein-Uhlenbeck process). A stationary fOU process (X_t) is defined as the stationary solution of the following SDE, known as the Fractional Ornstein-Uhlenbeck SDE :

$$dX_t = \nu dB_t^H - \kappa(X_t - m)dt \quad [1, \text{Section 3.1 Page 14}], \quad (6.2.3)$$

where $m \in \mathbb{R}$ and $\nu, \kappa > 0$.

Gatheral, Jaisson and Rosenbaum subsequently proposed the following model

$$\sigma_t = \exp(X_t), \quad t \in [0, T], \quad [1, \text{Equation 3.4 Page 15}] \quad (6.2.4)$$

where (X_t) satisfies the fractional SDE (6.2.3) for some appropriate choice of parameters. Intuitively, κ represents the speed of mean reversion towards the long-term mean m .

Remark 6.2.3. Under the RFSV model (6.2.4), if $\kappa = 0$ we observe that $X_t - X_s = \nu(B_t^H - B_s^H)$, and thus we recover our non-stationary model described by (6.2.1).

Furthermore, the solution to (6.2.3) can be written in the following explicit form

$$X_t = \nu \int_{-\infty}^t e^{-\kappa(t-s)} dB_s^H + m \quad [1, \text{Equation 3.3 Page 14}]. \quad (6.2.5)$$

Remark 6.2.4. The stochastic integral with respect to fBm in (6.2.5) is a path-wise Riemann-Stieltjes integral.

The RFSV model specified by (6.2.4) is a stationary model [1, Section 3.1 Page 15]. Furthermore, for $\kappa \ll \frac{1}{T}$, we observe that $\log(\sigma_t)$ behaves locally as a fBm [1, Section 3.1 Page 15]. This notion is formalised by the following proposition:

Proposition 6.2.5. Let B^H be a fBm and X^κ be the process defined by (6.2.5) for a given $\kappa > 0$. Moreover, let $q > 0, t > 0, \Delta > 0$. Then, as κ tends to zero,

$$1 \quad \mathbb{E} \left[\sup_{t \in [0, T]} |X_t^\kappa - X_0^\kappa - \nu B_t^H| \right] \rightarrow 0 \quad [1, \text{Proposition 3.1 Page 15}].$$

$$2 \quad E \left[|X_{t+\Delta}^\kappa - X_t^\kappa|^q \right] \rightarrow \nu^q K_q \Delta^{qH} \quad [1, \text{Corollary 3.1 Page 15}].$$

Proof. A full proof can be found in Appendix A.1/2 of 'Volatility is Rough' [1, Appendix A.1/2 Page 32]. \square

From Proposition (6.2.5) we observe that if $\kappa \ll \frac{1}{T}$, then we can assume that the log-volatility process behaves as a fBm within our interval of interest $[0, T]$. Furthermore, from part 2 of proposition (6.2.5), we can see that the scaling property of fBm, is approximately reproduced by the fOU process for very small κ [1, Corollary 3.1 Page 15]. Therefore the RFSV model allows us to have a stationary model, which can also replicate the empirical smoothness of the volatility process and the scaling behavior of fBm. With this in mind, it seems theoretically suitable to model the log-volatility process in such a way as described by (6.2.4).

Remark 6.2.6. Long-memory had previously been well regarded as a stylised fact of volatility. However, Gatheral et al. [1, Section 4 Page 23], showed that their RFSV model, which did not have long-memory by construction, would still seem to have the property of long-memory when analysed by classic estimation procedures used to test for long-memory. They therefore propose that perhaps these classical estimation procedures are not entirely accurate, and perhaps volatility does not have long-memory. This serves as additional justification for their RFSV model, which utilises $H < \frac{1}{2}$, and does not have long-memory by design.

6.3 Comparison with the FSV Model

As mentioned before, the RFSV model is a particular case of the FSV model. The main difference, however, is that for their RFSV model in line with the empirical observations of the smoothness of the volatility process, we take $H < \frac{1}{2}$ and $\kappa \ll 1/T$, while the FSV model takes $H > \frac{1}{2}$, to satisfy

the long memory assumption. This difference in the value of H has a large number of implications on the relative behaviour of the models.

As mentioned before in Section 3.4, Fukasawa found that a stochastic volatility model driven by a fBm B^H , generates a term structure of ATM forward skew of the form $\psi(\tau) \sim \tau^{H-\frac{1}{2}}$, for small τ , where $\tau = T - t$, the time to maturity. From this we can see that if we take $H > \frac{1}{2}$, as in the FSV model, then we obtain a term structure that is increasing for small τ . Such a term structure is inconsistent with the general empirically observed term structure.

In contrast, Gatherl et al. showed that the RFSV is both able to reproduce a realistic term structure of the ATM Forward volatility skew, while also matching the observed smoothness of the realised volatility process [1, Section 3.1 Page 14]. Furthermore, the choice of $H < \frac{1}{2}$ allows the model to have 'mean-reverting' properties, consistent with those observed empirically.

Ultimately, investigations into the RFSV model led to the development of a specific case of the RFSV model, known as the Rough Bergomi model. This will be the main model we explore within this work, and we will begin by introducing the Rough Bergomi model in the next chapter.

Chapter 7

The Rough Bergomi Model

In this chapter we present the main model of this paper, the Rough Bergomi Model. We shall describe both the pathway taken to specify the model, and the specification of the model itself.

7.1 The Path to the Rough Bergomi Model

Building upon the work of Gatheral et al. in [1], Bayer et al. showed how the RFSV model can lead to a natural model for option pricing, in their paper 'Pricing Under Rough Volatility' [2].

Consider the following non-stationary RFSV model, with $\kappa = 0$, and $H < \frac{1}{2}$:

$$\log \sigma_{t+\Delta} - \log \sigma_t = \nu(B_{t+\Delta}^H - B_t^H). \quad (7.1.1)$$

Now, turning to our choice of representation for B^H , we shall initially use the following Mandelbrot-Van Ness representation:

$$B_t^H = C_H \left\{ \int_{-\infty}^t \frac{dW_s^{\mathbb{P}}}{(t-s)^\alpha} - \int_{-\infty}^0 \frac{dW_s^{\mathbb{P}}}{(-s)^\alpha} \right\} \quad [2, \text{Section 2.0 Page 8}], \quad (7.1.2)$$

where $\alpha = \frac{1}{2} - H$, and

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

Then, if we let $v_t = \sigma_t^2$ and substitute equation (7.1.2) into (7.1.1), we obtain the following model for v_t under \mathbb{P} :

$$\begin{aligned} \log v_u - \log v_t &= 2\nu C_H \left\{ \int_{-\infty}^u \frac{dW_s^{\mathbb{P}}}{(u-s)^\alpha} - \int_{-\infty}^t \frac{dW_s^{\mathbb{P}}}{(t-s)^\alpha} \right\} \\ &= 2\nu C_H \left\{ \int_t^u \frac{1}{(u-s)^\alpha} dW_s^{\mathbb{P}} + \int_{-\infty}^t \left[\frac{1}{(u-s)^\alpha} - \frac{1}{(t-s)^\alpha} \right] dW_s^{\mathbb{P}} \right\} \\ &= 2\nu C_H \{M_t(u) + Z_t(u)\} \quad [2, \text{Equation 2.2 Page 9}]. \end{aligned}$$

Here $Z_t(u)$ is \mathcal{F}_t -measurable, while $M_t(u)$ is independent of \mathcal{F}_t as it depends on the integral of W_s from t to u . Furthermore, $M_t(u)$ is a centred Gaussian.

If we now introduce the following Riemann-Liouville process

$$\tilde{B}_t^{\mathbb{P}}(u) := \sqrt{2H} \int_t^u \frac{dW_s^{\mathbb{P}}}{(u-s)^\alpha}, \quad (7.1.3)$$

we can observe that

$$M_t(u) = \frac{1}{\sqrt{2H}} \tilde{B}_t.$$

Therefore, taking $\eta = \frac{2\nu C_H}{\sqrt{2H}}$ we have

$$\log v_u - \log v_t = \eta \tilde{B}_t^{\mathbb{P}}(u) + \eta \sqrt{2H} Z_t(u),$$

and subsequently that

$$v_u = v_t \exp \left(\eta \tilde{B}_t^{\mathbb{P}}(u) + \eta \sqrt{2H} Z_t(u) \right). \quad (7.1.4)$$

Now as \tilde{B}^H is a centred Gaussian, $\exp(\tilde{B}^H(u))$ is a log-normal random variable. Therefore, as v_t and $Z_t(u)$ are \mathcal{F}_t -measurable, we have the following

$$\begin{aligned} \mathbb{E}^{\mathbb{P}} [v_u | \mathcal{F}_t] &= v_t \exp \left(\eta \sqrt{2H} Z_t(u) \right) \times \exp \left(\mathbb{V}(\nu \tilde{B}_t^{\mathbb{P}}(u)) \right) \\ &= v_t \exp \left\{ \eta \sqrt{2H} Z_t(u) + \frac{1}{2} \eta^2 E \left| \tilde{B}_t^{\mathbb{P}}(u) \right|^2 \right\} \quad [2, \text{Equation 2.4 Page 9}]. \end{aligned} \quad (7.1.5)$$

Now, substituting (7.1.5) into (7.1.4), we obtain

$$v_u = v_t \exp \left(\eta \tilde{B}_t^{\mathbb{P}}(u) + \eta \sqrt{2H} Z_t(u) \right) \quad (7.1.6)$$

$$\begin{aligned} &= v_t \exp \left\{ \eta \sqrt{2H} Z_t(u) + \frac{1}{2} \eta^2 E \left| \tilde{B}_t^{\mathbb{P}}(u) \right|^2 \right\} \times \exp \left\{ \eta \tilde{B}_t^{\mathbb{P}}(u) - \frac{1}{2} \eta^2 E \left| \tilde{B}_t^{\mathbb{P}}(u) \right|^2 \right\} \\ &= E^{\mathbb{P}} [v_u | \mathcal{F}_t] \mathcal{E} \left(\eta \tilde{B}_t^{\mathbb{P}}(u) \right) \quad [2, \text{Equation 2.4 Page 9}], \end{aligned} \quad (7.1.7)$$

where \mathcal{E} represents the Wick stochastic exponential.

From equation (7.1.6), we can see that v_u is non-Markovian, as we recall that $Z_t(u)$ depends on the full history of $W_t^{\mathbb{P}}$ up to time t . However, from (7.1.7) we see that the conditional distribution of v_u is only dependent on \mathcal{F}_t , through the term $E^{\mathbb{P}} [v_u | \mathcal{F}_t]$, which corresponds to the instantaneous variance forecast at time $t < u$ [2, Section 2 Page 9].

With equation (7.1.7) in mind, Bayer et al. proposed the following model, under the physical probability measure \mathbb{P} :

$$\begin{aligned} \frac{dS_u}{S_u} &= \mu_u du + \sqrt{v_u} dZ_u^{\mathbb{P}}, \\ v_u &= v_t \exp \left\{ \eta \tilde{B}_t^{\mathbb{P}}(u) + 2\nu C_H Z_t(u) \right\} \quad [2, \text{Equation 2.5 Page 10}] \end{aligned} \quad (7.1.8)$$

where $Z^{\mathbb{P}}$ is another Wiener process, which is correlated to the Wiener process $W^{\mathbb{P}}$ driving $Z_t(u)$, with correlation coefficient ρ . Moreover, μ_u is an appropriate drift term, $\mu = \mu(u, \omega)$.

7.1.1 Change of Measure

The pricing of derivatives is usually conducted under the risk-neutral measure \mathbb{Q} , an equivalent martingale measure ¹ $\mathbb{Q} \sim \mathbb{P}$ on $[t, T]$. Using such a measure \mathbb{Q} , we ensure that the discounted asset price process S_t is a martingale under \mathbb{Q} . Thus, in our case, and without loss of generality assuming interest rates are zero, we would have the following price process dynamics under \mathbb{Q} :

$$\frac{dS_u}{S_u} = \sqrt{v_u} dZ_u^{\mathbb{Q}}, \quad t \leq u \leq T.$$

Therefore, we now to find a suitable change of measure to obtain a model under \mathbb{Q} .

Utilising Girsanov's theorem, we obtain the following change of measure on $[t, T]$:

$$dZ_u^{\mathbb{Q}} = dZ_u^{\mathbb{P}} + \frac{\mu_u}{\sqrt{v_u}} du, \quad t \leq u \leq T. \quad (7.1.9)$$

We now turn our attention to a change of measure for $\tilde{B}^{\mathbb{P}}$, our Riemann-Liouville process. Recall that $\tilde{B}^{\mathbb{P}}$ is constructed from $W^{\mathbb{P}}$ a Wiener process, which was correlated with $Z_u^{\mathbb{P}}$ with correlation coefficient ρ , usually taken to be $\rho \in (-1, 0)$. Therefore, we can represent $W^{\mathbb{P}}$ in terms of $Z^{\mathbb{P}}$ and another Wiener process $Z'^{\mathbb{P}}$ that is independent of $Z^{\mathbb{P}}$. In doing so, we obtain the following relation:

$$W^{\mathbb{P}} = \rho Z^{\mathbb{P}} + \sqrt{1 - \rho^2} Z'^{\mathbb{P}}. \quad (7.1.10)$$

¹Two Measure \mathbb{P} and \mathbb{Q} on (Ω, \mathcal{F}) , are equivalent if for all $A \in \mathcal{F}$, $\mathbb{P}(A) = 0$ if and only if $\mathbb{Q}(A) = 0$

We shall use equation (7.1.10) to derive our change of measure for $\tilde{B}^{\mathbb{P}}$, by first finding a change of measure for $W^{\mathbb{P}}$. In order to do this we turn to the components of $W^{\mathbb{P}}$, from equation (7.1.10). Now, we already have a change of measure for $Z^{\mathbb{P}}$, thus we simply need a change of measure for $Z'^{\mathbb{P}}$. We shall utilise the general change of measure given by Bayer et al. [2, Section 2.1 Page 11], this is given by:

$$dZ'_u{}^{\mathbb{Q}} = dZ'_u{}^{\mathbb{P}} + \gamma_u du, \quad (7.1.11)$$

where γ is a suitably chosen adapted process of the form $\gamma = \gamma(u, \omega)$, for $u \in [t, T]$, called the 'market price of volatility risk' [2, Section 2.1 Page 11].

We proceed by substituting (7.1.9) and (7.1.11) into (7.1.10), to obtain a change of measure for $W^{\mathbb{P}}$,

$$\begin{aligned} dW^{\mathbb{Q}} &= \rho dZ^{\mathbb{Q}} + \sqrt{1 - \rho^2} dZ'^{\mathbb{Q}} \\ &= dW_u^{\mathbb{P}} + \left(\frac{\rho \mu_u}{\sqrt{v_u}} + \sqrt{1 - \rho^2} \gamma_u \right) du. \end{aligned}$$

Which can be represented in the form

$$dW_s^{\mathbb{P}} = dW_s^{\mathbb{Q}} + \lambda_s ds \quad [2, Equation 2.6 Page 11].$$

Here, the choice of λ_s specifies our change of measure. We shall choose to take a simple change of measure, where $\lambda_s = \lambda(s)$ is a deterministic function of s .

As we now have the required change of measure terms, we can take our expression for v_u under $\mathbb{P}(7.1.7)$, and write it in under \mathbb{Q} as

$$\begin{aligned} v_u &= \mathbb{E}^{\mathbb{P}} [v_u | \mathcal{F}_t] \exp \left\{ \eta \sqrt{2H} \int_t^u \frac{1}{(u-s)^\gamma} dW_s^{\mathbb{P}} - \frac{\eta^2}{2} (u-t)^{2H} \right\} \\ &= \mathbb{E}^{\mathbb{P}} [v_u | \mathcal{F}_t] \mathcal{E} \left(\eta \tilde{B}_t^{\mathbb{Q}}(u) \right) \exp \left\{ \eta \sqrt{2H} \int_t^u \frac{\lambda(s)}{(u-s)^\gamma} ds \right\} \quad [2, Equation 2.8 Page 11] \quad (7.1.12) \\ &= \xi_t(u) \mathcal{E} \left(\eta \tilde{B}_t^{\mathbb{Q}}(u) \right), \end{aligned}$$

where $\xi_t(u) = \mathbb{E}^{\mathbb{P}} [v_u | \mathcal{F}_t]$, known as the forward variance curve. We can decompose ξ_t as follows

$$\xi_t(u) = \mathbb{E}^{\mathbb{P}} [v_u | \mathcal{F}_t] \exp \left\{ \eta \sqrt{2H} \int_t^u \frac{1}{(u-s)^\gamma} \lambda(s) ds \right\} \quad [2, Section 3 Page 13],$$

where we recognise that the first term depends on the history of the driving Brownian motion, and the latter term depends on the price of risk $\lambda(s)$.

7.2 The Rough Bergomi Model

Now that we have the relevant dynamics under \mathbb{Q} , we can specify the Rough Bergomi model of Bayer et al. [2].

Let S_t denote our price process, and let v_t be our variance process, then we specify the Rough Bergomi by the following dynamics

$$S_t = S_0 \mathcal{E} \left(\int_0^t \sqrt{v_u} dZ_u^{\mathbb{Q}} \right) \quad (7.2.1)$$

$$v_u = \xi_0(u) \mathcal{E} \left(\eta \sqrt{2H} \int_0^u \frac{1}{(u-s)^\gamma} dW_s^{\mathbb{Q}} \right) = \xi_0(u) \mathcal{E} \left(\eta \tilde{B}_u^{\mathbb{Q}} \right) \quad [2, Section 4 Page 14]. \quad (7.2.2)$$

Remark 7.2.1. Recall that $Z_t^{\mathbb{Q}} = \rho W_t^{\mathbb{Q}} + \sqrt{1 - \rho^2} W'_t{}^{\mathbb{Q}}$. Here ρ allows us to model the correlation between the price process and the volatility process. This correlation is usually observed to be an negative correlation and thus we typically take $\rho \in (-1, 0)$.

We note here that this (1-factor) Rough Bergomi Model is a non-Markovian generalisation of the 1-factor Bergomi Model (see [30]). It is non-Markovian in the instantaneous variance v_t as $\mathbb{E}^{\mathbb{Q}}[v_u|\mathcal{F}_t] \neq \mathbb{E}^{\mathbb{Q}}[v_u|v_t]$. However, it is Markovian in the infinite-dimensional state vector $\mathbb{E}^{\mathbb{Q}}[v_u|\mathcal{F}_t] = \xi_t(u)$ [2, Section 3 Page 13].

It is also interesting to note that, in the Rough Bergomi model, options prices when viewed from time t only depends on the forward variance curve $\xi_t(u)$ and the three parameters H, η , and ρ [2, Remark 3.1 Page 14].

7.2.1 The Bergomi Model

In order to better understand the Rough Bergomi, we shall briefly explore the traditional Bergomi Model.

Bergomi [30], in an attempt to better capture forward volatility and forward skew risks accurately, decided to model the dynamics of forward variance, instead of instantaneous variance. He proposed a general framework for modelling forward variances, which when combined with appropriate dynamics of the price process, result in the Bergomi Model.

Bergomi proposed various models, with the differences lying in the number of Brownian motions used within the dynamics of the forward variance process. Though the models do differ in their specification of the forward variance dynamics, all of Bergomi's models have the following dynamics for the price process :

$$dS_t = (r - q)S_t dt + \sqrt{\xi^t(t)}S_t dW_t^S \quad [30, Section 3.1 Page 4].$$

Remark 7.2.2. In each of the variations of the Bergomi model, the Brownian motion driving the price process will be correlated with each of the Brownians driving the forward variance process, each with a potentially unique coefficient. Furthermore, in the multi-factor case, the Brownian motions within the SDE for the forward variance process, will have a correlation matrix specifying the correlation between each pair of Brownian motions.

7.2.2 1-factor model

The one-factor Bergomi model utilises a single Brownian motion in the dynamics of the forward variance curve $\xi^T(t)$.

With the requirement that $\xi^T(t)$ is driftless, Bergomi chooses to further assume that $\xi^T(t)$ is log-normally distributed and that the volatility of $\xi^T(t)$, ω is a function of $T-t := \rho$. Consequently, Bergomi specifies the following SDE for ξ_t :

$$d\xi^T = \omega(T-t)\xi^T dW_t \quad [30, Section 2.1 Page 2]. \quad (7.2.3)$$

Bergomi then proceeds to specify that $\omega(\rho) = \omega e^{-k\rho}$. With this choice, applying Ito's lemma, the solution to the SDE is given by :

$$\xi^T(t) = \xi^T(0)e^{\left(\omega e^{-k(T-t)}X_t - \frac{\omega^2}{2}e^{-2k(T-t)}\mathbb{E}^{\mathbb{Q}}[X_t^2]\right)} \quad [30, Equation 2.1 Page 2], \quad (7.2.4)$$

where X_t is an Ornstein-Uhlenbeck process with the following dynamics:

$$\begin{aligned} dX_t &= -kX_t dt + dW_t \\ X_0 &= 0. \end{aligned}$$

Consequently, X_t and its variance are given by:

$$\begin{aligned} X_t &= \int_0^t e^{-k(t-u)} dW_u \\ \mathbb{E}[X_t^2] &= \frac{1 - e^{-2kt}}{2k}. \end{aligned}$$

Using this, the solution of the SDE given by equation (7.2.3) is as follows:

$$\xi_t^T = \xi_0^T \exp \left(\omega e^{-k(T-t)} X_t - \frac{\omega^2}{2} e^{-2k(T-t)} \mathbb{E}[X_t^2] \right).$$

In order to generate the Forward Variance curve, Bergomi suggests the following procedure [30, Section 2.1 Page 3]:

- 1 Let x_δ be a centred Gaussian such that $\mathbb{E}[x_\delta^2] = \frac{1-e^{-2k\delta}}{2k}$.
- 2 Start from known values of X_t and $\mathbb{E}(X_t^2)$ at time t , and calculate $\xi^T(t)$ using equation (7.2.4).
- 3 Generate the Forward Variance curve incrementally, using time steps of size δ using the following relationship

$$\begin{aligned} X_{t+\delta} &= e^{-k\delta} X_t + x_\delta \\ \mathbb{E}[X_{t+\delta}^2] &= e^{-2k\delta} \mathbb{E}[X_t^2] + \frac{1 - e^{-2k\delta}}{2k}, \end{aligned}$$

and equation (7.2.4).

Remark 7.2.3. We can see that through our choice of an exponential kernel function ω , we have obtained a Markovian model, with all $\xi^t(t)$ being a function of a single Wiener process W [30, Section 2.1 Page 3].

N-factor model

In order to have more flexibility to fit various term-structures of forward volatility, Bergomi also proposed the more general multi-factor model [30, Section 2.2 Page 3] The general Bergomi model framework utilises N-Brownian motions within the dynamics of the forward variance curve $xi_t^T = \mathbb{E}^{\mathbb{Q}}(v_T | v_t)$. The N-factor model has the following SDE:

$$d\xi_t^T = \omega \alpha_\omega \xi_t^T \sum_i \omega_i e^{-\kappa_i(T-t)} dW_t^i \quad [30, Section 2.4 Page 4] \quad (7.2.5)$$

where α_ω is a normalising factor. The N Brownian motions are correlated using a correlation matrix such that $\rho_{i,j}$ is the correlation between W^i and W^j .

The solution of (7.2.5) is then given by

$$\xi_t^T = \xi_0^T \exp \left(\omega \sum_i \omega_i e^{-k_i(T-t)} X_t^i - \frac{\omega^2}{2} \sum_{ij} \omega_i \omega_j e^{-(k_i+k_j)(T-t)} \mathbb{E}^{\mathbb{Q}} [X_t^i X_t^j] \right) \quad [31, Equation 2.1 Page 2],$$

where X_t^i are N Ornstein-Ühlenbeck processes defined by the following dynamics:

$$dX_t^i = -k_i X_t^i dt + dW_t^i, \quad X^i(0) = 0.$$

As proposed by Bayer et al. [2, Section 3.1 Page 14], now that the Rough Bergomi model exists, we can view the traditional N -factor Bergomi model as a Markovian approximation to the Rough Bergomi Model. The N -factor Bergomi model has less realistic exponential kernels, that can be viewed as approximating the power-law kernel present in the Rough Bergomi model.

Now that we have introduced the Rough Bergomi model and its predecessor the Bergomi model, we proceed in the next chapter to investigate how we utilise the Rough Bergomi model for the purpose of derivatives pricing.

Chapter 8

Simulation Techniques

This chapter will detail the relevant simulation techniques used to simulate the Rough Bergomi model, with an emphasis on the techniques used to simulate the Riemann-Liouville process driving it.

8.1 Preliminaries

Now that we have defined the Rough Bergomi model, we can proceed with considering how to utilise the model for derivatives pricing. Unfortunately, because of the non-Markovian nature of the model, we cannot derive a semi-closed or closed form solution under Itô calculus nor Heston's framework [32, Section 2.1 Page 5]. Therefore, in order to be able to price derivatives, we must be forced to simulate the price process of the Rough Bergomi model using Monte Carlo simulations.

8.1.1 Monte Carlo Simulations

Consider a European call option with strike K and maturity T . We can express the price C_t of this call option at time t , as

$$C_t = e^{-r(T-t)} \mathbb{E}^{\mathbb{Q}} \left[(S_T - K)^+ \right].$$

If we cannot calculate the exact value of $\mathbb{E}^{\mathbb{Q}} \left[(S_T - K)^+ \right]$, we can construct an estimate of the price of the options via Monte Carlo simulations.

This involves generating m sample paths of the stock price process S , under the appropriate dynamics and risk-neutral measure. Then, we can construct the following estimator

$$\hat{C}(t) = e^{-r(T-t)} \frac{1}{m} \sum_{i=1}^m ((S_T)_i - K)^+, \quad (8.1.1)$$

where $(S_T)_i$ denotes the i -th simulated path.

This estimator converges to the true value of C_t at a rate of convergence of $O(m^{-1/2})$ [32, Section 3.1 Page 5].

Now in order to construct the stock price path, we first must generate paths of the Brownian motion and fBm driving our model of choice, and subsequently the variance and price processes.

8.1.2 Dependence Structure

With this in mind, we turn our attention back to the Rough Bergomi model. Recall that, in contrast to a standard Brownian motion, fBm has correlated increments. Thus, when simulating the paths of the fBm we require a much more complicated covariance matrix than would be necessary for simulating a standard Brownian motion. When we discretise our time interval, we will require the covariance of the fBm at each point in time, with all other points in time. Thus, in order to simulate the fBm and subsequently the price and variance processes, we need to have the dependence structure of the fBm with itself, and also with the Brownian motion driving the price-process.

Remark 8.1.1. Henceforth all formulae stated will be under the Risk-Neutral Measure \mathbb{Q} . Therefore, we shall drop the explicit reference to \mathbb{Q} .

Taking the initial time as $t = 0$, we recall the specification of the Rough Bergomi model as:

$$S_t = S_0 \mathcal{E} \left(\int_0^t \sqrt{v_u} dZ_u \right)$$

$$v_u = \xi_0(u) \mathcal{E} \left(\eta \sqrt{2H} \int_0^u \frac{1}{(u-s)^\gamma} dW_s \right) = \xi_0(u) \mathcal{E} \left(\eta \tilde{B}_u \right).$$

We shall proceed, by obtaining the various dependence structures of our processes. Taking ρ as our correlation coefficient, the covariances of the Brownian motion Z with the Riemann-Liouville process \tilde{B} , for $v \geq u$ are given by

$$\mathbb{E} \left[\tilde{B}_v Z_u \right] = \rho D_H \left[v^{H+\frac{1}{2}} - (v-u)^{H+\frac{1}{2}} \right],$$

$$\mathbb{E} \left[Z_v \tilde{B}_u \right] = \rho D_H u^{H+\frac{1}{2}} \quad [2, \text{Section 4 Page 15}],$$

where we define

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

We can combine these two formulae to obtain:

$$\mathbb{E} \left[\tilde{B}_v Z_u \right] = \rho D_H \left[v^{H+\frac{1}{2}} - (v - \min(u, v))^{\frac{1}{2}} \right] \quad [2, \text{Section 4 Page 15}]. \quad (8.1.2)$$

For convenience we shall also restate (see (4.5.4)) the covariance structure of the Riemann-Liouville process, where, for $v \geq u$,

$$\mathbb{E} \left[\tilde{B}_v \tilde{B}_u \right] = u^{2H} G \left(\frac{v}{u} \right),$$

where, for $x \geq 1$,

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

$$= \frac{2H}{\frac{1}{2} + H} x^{\frac{1}{2}-H} {}_2F_1 \left(1, \frac{1}{2} - H, \frac{3}{2} + H, x \right),$$

where ${}_2F_1(\cdot)$ is again the Gaussian Hypergeometric function.

Finally, as Z is a Wiener process, we have that

$$\mathbb{E}[Z_u Z_v] = u \wedge v.$$

Now that we have these dependence structures, we can tackle the simulation of the price path of the Rough Bergomi Model. The main challenge of simulating the Rough Bergomi model is finding an efficient method to simulate the Riemann-Liouville process that is driving the variance process. As alluded to above, the simulation of a type 1 or type 2 (Riemann-Liouville) fBm is much more computationally intensive than a standard Brownian motion, as the increments of the fBm are not independent, thus we must correlate multiple Normal random variables appropriately to form the fBm. The next section will explore different method for simulating the Riemann-Liouville process.

8.2 Simulating the Riemann-Liouville process

Here we will present 3 different methods for simulating the Riemann-Liouville process, used in the Rough Bergomi model. The first of these methods involves utilising the Cholesky decomposition of the covariance matrix of the Riemann-Liouville process. This method is an exact method with

no approximation. On the other hand, the second method known as the 'Hybrid scheme', involves certain integral approximations. While this method is not exact, it does improve the speed of computation. Similarly, the third method also involves an approximation, and is not exact in nature.

We must note that there are many other ways to simulate the Riemann-Liouville process. However, not all methods are suitable for the purpose of the stochastic volatility models. In the Rough Bergomi, as in many other fractional stochastic volatility models, the Brownian motion driving the price-process is correlated with the Brownian motion driving the Riemann-Liouville process. We therefore require a method to generate the Riemann-Liouville process that explicitly utilises the integral form of a Riemann-Liouville process, with respect to a standard Brownian motion. This will enable us to model the proper dependence structure between the two Brownian motions.

8.2.1 Cholesky Decomposition Method

The Cholesky decomposition is a decomposition of a Hermitian positive semi-definite matrix. It is used frequently within Monte Carlo simulations as a way of decomposing covariance matrices, as they satisfy the aforementioned criteria. The covariance matrix is decomposed, giving a lower-triangular matrix L , we then take a Normal random vector v , and calculate Lv to obtain a vector of correlated random variables.

Definition 8.2.1 (Cholesky Decomposition of Real-valued Matrix). Let A be a $n \times n$ positive semi-definite matrix with entries in \mathbb{R} . Then the Cholesky decomposition of A exists and is

$$A = LL^T,$$

where L is a $n \times n$ lower-triangular matrix.

Remark 8.2.2. There are multiple algorithms for computing the Cholesky Decomposition of a target matrix, examples of these can be found in [33].

The Cholesky Decomposition method utilises the Cholesky decomposition of the covariance matrix of the process that is to be simulated, in our case the Riemann-Liouville process defined by equation (7.1.3). We consider a discrete realisation of our Riemann-Liouville process $\tilde{B}_{t_0}^H, \tilde{B}_{t_1}^H, \dots, \tilde{B}_{t_n}^H$ on time grid $0 = t_0 < t_1 < \dots < t_n = T$, with time step $h = \frac{T}{n}$.

We now proceed by constructing the $n \times n$ covariance matrix for \tilde{B}_t^H , utilising the dependence structure specified in Section 8.1.2. The covariance matrix is constructed as follows:

$$\Gamma = \begin{bmatrix} \mathbb{E} \left[\tilde{B}_{t_1}^H \tilde{B}_{t_1}^H \right] & \mathbb{E} \left[\tilde{B}_{t_1}^H \tilde{B}_{t_2}^H \right] & \dots & \mathbb{E} \left[\tilde{B}_{t_1}^H \tilde{B}_{t_n}^H \right] \\ \mathbb{E} \left[\tilde{B}_{t_2}^H \tilde{B}_{t_1}^H \right] & \mathbb{E} \left[\tilde{B}_{t_2}^H \tilde{B}_{t_2}^H \right] & \dots & \mathbb{E} \left[\tilde{B}_{t_2}^H \tilde{B}_{t_n}^H \right] \\ \vdots & \vdots & \ddots & \vdots \\ \mathbb{E} \left[\tilde{B}_{t_n}^H \tilde{B}_{t_1}^H \right] & \mathbb{E} \left[\tilde{B}_{t_n}^H \tilde{B}_{t_2}^H \right] & \dots & \mathbb{E} \left[\tilde{B}_{t_n}^H \tilde{B}_{t_n}^H \right] \end{bmatrix}.$$

Γ is a symmetric positive definite matrix, and thus we can compute its Cholesky decomposition

$$\Gamma = LL^T$$

where $L \in \mathbb{R}^{n \times n}$ is a lower triangular matrix with positive real entries. L is of the form

$$L = \begin{pmatrix} l_{1,1} & 0 & \dots & 0 \\ l_{2,1} & l_{2,2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n,1} & l_{n,2} & \dots & l_{n,n} \end{pmatrix}.$$

If we then generate a Normal Random vector $V^T = (v_1, \dots, v_n)$, where v_i are i.i.d samples of standard Normal random variables, we can compute

$$X_i^H = \sum_{k=0}^n l_{i,k} v_k.$$

Then the vector $(0, X_1^H, X_2^H, \dots, X_n^H)$ is a sample path of the Riemann-Liouville process. We can also represent this path as $(0, (LV)_1, (LV)_2, \dots, (LV)_n)$ where LV is a vector computed by the matrix product of L and V .

When simulating multiple paths, we proceed in a similar fashion. To compute m paths of the Riemann-Liouville process, generate m normal random vectors V_i , then store these vectors in a matrix $A = [V_1, V_2, \dots, V_m] = (v_{i,j})$. We then compute

$$X = LA,$$

and append a row of zeros as the new 1st row of X . Then X^T is a $m \times (n+1)$ matrix consisting of m paths of the Riemann-Liouville process, with each path being a row of X^T .

To summarise, the algorithm for computing m paths of the Riemann-Liouville process is given below:

- 1 Construct an evenly spaced discrete time-grid $0 = t_0 < t_1 < \dots < t_n = T$ on our time interval of interest $[0, T]$, with time-step $h = \frac{T}{n}$.
- 2 Construct the covariance matrix Γ for our given time-steps and Hurst parameter H .
- 3 Find the Cholesky decomposition $\Gamma = LL^T$ (see [33] for a range of methods to do so).
- 4 Generate Random Normal Matrix A of size $n \times m$.
- 5 Compute $X = LA$.
- 6 Append a row of zeros as the first row of X .
- 7 X^T now contains m paths of our Riemann-Liouville process.

Remark 8.2.3. The $m \times n$ matrix of increments of the Wiener process W that drives the Riemann-Liouville process, can be obtained as $\Delta W = \sqrt{h}A^T$.

While this method does produce a discrete process with the exact covariance structure specified, it is very computationally intensive. The time complexity of this simulation method is of $\mathcal{O}(n^3)$ [32, Section 3.2.1 Page 7]. Furthermore, this method is also demanding on memory, as the covariance matrix grows with space complexity of $\mathcal{O}(n^2)$. However, a benefit of this method is that the generation of the covariance matrix and subsequent Cholesky decomposition need only be done once, even when many sample paths are required.

8.2.2 Hybrid Scheme Method

The Hybrid scheme and its variants are methods introduced by Bennedsen, Lunde and Pakkanen in 2017 [34], as a way to approximately simulate a class of processes known as 'Brownian semi-stationary' (BSS) processes. For the purpose of the simulating the Riemann-Liouville process, we are specifically interested in the variant of the scheme used to simulate 'Truncated Brownian semi-stationary' (TBSS) processes.

Definition 8.2.4 (Brownian Semistationary process). Let $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t \in \mathbb{R}}, \mathbb{P})$ be a filtered probability space, supporting a standard Brownian motion W_t , and fulfilling the usual conditions for such a space.

- i. Then a Brownian semistationary process is a stochastic process of the form

$$X_t = \int_{-\infty}^t g(t-s)v(s)dW_s \quad [34, \text{Equation 2.1 Page 934}], \quad (8.2.1)$$

where $v = (v-t)_{t \in \mathbb{R}}$ is an $(\mathcal{F}_t)_{t \in \mathbb{R}}$ -predictable process with locally bounded trajectories that drive the volatility of the process, and $g : (0, \infty) \rightarrow [0, \infty)$ is a Borel-measurable function [34, Section 2.1 Page 934].

- ii. Then a truncated Brownian semistationary process is a stochastic process of the form

$$Y_t = \int_0^t g(t-s)v(s)dW_s \quad t \geq 0 \quad [34, \text{Equation 2.14 Page 943}], \quad (8.2.2)$$

where v and g fulfil the same conditions as (8.2.1). Y_t is known as a TBSS as it is obtained from truncating the stochastic integral of a BSS such as X_t , at 0.

If we now turn our attention back to the Riemann-Liouville process used within the Rough Bergomi model (see (7.1.3)), we can see that

$$\tilde{B}_t^H := \sqrt{2H} \int_0^t (t-s)^{H-\frac{1}{2}} dW_s, \quad (8.2.3)$$

is in fact a TBSS, with kernel function $g(x) = \sqrt{2H}x^{H-\frac{1}{2}}$ and $v = 1$.

For a description of how the Hybrid scheme operates, we utilise the words of Bennedsen et al., who stated that the Hybrid scheme works by "approximating the kernel function by a power function near zero and by a step function elsewhere. The resulting approximation of the process is a combination of Wiener integrals of the power function and a Riemann sum" [34, Abstract Page 931].

For the purpose of generating our Riemann-Liouville process (8.2.3), the basis of the scheme proceeds in the following way [34, Section 2.5 Page 943].

Construct a time grid $\pi_t^n := \{t, t - \frac{1}{n}, t - \frac{2}{n}, \dots\}$ for the discretisation of the Riemann-Liouville process, and let $\kappa \geq 1$ be an integer. Then, the discretisation of the Riemann-Liouville process (8.2.3) can be represented as

$$Y_t = \sum_{k=1}^{\infty} \sqrt{2H} \int_{t-\frac{k}{n}}^{t-\frac{k}{n}+\frac{1}{n}} (t-s)^{H-\frac{1}{2}} dW_s \quad [32, Section 3.3.2 Page 8].$$

Now, for $k \leq \kappa$, we do not proceed with any further approximation for $g(t-s) = (t-s)^{H-\frac{1}{2}}$. However for $k > \kappa$, we approximate

$$(t-s)^{H-\frac{1}{2}} \approx \left(\frac{b_k}{n}\right)^{H-\frac{1}{2}}, \quad t-s \in \left[\frac{k-1}{n}, \frac{k}{n}\right]$$

where $b_k \in [k-1, k]$. Applying these approximations, we obtain

$$Y_t \approx \sqrt{2H} \left(\sum_{k=1}^{\kappa} \int_{t-\frac{k}{n}}^{t-\frac{k}{n}+\frac{1}{n}} (t-s)^{H-\frac{1}{2}} dW_s + \sum_{k=\kappa+1}^{\infty} \left(\frac{b_k}{n}\right)^{H-\frac{1}{2}} \int_{t-\frac{k}{n}}^{t-\frac{k}{n}+\frac{1}{n}} dW_s \right) \quad [32, equation 19 Page 7]. \quad (8.2.4)$$

At this point we see that the second sum in (8.2.4) is infinite. Therefore, in order to have a feasible numerical approximation, we truncate this second sum at an appropriate value $N_n \geq \kappa + 1$, so that both sums have the same number of terms in total.

For the choice of b_k , Bennedsen et al. provide the formulae for the values of b_k that minimizes the mean square error induced by the discretisation. These optimal b_k are given by

$$b_k^* = \left(\frac{k^{H+\frac{1}{2}} - (k-1)^{H+\frac{1}{2}}}{H + \frac{1}{2}} \right)^{\frac{1}{H-\frac{1}{2}}}, \quad k \geq \kappa + 1 \quad [34, Proposition 2.8 Page 941].$$

By choosing $b_k = b_k^*$ and truncating the second sum as described, we arrive at the Hybrid scheme approximation

$$Y_t^n := \check{Y}_t^n + \hat{Y}_t^n,$$

where

$$\check{Y}_t^n = \sqrt{2H} \left(\sum_{k=1}^{\kappa} \int_{t-\frac{k}{n}}^{t-\frac{k}{n}+\frac{1}{n}} (t-s)^{H-\frac{1}{2}} dW_s \right), \quad (8.2.5)$$

$$\hat{Y}_t^n = \sum_{k=\kappa+1}^{N_n} \left(\frac{b_k^*}{n}\right)^{H-\frac{1}{2}} \int_{t-\frac{k}{n}}^{t-\frac{k}{n}+\frac{1}{n}} dW_s = \sum_{k=\kappa+1}^{N_n} \left(\frac{b_k^*}{n}\right)^{H-\frac{1}{2}} \left(W_{t-\frac{k}{n}+\frac{1}{n}} - W_{t-\frac{k}{n}}\right) \quad [34, section 2.3 Page 938]. \quad (8.2.6)$$

Let us for now consider a practical implementation of the first order of approximation only, corresponding to $\kappa = 1$. In this case to apply the Hybrid scheme to the simulation of our Riemann-Liouville process \tilde{B}_t^H we follow the following procedure:

Algorithm 8.2.5 (Hybrid scheme with $\kappa = 1$). 1 Create an equidistant time-grid $t_i = \frac{iT}{n}$, $i = 0, 1, 2, \dots, n$, with n time-steps in the interval of interest $[0, T]$. On this time-grid, the Hybrid scheme will generate Y_i , $i = 0, 1, \dots, n$.

2 Now (with $\kappa = 1$), the numerical Hybrid scheme for the Riemann-Liouville process can be written as

$$Y_i = \sqrt{2H} \left(W_{\max\{i-1,0\},1} + \sum_{k=2}^i \left(\frac{b_k^*}{n} \right)^{H-\frac{1}{2}} W_{i-k,2} \right) \quad [32, Equation 20 Page 8] \quad (8.2.7)$$

where for each $i = 0, 1, 2, \dots, n$, $W_{i,1}$ and $W_{i,2}$, are two i.i.d random vectors from a centred bivariate normal distribution with covariance matrix Γ

$$\Gamma = \begin{bmatrix} \frac{1}{n} & \frac{1}{(H+\frac{1}{2})n^{(H+\frac{1}{2})}} \\ \frac{1}{(H+\frac{1}{2})n^{(H+\frac{1}{2})}} & \frac{1}{2Hn^{2H}} \end{bmatrix} \quad [32, Section 3.2.2 Page 8].$$

Thus, we now generate the required matrix Γ , then draw $W_{i,1}$ and $W_{i,2}$ for each i , from the multivariate Gaussian distribution $N(\mathbf{0}, \Gamma)$.

3 Then we compute each Y_i using our vectors $W_{i,1}$ and $W_{i,2}$, and equation (8.2.7).

Remark 8.2.6. As laid out by Matas and Pospisil [32, Section 3.2.2 Page 8], we can represent the right-hand sum of equation (8.2.7) by

$$\sum_{k=2}^i \left(\frac{b_k^*}{n} \right) W_{i-k,2}^n = \sum_{k=1}^i \Gamma_k \Xi_{i-k} = (\Gamma \star \Xi)_i \quad [32, Section 3.2.2 Page 9],$$

where $\Gamma \star \Xi$ stands for discrete convolution and

$$\Gamma_k := \begin{cases} 0, & k = 1, \\ \left(\frac{b_k^*}{n} \right)^{H-\frac{1}{2}}, & k = 2, \dots, i, \end{cases}$$

$$\Xi_k := W_{k,2}, \quad k = 0, 1, \dots, n-1.$$

Then we can represent the approximation for each Y_i from the Hybrid scheme as

$$Y_i = \sqrt{2H} (W_{\max\{i-1,0\},1} + (\Gamma \star \Xi)_i).$$

This is a useful form in particular because the discrete convolution can be effectively calculated using the Fast Fourier Transform (see [35, Section 3.3.4]). Furthermore, under these circumstances, the complexity of the time complexity of the Hybrid scheme is $\mathcal{O}(n \log n)$ [34, Remark 3.2 Page 947].

8.2.3 Rough Donsker Scheme

The Rough Donsker (rDonsker) scheme was introduced by Horvath, Jacquier and Muguruza in 2017 [21], as an alternative approach to the Hybrid scheme (see 8.2.2). The rDonsker scheme is based on an extension of Donsker's theorem (see [36]). As described by Horvath et al. [21, Introduction Page 1], the rDonsker scheme produces an approximating sequence for a Brownian semistationary process, while also keeping track of the underlying Brownian motion. This allows one to easily obtain the path of the driving Brownian motion, and correlate this Brownian motion with additional Brownian motions in a stochastic volatility models as needed. A thorough description of the appropriate theory and convergence of the scheme is laid out in chapters 1-3 of [21, Chapters 1-3], which we encourage the reader to review.

Henceforth, we shall proceed by laying out a practical algorithm for simulating our Riemann-Liouville process \tilde{B}_t^H using the rDonsker scheme. This algorithm was proposed by Horvath, Jacquier and Muguruza as 'Algorithm 3.3' [21, Algorithm 3.3 Page 12], and has been adapted as per the authors instructions for our specific choice of process.

Algorithm 8.2.7 (Simulation of Riemann-Liouville process (8.2.3) using rDonsker Scheme). Consider an equidistant time partition $\frac{ti}{n}$, $i = 0, 1, 2, \dots, n$, for a fixed $n \in \mathbb{N}$, on our interval of interest $[0, T]$. On this time-grid we shall generate m paths of Riemann-Liouville process \tilde{B}_t^H . Let Y_i^j denote the j -th discrete approximation $j = 1, \dots, m$ of \tilde{B}_t^H , evaluated at t_i , $i = 0, 1, 2, \dots, n$. Furthermore, as in the Hybrid scheme setting, let $g(x) = \sqrt{2H}x^{H-\frac{1}{2}}$. Then we proceed as follows:

1 Simulate two $\mathcal{N}(0, 1)$ matrices $\{\xi_{j,i}\}_{j=1,\dots,m, i=1,\dots,n}$ and $\{\zeta_{j,i}\}_{j=1,\dots,m, i=1,\dots,n}$, with $\text{corr}(\xi_{j,i}\zeta_{j,i}) = \rho$.

2 Denote

$$\Delta W_i^j = \sqrt{\frac{T}{n}} \zeta_{j,i}, \quad i = 1, \dots, n, \text{quad}, j = 1, \dots, m.$$

3 Simulate m paths of the Riemann-Liouville process Y by

Remark 8.2.8. Note that the Normal Matrix $\xi_{j,i}$, will be used later to obtain the increments of the Brownian motion within the log-price process simulation. It is more convenient to generate two matrices with correlation coefficient ρ at inception here, rather than correlating two processes later.

$$Y_i^j = \sum_{k=1}^i g(t_{i-k+1}) \Delta W_k^j = \sum_{k=1}^i g(t_k) \Delta W_{i-k+1}^j, \quad i = 1, \dots, n, \quad \text{and} \quad j = 1, \dots, m,$$

As in the Hybrid scheme case, we complete this step using discrete convolution with complexity $\mathcal{O}(n \log n)$ (see [21, Appendix B Page 26]).

8.2.4 Comparing Simulation Methods for the Riemann-Liouville process

Accuracy of computation

As both the rDonsker and Hybrid scheme methods are approximations, it is prudent to investigate whether the paths produced by these schemes behave in the way we would expect the Riemann-Liouville process to behave. Matas and Pospisil [32, Section 4.1 Page 11], conducted such an investigation, comparing the absolute sample moments of Riemann-Liouville paths generated by the Cholesky decomposition, rDonsker and Hybrid schemes, with the theoretical moments of the Riemann-Liouville process. Here the Cholesky decomposition method is taken as a benchmark, as the discrete-time paths generated by it will have the exact covariance structure of a discrete-time Riemann-Liouville process, and thus should be our 'best' approximation to the continuous time Riemann-Liouville process. Rather remarkably, Matas and Pospisil found that the sample paths generated by the approximate schemes (rDonsker and the Hybrid scheme), were of a similar quality to those generated by the exact Cholesky decomposition method [32, Section 4.1 Page 12]. As all these methods can be considered to generate paths which resemble a Riemann-Liouville process equally well, we must turn to another metric to deduce which method of simulation may be more appropriate or optimal.

Speed of computation

From the descriptions of the methods above, we can already compare the asymptotic time complexity of the simulation of one path of the Riemann-Liouville process for our various methods. Under big \mathcal{O} notation, and with n being the number of time-steps within our interval of interest, we saw that the Cholesky decomposition method had a asymptotic time complexity of $\mathcal{O}(n^3)$, whereas the Hybrid-scheme and rDonsker method shared an asymptotic time complexity of $\mathcal{O}(n \log n)$. Therefore, we can see that the approximate methods are more efficient than the exact Cholesky decomposition method. However, as noted by Matas and Pospisil [32, Section 4.1 Page 13], the asymptotic time complexity does not convey which method is superior at generating m paths when m is large. We must then, for each method, consider the effect of increasing the number of paths m , rather than the number of time-steps n .

If we first consider the Cholesky decomposition method, we can see that this method is easily vectorised. In fact, through the use of matrix notation, the method is vectorised implicitly [32,

Section 4.1 Page 13]. When we perform the Cholesky decomposition method, we only need to compute the required covariance matrix Γ and subsequent Cholesky decomposition once. Furthermore, the size of the covariance matrix is only affected by a change in n , and not a change in m . Once we have computed the Cholesky decomposition $\Gamma = LL^T$, we must matrix multiply L with A , the $n \times m$ Normal random matrix. Here, we see that the size of A is affected by an increase in m , and subsequently we can deduce that increasing the number of paths m will increase the time of computation, through the time required to generate A and the time required for the matrix multiplication.

Conversely, neither the Hybrid scheme nor the rDonsker scheme is easily vectorised, due to the discrete convolutions that is required [32, Section 4.1 Page 13]. Therefore, in order to compute m paths, we must loop the computation of a single path m times. Therefore, the runtime for m paths should increase linearly in m .

Matas and Pospisil [32, Section 4.2 Page 14] subsequently performed an empirical investigation into the run-times of the aforementioned simulation methods. The authors measured the run-time of producing m paths with n steps of the Riemann-Liouville process, for $100 \leq m \leq 150,000$ and $250 \leq n \leq 10,000$. We summarise their findings below [32, Section 4.2 Page 14]:

- 1 For $m < 100$ we see that the rDonsker scheme is the fastest for all n considered, closely followed by the Hybrid scheme.
- 2 As m increases, we see that the Cholesky decomposition becomes the fastest. If we denote the value of m for which the Cholesky decomposition becomes the fastest as m^* , we see that m^* varies on n , with m^* being in the range $1000 < m^* < 10,000$.
- 3 For $m \geq 10,000$ the Cholesky decomposition is the fastest, for all n considered, followed by the rDonsker.

For the purpose of derivatives pricing, we typically use at least 50,000 paths. Therefore, for any m of practical use we can see that the Cholesky decomposition method is in fact the fastest, for any n .

Although the asymptotic time complexity of the Cholesky decomposition method is much larger than that of the rDonsker and Hybrid schemes, we saw that once large number of paths m are considered, that the Cholesky decomposition method becomes the fastest. Therefore, given that the Hybrid scheme and rDonsker method are approximations and slower than the Cholesky decomposition method, it seems appropriate to utilise the Cholesky decomposition for most cases. We do note that the rDonsker or Hybrid scheme may be more optimal in a situation where we need to generate few paths m , but with a high granularity.

8.2.5 Choice of Riemann-Liouville process simulation technique

While the analysis of Matas and Pospisil [32, Figure 2 Page 15] suggests that for the Cholesky decomposition may be faster than the rDonsker and Hybrid scheme method, we shall not proceed with this method due to memory constraints. Memory constrains come about when utilising the Cholesky decomposition method due to the generation of the large covariance matrix. Unfortunately, due to memory constrains on each core we have available, we are forced to make use of the less memory-demanding techniques such as the rDonsker method or the Hybrid scheme. Now, left with the choice of the two remaining schemes, we choose to utilise the Hybrid scheme at this point, and leave an exploration of the rDonsker scheme to further research. We choose the Hybrid scheme due to the availability of literature on the method, from which we can corroborate our results with.

8.3 Simulating the Rough Bergomi Model

Once we have simulated the Riemann-Liouville process \tilde{B}_t^H using one of the aforementioned methods, we are left to simulate the variance process, and subsequently the price process of the model. For this, we will rely on using the Euler scheme, and proceed by utilising the following algorithm.

Algorithm 8.3.1 (Simulation of the Rough Bergomi Model). Again, we consider the same equidistant time partition $0 = t_0 < t_1 < \dots < t_n = T$ on our time interval of interest $[0, T]$, with time-step

$h = \frac{T}{n}$, for a fixed $n \in \mathbb{N}$, as we used for the generation of the Riemann-Liouville process. On this time-grid we shall generate m paths of the variance process and price process of the Rough Bergomi model. Furthermore, let Y_i^j, v_i^j, S_i^j denote the j -th discrete approximation $j = 1, \dots, m$ of the Riemann-Liouville process \tilde{B}_t^H , variance process v_t and price process S_t respectively, evaluated at $t_i, i = 0, 1, 2, \dots, n$. For the purpose of this algorithm all i will be $i = 0, 1, 2, \dots, n$, and all j will be $j = 1, \dots, m$, and without loss of generality we consider the risk-free rate $r = 0$ for the time-being.

- 1 Simulate m sample paths of the Riemann-Liouville process Y_i^j , by one of the aforementioned simulation methods (we shall utilise the Hybrid scheme method). Then, for each method we also store the increments of the Wiener process driving the Riemann-Liouville process, we denote these as ΔW_i^j .
- 2 At each time-step t_i compute m values of the variance process using $v_{t_i}^j = \xi_0(t_i)\mathcal{E}(\eta Y_i^j)$.
- 3 Simulate m paths of an independent Wiener process $W_i'^j$, with increments $\Delta W_i'^j$. Then construct

$$Z_i^j = \rho W_i^j + \sqrt{1 - \rho^2} W_i'^j,$$

whose increments we denote as ΔZ_i^j . We note that this step was already implicitly performed in Step 1 of the rDonsker Algorithm (8.2.7), thus we can simply utilise the stored increments within the second Normal matrix we generated.

- 4 Utilise the forward Euler scheme to simulate m sample paths of our price-process S by

$$S_{i+1}^j = S_i^j + \sqrt{v_i^j} S_i^j (\Delta Z_i^j).$$

8.3.1 Pricing Multiple Options

For the purpose of calibration, and also just for general use, it is useful to price multiple vanilla options using a single set of simulations.

Consider the following example, where we wish to price 2 call options, C_1 and C_2 , with strikes k_1 and k_2 respectively, and maturities T_1 and T_2 respectively. We could proceed with the simple approach of simulating m paths of the price-process up to T_1 and pricing C_1 , and then simulating another m paths to price C_2 . However, it is much more efficient to simply simulate m paths of the price-process up to $\max(T_1, T_2)$, and store S_{T_1} and S_{T_2} for each path. Then we can evaluate the payoff of each option at their respective maturities, and price each option using one single set of m paths. The efficiency of this techniques becomes even more apparent as we add more options. Consider the case where we would like to price options C_1, \dots, C_n , where there are $\frac{n}{k}$ strikes considered for each of the k maturities. We simply simulate m paths of the Rough Bergomi price process up to $\max(T_1, \dots, T_k)$, and store S_{T_i} for each of the m paths at each of the k maturities. Then for each path we can evaluate each of the $\frac{n}{k}$ payoffs at each maturity, and finally compute the prices of the n options. Therefore, we can price an arbitrarily large number of options with a single set of m simulations, provided that we have enough available memory to store the values of the price-process as each desired maturity. We can think of this technique as pricing a maturity-strike grid of options all at once, and will refer to it as 'grid-pricing' throughout this paper.

8.3.2 Parallelisation

As we will be generating a large number of sample paths for the purpose of Monte Carlo simulation, it is extremely useful to consider parallelisation. To put it briefly, we shall distribute the m sample paths that we wish to simulate into b batches, with each batch running on a different core. This works well with Monte Carlo price simulations, as no information is needed from the other $m - 1$ paths, for the purpose of generating one path. Once all the $\frac{m}{b}$ paths are generated for each batch, we simply store the values of the paths at the times of interest, typically the maturities of the options we wish to simulate. We can then take a batch-wise average of each payoff we wish to evaluate, and then take a final average of the b batch-wise averages, to generate our final price estimate.

Parallelisation is also easily applied to the grid-pricing methodology. We again split our required m simulations into b batches. Once each batch generates a price-grid of options, we simply take

a component-wise average of all b price-grids, to obtain our final price-grid of Monte Carlo prices. For the rest of this paper, we shall be parallelising our Monte Carlo simulations in this way, to speed up the computation.

8.3.3 Variance reduction - "Turbocharging" Monte Carlo

Unfortunately, simulating rough volatility models by Monte Carlo simulations can be time-demanding, as the estimators produced have a high variance and thus a large number of simulations is needed to have precision. To remedy this issue, we can consider a range of variance reduction techniques, to reduce the variance of the estimator, so that we require less simulations to reach our desired level of precision. Notable variance reduction techniques include antithetic sampling, importance sampling, and control variates. These are all standard techniques that have been implemented in various ways across different models. In this subsection we will briefly explore a notable variance reduction technique designed specifically for Rough Volatility models, that was proposed by McCrickerd and Pakkanen in 2018 [37].

McCrickerd and Pakkanen [37] refer to their variance reduction technique as "turbocharging". The idea of "turbocharging", in the context of option pricing, is to construct a mixed estimator for the price of an option, instead of using the standard Monte Carlo estimator (8.1.1).

Suppose we are in the framework of Section 8.1.1, and again wish to price the European Call Option with strike K and maturity T .

In the context of the Rough Bergomi model, this mixed estimator is calculated by first considering the orthogonal separation of the price process S_t , into S_t^1 and S_t^2 where

$$\begin{aligned} S_t^1 &:= \mathcal{E} \left(\rho \int_0^t \sqrt{v_u} dW_u^1 \right) \\ S_t^2 &:= \mathcal{E} \left(\sqrt{1 - \rho^2} \int_0^t \sqrt{v_u} dW_u^2 \right) \quad [37, \text{Section 2 Page 5}] \end{aligned}$$

Remark 8.3.2. W_t^1 is the Brownian motion driving the Riemann-Liouville process used in the Rough Bergomi Model.

We can see that this notion of orthogonal separation comes about naturally, by writing the Rough Bergomi model (with $S_0 = 1$) in the following way:

$$\begin{aligned} S_t &= \mathcal{E} \left(\int_0^t \sqrt{v_u} d(\rho W_u^1 + \sqrt{1 - \rho^2} W_u^2) \right) \\ v_t &= \xi_0(t) \mathcal{E} \left(\eta \tilde{B}_t^H \right) \\ \tilde{B}_t^H &:= \sqrt{2H} \int_0^t (t - u)^{H - \frac{1}{2}} dW_u^1 \end{aligned}$$

Then, the "turbocharging" mixed MC estimator of McCrickerd and Pakkanen, for m sample paths, is defined as:

$$\begin{aligned} \tilde{C}(t) &= \frac{1}{m} \sum_{i=1}^m (X_i + \hat{\omega} Y_i) - \hat{\omega} \mathbb{E}[Y] \\ X &= \text{BS} \left(S_t^1, K, T, (1 - \rho^2) \int_0^t v_u du, t \right) \\ Y &= \text{BS} \left(S_t^1, K, T, \rho^2 \left(\hat{Q} - \int_0^t v_u du \right), t \right), \quad [32, \text{Equation 21 Page 11}], \end{aligned}$$

where $\text{BS}(\cdot)$ is the standard Black-Scholes formula for a call option, where we note the integral expressions are being used as the volatility value within the formulae. Furthermore, $\hat{\omega}$ and \hat{Q} are constants computed post-simulation, with $\hat{\omega}$ in particular being calculated from sampled X_i and Y_i values. $\hat{\omega}$ and \hat{Q} are defined as follows:

$$\hat{\omega} = -\frac{\sum_{i=1}^m (X_i - \hat{X})(Y_i - \hat{Y})}{\sum_{i=1}^m (Y_i - \hat{Y})^2}$$

$$\hat{Q} = \max \left\{ \left(\int_0^t v_u \, du \right)_i : i = 1, \dots, m \right\}, \quad [32, \text{Equation 23 Page 11}].$$

The mixed estimator $\bar{C}(t)$ is always a biased estimator of $C(t)$, as the Black-Scholes formulae is non-linear [32, Section 3.3 Page 11]. However, McCrickerd and Pakkanen state that, for $m > 1000$ the bias is not practically significant [37, Section 3.2 Page 10]. Furthermore, McCrickerd and Pakkanen found that their "turbocharged" estimator reduces the run-time of Monte Carlo simulations, until a specific standard deviation of estimator is reached, by up to 34 times, when compared to the base Monte Carlo estimator [37, Section 3.2 Page 10].

In their 2018 paper [37], McCrickerd and Pakkanen applied their "Turbocharging" method to the simulation of the Rough Bergomi model, where the Hybrid scheme with $\kappa = 1$ was used for the generation of the Riemann-Liouville process. However, we note that the turbocharging method is not meant to be exclusively used in conjunction with the Hybrid method [32, Section 3.3 Page 11]. The turbocharging variance reduction technique can be utilised with a simulation of the Riemann-Liouville process by the rDonsker method or the Cholesky decomposition method.

While the use of 'turbocharging' does seem promising, the recent work of Matas and Pospíšil [32] brought some drawbacks of the method to light. Matas and Pospíšil found that, with certain parameter choices, the 'turbocharging' methodology produced negative prices for vanilla options [?], suggesting that the method is malfunctioning in some way. Furthermore, Matas and Pospíšil found that for deep OTM options, the variance was actually increased when compared to the standard Hybrid scheme, and that the bias of the price estimates was large [32, Figure 5 Page 19]. Therefore, as a result of this analysis, we have decided not to utilise the 'turbocharging' methodology in its current state, due to its lack of tractability, the potential presence of bias, and the potential presence of negative prices.

8.4 Market Information

Up to now, we have not included the use of any market information into the simulation of the Rough Bergomi model. In order for us to calibrate the model, and for us to utilise it for the pricing of derivatives, we must incorporate appropriate market information into the simulation of the Rough Bergomi model. Furthermore, for the purpose of calibration, we will make use of information provided by the market implied volatility surface of our chosen asset.

Given a specific date, which we shall utilise as our initial time $t_0 = 0$, and a specific asset, the first and most obvious piece of market information we will utilise is the spot price of the asset. We will take the closing price of our asset on this date as our initial value S_0 . Moreover, we choose to take the closing value of any market information on a specified date as the value for that date. The spot value is integral to our diffusion as it specifies the initial value of our price-process. Aside from the spot value, we require knowledge of the Forward curve, Discount curve and Forward variance curve for the simulation of both our price and variance diffusion processes.

8.4.1 Forward Curve

Up to this point we had largely ignored the effect of interest rates within our diffusion for the price-process. We rationalised this by choosing to only consider deterministic interest rates, and then without loss of generality we set these rates equal to zero. Unfortunately, real markets rarely have interest rates equal to zero, and thus in order for our model to properly simulate our chosen assets price process we must utilise the appropriate interest rate. In a simplified market, the constant risk-free rate is the drift coefficient of our assets price-process once we have changed from the real-world probability measure \mathbb{P} to the risk-neutral measure \mathbb{Q} . However, in real equity markets we do not in practice observe a constant risk-free rate, instead the risk-free rate changes over time. Therefore, for the purpose of using the Rough Bergomi model for pricing equity derivatives, we continue under the assumption of deterministic interest rates, but we no longer operate under the assumption of a constant risk free rate. To complicate things more, equity markets also often operate with the presence of dividends and borrow rates. Fortunately, all of the information about

the borrow rate of an asset, its dividends, and the risk-free rate can be approximated from the forward curve of the chosen asset.

Remark 8.4.1. We shall consider dividends in the form of continuous dividends, such that we avoid large jumps in our price process.

We will work under the assumption that at $t = 0$, the risk-free rate r_t , the borrow rate b_t and the continuous dividend q_t , are all built into the forward curve of our asset for $t \in [0, T]$.

Now, with the inclusion of non-zero r, b, q , in the Rough Bergomi model under \mathbb{Q} we have that:

$$dS_t = \mu_t S_t dt + \sqrt{v_t} S_t dZ_t,$$

where $\mu_t = r_t - b_t - q_t$.

Let us denote the forward price of S at time T , viewed at time t , as $F(t, T)$. Then, given deterministic r, b, q we have that

$$F(t, T) = S_t \times \exp\left(\int_t^T \mu_s ds\right)$$

Now, let π^n be the time partition defined as the sequence t_1, t_2, \dots, t_n , such that $t_i < t_{i+1}$ for all i , $0 = t_0$, $t_n = T$, and $t_{i+1} - t_i = \frac{T}{n}$. The sequence π^n will be the discrete time-grid used for our asset diffusion. Now, at time $t = 0$, we can observe the forward price $F(0, t)$ of our asset, for discrete times $t \in \{\bar{t}_1, \bar{t}_2, \dots, \bar{t}_n\} \in [0, T]$ where T is our final date of interest. Forward prices may only be observable from the market at discrete dates, therefore we shall construct a forward curve such that we have an approximate forward value at each of the time-steps of our diffusion process. Given discrete forward prices $\{F(0, t) : t \in \{\bar{t}_1, \bar{t}_2, \dots, \bar{t}_n\}\}$, we can construct a forward curve through interpolation. The exact choice of methodology for the construction of the forward curve is outside of the scope of this paper. We shall assume that we have obtained a continuous forward curve, through an appropriate interpolation or bootstrapping scheme. The continuous forward curve will contain approximate forward prices for all $t \in [0, T]$, and from this we can extract a forward value for each time-step of our diffusion. Now that we have $F(0, t_i)$ for all t_i in our diffusion-time grid, we assume the drift term μ_{t_i} is constant on the interval $[t_i, t_{i+1})$ for all i , then we can approximate the discrete drift term μ_{t_i} via the following:

$$\begin{aligned} F(0, t_{i+1}) &= S_0 \times \exp\left(\int_0^{t_{i+1}} \mu_s ds\right) = S_0 \times \exp\left(\int_0^{t_i} \mu_s ds + \int_{t_i}^{t_{i+1}} \mu_s ds\right) \\ \implies \frac{F(0, t_{i+1})}{F(0, t_i)} &= \frac{S_0 \times \exp\left(\int_0^{t_i} \mu_s ds + \int_{t_i}^{t_{i+1}} \mu_s ds\right)}{S_0 \times \exp\left(\int_0^{t_i} \mu_s ds\right)} = \exp\left(\int_{t_i}^{t_{i+1}} \mu_s ds\right) \\ \implies \ln\left(\frac{F(0, t_{i+1})}{F(0, t_i)}\right) &= \int_{t_i}^{t_{i+1}} \mu_s ds. \end{aligned}$$

Then, as we assumed μ_{t_i} is constant on the interval $[t_i, t_{i+1})$, we have that

$$\ln\left(\frac{F(0, t_{i+1})}{F(0, t_i)}\right) = \int_{t_i}^{t_{i+1}} \mu_s ds = \mu_{t_i} \times (t_{i+1}) - \mu_{t_i} \times (t_i) = \mu_{t_i} \Delta t \quad (8.4.1)$$

where $\Delta t = t_{i+1} - t_i$.

While this is still an approximation, as we have assumed that μ_t is constant on the interval $[t_i, t_{i+1})$, we shall use (8.4.1) and our interpolated forward curve to obtain the drift term for each time interval in our Euler scheme for the assets price-process. Therefore, we were able to incorporate all of the information about r, b , and q , all from a set of forward prices observable in the market at $t = 0$.

8.4.2 Variance Swap Curve

The Rough Bergomi Model, being a modification of the Bergomi model, is a forward variance model, requiring as an input, the forward variance curve at time $t = 0$. In order to obtain the

forward variance curve, we make use of the forward variance swap curve, constructed from the fair-strikes of variance swaps for increasing maturities. As stated by Bayer et al. [2, Section 5.1 Page 17], "Variance swaps are actively traded so in principle, computation of the forward variance swap curve should be straightforward". Unfortunately, reliable variance swap quotes are only obtainable for a selection of dates. Thus, in order to construct a variance swap curve, we need to find a way to estimate the fair strike of variance swaps for intermediary maturity dates. One way to construct the forward variance swap curve is perform a monotonic spline to interpolate between existing variance swap fair strikes, as done so in [2, Section 5.1 Page 17]. Another way to construct the curve is to price variance swaps at the intermediary maturity dates that we are interested in, using information obtained from the implied volatility surface, and then perform a suitable interpolation technique. There are numerous other ways to construct such the forward variance swap curve, including techniques which derive the curve directly from observed vanilla options (see [38]), however the specifications of these methods are out of the scope of this work. For the purpose of this work, we proceeded with the monotonic spline interpolation laid out by Bayer et al. in [2, Section 5.1 Page 17].

Once we have obtained an interpolated curve of the fair strikes of variance swaps for varying maturities within our time span of interest $[0, T]$, we can extract forward variance curve, via the following relation

$$\xi^T(t) = \partial_T V_t^T, \quad T \geq t \geq 0 \quad [39, \text{Section 1.1 Page 3}]$$

where we recall from (2.3.1) that V_t^T is the price of a variance swap with maturity T viewed at time t , and we recall from (7.1.12) that $\xi^T(t)$ is the forward variance curve. For a more in-depth discussion of forward variance dynamics see [39].

8.4.3 Discount Curve

Now, given that we have completed the simulation of our assets price-process, and computed payoffs of interest, we are left to discount the payoffs back to $t = 0$. Unfortunately, we no longer work under the assumption of a constant risk-free rate, and thus in order to do this, we need to know the discount curve of our market. Discount curves can be constructed via observing certain interest rate products, such as swaps, and bootstrapping a curve together. The specifications of the various methodologies of bootstrapping a discount curve are out of the scope of this paper, and thus we shall work under the pretense that we have obtained a bootstrapped discount curve containing discount factors for each time-step in π^n , the time-grid for our discrete diffusion. With this curve at hand, we can simply utilise the appropriate discount factor for each payoff of interest, discounting that payoff back to $t = 0$ to obtain our price estimate for the derivative.

8.4.4 Implied volatility surface

The market-implied volatility surface, for a given asset, is important for the construction of the variance swap curve (as described in 8.4.2), and for calibration of the model. For the purpose of model calibration, the market implied volatility surface is used to assess the 'goodness of fit' of the model generated implied volatility surface. The exact cost function and calibration methodologies will be covered later in 10. Certain calibration methodologies may require a smooth, arbitrage-free, parameterised surface, such as the one provided by the SVI parameterised volatility surface (see [40]), while other calibration methodologies simply require implied volatilities on a discrete strike by maturity grid. We need at minimum a selection of implied volatility estimates, whether they be interpolated or directly observed in the market, for each of the options we will use for our calibration. Again, the construction of an implied volatility surface is outside of the scope of this paper, and we choose to operate under the pretense that we have been provided with a suitable implied volatility surface for our asset at $t = 0$. Then, we may utilise the surface for the purpose of calibrating our Rough Bergomi model.

With market data at hand and a simulation method chosen, we can begin to simulate the Rough Bergomi model. Before we delve into calibrating the model, we shall explore the effects of the parameters of the mode, through the use of our chosen simulation technique.

Chapter 9

Parameters of the Rough Bergomi Model

The Rough Bergomi model has been praised for its ability to fit a wide range of shapes of volatility surfaces with just three parameters: H , η and ρ . Before we proceed the calibration of the model, we shall explore the effect of each parameter on the behaviour of the model.

9.1 Exploring the Effect of the Rough Bergomi Parameters

In order to explore the effect of each parameter, we have generated volatility smiles and skews, while varying individual parameters of the Rough Bergomi model or combinations of them. All of the Riemann-Liouville process paths utilised in this analysis were generated using the Hybrid scheme method. Traditional Monte Carlo option price estimators were used to calculate the price of each option, from which the implied volatilities and subsequently skews were calculated. Each option price was calculated using 200,000 simulated stock price paths, with each path using daily time-steps for 252 trading days per year, up to a time horizon of 5 years. Moreover, for the purpose of this exploratory investigation, a flat forward variance curve was utilised, and deterministic interest rates of $r = 0$ were used, unless otherwise stated.

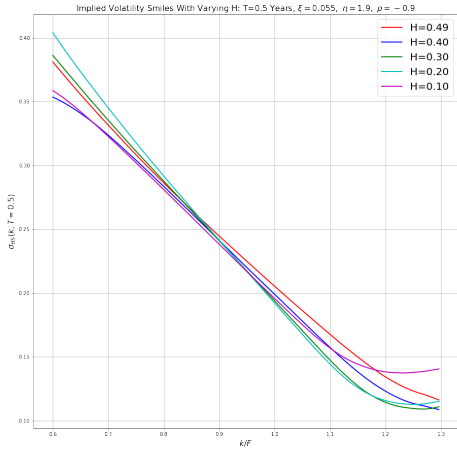
9.1.1 The effect of H

In each of the following plots, we fix ρ, η, ξ and vary H .

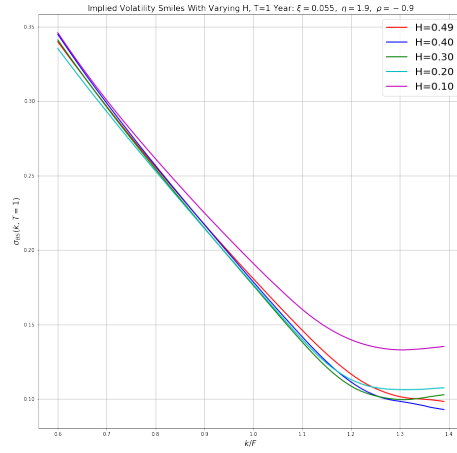
Unfortunately, it is hard to deduce much about the effect of H from Figure 9.1. Nonetheless, we can begin to see that the minimum of each smile moves upwards as H decreases. In order to deduce more about the effect of H , we turn to the term structure of the model generated ATM forward skew. For the following plots of the ATM Skew, the forward variance curve, discount curve, forward price curve, and stock initial value, used in the diffusion for both the variance process and asset price process were all constructed from data for the SPX Index on the 30th May 2022.

Figure 9.2 shows the effect of varying H on the term structure of the model generate ATM forward skew. From Figure 9.2(a) we can see that as we begin to decrease H , we see that the model delivers a lower long-term skew, with a steeper increase in the term structure of the short-term skew, and a higher value for the skew in the extreme short-term. However, we do see slightly different behaviour once $H \leq 0.10$. Figure 9.2(b) and Figure 9.2(c) were then plotted to explore the difference in the behaviour of the term structure of the skews, above and below $H = 0.10$ respectively. From Figure 9.2(b) we can confirm the behaviour we saw in Figure 9.2(a), with decreasing values of H leading to a sharper decay in skew for small τ . Turning to Figure 9.2(c) we begin to see that further decreasing the value of H , again leads to a lower long-term skew, but also leads to a noticeably less steep term-structure in the short term, with the skew reaching a lower maximum value in the extreme-short term. We conclude that the most notable effect of H , is to control the steepness of the term-structure. However we note that for $H \leq k$, with $k \approx 0.10$ for our parameter choices, we see that a decrease in H no longer increasing the steepness of the term-structure for small τ , but in fact decreases the steepness.

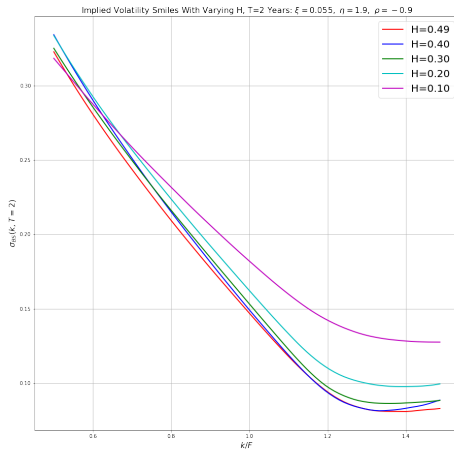
In order to further investigate how H affects the ATM Forward skew, we can search for a closed form approximation of the ATM skew. Under the special case of a flat variance curve,



(a) $T = 6$ months.

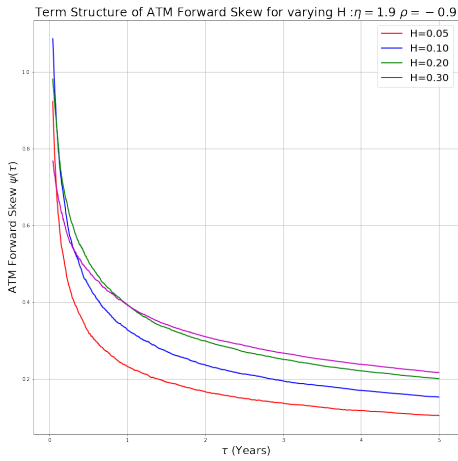


(b) $T = 1$ Year.

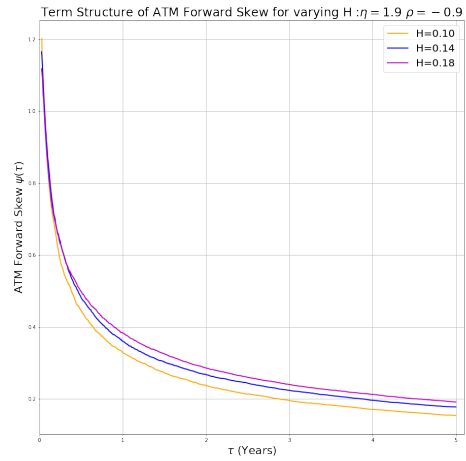


(c) $T = 2$ Years.

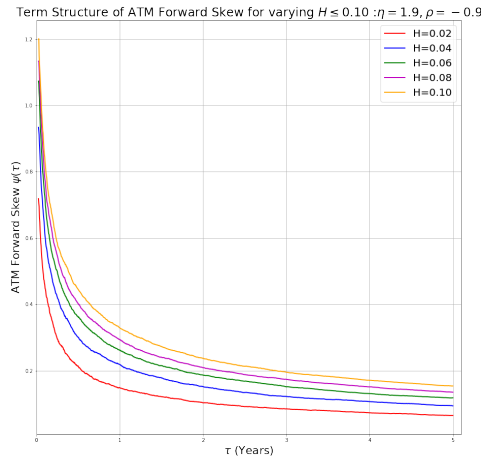
Figure 9.1: Implied volatility smiles generated by the Rough Bergomi Model (Hybrid scheme method), with varying values of H , and fixed $\rho = -0.9$, $\eta = 1.9$, $\xi = 0.055$



(a) ATM forward skew, $0 < H < 0.5$.



(b) ATM forward skew, $0.10 \leq H < 0.5$.



(c) ATM forward skew, $H \leq 0.10$.

Figure 9.2: Term structures of Rough Bergomi (Hybrid scheme method) ATM forward skew, with varying values of H , and fixed $\rho = -0.9$, $\eta = 1.9$, ξ constructed from SPX Data from the 30th May 2022 (Note: The shortest time to maturity considered here is 5 business days).

where $\xi_0(u) = \bar{\sigma}^2$, Gatheral et al. [2, Section 7.1 Page 28] derived the following expression for the ATM forward volatility in the Rough Bergomi model, up to the second order in η :

$$\psi(\tau) = \frac{\rho\eta}{2} E_H \frac{1}{\tau^{\frac{1}{2}-H}} + \frac{1}{4} \rho^2 \eta^2 \bar{\sigma} \tau^{2H} \left[\frac{D_H^2}{1+H} \left(1 + \frac{\Gamma(H + \frac{3}{2})^2}{\Gamma(2H+3)} \right) - \frac{3}{2} E_H^2 \right], \quad (9.1.1)$$

where $E_H = \frac{D_H}{H+\frac{3}{2}}$ and $D_H = \frac{\sqrt{2H}}{H+\frac{1}{2}}$ [2, Equation 7.6 Page 28].

With this expression at hand, we can plot the value of the ATM Forward Skew for varying values of H , and gain further insight into how H affects the value of the skew, particularly for small τ .

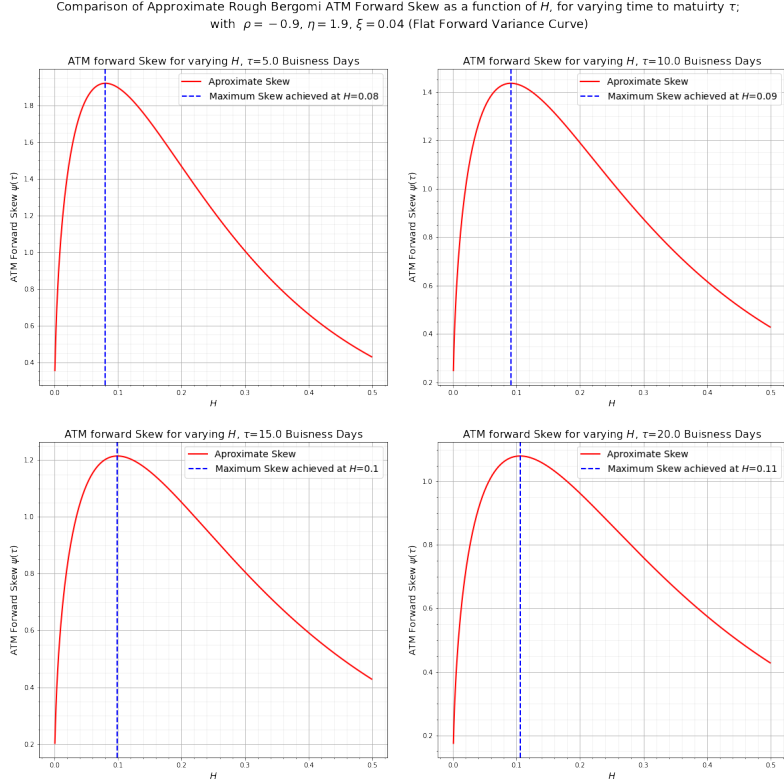


Figure 9.3: Plots of the approximate ATM forward skew as a function of H , for varying τ . Each plot has the value of H that maximises $\psi(\tau)$. These values were generated using equation (9.1.1), with $\rho = -0.9$, $\eta = 1.9$, $\xi = 0.04$ (flat forward variance curve)

Figure 9.3 shows clearly that, for small τ , the approximate ATM forward skew decreases rapidly once H passes a value around 0.10 as it approaches 0. While we must note that this is only an approximation to the special case of a flat variance curve, we do see that Figure 9.3 explains the change in behaviour we see as we decrease H past 0.10 that we saw in Figure 9.2(a). This phenomena is important as one of the primary goals of the Rough Bergomi model is to fit the rapid increase observed in the term structure of the ATM forward skew. Without knowledge of this phenomena, a practitioner may naively choose an extremely low value of H , in an attempt to maximise the short-term ATM forward skew.

Furthermore, if we let the value of H that maximises the approximate forward skew at maturity τ be $H^*(\tau)$, a function of τ , then from Figure 9.3 we see that H^* is increasing in τ . We can then plot $H^*(\tau)$ against τ , to learn more about the behaviour of $H^*(\tau)$.

Figure 9.4 shows the value of $H^*(\tau)$ as τ increases. Initially we only considered $H^* \in (0, \frac{1}{2})$ as this is the applicable range of H for the Rough Bergomi model. While the approximation formulae given by equation (9.1.1) was derived specifically for the Rough Bergomi model, where $H \in (0, \frac{1}{2})$, it is interesting to gently explore the behaviour of $H^*(\tau) \in (0, 1)$. In both cases, we see that in order to maximise $\psi(\tau)$ at increasingly longer maturities, we must increase H . Furthermore, in

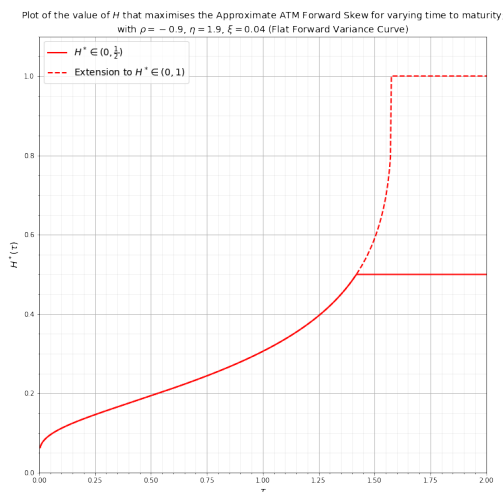


Figure 9.4: Plot of $H^*(\tau)$ as a function of τ , considering both $H^* \in (0, \frac{1}{2})$ and $H^* \in (0, 1)$. These values were generated using equation (9.1.1), with $\rho = -0.9$, $\eta = 1.9$, $\xi = 0.04$ (flat forward variance Curve)

both cases we see the value of $H^*(\tau)$ increases approaching 0.5 and 1 respectively. These plot also serve to explain the decrease in long-term ATM forward skew that we saw when decreasing H in Figure 9.2(a).

To conclude, the main effect of H is to control the decay of the ATM forward skew for small τ . Starting from $H = 0.49$, decreasing H leads to a steeper term structure of the ATM forward skew for small τ , up until a point, after which further decreases in the value of H lead to a shallower term structure of the ATM forward skew for small τ . Further analysis into the behaviour of the Rough Bergomi model for extremely small H can be found in [41] and [42].

9.1.2 The effects of ρ and η

The effects of ρ and η on the Rough Bergomi model are intertwined and thus we shall consider them together. Fortunately, in contrast to what we experienced when investigating the effect of H , we can gain a large insight into the effects of ρ and η through plotting volatility smiles.

From Figure 9.5(a) we can see that the minimum of the smile decreases in value and moves further along the strike axis as ρ decreases. Turning to Figure 9.5(b) we can see that a decrease in η causes a similar movement in the smile, with the minimum once again dropping and moving to the right. Finally, from Figure 9.5(c), we see that, even with $\rho \times \eta$ fixed, a decrease in ρ still results in a move of the minimum of each smile to the right, with the value of said minimum decreasing.

When consider ρ , it is tempting to think of it as the correlation between the process driving the price-process and the process driving the variance process, as it is in many other traditional stochastic volatility models. However, we must recall that this it not true in our case. Recalling equation (7.2.1), we see that ρ is the correlation coefficient between Z_t the Brownian motion driving the price-process, and W_t the Brownian motion driving the Riemann-Liouville process. With this in mind, if we wish to calculate the correlation between Z_t and B_t^H we can proceed as follows:

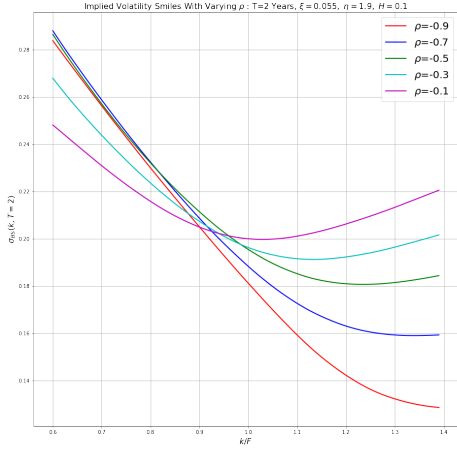
From equation (8.1.2), we can substitute $u = v = t$ to obtain the following

$$\mathbb{E} [B_t^H Z_t] = \rho D_H t^{H+\frac{1}{2}},$$

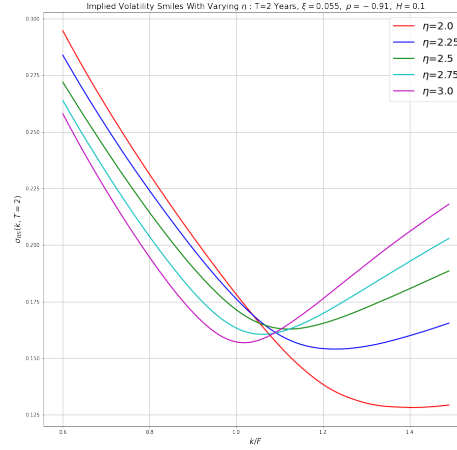
where we recall $D_H = \frac{\sqrt{2H}}{H+\frac{1}{2}}$. Now, from remark 4.5.9 we know that $\text{Var}(B_t^H) = t^{2H}$, and we also recall that for a Brownian motion we have that $\text{Var}(Z_t) = t$.

Then, we can calculate the correlation between Z_t and B_t^H as

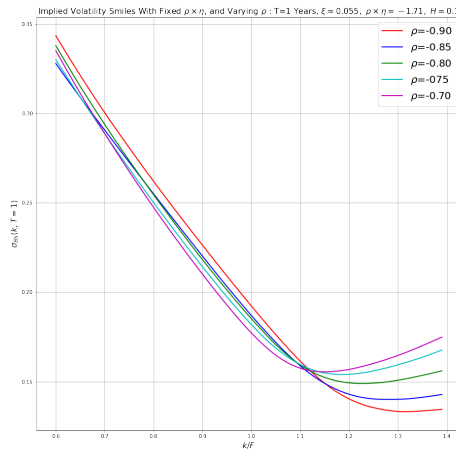
$$\rho_{B_t^H, Z_t} = \frac{\mathbb{E} [B_t^H Z_t]}{\text{Var}(B_t^H) \text{Var}(Z_t)} = \frac{\rho D_H t^{H+\frac{1}{2}}}{t^H t^{\frac{1}{2}}} = \rho D_H.$$



(a) Smiles with varying ρ .
 $T = 2$ years, $H = 0.1, \eta = 1.9$,
 $\xi = 0.055$ (Flat Variance Curve)



(b) Smiles with varying η .
 $T = 2$ years, $H = 0.1, \rho = -0.91$,
 $\xi = 0.055$ (Flat Variance Curve).



(c) Smiles with fixed product $\rho \times \eta$ and varying ρ .
 $T = 1$ year, $\rho \times \eta = -1.71$, $H = 0.1$.

Figure 9.5: Implied Volatility Smiles generated by the Rough Bergomi Model (Hybrid scheme method), with varying values of ρ or η . Note: Smiles for a wider range of maturities are included in Appendix A.1, they have been omitted here as a single maturity is sufficient to show the desired effects.

Therefore we can see that the actual correlation between the processes driving the price-process and variance process is equal to $\rho \times D_H$. If we plot D_H against H we can begin to gain some insight as to the behaviour of $\rho_{B_t^H, Z_t}$, as H varies.

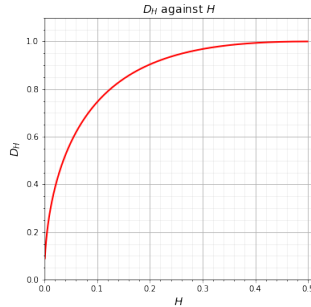


Figure 9.6: Plot of D_H against H

From Figure 9.6 we can see that D_H increases with H . The 'spot-vol' correlation is being scaled by the factor D_H which is dependent on the smoothness of the Riemann-Liouville process. In the case that $H = \frac{1}{2}$, we see that $D_H = 1$, and thus $\rho_{B_t^{0.5}, Z_t} = \rho$. Therefore, in this case we do have that ρ is the 'spot-vol' correlation. On the other hand, as we decrease H below 0.10, we will have a rapid decrease in D_H and subsequently a rapid decrease in the 'spot-vol' correlation $\rho_{B_t^H, Z_t}$.

From this chapter we have gained an idea of the effects of each of the 3 parameters in the Rough Bergomi model. This knowledge will be useful for the purpose of calibrating the Rough Bergomi model, which we shall cover in the next chapter.

Chapter 10

Calibration of the Rough Bergomi Model

Now that we have covered the effects of each parameter within the Rough Bergomi model, we can begin to examine how we will calibrate the model. With proper knowledge of the effect of each parameter, we could in effect make educated guesses for the appropriate value of each parameter. This notion is corroborated by a quote from Bayer et al. stating that 'Rough Bergomi parameters can be guessed in practice' [2, Section 5 Page 16]. Furthermore, Bayer et al. [2, Section 5.2 Page 18] showed, with proper knowledge of the effects of each parameter, that 'guessing' the parameters of the Rough Bergomi model provides a relatively good fit to the implied volatility surfaces of SPX options. Building on this, they proposed certain methods to estimate H directly from the term structure of the ATM forward skew. However, for the purpose of this paper we shall proceed with calibration methods that endeavor to systematically obtain a good fit to the market-implied volatility surface. This should allow us to consistently achieve a good fit to a wide range of volatility surfaces.

10.1 Existing Methods

The existing literature on calibration of the Rough Bergomi presents two main styles of calibration methodologies: Numerical minimisation of a chosen cost function, or direct fitting to a market observable quantity with a closed form solution under the model. We shall briefly explore an example of the latter style of calibration, before proceeding with a description of the main calibration methodologies used in this work, both of which belong to the former style of calibration methodologies.

10.1.1 Variance Swap Calibration

In his 2018 presentation, Gatheral [43], proposed the idea of calibrating the Rough Bergomi through market observable quantities, such as variance swaps, that could also be calculated via a closed form formulae under the Rough Bergomi model. Variance swaps were deemed an appropriate quantity, as they can in principle be observed from an asset's implied volatility surface. Furthermore, based on the expectation decomposition formulae of Alòs [44, Theorem 2.2 Page 406], Gatheral set out the theoretical basis to calibrate the Rough Bergomi, from leverage swaps [43, Slide 60-66]. Gatheral showed that leverage swaps, which are products that can be decomposed into variance and gamma swaps, can, in theory, be computed in any stochastic volatility model, and they can also be estimated from the market-implied volatility surface [43, Slide 66]. While promising, this calibration scheme will not be utilised for the calibration of the Rough Bergomi model in this paper.

For the purpose of this work, we shall proceed with the following two calibration methodologies, a Sequential Least Squares Programming minimisation of a chosen cost function, and a novel Sequential Refined Grids Calibration method. Both of these methodologies seek to minimise a chosen cost function, with the latter of the methods searching for a minimising value on a discrete parameter space, rather than searching for a global minimum. For the purpose of both methods, we shall introduce the setup and choice of our chosen cost function.

10.2 Calibration Setup & Cost function

Our goal is to calibrate the 3 parameters of the Rough Bergomi model, such that the model generated implied volatility surface at inception, is similar to that of the market implied volatility surface. While models such as the local-volatility can fit the market volatility surface perfectly, the Rough Bergomi model may not be able to fit every shape of surface exactly. However, we hope that the Rough Bergomi model can fit a wide range of surfaces better than a traditional stochastic volatility model, while also delivering more realistic long-term dynamics when compared to a local-volatility model.

As we are attempting to fit the market implied-volatility surface, it is an obvious choice to try and minimise the difference between the implied volatilities generated by the Rough Bergomi model, and the market implied volatilities. In order to calibrate in a practical and timely manner, we choose to minimise the difference between market implied volatilities, σ_{Market} , and model generated implied volatilities, σ_{RB} , on a selection of 30 points on the respective surfaces, corresponding to 30 unique strike-maturity pairings.

We choose to minimise the following cost function

$$c(H, \rho, \eta, G) = \sum_{i=1}^{30} (\sigma_{i,\text{Market}} - \sigma_{i,\text{RB}})^2, \quad (10.2.1)$$

where G is our collection of 30 options.

Our choice of G will be determined by which areas of implied volatility surface we are most interested in fitting to. A specification of G that includes only ATM options would naturally lead to a surface that has a closer fit ATM, and conversely if we wished to fit the wings of the implied-volatility surface better, we would choose G to include significantly OTM options. We can represent our choice of options G , on a maturity-strike matrix, with a 1 signalling that the options is within G , and a 0 signalling that option is not included. We will proceed by using the choice of grid G , represented as both a matrix and a table:

$$G_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}, \quad (10.2.2)$$

Maturity	Strike (as % of forward)									
	60%	70%	80%	80%	100%	110%	120%	130%	140%	
1 Month					x					
3 Months				x	x	x				
6 Months			x	x	x	x	x			
1 Year	x	x	x	x	x	x	x	x	x	
2 Years		x	x	x	x	x	x	x		
5 Years			x	x	x	x	x			

Table 10.1: Table representing the choice of options in grid G_1 . Cells in green with an x in them indicate options we will consider in G_1 , for the purpose of calibrating our model

G_1 is a fairly balanced grid of options, placing a higher emphasis on options near the money, but also ignoring extremely OTM or ITM options for short-expiries. G_1 is used for fitting the whole surface, with a larger emphasis on the 'centre' of the surface. With our choice of cost function, equation (10.2.1), we have chosen to weight each option within G equally. If we wished to achieve a more precise fit on certain areas of the grid, we could weight those points more heavily. This can be achieved by multiplying a constant factor w_i to each $(\sigma_{i,\text{Market}} - \sigma_{i,\text{RB}})^2$.

Now for a fixed G , our cost function, $c(H, \rho, \eta, G)$ (equation (10.2.1)) becomes a function of H, ρ and η , and thus we seek the triplet of parameters that minimises $c(H, \rho, \eta)$.

To proceed, we will use the Rough Bergomi model and the grid-pricing method to generate options prices for the specified strikes and maturities in the chosen G that we consider. We at

this point must note that we shall only consider OTM options for the purpose of our calibration. Thus for strikes below 100% of the forward, we price a put option, and for all other strikes we price a call option. This is motivated by a remark from McCrickerd and Pakkanen, stating that "The implied volatilities generated when exclusively considering call or put option estimators are significantly noisier when they are respectively in-the-money" [37, Footnote 1 Page 2]. The variance of the Monte Carlo estimators for option prices in the Rough Bergomi model are already very high, thus we choose exclusively OTM options in an attempt to reduce this variance. For the purpose of calibration we are very concerned with reducing the variance of the price estimate, as these translate to a high variance in the implied volatilities, which subsequently makes it difficult for any minimisation scheme to find the minimum of our cost function. This is in-fact one of the main 'issues' with rough volatility models currently, the variance of the estimators is rather high, and as a result of this many sample paths are required to get the variance down to a level that allows resolution of the minimum of the cost function.

Once we have obtained our selection of option prices, we utilise the Black-Scholes formula (2.2.2) to solve for the implied volatilities. We then can compute our cost function, comparing the model generated implied volatilities observed directly from the market, or from a synthesised arbitrage free volatility surface. As we are now able to obtain a cost function value for a given triplet (H, ρ, η) , we can proceed to selecting an appropriate minimisation algorithm.

10.3 High Variance Issues

Here we briefly expand upon the issue of the high variance of the traditional Monte Carlo price estimates under the Rough Bergomi model. As alluded to above, the Rough Bergomi model, due to the inclusion of a fBm within the variance process, produces Monte Carlo price estimates which have a high variance. Let us consider the case of simulating a Rough Bergomi model with a fixed parameter triplet, and utilising this model to price a call option of fixed strike k and fixed maturity T . We utilise m paths in the simulation of the model, and receive a price $C(T, k)$ as an output. Traditionally in other volatility models, $m = 200,000$ would have been a sufficient number of paths such that on subsequent repricing of the same call option, we would have a low variance in the Monte Carlo price estimate. Unfortunately this is not the case in the Rough Bergomi model, as the variance of the paths is so high. We performed an analysis of this to obtain an empirical estimate for the standard deviation of repeated Monte Carlo pricings of a call option with maturity of $T = 5$ years, and a strike of 100% of the forward.

H	σ of $C(T = 5, k = 100\%)$ (as % of mean)
0.1	0.34
0.15	0.32
0.2	0.27

Table 10.2: Table of standard deviation of 20 Monte Carlo pricings of a call option with maturity of 5 Years and strike of 100% of the forward, for varying H . Each pricing utilised 200,000 paths, and the standard deviation is expressed as a % of the mean price for each batch of 20 pricings. Parameters used : $\rho = -0.7, \eta = 2$. SPX data from 30th May 2022 was utilised for these simulations.

We can see from Table 10.2 that the standard deviation is rather high, especially for $H = 0.1$. We also see that the standard deviation increases as the fBm within the Rough Bergomi model gets rougher. It is far from ideal that, even after utilising 200,000 simulation paths, 20 subsequent price estimates for a single option have a standard deviation of as much as 0.34% of the mean of the 20 prices. This will prove to be an issue for the purpose of calibration, and for the purpose of merely generating consistent derivatives prices. The high variance of the price estimates will in turn translate to a high variance in the implied volatilities obtained from each options price, and this variance will in turn affect the evaluation of the cost function used for calibration. In order to investigate the magnitude of this effect, we fixed a parameter triplet for the Rough Bergomi model and calculated our cost function (10.2.1) over our calibration grid G_1 .

H	σ of $c(H, \rho, \eta, G_1)$ (as % of mean)
0.1	9.53
0.15	2.82
0.2	1.23
0.3	0.54

Table 10.3: Table of standard deviation of 20 evaluations of cost function $c(H, \rho, \eta, G_1)$, for varying H . Each pricing within a cost function evaluation utilised 200,000 paths, and the standard deviation is expressed as a % of the mean cost function value for each batch of 20 evaluations. The SPX volatility surface on the 30th May 2022 was utilised for the market volatilities within the cost function, and appropriate market data was also taken from this index on the specified date. Parameters used : $\rho = -0.9, \eta = 1.9$.

Now from Table 10.3 we can see that this variance is going to be a large issue when performing calibration. We note here that $H = 0.1$ is close to the minimising value of H for these evaluations of the cost function on the SPX surface on the 30th May 2022, as we shall discover later in Chapter 11. We see that as H decreases, the standard deviation of the cost function evaluations reaches as high as 9.5% of the mean cost function evaluation. This results in an extremely high noise to signal ratio for any minimisation scheme, which will make it hard for the scheme to find the true minimum. We cannot at this time differentiate as to whether the standard deviation, as a % of the mean cost function evaluation, is increasing as the paths of the fBm are becoming increasingly rough, or because we are sufficiently close to the minimising value. Unfortunately, as we saw from our estimates for H in Chapter 5, we will likely find that H is close to 0.1 for many indices that we wish to simulate, and thus this high standard deviation in the cost function evaluations is likely something that we will encounter during the calibration of any index we considered.

We will see, in Chapter 11, that this variance in the cost function will be one of the main issues we face. Nevertheless, we still for now proceed by attempting to minimise our chosen cost function. We now turn to the choice of an appropriate minimisation scheme.

10.4 Minimisation Algorithms

In order to select an appropriate minimisation algorithm, it would serve us well to understand the behaviour of our chosen cost function $c(H, \rho, \eta)$. The cost function c returns a value for each choice of triplet (H, ρ, η) , and thus we can consider the four-dimensional surface, with co-ordinates given by $(H, \rho, \eta, c(H, \rho, \eta))$. Unfortunately, it is rather hard to visualise such a surface in four-dimensions. Therefore, we shall resort to fixing one parameter value, for example $H = \bar{H}$, and plotting the surface with co-ordinates given by $(\rho, \eta, c(\bar{H}, \rho, \eta))$. In plotting these surfaces, we are in particular looking for the presence of large local-minima, which may prevent certain gradient-based minimisation schemes from finding the global-minimum.

The following surfaces were generated by simulating the Rough Bergomi model using the Hybrid scheme, with market data taken from the 30th May 2022 for the SPX. Moreover, grid G_1 was utilised in the specification of the cost function c .

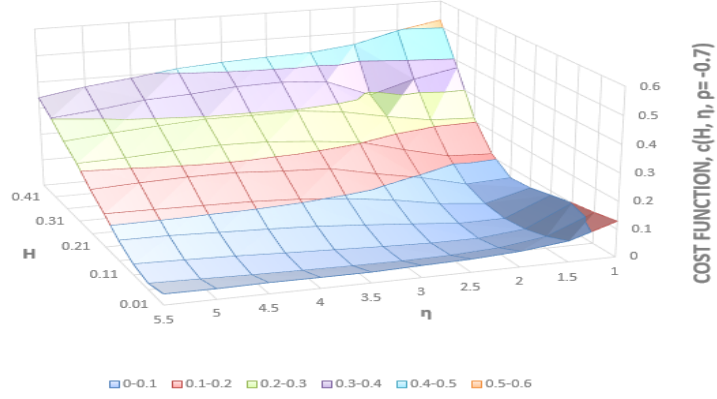
Remark 10.4.1. The cost function grids used to generate these surfaces are in Appendix A.2.

From Figure 10.1 we can observe that the surfaces plotted are relatively smooth, suitably convex, and do not contain any noticeable local minima. While these surfaces are not the actual four-dimensional cost function surface we are interested in, and plotting one specific set of three-dimensional surfaces is certainly not an exhaustive investigation, we can still use these surfaces plotted as weak evidence to suggest that a gradient-based minimisation algorithm may be feasible.

10.5 SLSQP Minimisation

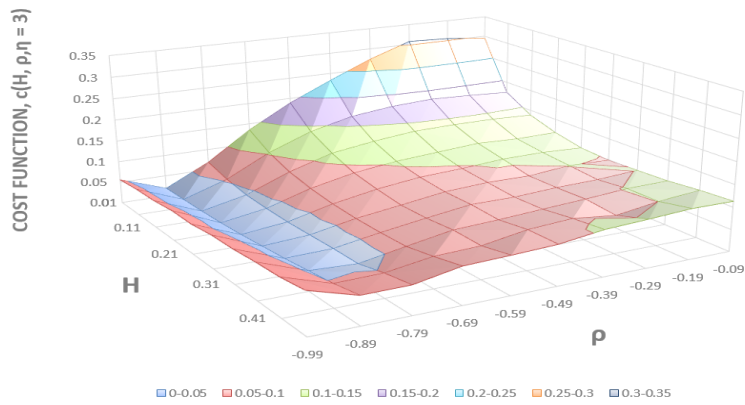
We, shall proceed with using the Sequential Least Squares Programming (SLSQP) scheme for the minimisation of our cost function. The SLSQP algorithm, proposed by Dieter Kraft [45], minimises a function of several variables, and can be used with combinations of bounds, equality and inequality constraints. For SLSQP we require that the cost function is at least twice continuously differentiable, and the surfaces from Figure 10.1 suggest that this is true. SLSQP uses a method

SPX - Cost Function Plot: H against η , with $\rho=-0.7$



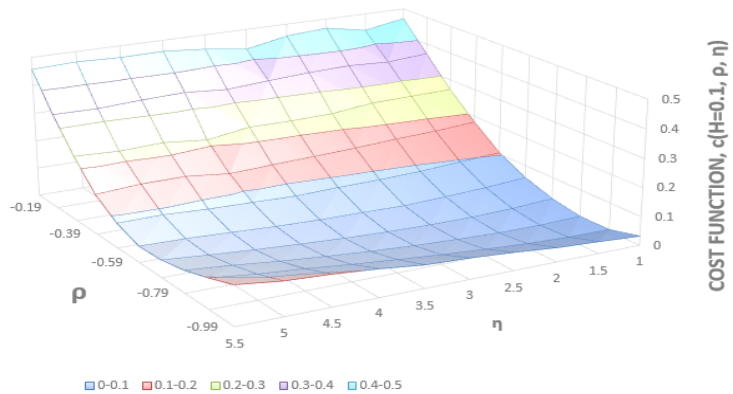
(a) Cost Function surface for varying values of H and η , with fixed $\rho = -0.7$

SPX - Cost Function Plot: H against ρ , with $\eta=3$



(b) Cost Function surface for varying values of H and ρ , with fixed $\eta = 3$

SPX - Cost Function Plot: ρ against η , with $H=0.1$



(c) Cost Function surface for varying values of ρ and η , with fixed $H = 0.1$

Figure 10.1: Cost function surfaces for fit to the SPX implied volatility surface on 30th May 2022, using grid G_1 , for varying parameter values. The Hybrid scheme method was used to simulate the Rough Bergomi model which was utilised for these cost function evaluations.

known as Sequential Quadratic Programming (SQP), which is not a single algorithm itself, but rather a method from which numerous minimisation algorithms have evolved. For a detailed explanation of SQP and SLSQP see [46]. SQP can be viewed as a natural extension of Newton-like methods [46, Section 1 Page 2], and as such can be expected to share some of the same characteristics of Newton-like methods; such as rapid convergence when iterations are close to the minima, but potentially erratic behaviour far from the minima.

Now that we have chosen our minimisation scheme, we must define our domain of calibration. For the purpose of the parameters of the Rough Bergomi model for equity derivatives, we shall define the calibration domain \mathcal{D} as

$$\begin{aligned}\mathcal{D} &= [H_{\min}, H_{\max}] \times [\rho_{\min}, \rho_{\max}] \times [\eta_{\min}, \eta_{\max}] \\ &= [0.01, 0.49] \times [-0.99, -0.01] \times [0.5, 5.5]\end{aligned}\tag{10.5.1}$$

With this in mind, we describe the calibration process as follows:

- Algorithm 10.5.1** (SLSQP Calibration Method). 1 Simulate 30 options specified by choice of grid G , using both grid-pricing and the Hybrid scheme method.
- 2 Solve for the implied volatility of each option.
- 3 Use SLSQP to minimise the cost function $c(H, \rho, \eta)$ to obtain the minimising triplet (H^*, ρ^*, η^*) .

This calibration method was used to calibrate the Rough Bergomi model to the SPX implied volatility surface, with data from the 30th May 2022. Before we proceed to calibration results, we shall introduce an additional calibration method.

10.6 Sequential Refined Grids Calibration Method

We now present a novel calibration method, which to the best of our knowledge is not published in the existing literature. We call this method the ‘Sequential Refined Grids’ (SRFG) method, and present it as general framework for a class of methods that differ depending on the number of cycles and grid refinements considered.

The methodology consists of sequential calibration of the model parameters on increasingly refined discrete parameter grids. An important principle of this method is that the minimum of each grid is included as a point within the next grid considered, this ensures that the minimising value in a grid is larger than that of all subsequent grids. As a result of this, we can guarantee that the minimising value on the final discrete grid is the minimum of all discrete parameter combinations over all grids considered.

Algorithm 10.6.1 (Sequential Refined Grids Calibration). This method can be applied with numerous choices of cost functions, however, we shall proceed with the cost function defined by equation (10.2.1). We utilise the domain of calibration \mathcal{D} as specified by (10.5.1).

- 1 **First Calibration Cycle:** In the first calibration cycle, an optimisation is done in (ρ, η) , with H set to an appropriate value. The initial value of H could be chosen from an analysis of the roughness of the realised variance of the asset in question (see Chapter 5) or from previous calibrated values of H .

1(a) Fix the value of $H = H_0$ to a suitable value.

1(b) Construct a coarse grid, $\overline{\mathcal{D}}_{\rho, \eta}$, of size $N_1 \times M_1$, defined via the discretisation of the intervals $[\rho_{\min}, \rho_{\max}], [\eta_{\min}, \eta_{\max}]$:

$$\overline{\mathcal{D}}_{\rho, \eta} := \left\{ \rho_{\min} + \left(\frac{i}{N_1 - 1} \times \Delta\rho \right) \right\}_{i=0}^{N_1-1} \times \left\{ \eta_{\min} + \left(\frac{j}{M_1 - 1} \times \Delta\eta \right) \right\}_{j=0}^{M_1-1}$$

with $\Delta\rho := (\rho_{\max} - \rho_{\min}) = 0.98$, $\Delta\eta := (\eta_{\max} - \eta_{\min}) = 5$ For convenience we denote, $l := \lfloor \frac{1}{N_1-1} \times \Delta\rho \rfloor$ and $k := \lfloor \frac{1}{M_1-1} \times \Delta\eta \rfloor$.

- 1(c) For each point $(\rho_i, \eta_j) \in \overline{\mathcal{D}}_{\rho, \eta}$ simulate 30 options prices specified by options grid G , using the Rough Bergomi model with parameter choices (H_0, ρ_i, η_j) , using the Hybrid scheme method. Solve for the implied volatility of each option, and subsequently calculate the value of the cost function $c_{i,j}(H_0, \rho_i, \eta_j)$ and store its value.

- 1(d) Once all points (ρ_i, η_j) have been considered, find the minimum value of $c_{i,j}(H_0, \rho_i, \eta_j)$ and denote

$$(\bar{\rho}, \bar{\eta}) := \arg \min_{\rho_i, \eta_j \in \bar{\mathcal{D}}_{\rho, \eta}} c(H_0, \rho_i, \eta_j,)$$

- 1(e) Let N_2, M_2 both be odd integers. Then, we construct a refined grid, $\underline{\mathcal{D}}_{\rho, \eta}$ of size $N_2 \times M_2$, defined via the discretisation of intervals of length $2l$ and $2k$ respectively, centred around $\bar{\rho}, \bar{\eta}$:

$$\underline{\mathcal{D}}_{\rho, \eta} := \left\{ \bar{\rho} - l + \left(\frac{i \times 2l}{N_2 - 1} \right) \right\}_{i=0}^{N_2-1} \times \left\{ \bar{\eta} - k + \left(\frac{j \times 2k}{M_2 - 1} \right) \right\}_{j=0}^{M_2-1}.$$

By taking N_2, M_2 as odd integers, the point $(\bar{\rho}, \bar{\eta})$ is included in $\underline{\mathcal{D}}_{\rho, \eta}$ by construction. This ensures that

$$\arg \min_{\rho_i, \eta_j \in \underline{\mathcal{D}}_{\rho, \eta}} c(H_0, \rho_i, \eta_j,) \leq \arg \min_{\rho_i, \eta_j \in \bar{\mathcal{D}}_{\rho, \eta}} c(H_0, \rho_i, \eta_j,).$$

If we have, for a point $(\rho_i, \eta_j) \in \underline{\mathcal{D}}_{\rho, \eta}$, that $(\rho_i, \eta_j) \notin [\rho_{\min}, \rho_{\max}] \times [\eta_{\min}, \eta_{\max}]$, we simply ignore the point and do not include it in our grid $\underline{\mathcal{D}}_{\rho, \eta}$.

- 1(f) Repeat steps 1(c)-1(d), and denote

$$(\underline{\rho}, \underline{\eta}) := \arg \min_{\rho_i, \eta_j \in \underline{\mathcal{D}}_{\rho, \eta}} c(H_0, \rho_i, \eta_j,).$$

- 2 **Second Calibration Cycle:** We now optimise over (H, ρ) with η fixed to $\eta = \underline{\eta}$ obtained from 1(f). We let $\Delta\rho = \min(|\underline{\rho} - \rho_{\min}|, |\underline{\rho} - \rho_{\max}|)$ and $\Delta H = \min(|H_0 - H_{\min}|, |H_0 - H_{\max}|)$.

- 2(a) Let N'_1, M'_1 be odd integers. Then we construct a coarse grid $\bar{\mathcal{D}}^1_{H, \rho}$, of size $N'_1 \times M'_1$, defined via the discretisation of the intervals, $[H_0 - \frac{1}{2}\Delta H, H_0 + \frac{1}{2}\Delta H], [\underline{\rho} - \frac{1}{2}\Delta\rho, \underline{\rho} + \frac{1}{2}\Delta\rho]$:

$$\bar{\mathcal{D}}^1_{H, \rho} := \left\{ H_0 - \frac{1}{2}\Delta H + \left(\frac{i}{N'_1 - 1} \times \Delta H \right) \right\}_{i=0}^{N'_1-1} \times \left\{ \underline{\rho} - \frac{1}{2}\Delta\rho + \left(\frac{j}{M'_1 - 1} \times \Delta\rho \right) \right\}_{j=0}^{M'_1-1}.$$

This is a grid, centred around $(H_0, \underline{\rho})$. We have by construction chosen to include the point $(H_0, \underline{\rho})$, so that the minimising triplet from 1(f) $(H_0, \underline{\rho}, \underline{\eta})$ is a discrete point within $\bar{\mathcal{D}}^1_{H, \rho}$, when we extend to the triplet by including the fixed value $\underline{\eta}$. Therefore, we ensure that

$$\arg \min_{H_i, \rho_j \in \bar{\mathcal{D}}^1_{H, \rho}} c(H_i, \rho_j, \underline{\eta},) \leq \arg \min_{\rho_i, \eta_j \in \underline{\mathcal{D}}_{\rho, \eta}} c(H_0, \rho_i, \eta_j,).$$

For convenience we denote, $l' := |\frac{1}{N'_1-1} \times \Delta H|$ and $k' := |\frac{1}{M'_1-1} \times \Delta\rho|$.

- 2(b) For each point $(H, \rho_j) \in \bar{\mathcal{D}}^1_{H, \rho}$ simulate 30 options prices specified by options grid G , under the Rough Bergomi model with parameter choices $(H_i, \rho_j, \underline{\eta})$, using the Hybrid scheme method. Solve for the implied volatility of each option, and subsequently calculate the value of the cost function $c_{i,j}(H_i, \rho_j, \underline{\eta})$ and store its value.

- 2(c) Once all points (H_i, ρ_j) have been considered, find the minimum value of $c_{i,j}(H_i, \rho_j, \underline{\eta})$ and denote

$$(\bar{H}, \bar{\rho}') := \arg \min_{H_i, \rho_j \in \bar{\mathcal{D}}^1_{H, \rho}} c(H_i, \rho_j, \underline{\eta},).$$

- 2(d) Let N'_2, M'_2 both be odd integers. Then, we construct a refined grid, $\underline{\mathcal{D}}'_{H, \rho}$ of size $N'_2 \times M'_2$, defined via the discretisation of intervals of length $2 \times l'$ and $2 \times k'$ respectively, centred around $\bar{H}, \bar{\rho}'$:

$$\underline{\mathcal{D}}'_{H, \rho} := \left\{ \bar{H} - l' + \left(\frac{i \times 2l'}{N'_2 - 1} \right) \right\}_{i=0}^{N'_2-1} \times \left\{ \bar{\rho}' - k' + \left(\frac{j \times 2k'}{M'_2 - 1} \right) \right\}_{j=0}^{M'_2-1}.$$

By taking N'_2, M'_2 as odd integers, the point $(\bar{H}, \bar{\rho}')$ is included in $\underline{\mathcal{D}}'_{H,\rho}$ by construction. This ensures that

$$\arg \min_{H_i, \rho_j \in \underline{\mathcal{D}}'_{H,\rho}} c(H_i, \rho_j, \underline{\eta}) \leq \arg \min_{H_i, \rho_j \in \overline{\mathcal{D}}'_{H,\rho}} c(H_i, \rho_j, \underline{\eta}).$$

If we have, for a point $(H, \rho_j) \in \underline{\mathcal{D}}'_{H,\rho}$, that $(H_i, \rho_j) \notin [H_{\min}, H_{\max}] \times [\rho_{\min}, \rho_{\max}]$, we simply ignore the point and do not include it in our grid $\underline{\mathcal{D}}'_{H,\rho}$.

2(e) Repeat steps 2(b)-2(c), and denote

$$(\underline{H}, \underline{\rho}') := \arg \min_{H_i, \rho_j \in \underline{\mathcal{D}}'_{H,\rho}} c(H_i, \rho_j, \underline{\eta}).$$

3 We then have a triplet of parameters $(\underline{H}, \underline{\rho}', \underline{\eta})$, which we take as our final calibrated parameters.

By the construction of our grids, we have ensured that the triplet $(\underline{H}, \underline{\rho}', \underline{\eta})$ will be the minimising set of parameters for the cost function, for all of the triplets we have considered over the associated 4 grids. This ensures that we do not necessarily need to store all of the cost function values for each triplet as we move from one grid to another, as we can be certain the final value will be the minimising value for the discrete set of values considered.

Remark 10.6.2. This scheme can be adapted to include more rounds of calibration, or a larger number of increasingly refined grids per cycle. For example, a third calibration cycle could be performed, optimising over a grid of (H, η) while fixing $\rho = \underline{\rho}'$. Various cycles and grids can be added, though one must take care to include the minimising triplet from a previous grid, into the next grid, if they want to ensure the final triplet is the minimum over the discrete values considered. We have chosen to use 2 calibration cycles, with 2 grids in each cycle, as we deemed this to provide a reliable level of calibration accuracy in terms of minimising the cost function. Further numerical investigations could be done into the computation time and calibration accuracy, of various combinations of the number of calibration cycles and the number of increasingly refined grids per cycle.

While this method does not guarantee we find a minimum of the cost function over our parameter space \mathcal{D} , it does lead to a minimising value on the final discrete refined grid, and this value is the minimising value of all discrete parameter values considered. The method should not be viewed as a proper minimisation scheme, as we are in effect just using brute force to try and find a minimising parameter triplet on a discrete parameter cube. However, the method does still seem to have some merits, specifically in the context of calibrating the Rough Bergomi model, as traditional minimisation schemes do not seem to have been very effective in both our testing and the current literature. The benefit of this methodology is that, if we fix the size of each grid considered to consist of N (N being an odd integer) points, then we are only required to simulate the Rough Bergomi model a total of $4 \times N^2$, to obtain our final optimising triplet. A notable benefit of this method is that it is easily parallelised, with each core being able to compute the cost function of a point on each grid in parallel, before recombining the values to find the minimum. This is in contrast to the SLSQP method, which requires information from the previous cost function analysis before a subsequent analysis can be conducted. Given sufficient cores, we can compute each grid in approximately the same amount of time as one cost function evaluation, and thus we can evaluate k grids in a similar amount of time as k iterations in the SLSQP method. This becomes even more powerful when we consider that as we are utilising grid pricing, where each cost function evaluation over our grid G of options only requires one set of m paths and thus is computed in approximately the same amount of time as the computation of one option up to a final maturity time of T , if we were not using the grid-pricing method. Therefore, with k grids being utilised, we can calibrate the entire model in approximately the time required to generate k options prices for a maturity of T if we were not using grid-pricing. A disadvantage of this method is that we evaluate the cost function a large number of times far from the discrete minimum of each grid. However, the decreased computation time from parallelisation of this method does seem to alleviate this issue, at least from a computation time point of view. Further development into this model could try to reduce the number of 'wasted' cost function evaluations, perhaps by integrating algorithms which restrict cost function evaluations in certain areas of the grid if the minimum

value is deemed unlikely to be in such an area. Figure 10.2 shows a visual representation of the method, for the case where we use 2 calibration cycles consisting of 2 grids each, and is shown to aid the readers understanding of the method.

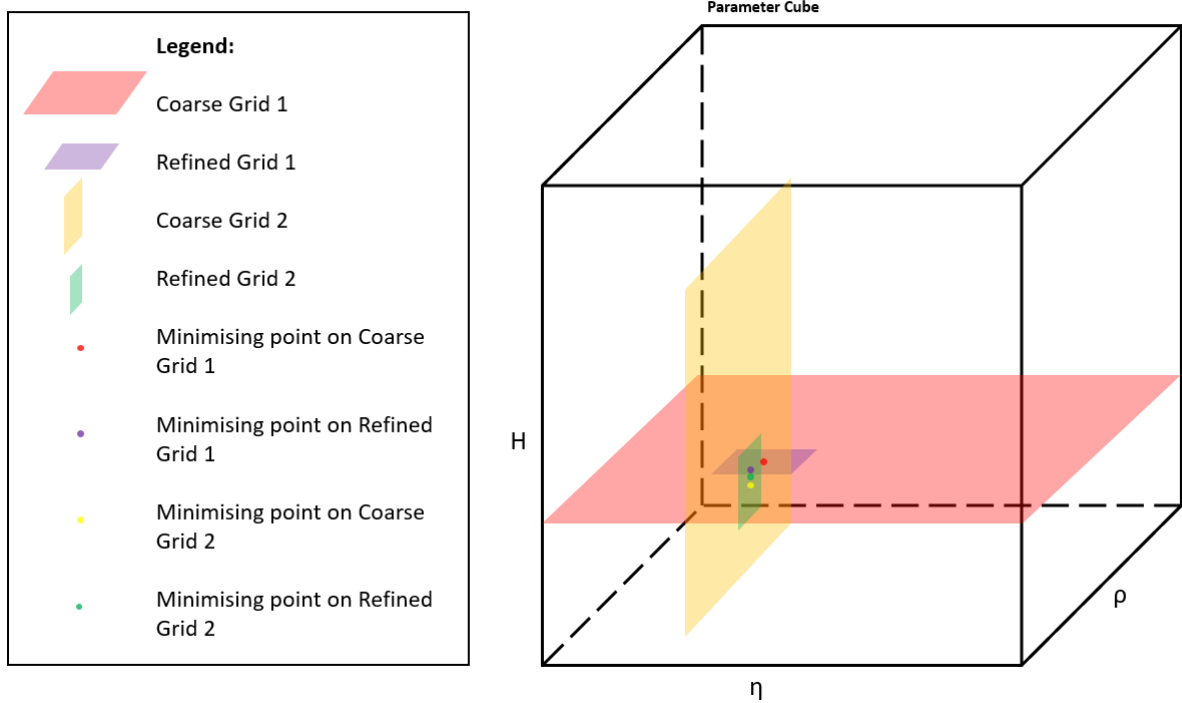


Figure 10.2: Visual representation of the SRFG Calibration Method of H, ρ, η on the parameter domain \mathcal{D} represented by a parameter cube. Here we have utilised 2 calibration cycles, with 2 grids per cycle.

We utilised both the SLSQP and SRFG method to calibrate the Rough Bergomi model to the SPX volatility surface on the 30th May 2022, with the grid G_1 . We shall now proceed to explore the results of these calibrated models.

Chapter 11

Calibration results

We will now show results from the calibration of the Rough Bergomi model, utilising both the SLSQP calibration method, and the SRFG method. In both methods, we utilise Grid-pricing and the Hybrid scheme to generate the option prices, and the cost function chosen is specified by (10.2.1). Furthermore, both calibration methods were performed on the SPX volatility surface on the 30th May 2022, and used market data from the same date.

11.1 SLSQP Calibration

The following results were obtained through SLSQP calibration, on the stike-maturity grid, G_1 , specified by (10.2.2). For the purpose of calibration, 200,000 paths were used to price each grid of options.

H	0.0996
ρ	-0.848
η	1.991
Final Cost Function Value, $c(H, \rho, \eta, G_1)$	0.00112
Mean Squared Error	0.0000373

Table 11.1: Calibration results from SLSQP calibration of Rough Bergomi Model to SPX Implied volatility surface on the 30th May 2022, utilising options grid G_1 .

Table 11.1 displays the values of our calibrated parameters, the corresponding final cost function value, and the corresponding mean squared error of the model generated implied volatilities compared to the market implied volatilities. Unfortunately, as mentioned before, the high variance of the prices and subsequent implied volatilities generated by the Rough Bergomi model, makes it hard for the SLSQP scheme to obtain the minimising parameter triplet. The SLSQP scheme rapidly approached a neighbourhood of our minimising parameter triplet, but then the scheme found it difficult to find a precise minimising point, as the variance of the cost function value is high even within a narrow range of parameters. To remedy this, we would like to utilise more paths per evaluation of the cost function, so that we would see a decrease in the variance, and the SLSQP scheme would be able to find a minimising point more quickly and accurately. Unfortunately, we could not utilise more than 200,000 paths while simulating up to a maturity of 5 years, due to memory constraints. As we lower the maximum maturity, we will be able to utilise more paths, and thus we should see that the cost function can be minimised more quickly and to a lower value.

Now that we have obtained calibrated parameters for the Rough Bergomi model, we can use the model to price a set of options, and compare the implied volatilities obtained from the model, with the market implied volatilities. We note that, we will now generate model-delivered implied volatilities for points on the strike \times maturity plane that were not necessarily considered in our calibration grid.

Maturity	Strike	Absolute Error
1 Month	70%	0.00421
	80%	0.01837
	90%	0.01037
	100%	0.01662
	110%	0.00834
	120%	0.00251
	130%	0.01659
	3 Months	60%
70%		0.00232
80%		0.00380
90%		0.00025
100%		0.00504
110%		0.00482
120%		0.00395
130%		0.00275
6 Months	140%	0.00828
	60%	0.00551
	70%	0.00354
	80%	0.00591
	90%	0.00196
	100%	0.00172
	110%	0.00033
	120%	0.01075
1 Year	130%	0.01412
	140%	0.01061
	60%	0.00356
	70%	0.00636
	80%	0.00589
	90%	0.00337
	100%	0.00164
	110%	0.00402
2 Years	120%	0.00847
	130%	0.00883
	140%	0.00500
	60%	0.00807
	70%	0.00777
	80%	0.00632
	90%	0.00405
	100%	0.00205
5 Years	110%	0.00180
	120%	0.00238
	130%	0.00155
	140%	0.00129
	60%	0.00205
	70%	0.00092
	80%	0.00399
	90%	0.00679
1 Year	100%	0.00831
	110%	0.01011
	120%	0.01074
	130%	0.01047
	140%	0.01031

Table 11.2: Table of absolute error of the SLSQP calibrated Rough Bergomi model delivered implied volatility estimates, compared to the market implied volatility values. Strikes are expressed as % of the forward price. In this simulation, the Rough Bergomi utilised the calibrated parameters, $H = 0.0996$, $\rho = -0.847$, $\eta = 1.991$, and 200,000 simulation paths were used. Market data and the implied volatility surface were taken from the SPX index on the 30th May 2022.

Maturity	Mean Absolute Error	Standard Deviation of Absolute Error
1 Month	0.0110	0.00636
3 Months	0.00544	0.00512
6 Months	0.00605	0.00478
1 Year	0.00524	0.00239
2 Years	0.00392	0.00276
5 Years	0.00708	0.00386

Table 11.3: Table of mean absolute error and standard deviation of absolute error for each maturity considered in Table 11.2.

Strike (As % of Forward)	Mean Absolute Error	Standard Deviation of Absolute Error
60%	0.00739	0.00633
70%	0.00418	0.00356
80%	0.00738	0.00549
90%	0.00447	0.00362
100%	0.00590	0.00586
110%	0.00494	0.00374
120%	0.00647	0.00398
130%	0.00905	0.00601
140%	0.00710	0.00394

Table 11.4: Table of mean absolute error and standard deviation of absolute error for each strike considered in Table 11.2.

Table 11.2 shows the absolute error of the model delivered implied volatility compared to the market implied volatility, for a set of options. The mean of the absolute errors is 0.00628 and the standard deviation is 0.00462. Moreover, Table 11.3 and Table 11.4 show the mean and standard deviations of the absolute errors for each maturity and strike considered, respectively. This allows us to observe how the distribution of absolute error varies by both strike and maturity. From Table 11.1, recalling our grid of options G_1 which we used for our calibration, we can see that the implied volatilities for options that were not present within the calibration grid have higher absolute errors. This is to be expected. Our calibration scheme sought to minimise the error over these specific points on the implied volatility surfaces, thus it makes intuitive sense that upon re-simulation with calibrated parameters, that these points will have a lower absolute error. Furthermore, from Table 11.3, we can see that the 1 month maturity has the highest mean absolute error, this corroborates our previous statement, as the 1 month maturity had the least number of options present within our calibration grid G_1 . Unfortunately, not much can be deduced from Table 11.4, with no clear trend being present. We likely can attribute this to the high variance of the option price and subsequent implied volatility estimate of our current version of the Rough Bergomi model. Ultimately, most of the trends we see in Tables 11.2-11.4 are due to our choice of calibration grid G , and should not differ between the SLSQP method and the SRFG method. While we again expect to see larger errors for options not considered in the calibration grid, it is more interesting to look at the difference in magnitude of the absolute errors between the two methods, corresponding to a difference in goodness of fit. We shall explore a comparison of the two calibration methodologies later.

Given the data presented in Tables 11.6-11.4, the plotting of smiles would provide us with little additional information about the goodness of fit of the model. Nevertheless, we shall plot a selection of smiles for the maturities we considered, to allow us to visualise the fit of the model delivered implied volatilities to the market implied volatility surface.

Figure 11.1 shows the model delivered and market smiles for varying maturities. By plotting these smiles we can easily visualise the data from Table 11.2, and see the different shapes of smiles present for each maturity. We can again see that the model delivered implied volatilities for options considered within our grid G_1 are closer to that of the market implied volatilities, corroborating what we learnt from Tables 11.6-11.4.

As discussed in Chapter 3, one of our main goals with the Rough Bergomi model was to better fit the term structure of the ATM forward volatility skew. Therefore, it is only natural that we should plot term structures of the skew generated by our calibrated model, and see how well they fit the market term structure. Furthermore, it is also useful to plot the term structure of the ATM volatility delivered by our calibrated model, as this is not only a quantity closely related to the skew, but also an important quantity for pricing exotic options such as cliquet options (see 2.3.1).

SLSQP Calibration on G_1 : Comparison of Model Delivered and Market Implied Volatility Smiles for Varying Maturities

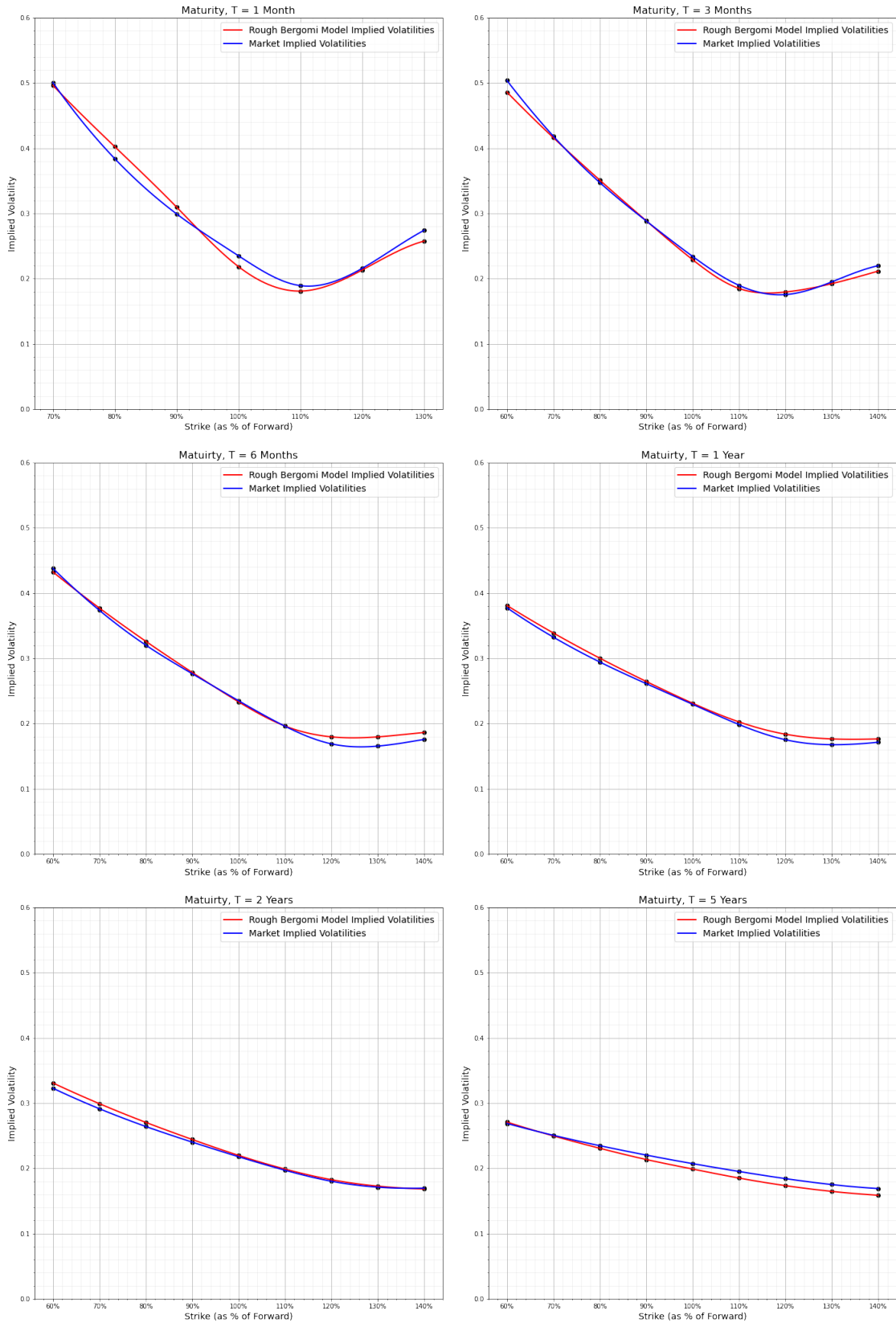
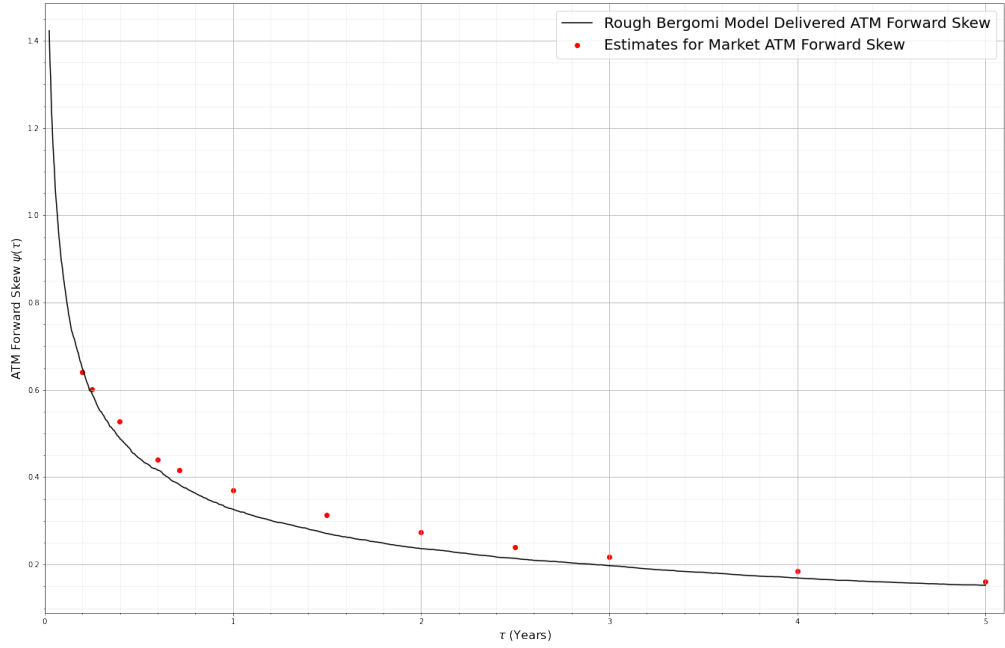


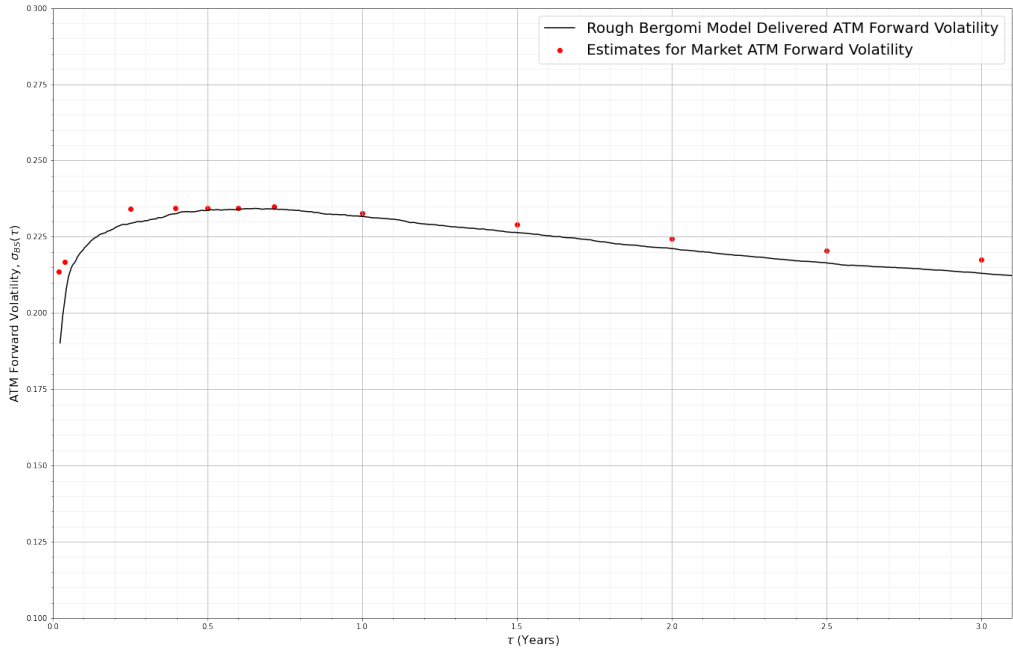
Figure 11.1: Plots of volatility smiles for varying maturities, for both market implied volatilities, and SLSQP calibrated Rough Bergomi model delivered implied volatilities. The Rough Bergomi model was simulated with the following parameters obtained from the SLSQP calibration to the SPX volatility surface on the 30th May 2022: $H = 0.0996$, $\rho = -0.848$, $\eta = 1.991$.

Term Structure of SLSQP Calibrated Rough Bergomi Model Delivered ATM Forward Skew With Estimates for Market ATM Forward Skew



(a) Term structure of SLSQP calibrated Rough Bergomi model delivered ATM forward skew with estimates for market ATM forward skew.

Term Structure of SLSQP Calibrated Rough Bergomi Model Delivered ATM Forward Volatility With Estimates for Market ATM Forward Volatility



(b) Term structure of SLSQP calibrated Rough Bergomi model delivered ATM forward volatility with estimates for market ATM forward volatility.

Figure 11.2: Term structures of ATM forward volatility and skew from SLSQP calibrated Rough Bergomi Model, with $H = 0.0996$, $\rho = -0.848$, $\eta = 1.991$. Market data from the SPX volatility surface on the 30th May 2022 was used to plot market estimates for ATM forward volatility and skew, and for calibration of the Rough Bergomi model.

Figure 11.2 displays the calibrated model-generated term structure of the ATM forward skew and the ATM forward volatility. From Figure 11.2(a) we see that the calibrated model fits the market skew well initially in the very short term, however, it then has a larger error for maturities of up to 3 years, after which we then see the error decrease. On the other hand, from Figure 11.2(b), we see an initially poor fit for very short maturities, followed by an improvement in fit for maturities between 5 months and 1.5 years, and then a decrease in the quality of fit.

Overall, we can see that the Rough Bergomi model calibrated by the SLSQP method is able to fit the market-implied volatility surface well, fitting empirical smiles and skews decently, with just 3 parameters. The fit could be better, but unfortunately, the SLSQP method had difficulty finding the true minimising parameter triplet due to the large amount of noise present in each cost function evaluation, coming from the variance of the Monte Carlo estimator for the Rough Bergomi model. Moreover, even with knowledge of the true minimising parameters of our cost function, due to the high levels of variance present in the Rough Bergomi model, we cannot guarantee that upon re-simulation we will have an optimal fit. We will expand more upon this in section 11.3.

11.2 Sequential Refined Grids Calibration

We now turn to calibration results from the SRFG calibration method. We utilised the 4 grid variant of the SRFG method, with $N_1, M_1 = 10$, and $N_2, M_2, N'_1, M'_1, N'_2, M'_2 = 11$, on domain \mathcal{D} (see (10.5.1)), and $H_0 = 0.1$. Furthermore, 200,000 paths were used to simulate each grid of options, and all market data was taken from the SPX volatility surface on the 30th May 2022.

H	0.0946
ρ	-0.896
η	1.944
Final Cost Function Value, $c(H, \rho, \eta, G_1)$	0.00104
Mean Squared Error	0.0000345

Table 11.5: Calibration results from SRFG calibration of Rough Bergomi Model to SPX Implied volatility surface on the 30th May 2022, utilising options grid G_1 .

From Table 11.5, we see that the SRFG method actually resulted in a lower final cost function value than the SLSQP method, corresponding to a lower mean squared error over the grid G_1 . Unfortunately, as we will see, this lower mean squared error on calibration does not always translate to a better fit upon re-simulation. This is due to the high variance of the Rough Bergomi prices, which even with fixed parameters, produce varying price and implied volatility estimates upon each re-simulation (see 10.3). Therefore, even if we have two parameter triplets $p_1 = (H_1, \rho_1, \eta_1)$ and $p_2 = (H_2, \rho_2, \eta_2)$ with $c(p_1) < c(p_2)$, we do not necessarily get that a Rough Bergomi model with parameters p_1 will fit the market better than p_2 upon re-simulation. It seems that due to the high variance we are currently presented with, we are only able to calibrate the model to a certain level of fit, where the noise generated from the variance of the price estimations prevents minimisation schemes from finding the true minimum. With this in mind, the SRFG method seems more effective given the current circumstances than the SLSQP method. Both methods are able to calibrate to a similar level of accuracy, which is limited by the variance of the price estimation, which is in turn limited by memory constraints. However, the SRFG method with 4 grids can be parallelised such that it can run much faster than the SLSQP method, and therefore, we can obtain a similar level of fit in significantly less time.

Upon re-simulation using the calibrated parameters from the SRFG method, we obtain Table 11.6, displaying the absolute errors on the options considered. The mean of the absolute errors is 0.00655 and the standard deviation is 0.00488. We note that, compared to the absolute errors when we re-simulated the SLSQP method, the SRFG method produced higher mean absolute errors and a higher standard deviation of absolute error, even though the cost function was lower. This, as suggested above, supports the idea that given the high variance of the price estimates from the Rough Bergomi model, a lower calibration cost function value does not necessarily guarantee a better fit upon re-simulation, if we only consider one re-simulation attempt.

We once again display analogous tables and figures to those of the SLSQP method. The general trends of these tables and figures seem to be consistent with those of the SLSQP method, reinforcing the notion that the trends we see with regard to the fit at each strike and maturity

Maturity	Strike	Absolute Error	
1 Month	70%	0.00033	
	80%	0.02139	
	90%	0.01320	
	100%	0.01665	
	110%	0.01262	
	120%	0.00564	
	130%	0.01774	
	3 Months	60%	0.01700
70%		0.00046	
80%		0.00563	
90%		0.00221	
100%		0.00406	
110%		0.00784	
120%		0.00299	
130%		0.00951	
6 Months	60%	0.00653	
	70%	0.00383	
	80%	0.00719	
	90%	0.00355	
	100%	0.00042	
	110%	0.00066	
	120%	0.00556	
	130%	0.00679	
1 Year	60%	0.00373	
	70%	0.00710	
	80%	0.00716	
	90%	0.00479	
	100%	0.00303	
	110%	0.00427	
	120%	0.00594	
	130%	0.00325	
	140%	0.00306	
	2 Years	60%	0.00982
		70%	0.00989
		80%	0.00867
		90%	0.00649
		100%	0.00449
110%		0.00342	
120%		0.00229	
130%		0.00098	
5 Years	60%	0.00612	
	70%	0.00514	
	80%	0.00245	
	90%	0.00038	
	100%	0.00305	
	110%	0.00600	
	110%	0.00798	
	120%	0.00926	
	130%	0.01020	
	140%	0.01149	

Table 11.6: Table of absolute error of the SRFG calibrated Rough Bergomi model delivered implied volatility estimates, compared to the market implied volatility values. Strikes are expressed as % of the forward price. In this simulation, the Rough Bergomi utilised the calibrated parameters, $H = 0.0996$, $\rho = -0.848$, $\eta = 1.991$, and 200,000 simulation paths were used. Market data and the implied volatility surface were taken for the SPX index on the 30th May 2022.

are more heavily affected by the choice of our grid G . We again see that options considered in our grid have a lower absolute error. Overall we again see that the Rough Bergomi model with SRFG calibrated parameters fits the market implied volatility surface well, fitting empirical smiles and skews. The quality of fit of the SRFG model does not seem to be notably different from the SLSQP calibrated model, and the difference between the mean error of both models is likely due to the variance of the price estimators, rather than any difference in the calibration schemes.

Maturity	Mean Absolute Error	Standard Deviation of Absolute Error
1 Month	0.0125	0.00730
3 Months	0.00727	0.00587
6 Months	0.00412	0.00257
1 Year	0.00470	0.00166
2 Years	0.00580	0.00325
5 Years	0.00622	0.00380

Table 11.7: Table of Mean absolute error and standard deviation of absolute error for each maturity considered in Table 11.6.

Strike (As % of Forward)	Mean Absolute Error	Standard Deviation of Absolute Error
60%	0.00844	0.00529
70%	0.00401	0.00381
80%	0.00840	0.00699
90%	0.00555	0.00403
100%	0.00578	0.00564
110%	0.00613	0.00422
120%	0.00528	0.00248
130%	0.00808	0.00592
140%	0.00779	0.00570

Table 11.8: Table of Mean absolute error and standard deviation of absolute error for each strike considered in Table 11.6.

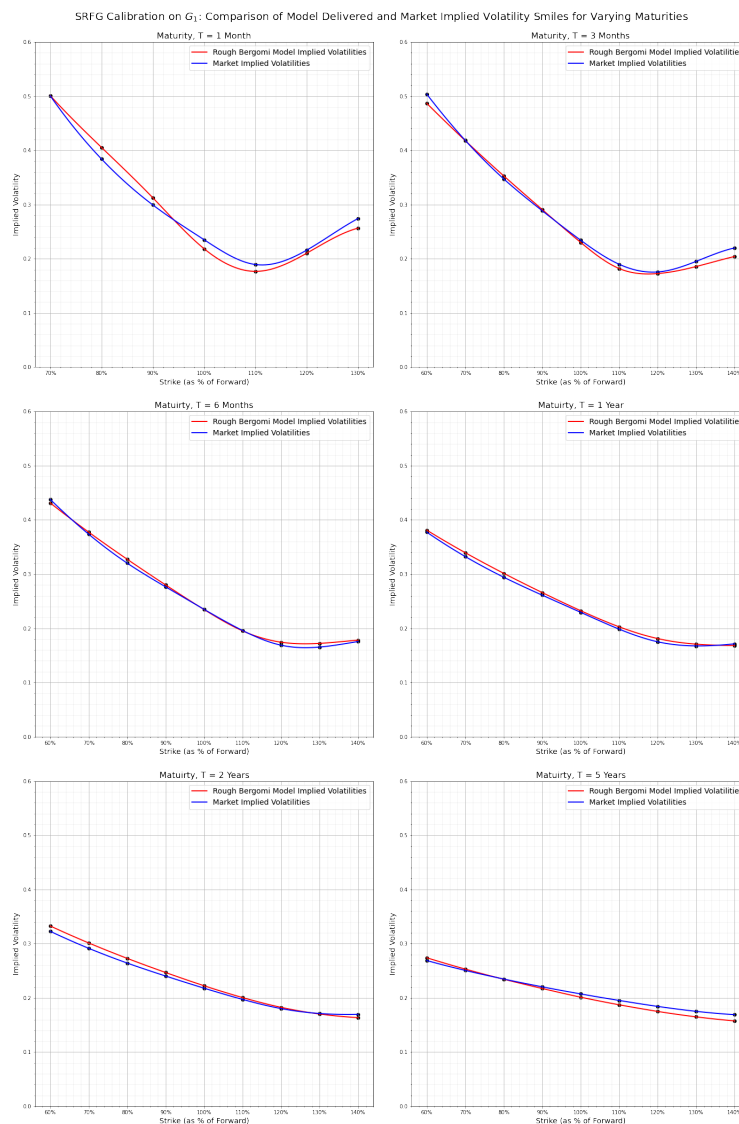
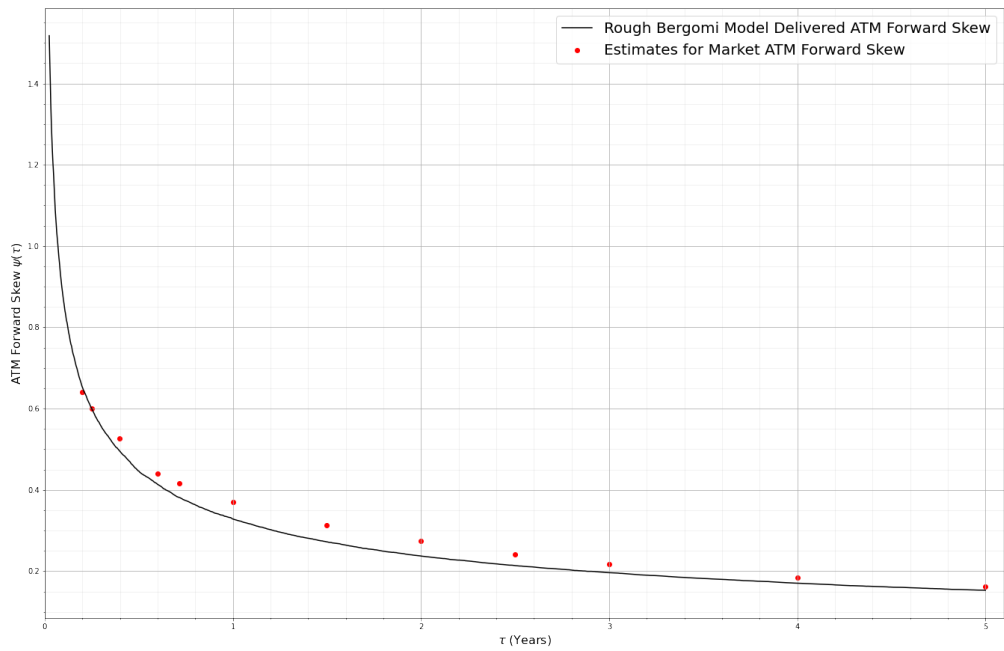


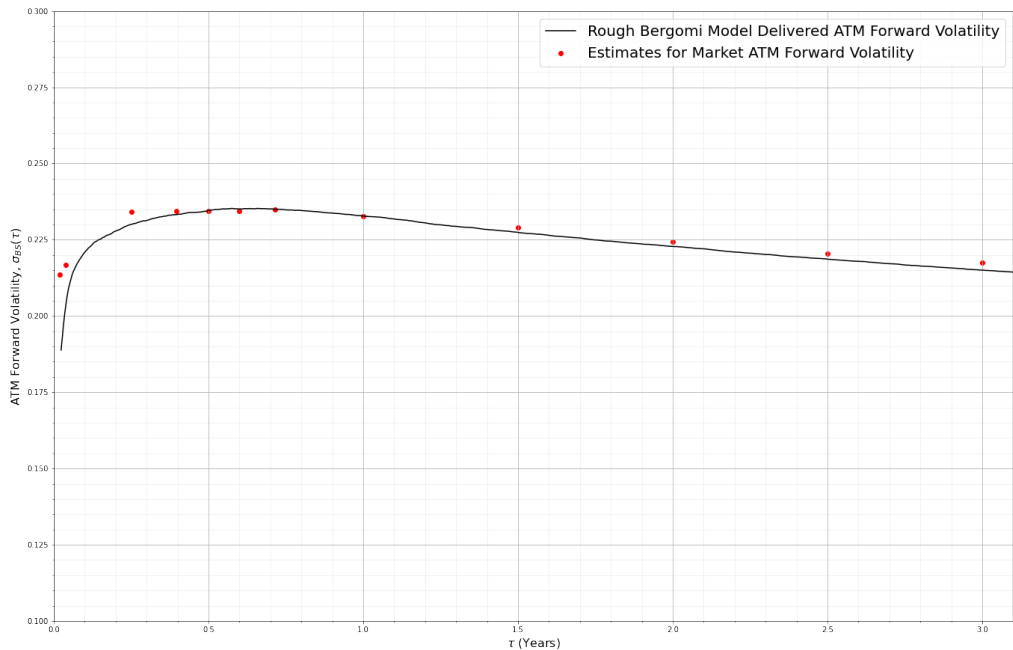
Figure 11.3: Plots of volatility smiles for varying maturities, for both market implied volatilities, and SRFG calibrated Rough Bergomi model delivered implied volatilities. The Rough Bergomi model was simulated with the following parameters obtained from the SRFG calibration to the SPX volatility surface on the 30th May 2022: $H = 0.0946$, $\rho = -0.896$, $\eta = 1.944$.

Term Structure of SRFG Calibrated Rough Bergomi Model Delivered ATM Forward Skew With Estimates for Market ATM Forward Skew



(a) Term Structure of SRFG Calibrated Rough Bergomi Model Delivered ATM Forward Skew With Estimates for Market ATM Forward Skew.

Term Structure of SRFG Calibrated Rough Bergomi Model Delivered ATM Forward Volatility With Estimates for Market ATM Forward Volatility



(b) Term Structure of SRFG Calibrated Rough Bergomi Model Delivered ATM Forward Volatility With Estimates for Market ATM Forward Volatility.

Figure 11.4: Term structures of ATM Forward volatility and skew from SRFG calibrated Rough Bergomi Model, with Rough Bergomi Model, with $H = 0.0946$, $\rho = -0.896$, $\eta = 1.944$. Market data from the SPX volatility surface on the 30th May 2022 was used to plot market estimates for ATM forward volatility and skew, and the market data was used for calibration.

11.3 Comparison of SLSQP and SRFG methods

Analysing the results from Section 11.1 and 11.2 we see that both schemes manage to minimise the cost function down to a similar level. The SLSQP method resulted in a final mean squared error of 3.73×10^{-5} of over the 30 options considered, whereas the SRFG scheme resulted in a final mean squared error of 3.45×10^{-5} . While both schemes manage to get this mean squared error down to a low level, there is still a fundamental issue with the calibration of the Rough Bergomi model through the minimisation of our chosen cost function. As alluded to earlier, our main issue is that, given the constraint of a maximum of 200,000 paths, the Rough Bergomi model produces Monte Carlo price estimates that still have a high level of variance. This high variance of the price estimates results in a high variance within the implied volatility estimates, and consequently within our cost function evaluations. Therefore, both schemes find it hard to detect the true minimising parameters of the cost function, due to the presence of this large amount of noise. The SLSQP scheme performs repeated evaluations of the cost function with small variations in the parameters. The variance in the cost function value at each evaluation is high enough such that once the cost function is minimised down to a magnitude of 10^{-3} , it has difficulty determining which direction to move the parameter estimate towards. Similarly, the SRFG method evaluates each parameter combination on its grids only once, and thus the minimum value for each grid can easily change upon reevaluation of the cost function over the grid. The minimising triplet obtained from each grid, especially for the more refined grid, is thus not necessarily the actual minimising triplet over that grid. Essentially, both schemes are able to successfully find a neighbourhood of the true minimising parameter triplet, but once they are here the noise is too great to allow them to find a more precise estimate. Furthermore, as displayed in Section 10.3, upon re-simulation of the model, we have a high variance in the price estimates, which can lead to a large variance of the fit of the model to the market implied volatility surface. Even if the parameters chosen were the true minimising parameters of our cost function, a simulation of the Rough Bergomi model could likely provide a worse fit than another simulation with other parameters.

Ultimately the issues regarding fit are currently to do with this variance problem, and this should be the first thing we endeavour to address. The obvious solution is to utilise more simulation paths, and if memory constraints were not present then we would have utilised this approach. Variance reduction techniques would also be extremely useful here, however, as discussed we chose not to utilise the 'turbocharging' technique due to the bias in the estimators. The addition of an appropriate variance reduction technique that is both tractable and without large bias, would greatly enhance the feasibility of calibration of the Rough Bergomi method in this way. With effective variance reduction techniques and more simulation paths, we could reduce the variance of the price estimates sufficiently such that the signal-to-noise ratio of our cost function evaluations would improve significantly. This would allow the SLSQP method would be able to find more accurate minimising parameters for the Rough Bergomi model. Furthermore, if we managed to reduce the variance down to a suitable level, the SLSQP methodology would likely become much more effective than the 4 grid variant of the SRFG method. With the variance reduced, we will still encounter the following issue with the SRFG method. That issue being, that if we perturb the initial guess for H_0 in the SRFG method, then the calibration will produce different parameter values, as all of the grids have been shifted. The SRFG method's parameter outputs are linked to the chosen grids, whereas the SLSQP method should produce similar parameter outputs provided we choose an appropriate starting value for our parameters. Furthermore, even after reducing the variance of the Rough Bergomi model, the SRFG method would likely only take us to a neighbourhood of the minimising parameters, due to the discrete nature of the grids it utilises, whereas the SLSQP method should be able to output the actual minimising parameters. Overall, due to the currently present high variance of the model, the SRFG method is able to produce a similar level of calibration in much less time. However, as improvements are made to the variance reduction of the model, we should see that the SLSQP method provides a much better quality of fit.

On the other hand, it is likely also worth exploring other calibration methodologies in more detail. Additional calibration methodologies, such as fitting to market quantities such as variance swaps (see 10.1.1), may prove to be more effective than minimising our chosen cost function. We thus await more research on the efficacy of such methods.

Despite the issues with the current calibration methodologies, the Rough Bergomi model still shows large signs of promise. We were able to achieve a decent fit to the market-implied volatility

surface, with only 3 parameters. The model is able to not only fit smiles from different maturities but also able to reproduce realistic term structures of ATM forward skew, an issue that has long plagued other stochastic volatility models.

Conclusion

We have displayed a thorough overview of the Rough Bergomi model, from the motivation behind its creation to the calibration of the model to the SPX volatility surface. The Rough Bergomi model has the power and flexibility to fit numerous shapes of volatility surfaces, with just three parameters. Furthermore, the Rough Bergomi model can produce realistic term structures of ATM forward skew and produces realistic consistent dynamics for the model delivered implied volatility surface. The model is thus able to combine the benefits of both local volatility models and stochastic volatility models, without the need for a model add-on or adjustment.

We displayed two calibration methodologies for the Rough Bergomi model, an SLSQP based calibration scheme, and a novel SRFG scheme. Due to the high variance of the Rough Bergomi model's price estimates, both schemes were able to calibrate to a similar level, both providing a good fit to the implied volatility surface in the form of smiles and term structures of skews. With advances in variance reduction techniques, we expect faster and more accurate calibration of the model to become possible.

Further research into novel variance reduction techniques or additional calibration schemes will help to allow for the Rough Bergomi model to find its place within production environments in industry. Furthermore, the multi-factor Rough Bergomi model (see [47]) is a particular area of interest for further research. This multi-factor model combines the benefits of a more realistic power-law kernel, obtained through the use of fBms, and the presence of additional factors for the purpose of flexibility. The flexibility gained from the inclusion of multiple fBm factors allows the model to fit the implied volatility surface of equity indices and of the Volatility Index (VIX), which the Rough Bergomi model currently struggles with [2, Chapter 9 Page 36]. Unfortunately, the multi-factor model is even more difficult to calibrate, due to the large number of parameters present. Nonetheless, some interesting developments have been made in the calibration of the multi-factor Rough Bergomi model, most notably the work of Jacquier et al. [47], which presented explicit formulae for the ATM volatility and skew, in terms of the parameters of the model, thus allowing for efficient calibration of the model.

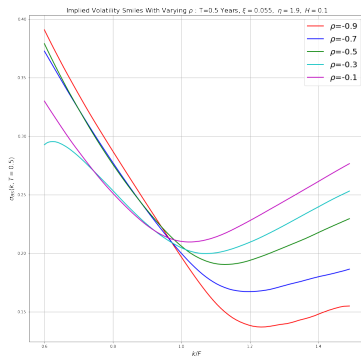
Although efficient calibration of the Rough Bergomi and multi-factor Rough Bergomi model still remains partially unsolved, we remain overtly optimistic about future research. Rough volatility models remain a thriving area of academic interest, and with further research, they will without doubt become increasingly important models for the purposes of derivatives pricing, particularly for exotic derivatives.

Appendix A

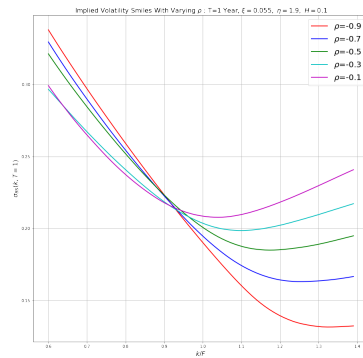
Additional Figures

A.1 Effect of Rough Bergomi Parameters

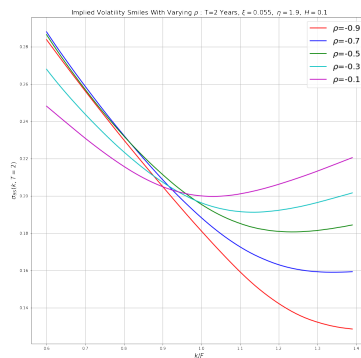
Here we display some additional figures generated for the investigation into the effect of the parameters of the Rough Bergomi model, performed in Chapter 9.



(a) Smiles with varying ρ .
 $T = 6$ months, $H = 0.1, \eta = 1.9$,
 $\xi = 0.055$ (Flat Variance Curve).

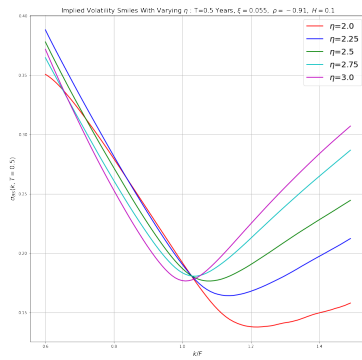


(b) Smiles with varying ρ .
 $T = 1$ year, $H = 0.1, \eta = 1.9$,
 $\xi = 0.055$ (Flat Variance Curve)

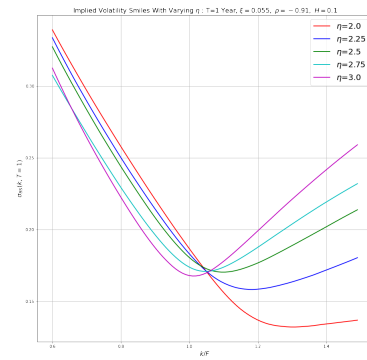


(c) Smiles with varying ρ .
 $T = 2$ years, $H = 0.1, \eta = 1.9$,
 $\xi = 0.055$ (Flat Variance Curve)

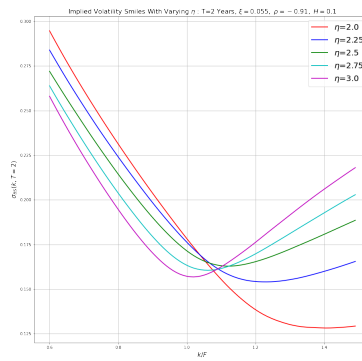
Figure A.1: Implied Volatility Smiles generated by the Rough Bergomi Model (Hybrid scheme Method), with varying values of ρ .



(a) Smiles with varying η .
 $T = 6$ months, $H = 0.1, \rho = -0.91$,
 $\xi = 0.055$ (Flat Variance Curve).



(b) Smiles with varying η .
 $T = 1$ year, $H = 0.1, \rho = -0.91$,
 $\xi = 0.055$ (Flat Variance Curve)

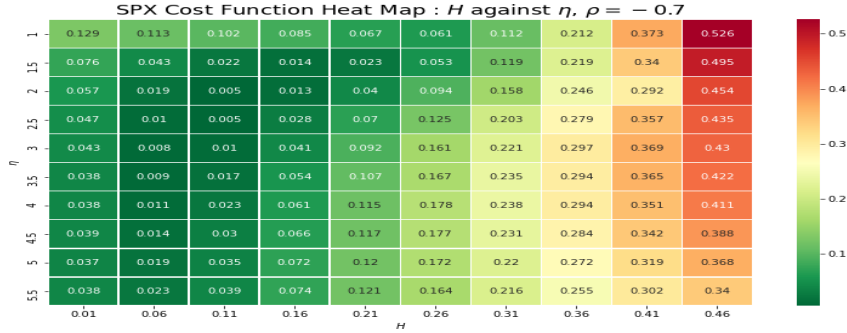


(c) Smiles with varying η .
 $T = 2$ years, $H = 0.1, \rho = -0.91$,
 $\xi = 0.055$ (Flat Variance Curve)

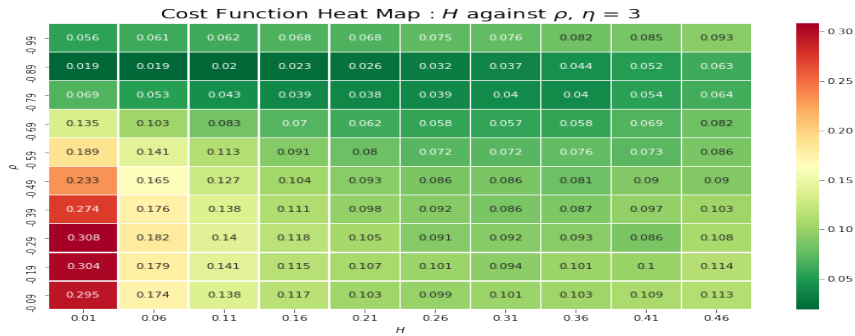
Figure A.2: Implied Volatility Smiles generated by the Rough Bergomi Model (Hybrid scheme Method), with varying values of η .

A.2 Cost Function Heat Maps

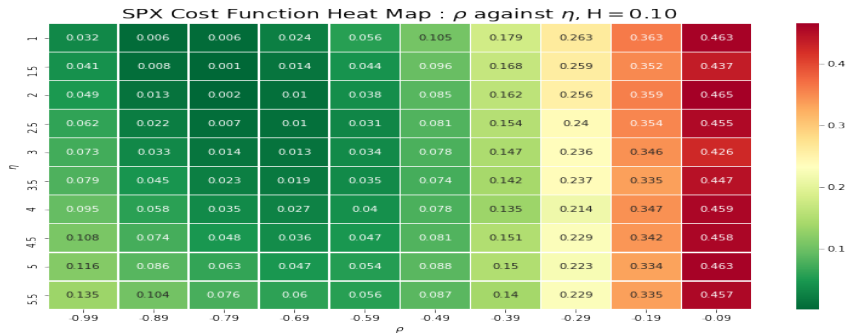
Here we display heat maps of the values of the cost function on discrete parameter grids, these values were used to generate the cost function surfaces (see 10.1) in Chapter 10.



(a) Heat map of Cost Function values for varying values of H and η , with fixed $\rho = -0.7$



(b) Heat map of Cost Function values for varying values of H and ρ , with fixed $\eta = 3$



(c) Heat map of Cost Function values for varying values of ρ and η , with fixed $H = 0.1$

Figure A.3: Heat maps of cost function values for fit to the SPX implied volatility surface on 30th May 2022, using grid G_1 , for varying parameter values.

Bibliography

- [1] Jim Gatheral, Thibault Jaisson, and Mathieu Rosenbaum. Volatility is rough. *Quantitative finance*, 18(6):933–949, 2018.
- [2] Christian Bayer, Peter Friz, and Jim Gatheral. Pricing under rough volatility. *Quantitative Finance*, 16(6):887–904, 2016.
- [3] Georgiy Shevchenko. Fractional brownian motion in a nutshell. *arXiv preprint arXiv:1406.1956*, 2014.
- [4] Fabienne Comte and Eric Renault. Long memory in continuous-time stochastic volatility models. *Mathematical finance*, 8(4):291–323, 1998.
- [5] Fischer Black and Myron Scholes. The pricing of options and corporate liabilities. *Journal of political economy*, 81(3):637–654, 1973.
- [6] Henri Berestycki, Jérôme Busca, and Igor Florent. Computing the implied volatility in stochastic volatility models. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 57(10):1352–1373, 2004.
- [7] Lorenzo Bergomi. *Stochastic volatility modeling*. CRC press, 2015.
- [8] Steven L. Heston. A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options. *The Review of Financial Studies*, 6(2):327–343, 04 1993.
- [9] Milan Mrázek and Jan Pospíšil. Calibration and simulation of heston model. *Open Mathematics*, 15(1):679–704, 2017.
- [10] Hansjörg Albrecher, Philipp Mayer, Wim Schoutens, and Jurgen Tistaert. The little heston trap. *Wilmott*, (1):83–92, 2007.
- [11] Masaaki Fukasawa. Asymptotic analysis for stochastic volatility: martingale expansion. *Finance and Stochastics*, 15:635–654, 2011.
- [12] Guy Jumarie. Schrödinger equation for quantum fractal space–time of order n via the complex-valued fractional brownian motion. *International Journal of Modern Physics A*, 16(31):5061–5084, 2001.
- [13] Ilkka Norros. On the use of fractional brownian motion in the theory of connectionless networks. *IEEE Journal on selected areas in communications*, 13(6):953–962, 1995.
- [14] Jean Picard. Representation formulae for the fractional brownian motion. *Séminaire de probabilités XLIII*, pages 3–70, 2011.
- [15] Yuliya Mishura and Iuliia S Mishura. *Stochastic calculus for fractional Brownian motion and related processes*, volume 1929. Springer Science & Business Media, 2008.
- [16] Céline Jost. Integral transformations of volterra gaussian processes. 2007.
- [17] René L Schilling. *Measures, integrals and martingales*. Cambridge University Press, 2017.
- [18] Christopher C Bernido, Maria Victoria Carpio-Bernido, Martin Grothaus, Tobias Kuna, Maria João Oliveira, José Luís Da Silva, et al. *Stochastic and infinite dimensional analysis*. Springer, 2016.

- [19] Ioannis Karatzas and Steven Shreve. *Brownian motion and stochastic calculus*, volume 113. Springer Science & Business Media, 1991.
- [20] *NIST Digital Library of Mathematical Functions*. <http://dlmf.nist.gov/>, Release 1.1.4 of 2022-01-15. F. W. J. Olver, A. B. Olde Daalhuis, D. W. Lozier, B. I. Schneider, R. F. Boisvert, C. W. Clark, B. R. Miller, B. V. Saunders, H. S. Cohl, and M. A. McClain, eds.
- [21] B Horvath, A Jacquier, and A Muguruza. Functional central limit theorems for rough volatility (2017). *arXiv preprint arXiv:1711.03078*.
- [22] Benoit B Mandelbrot and John W Van Ness. Fractional brownian motions, fractional noises and applications. *SIAM review*, 10(4):422–437, 1968.
- [23] Georgii Moiseevich Molchan and Yu I Golosov. Gaussian stationary processes with asymptotically a power spectrum. In *Doklady Akademii Nauk*, volume 184, pages 546–549. Russian Academy of Sciences, 1969.
- [24] SC Lim. Fractional brownian motion and multifractional brownian motion of riemann-liouville type. *Journal of Physics A: Mathematical and General*, 34(7):1301, 2001.
- [25] Mathieu Rosenbaum. First order p-variations and besov spaces. *Statistics & Probability Letters*, 79(1):55–62, 2009.
- [26] Mathieu Rosenbaum. A new microstructure noise index. *Quantitative Finance*, 11(6):883–899, 2011.
- [27] Jean Jacod. Asymptotic properties of realized power variations and related functionals of semimartingales. *Stochastic processes and their applications*, 118(4):517–559, 2008.
- [28] Ole E Barndorff-Nielsen and Neil Shephard. Econometric analysis of realized volatility and its use in estimating stochastic volatility models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 64(2):253–280, 2002.
- [29] Asger Lunde Neil Shephard Heber, Gerd and Kevin Sheppard. Oxford-man institute’s realized library, 2009.
- [30] Lorenzo Bergomi. Smile dynamics ii. *Available at SSRN 1493302*, 2009.
- [31] Lorenzo Bergomi. Smile dynamics iii. *Available at SSRN 1493308*, 2008.
- [32] Jan Matas and Jan Pospíšil. On simulation of rough volterra stochastic volatility models. *arXiv preprint arXiv:2108.01999*, 2021.
- [33] Haw-ren Fang and Dianne P O’leary. Modified cholesky algorithms: a catalog with new approaches. *Mathematical programming*, 115:319–349, 2008.
- [34] Mikkel Bennedsen, Asger Lunde, and Mikko S Pakkanen. Hybrid scheme for brownian semistationary processes. *Finance and Stochastics*, 21:931–965, 2017.
- [35] Stéphane Mallat. *A wavelet tour of signal processing*. Elsevier, 1999.
- [36] Monroe David Donsker. *An invariance principle for certain probability limit theorems*. 1951.
- [37] Ryan McCrickerd and Mikko S Pakkanen. Turbocharging monte carlo pricing for the rough bergomi model. *Quantitative Finance*, 18(11):1877–1886, 2018.
- [38] Yingzi Zhu and Jin E Zhang. Variance term structure and vix futures pricing. *International Journal of Theoretical and Applied Finance*, 10(01):111–127, 2007.
- [39] Sidi Mohamed Ould Aly. Forward variance dynamics: Bergomi’s model revisited. *Applied Mathematical Finance*, 21(1):84–107, 2014.
- [40] Jim Gatheral and Antoine Jacquier. Arbitrage-free svi volatility surfaces. *Quantitative Finance*, 14(1):59–71, 2014.

- [41] Martin Forde and Hongzhong Zhang. Asymptotics for rough stochastic volatility models. *SIAM Journal on Financial Mathematics*, 8(1):114–145, 2017.
- [42] Omar El Euch, Masaaki Fukasawa, Jim Gatheral, and Mathieu Rosenbaum. Short-term at-the-money asymptotics under stochastic volatility models. *SIAM Journal on Financial Mathematics*, 10(2):491–511, 2019.
- [43] Jim Gatheral, Masaaki Fukasawa, Thibault Jaisson, and Mathieu Rosenbaum. Rough volatility: An overview. *Global Derivatives*, 2018.
- [44] Elisa Alòs. A decomposition formula for option prices in the heston model and applications to option pricing approximation. *Finance and Stochastics*, 16:403–422, 2012.
- [45] Dieter Kraft. A software package for sequential quadratic programming. *Forschungsbericht-Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt*, 1988.
- [46] Paul T Boggs and Jon W Tolle. Sequential quadratic programming. *Acta numerica*, 4:1–51, 1995.
- [47] Antoine Jacquier, Aitor Muguruza, and Alexandre Pannier. Rough multifactor volatility for spx and vix options. *arXiv preprint arXiv:2112.14310*, 2021.