



UNIVERSITÀ
DI PAVIA

Department of Economics and Management

Master of Science in Finance

Ultra-fast Calibration of Affine and Quadratic Stochastic Volatility Models

Supervisor:

Prof. Riccardo Brignone

Candidate:

Federico Crivellari

Student ID: 551612

Academic Year 2025–2026



UNIVERSITÀ
DI PAVIA

Dipartimento di Scienze Economiche e Aziendali

Corso di Laurea Magistrale in Finance

Calibrazione Ultra-Rapida dei Modelli a Volatilità Stocastica Affini e Quadratici

Relatore:

Prof. Riccardo Brignone

Candidato:

Federico Crivellari

Matricola: 551612

Anno Accademico 2025–2026

Abstract

The thesis presents an efficient methodology for calibrating the Heston and Ornstein–Uhlenbeck stochastic volatility models to quoted European option prices based on the analytical gradient of the characteristic function. The methodology is tested on both real and simulated data. The contributions of this thesis can be summarized as follows. First, I identify an identifiability issue related to the mean–reversion rate and long–term variance parameters of the Heston model. Second, to overcome this issue, I propose a reparameterization of the Heston model that ensures greater stability of the optimal solution, reduces the number of calibration failures on real–world data, and achieves calibration speeds almost ten times faster than those of the original algorithm. Third, I show that the same identifiability issue does not arise in the Ornstein–Uhlenbeck model. The result is that the model is easier to calibrate and the proposed algorithm consistently finds the global optimum on real data, even without reparameterization. Finally, building on these findings, I propose a model selection procedure that ensures successful calibration on every day in the sample.

Abstract–Italian

La tesi presenta una metodologia efficiente per calibrare i modelli a volatilità stocastica di Heston e di Ornstein–Uhlenbeck sui prezzi delle opzioni europee quotate, attraverso il gradiente analitico della funzione caratteristica. Tale metodologia è stata testata sia su dati simulati sia su dati reali. I principali contributi di questa tesi possono essere così riassunti. In primo luogo, ho individuato un problema di identificabilità tra il tasso di mean–reversion e la varianza di lungo periodo del modello di Heston. In secondo luogo, per superare tale problema, ho proposto una riparametrizzazione del modello, che garantisce una maggiore stabilità della soluzione di ottimo, riduce il numero di fallimenti della calibrazione su dati reali e consente di raggiungere tempi di calibrazione quasi dieci volte inferiori rispetto a quelli ottenuti dall’algoritmo originale. In terzo luogo, ho mostrato che lo stesso problema di identificabilità non si verifica nel modello di Ornstein–Uhlenbeck. Ne consegue che, esso risulta più semplice da calibrare e che l’algoritmo proposto individua sistematicamente l’ottimo globale sui dati reali, anche in assenza di riparametrizzazione. Infine, sulla base di questi risultati, ho proposto una procedura di model selection che garantisce il successo della calibrazione per ogni giornata contenuta nel campione.

Contents

1	Introduction	2
1.1	Previous Work and Recognized Difficulties	3
1.2	Proposed Methodology	4
2	The Models	7
2.1	Basic Notions	7
2.1.1	Black and Scholes Model	7
2.1.2	Stochastic Volatility Models	12
2.1.3	Option Pricing via Fourier Transform	14
2.2	Heston Stochastic Volatility Model	17
2.2.1	Model Dynamics	17
2.2.2	Characteristic Function	21
2.3	Ornstein–Uhlenbeck driven Stochastic Volatility Model	27
2.3.1	Model Dynamics	27
2.3.2	Characteristic Function	30
2.4	Comparison between OU and Heston Model	32
3	Calibration Methodology	34
3.1	Calibration Problem	34
3.2	Branch Switching of the CPF	36
3.3	Analytical Gradient	39
3.4	Integration Scheme	41
3.5	Optimization Algorithms	42
3.5.1	Problem Implementations	42
3.5.2	Optimization Method	43
3.6	Condition Number and Standard Deviations	44
4	Reparameterized Algorithms	48
4.1	Constrained Problem	48
4.2	Proposed Reparameterizations	49
4.3	Summary	52

5	Numerical Results	53
5.1	Simulated Data	53
5.1.1	Random Starting Point	54
5.1.2	Validation Procedure	56
5.1.3	Performance	56
5.2	Real-world Data	59
5.2.1	Data	60
5.2.2	A Case Study	61
5.2.3	Performance	65
5.2.4	Condition Number and Parameter Time Series	66
5.2.5	Identifiability Problem in the Heston Model	69
5.2.6	The η parameter	72
6	Model Selection	74
6.1	Summary of Discoveries	74
6.2	Condition Number Approach	75
7	Conclusions	78
	References	80
	Appendix: MATLAB Codes	82

Notations

$$\mathbb{F} := \{\mathcal{F}_t\}_{t \geq 0} : \mathcal{F}_s \subset \mathcal{F}_t \quad \forall t > s$$

$$\mathbb{R}_+ := [0, \infty)$$

$$m\mathcal{F}_t : \mathcal{F}_t\text{-measurable}$$

$$L^n(\mathbb{P}) := \left\{ X : \Omega \rightarrow \mathbb{R} : \mathbb{E}^{\mathbb{P}}(|X|^n) < \infty \right\}$$

$$\Lambda^n([0, T]) := \left\{ X = \{X_t\}_{t \geq 0} : X \text{ is progressively measurable and } \mathbb{P}\left(\int_0^T |X_t|^n dt < \infty\right) = 1 \right\}$$

$$\mathbb{M}^n([0, T]) := \left\{ X = \{X_t\}_{t \geq 0} : X \text{ is progressively measurable and } \mathbb{E}\left(\int_0^T |X_t|^n dt\right) < \infty \right\}$$

$$\mathcal{N}_m : \text{normal distribution of dimension } m$$

$$\|\cdot\| : l_2\text{-norm}$$

$$\log(\cdot) : \text{natural logarithm}$$

$$\mathbb{Q} \ll \mathbb{P} : \mathbb{Q} \text{ is absolutely continuous with respect to } \mathbb{P}$$

$$X \perp\!\!\!\perp Y : X \text{ and } Y \text{ are independent}$$

$$\mathbf{a.s.} : \text{almost surely}$$

Chapter 1

Introduction

The aim of this work is to address the problem of option pricing under stochastic volatility models, which are widely used by practitioners because of their ability to fit the market data better than the classical Black–Scholes model. Indeed, this class of models typically depends on more than a single parameter and their structure depends on the type of option payoff considered. In particular, I explored the calibration problem.

Given an option pricing model that depends on a specific parameter set and given a set of quoted options, calibration consists in determining the parameter values such that the model is able to reproduce the market data as closely as possible. This amounts to solving a minimization problem.

This procedure is far from straightforward, as it involves several challenges. In the literature this problem, is known as *inverse problem*, which stems from the fact that, when studying a model from the theoretical point of view, we are accustomed to treat the parameters as given. In this sense, we typically use it to perform the valuation. In contrast with it, the calibration procedure is performed one step beforehand, representing a crucial operation for making the model applicable to pricing options in the market. The term *inverse* refers to the procedure through which, starting from market data associated with quoted European options, it is possible to infer the model parameters to be used to obtain a fair valuation of other types of options with more complex payoff structures, commonly referred to as *exotic options*.

Valuation is not the only use of these parameters, indeed, practitioners also use them to simulate market scenarios for risk management and to develop high–frequency trading strategies. Therefore, calibration must be performed as fast as possible to be used for practical trading and risk management; furthermore, the choice of the model is crucial because, it must ensure a good trade–off between accuracy and computational tractability.

It is worth emphasizing that there are two risks associated with the calibration: the *model risk* and the *calibration risk*. The former refers to the extent to which the model can be considered reliable; indeed, as reported by Ballotta (2024), practitioners need a trusted model, which, once calibrated, is able to provide consistent parameter estimates. The key issue is how to assess whether the model and the corresponding parameter set

identified are reliable. However, despite its importance, there is no consensus among researchers and practitioners on a unique methodology for performing the calibration. Consequently, it is not possible to state unambiguously whether a model is reliable. In particular, this is a consequence of arbitrary decisions regarding: how to construct the objective function, which metrics to use to evaluate the goodness of fit and which data to use for the optimization.

These choices represent the source of the risk, for this reason, given the particularly complex nature of the problem, it is also known as the *calibration conundrum*. The latter refers to the situation in which all the decisions listed above are made and is closely related to the structure of the objective function, because it typically exhibits non-convex or irregular behavior with multiple local minima. Consequently, the calibration problem is considered ill-posed. Furthermore, in this framework, the optimal solution may not be unique, therefore it is not possible to rule out the existence of other parameter sets which provide a better fit to the selected data. For these reasons, the *calibration risk* arises from non-optimal decisions regarding the method selected to solve the optimization problem.

In other words, depending on the structure of the objective function, it is necessary to find the appropriate optimization method, but once it has been selected, the problem is not solved yet; indeed, it is necessary to determine the initial parameter set to be provided to the algorithm.

There is no unanimous consensus on this choice either, as the solution identified has been observed to depend on the selected initial parameter set. This is again a consequence of the presence of multiple local minima in the objective function.

1.1 Previous Work and Recognized Difficulties

In the previous section, I listed the main risks associated with the model's calibration, stating that the model risk is particularly difficult to mitigate because it depends on many arbitrary decisions, while the calibration risk is related to the problem formulation and, consequently, to the structure of the objective function. For these reasons, I believe that the second one, it is easier to address. In line with this view, many authors have attempted to mitigate this risk by proposing well-founded procedures. For instance, Yeap et al. (2018) proposed generating random parameter vectors within predefined uniform bounds as starting points and then applying Nelder-Mead algorithms with a fixed number of iterations to reduce the computational cost. The parameter set associated with the lowest objective function values is selected as the starting point for the subsequent optimization stage. This procedure has the advantage of providing evidence of a global minimum by verifying whether the optimizer converges to the same optimal parameter set, regardless the starting point; however, I believe that applying it may give rise to some issues, for example, introducing arbitrary decisions regarding the width of the proposed bounds.

Other algorithms employed are represented by stochastic optimization methods, such as: particle swarm optimization, genetic algorithms and simulated annealing. They appear to achieve better convergence results, solving the problem related to the presence of multiple local minima, however, as reported by Cui et al. (2017), these methods are not computationally tractable, exhibiting significant drawbacks in most practical applications. More recently, Brignone et al. (2026) proposed using sequential Monte Carlo samplers to address the optimization problem, but their results show that even this methodology is too computational expensive. For this reason, I employed deterministic optimization algorithms, using the procedure proposed by Cui et al. (2017) and Brignone and Sgarra (2026).

Those papers proposed efficient calibration algorithms for, respectively, the Heston model and the Ornstein–Uhlenbeck driven stochastic volatility model. They formulated the calibration as an unconstrained nonlinear least squares problem and suggested to use the Levenberg–Marquardt method. However, what differentiates their procedure from the others is that the analytical gradient of the objective function is provided to the algorithm. The option price is achieved by applying the inverse Fourier transform, which involves computing the gradient of the characteristic function. For both models, this function was discontinuous due to branch switching in the complex power function and was therefore not analytically differentiable. However, Cui et al. (2017) derived an algebraically equivalent formulation for the characteristic function of the Heston model, while Brignone and Sgarra (2026) derived one for the Ornstein–Uhlenbeck model, thereby making them continuous.

Both algorithms were tested on simulated data, yielding a substantial improvement in terms of computational time. These results show that the algorithms meet the requirements of practical applications.

1.2 Proposed Methodology

Based on the results described in previous sections, I considered particularly useful and interesting the comparison between the calibration of the Heston model with that of the Ornstein–Uhlenbeck model, using the methodology proposed by Cui et al. (2017). Such a comparison is natural, as the two models exhibit very similar structures. As shown by Heston (1993), the underlying assumption of the model is that the volatility follows an Ornstein–Uhlenbeck process. From this one, it is possible to construct the corresponding variance process, which can be shown to exhibit the structure defined by Cox et al. (1985). Furthermore, the Heston model exhibits an affine structure, while that analyzed by Brignone and Sgarra (2026) has a quadratic one, which is more complicated than the former. In the literature, the second class of models is considered to be better able to replicate market data.

I performed the validation procedure described by Cui et al. (2017), in particular, simulating the market prices using the model under consideration and randomly

generating the initial guess for the selected optimizers. I decided to create 100 different initial sets of parameters in order to assess whether the solution found was sufficiently stable and, consequently, evaluating the convergence ability of the model. During this procedure, I observed that both models performed very well, with errors in identifying the minimum occurring in only 2% and 1.3% of the cases, respectively. However, for the Heston model, I found that, generating specific parameter sets, it exhibits an identifiability problem between κ and $\bar{v}$. Given the limited number of cases, during this step, I considered them as outliers.

I then proceeded to test the models on real-world data, structuring the analysis as follows. First, I calibrated both models on a single day and, subsequently, on a weekly basis over a period from 2007 to 2021. I immediately noticed that the calibrated parameters did not satisfy the domains imposed by the structure of the models; therefore, I concluded that different characterizations are necessary to make the Heston and Ornstein–Uhlenbeck models applicable on this kind of data.

For this purpose, I proposed the following two characterizations. The first consists of introducing constraints on the calibration problem. In this way, the optimizer is always able to find an optimal solution, which satisfies the conditions imposed on each parameter. The second is based on a geometric transformation of the search space in which calibration is performed. It consists on constructing a reparameterized calibration problem by defining new parameters to find, that are not subject to any constraint. Once the optimal parameter vector has been found, I then proceeded to recover the original optimal parameter set from it. I used a reparameterization such that the resulting parameter vector is unique and lies within the domains imposed by the models. By repeating the model validation under the two new proposed characterizations, I observed that the Heston model benefited from both approaches, with the number of calibration failures being reduced to below 1%. Therefore, I decided to employ both characterizations on real-world data. Considering the Ornstein–Uhlenbeck model, it benefited from the reparameterization, but not from the imposition of constraints, which increased the number of failures to 10% of the cases. From these results, I concluded that also the problem’s characterization depends on the model being calibrated.

At this point, I then repeated the calibration under both the constrained and the reparameterized approaches. Performing the single day test, I immediately noted that the condition number associated with the Heston model under the reparameterized calibration problem was two orders of magnitude lower than the one related to the constrained one. This shows that, the reparameterized characterization ensures more stability in the optimal solution found, through a locally better scaling of the approximated Hessian matrix. Indeed, the induced geometric transformation modifies the search space in such a way as to reduce the disparity in sensitivity of the objective function’s curvature to variations in the parameters.

Regarding the Ornstein–Uhlenbeck model, contrary to what was observed during the validation, the reparameterization worsened its performance and was therefore

immediately discarded. Instead, the introduction of constraints was able to preserve the strong stability of the optimal solution while maintaining high levels of accuracy.

Moving on to the weekly calibration, I computed also the standard errors of the parameters, because they are useful for assessing the degree of uncertainty associated to the components of the optimal solution found. The results showed that the most versatile model, namely the one with the lowest number of ill-conditioned days, is the Ornstein–Uhlenbeck one. However, its average computational time was greater than a second; in my opinion, this represents a disadvantage, as the model does not meet the maximum time requirements imposed by trading applications.

The Heston model, under the reparameterized calibration problem, registered a higher number of ill-conditioned days with respect to the one related to the Ornstein–Uhlenbeck model, but it achieved approximately 50% better value than that achieved applying the constrained approach, which turns out to be the less stable. Moreover, under the proposed reparameterization, the model can be calibrated approximately ten times faster than when it is applied the constrained characterization, achieving computational times suitable for trading applications. However, this reparameterization was not able to solve the identifiability issue between κ and $\bar{v}$; indeed, I discovered that there are days on which the randomly generated initial parameters fall within a broad flat valley, leading the optimizer to identify unreasonable parameter sets. To address this problem, I considered introducing a new reparameterization based on the product of κ and $\bar{v}$; however, this approach proved unsuccessful, introducing more problems than benefits.

To conclude, building on these findings, I selected the Heston model under the reparameterized calibration problem and the Ornstein–Uhlenbeck model to develop a new calibration approach based on the condition number. It is able to select the appropriate model to be calibrated on a trading day observing the stability of the optimal parameter set identified by the optimizer. In other words, I conjectured that it is possible to perform a model selection based on the condition number associated with the calibrated parameters. This approach identifies as the *appropriate model* the one associated with a condition number lower than a fixed threshold, giving greater importance to the stability of the solution obtained.

Chapter 2

The Models

In this chapter, I will exploit the mathematical foundations of both the Heston and the Ornstein–Uhlenbeck stochastic volatility models, aiming to represent which are the relations between them and the main differences.

2.1 Basic Notions

In this section, I introduce the well-known Black–Scholes (BS) model. The aim is to provide a basic framework before moving to the more sophisticated ones and to describe the reasons why BS model is inadequate for fitting the real-world data, thereby motivating the construction of more complicated models.

2.1.1 Black and Scholes Model

The Black–Scholes model is a continuous-time model defined on the filtered probability space $(\Omega, \mathcal{F}, \mathbb{F}, \mathbb{P})$, which satisfies the usual assumptions.

Definition 2.1.1 (Price process). *A price process is a stochastic process $\{P_t\}_{t \geq 0}$ such that*

- $P_t \in \mathbb{R}_+ \quad \forall t$
- P_t is $m\mathcal{F}_t$ $\forall t$

In the BS framework there are only two assets on the market:

- *Non-risky asset* $\{B_t\}_{t \geq 0}$, described by the following dynamics

$$dB_t = rB_t dt \tag{2.1}$$

- *Risky-asset (or underlying)* $\{S_t\}_{t \geq 0}$, which follows

$$dS_t = S_t (\mu dt + \sigma dW_t) \tag{2.2}$$

From these dynamics I can recover that, the parameters governing the model are:

- $r \in \mathbb{R}$: the short term risk free rate
- $\mu \in \mathbb{R}$: the instantaneous expected return
- $\sigma \in (0, \infty)$: the instantaneous volatility

Let $S_0, B_0 \in (0, \infty)$, using the techniques to solve SDEs, I obtain the following solutions:

$$B_t = B_0 e^{-rt}, \quad S_t = S_0 e^{\left(\mu - \frac{\sigma^2}{2}\right)t + \sigma W_t} \quad \forall t$$

The former does not present stochastic components, then it is deterministic; instead, the latter evolves stochastically and has some particular properties.

Proposition 2.1.1 (Properties of $\{S_t\}_{t \geq 0}$). *Let $\{S_t\}_{t \geq 0}$ follows (2.2) with $S_0 \in (0, \infty)$. Then*

- S_t is $\mathcal{M}_{\mathcal{F}_t}$ $\forall t$
- S_t has continuous trajectories a.s.
- Let $X_t = \log(S_t)$, then its law is $X_t \sim \mathcal{N}(\alpha, \beta)$, where

$$\alpha = \log(S_0) + \left(\mu + \frac{\sigma^2}{2}\right)t, \quad \beta = \sigma^2 t$$

- Let $0 \leq t_1 < \dots < t_n \leq T$ a uniform partition of the interval $[0, T]$
Let $Y_{t_k} = \log\left(\frac{S_{t_k}}{S_{t_{k-1}}}\right)$, then $Y_{t_k} \sim \mathcal{N}(\delta, \nu) \quad \forall k = 1, \dots, n$, where

$$\delta = \left(\mu + \frac{\sigma^2}{2}\right) \Delta_t, \quad \nu = \sigma^2 \Delta_t, \quad \Delta_t = t_k - t_{k-1} \quad \forall k = 1, \dots, n$$

To continue investigating the properties of the Black–Scholes market, I introduce the following notions.

Definition 2.1.2 (Strategy). *A strategy is a $\mathbb{R}^2$ -valued stochastic process $\xi = \left\{(\xi_t^B, \xi_t^S)\right\}_{t \geq 0}$ adapted to the filtration $\mathbb{F}$.*

Definition 2.1.3 (Portfolio). *A portfolio is a stochastic process $\{V_t(\xi)\}_{t \geq 0}$ adapted to the filtration $\mathbb{F}$, such that*

$$V_t(\xi) = \xi_t^B B_t + \xi_t^S S_t \quad \forall t \geq 0$$

Definition 2.1.4 (Discounted process). *Let $\{B_t\}_{t \geq 0}$ be defined as in (2.1). A discounted process is a stochastic process $\{\tilde{P}_t\}_{t \geq 0}$ such that*

$$\tilde{P}_t = \frac{P_t}{B_t} \quad \forall t \geq 0$$

Theorem 2.1.1 (Fundamental theorem of asset pricing). *A market is arbitrage-free if and only if there exists a probability measure $\mathbb{Q}$ equivalent to $\mathbb{P}$ such that the process $\{\tilde{S}_t\}_{t \geq 0}$ is a $\mathbb{Q}$ -martingale. Furthermore, if $\mathbb{Q}$ is unique, then the market is also complete.*

Proposition 2.1.2 (Self-financing strategy). *A strategy ξ is self-financing if and only if:*

- $\xi \in \Lambda^2([0, T])$
- $d\tilde{V}_t = \xi_t d\tilde{S}_t$

Applying the Definition 2.1.4 to the portfolio process, I obtain

$$\tilde{V}_t = \xi_t^B + \xi_t \tilde{S}_t$$

instead, considering the $\{S_t\}_{t \geq 0}$

$$d\tilde{S}_t = \tilde{S}_t [(\mu - r) dt + \sigma dW_t]$$

Definition 2.1.5 (Admissible strategy). *A strategy ξ is admissible if and only if:*

- ξ is self-financing
- $V_t(\xi) \geq 0 \quad \forall t \in [0, T]$
- $\sup_{t \in [0, T]} \tilde{V}_t(\xi) \in L^2(\mathbb{Q})$

The first property to investigate concerns the absence of arbitrage, by observing if the BS model satisfies the assumptions of Theorem 2.1.1.

I consider the discounted process $\{\tilde{S}_t\}_{t \geq 0}$, which follows

$$d\tilde{S}_t = \tilde{S}_t [(\mu - r) dt + \sigma dW_t]$$

It is possible to notice that, it is not a martingale under the real-world probability measure $\mathbb{P}$, since the component $\tilde{S}_t(\mu - r) \neq 0$; therefore I introduce a useful result.

Theorem 2.1.2 (Girsanov Theorem). *Let $\lambda = \{\lambda_t\}_{t \geq 0}$ be a stochastic process adapted to the filtration $\mathbb{F}$ with $\lambda \in \mathbb{M}^2([0, T])$, such that the process $\{L_t\}_{t \geq 0}$, defined by*

$$L_t = \exp \left(- \int_0^t \lambda_s dW_s - \frac{1}{2} \int_0^t \lambda_s^2 ds \right)$$

is a martingale. Then, under the probability measure $\mathbb{Q}$, which has density

$$\frac{d\mathbb{Q}}{d\mathbb{P}} = L_T$$

the process $\{W_t^{\mathbb{Q}}\}_{t \geq 0}$, defined by

$$W_t^{\mathbb{Q}} = W_t + \int_0^t \lambda_s ds$$

is a Brownian Motion under $\mathbb{Q}$.

Theorem 2.1.2 shows that it is possible to construct a Brownian Motion under a new probability measure starting from the one defined under the real-world probability.

Therefore, defining

$$dW_t^{\mathbb{Q}} = dW_t + \lambda_t dt$$

I rewrite the dynamic of $\{\tilde{S}_t\}_{t \geq 0}$ under $\mathbb{Q}$

$$d\tilde{S}_t = \tilde{S}_t [(\mu - r) dt + \sigma dW_t] = \tilde{S}_t [(\mu - r) dt + \sigma (dW_t^{\mathbb{Q}} - \lambda_t dt)]$$

By rearranging the expression, I obtain

$$d\tilde{S}_t = \tilde{S}_t [(\mu - r - \sigma \lambda_t) dt + \sigma dW_t]$$

At this point, in order to eliminate the drift term, it is necessary to set the following expression equal to zero:

$$\mu - r - \sigma \lambda_t = 0 \rightarrow \lambda_t = \lambda = \frac{\mu - r}{\sigma}$$

The process $\{\lambda_t\}_{t \geq 0}$ is constant and verifies the assumptions of the Theorem 2.1.2, so, under the measure $\mathbb{Q}$

$$d\tilde{S}_t = \sigma \tilde{S}_t dW_t^{\mathbb{Q}}$$

which is a martingale. Moreover, I can recover the corresponding dynamics for the underlying

$$dS_t = S_t (r dt + \sigma dW_t^{\mathbb{Q}})$$

Since $\lambda_t = \lambda$ is uniquely determined, then the measure $\mathbb{Q}$ is unique, so the second part of the Theorem 2.1.1 is satisfied, implying that the BS market is also complete.

Definition 2.1.6 (European Option). *An European option (or European option's payoff) h with maturity $T > 0$ is a random variable such that*

- h is $\mathcal{F}_T$
- $h \geq 0$ a.s.
- $h \in L^2(\mathbb{Q})$

In particular, throughout this work, I will consider only European call options, whose payoff is given by

$$h = (S_T - K)_+, \quad K \in [0, \infty)$$

It is worth emphasizing that h depends only from S_T because the strike price K is fixed, then I can consider $h = f(S_T)$.

Regarding the European put options, their price can be derived using the Put–Call parity relation.

Definition 2.1.7 (Replicating strategy). *A strategy ξ is replicating h if it is admissible and such that $V_T(\xi) = h$.*

Definition 2.1.6 and 2.1.7 are fundamental to introduce the following result, which is crucial for option pricing under the BS model.

Theorem 2.1.3 (Pricing theorem in BS). *Under the BS model, a European call option h can be replicated by a unique replicating strategy ξ and its value is equal to the one of the corresponding replicating portfolio $V_t(\xi)$, defined by*

$$V_t(\xi) = V_t = \mathbb{E}^{\mathbb{Q}} \left[e^{-r(T-t)} h \mid \mathcal{F}_t \right]$$

Since S_T can be written as

$$S_T = \frac{S_T}{S_t} S_t = S_t \exp \left[\left(r - \frac{\sigma^2}{2} \right) (T - t) + \sigma (W_T - W_t) \right]$$

and knowing that the increment $W_T - W_t$ is independent from $\mathcal{F}_t$ and S_t is $m\mathcal{F}_t$, then

$$V_t = C(t, S_t)$$

i.e. V_t is a Markovian function of time and the current state S_t , which represents the value taken by the underlying asset at the time of the valuation.

Now, it is possible to derive the BS formula. There are several ways to do that, and I selected the one related to the concept of replicating portfolio.

Starting from

$$C(t, S_t) = \mathbb{E}^{\mathbb{Q}} \left[e^{-r(T-t)} f(S_T) \mid \mathcal{F}_t \right]$$

using the freezing Lemma, denoting $S_t = s$:

$$C(t, s) = \mathbb{E}^{\mathbb{Q}} \left(e^{-r(T-t)} \left(s \exp \left[\left(r - \frac{\sigma^2}{2} \right) (T - t) + \sigma (W_T - W_t) \right] - K \right)_+ \right)$$

Through some algebraic manipulation, I rewrite the expression in the following more tractable form

$$C(t, s) = \mathbb{E}^{\mathbb{Q}} \left(s \exp \left[\sigma \sqrt{(T-t)} Z - \frac{\sigma^2}{2} (T-t) \right] - K e^{-r(T-t)} \right)_+, \quad Z \sim \mathcal{N}(0, 1) \quad (2.3)$$

It is necessary to study when the expression inside of (2.3) is different from zero; this happens if

$$s \exp \left[\sigma \sqrt{(T-t)} Z - \frac{\sigma^2}{2} (T-t) \right] > K e^{-r(T-t)}$$

Solving for Z , I obtain

$$Z > \frac{\log \left(\frac{K}{s} \right) + \left(\frac{\sigma^2}{2} - r \right) (T-t)}{\sigma \sqrt{T-t}}$$

Defining

$$d_2 = \frac{\log \left(\frac{s}{K} \right) + \left(r - \frac{\sigma^2}{2} \right) (T-t)}{\sigma \sqrt{T-t}}$$

the expression inside the expectation in (2.3) is greater than zero if and only if $Z > -d_2$.

Therefore, I have

$$C(t, s) = \int_{-d_2}^{+\infty} \left(s \exp \left[\sigma \sqrt{(T-t)} - \frac{\sigma^2}{2}(T-t) \right] - K e^{-r(T-t)} \right) \frac{e^{-\frac{z^2}{2}}}{\sqrt{2\pi}} dz$$

Using the additivity of the Riemann integral and the symmetry property of the distribution of Z :

$$C(t, s) = \int_{-\infty}^{d_2} \left(s \exp \left[\sigma \sqrt{(T-t)} - \frac{\sigma^2}{2}(T-t) \right] - K e^{-r(T-t)} \right) \frac{e^{-\frac{z^2}{2}}}{\sqrt{2\pi}} dz$$

Using the linearity property, I can study the behavior of two different integrals. From the second one, I recover

$$C(t, s) = \int_{-\infty}^{d_2} \left(s \exp \left[\sigma \sqrt{(T-t)} - \frac{\sigma^2}{2}(T-t) \right] \right) \frac{e^{-\frac{z^2}{2}}}{\sqrt{2\pi}} dz - K e^{-r(T-t)} N(d_2)$$

where

$$N(a) = \int_{-\infty}^a \frac{e^{-\frac{z^2}{2}}}{\sqrt{2\pi}} dz$$

In the remaining, by completing the square, and performing a change of variable $\tilde{z} = z - \sigma \sqrt{T-t}$, I obtain as upper bound of the integral $-d_1$, where $d_1 = d_2 + \sigma \sqrt{T-t}$. Computing it, I recover the so-called Black-Scholes equation:

$$C(t, s) = s N(d_1) - K e^{-r(T-t)} N(d_2) \quad (2.4)$$

From this brief discussion of the BS model, the most important point to recall is that the log-normality assumption allows the derivation of an analytical formula for the price of a European call option.

2.1.2 Stochastic Volatility Models

In the previous section, I introduced several concepts, therefore, before moving on to more complicated models, I want to summarize the fundamental assumptions of the BS model as follows:

1. in the market there are no transaction costs
2. there is no bid-ask spread
3. since ξ is $\mathbb{R}^2$ -valued, this means that short-selling is allowed and, moreover, it is possible to purchase and sell non-integer quantities of both non-risky and underlying asset
4. r represents the rate at which it is possible to lend and borrow money and is assumed to be non-stochastic and constant
5. the underlying does not pay dividends
6. the log-returns are i.i.d. and normally distributed

7. the underlying process has continuous trajectories a.s.
8. σ is constant

Considering the assumptions discussed above, they can be regarded as unrealistic. The first three completely ignore the microstructure of the market, while last three do not capture some of the features which characterize returns in real markets, such as:

1. nonlinear relationships involving them
2. leptokurtosis, meaning that the probability of extreme events does not tend to zero as rapidly as it does under the normal distribution
3. Jumps in the underlying asset price are not considered
4. imposing constant volatility, it is impossible to capture the *leverage effect* and the *volatility clustering*

The ones that are truly problematic are assumptions 6 and 8, indeed, as reported by Hull and White (1987), they lead to pricing issues, in particular, when $K \simeq 0.10S_0$, the BS model tends to

- increasingly overprice options as $T - t$ increases
- overprice at the money options

There are two class of models which aim to address these shortcomings:

- Mixed-Jump diffusion models (MJ)
- Stochastic Volatility Models (SV)

Merton (1976) described how to characterize the *jumps* in the trajectory of the underlying. In his model, the dynamics of $\{S_t\}_{t \geq 0}$ is

$$dS_t = \mu(t, S_t) dt + \sigma(t, S_t) dW_t + \zeta(t, S_{t-}) dJ_t, \quad S_0 = s_0$$

where:

- $\{J_t\}_{t \geq 0}$ is a compound jump process
- $\{S_t\}_{t \geq 0}$ is a càdlàg process i.e. $S_{t-} = \lim_{s \rightarrow t-} S_s$
- $\zeta(t, S_t)$ is the Jump impact function

Moreover, he defined $\lambda(t, S_t)$ as the Jump frequency function, which governs the frequency of the jumps occurring in the underlying asset's trajectory. These models are called *Mixed-Jump diffusion*.

Focusing on stochastic volatility models, which represent the core of this work, it can be stated that, rather than modifying the dynamics of the underlying asset, they aim to characterize its volatility by introducing an additional process. Generalizing their structure, a stochastic volatility model is composed of the following couple of SDEs:

$$\begin{aligned} dS_t &= \mu_S(t, S_t) dt + \sigma_S(t, S_t, v_t) dW_t^S \\ dv_t &= \mu_v(t, v_t) dt + \sigma_v(t, v_t) dW_t^v \end{aligned}$$

where the two Brownian Motions are related by

$$dW_t^S dW_t^v = \rho dt, \quad \rho \in [-1, 1]$$

Modifying the parameter ρ , it is possible to change the correlation between the two processes $\{W_t^S\}_{t \geq 0}$ and $\{W_t^v\}_{t \geq 0}$.

It is the different characterization of both the drift and volatility function that defines the different SV models developed in the literature. It is also possible to classify them based on the object that they aim to describe: the variance or the volatility process.

MJ and SV classes are not mutually exclusive; indeed, without considering the model's parsimony, they can be combined to provide a more comprehensive characterization of the stylized facts observed in financial markets. A representative example of this approach is the model proposed by Bates (1996).

2.1.3 Option Pricing via Fourier Transform

From this section onward, I will consider only stochastic volatility models.

This class of models do not admit a simple solution as in case of BS, because the log-returns are not normally distributed. Then, to compute

$$C(t, S_t) = \mathbb{E}^{\mathbb{Q}} \left[e^{-r(T-t)} f(S_T) \mid \mathcal{F}_t \right]$$

practitioners use Monte Carlo methods. However, there is another approach to pricing the European call options, which consists of using the analytical expression of the characteristic function (CHF) of $X_t = \log(S_t)$, which is, typically, available in closed form.

Definition 2.1.8 (Characteristic function). *Let $X : \Omega \rightarrow \mathbb{R}$ be a random variable on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and $f_X : \mathbb{R} \rightarrow \mathbb{R}_+$ be its probability density function. The function $\phi_X : \mathbb{R} \rightarrow \mathbb{C}$ defined by*

$$\phi_X(u) = \mathbb{E} \left(e^{iuX} \right) = \int_{-\infty}^{+\infty} e^{iux} f_X(x) dx, \quad u \in \mathbb{R}$$

is called characteristic function.

My aim, in this section, is to provide the derivation of a generalized version of the Black–Scholes formula.

Assumption 2.1.1. *Let $X_T = \log(S_T)$ and $\phi_X(u)$ be its CHF. Under the probability measure $\mathbb{Q}$, $\phi_X(u)$ admits an analytical expression.*

For simplicity, without loss of generality, I consider the derivation of

$$C(0, S_0) = C_0 = \mathbb{E}^{\mathbb{Q}} \left[e^{-rT} (S_T - K)_+ \mid \mathcal{F}_0 \right] = \mathbb{E}^{\mathbb{Q}} \left(e^{-rT} (S_T - K)_+ \right)$$

The previous expression can be rearranged as follows

$$C_0 = e^{-rT} \mathbb{E}^{\mathbb{Q}} \left(S_T \mathbf{1}_{\{S_T > K\}} \right) - K e^{-rT} \mathbb{E}^{\mathbb{Q}} \left(\mathbf{1}_{\{S_T > K\}} \right)$$

Using the definition of indicator function on the second term in the RHS, I can rewrite

$$C_0 = e^{-rT} \mathbb{E}^{\mathbb{Q}} \left(S_T \mathbf{1}_{\{S_T > K\}} \right) - K e^{-rT} \mathbb{Q}(S_T > K) \quad (2.5)$$

To compute $\mathbb{Q}(S_T > K)$, I will use the characteristic function.

Theorem 2.1.4 (Inversion Theorem). *Let $X : \Omega \rightarrow \mathbb{R}$ be a random variable on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$. Furthermore, let $F_X : \mathbb{R} \rightarrow [0, 1]$ be its cumulative distribution function and $\phi_X : \mathbb{R} \rightarrow \mathbb{C}$ be its characteristic function. If $\phi_X \in L^1(\mathbb{R})$, then F_X is absolutely continuous and has the following probability density function:*

$$f_X(x) := \frac{1}{2\pi} \int_{-\infty}^{+\infty} e^{-iux} \phi_X(u) du$$

The Theorem 2.1.4 also allows me to derive an expression for $F_X(x)$:

$$F_X(x) := \mathbb{P}(X \leq x) = \int_{-\infty}^x f_X(s) ds = \frac{1}{2} - \frac{1}{2\pi} \int_{-\infty}^{+\infty} \frac{e^{-iux}}{iu} \phi_X(u) du$$

After some computations, it is possible to obtain the two following expressions

$$f_X(x) := \frac{1}{\pi} \int_0^{+\infty} \Re \left(e^{-iux} \phi_X(u) \right) du$$

$$F_X(x) := \frac{1}{2} - \frac{1}{\pi} \int_0^{+\infty} \Re \left(\frac{e^{-iux}}{iu} \phi_X(u) \right) du$$

Using the definition of X_T and setting $k = \log(K)$, I can express (2.5) as

$$C_0 = e^{-rT} \mathbb{E}^{\mathbb{Q}} \left(e^{X_T} \mathbf{1}_{\{X_T > k\}} \right) - K e^{-rT} \mathbb{Q}(X_T > k)$$

Since

$$P_2 := \mathbb{Q}(X_T > k) = 1 - F_{X_T}(k)$$

and applying the Theorem 2.1.4, I have

$$P_2 := \frac{1}{2} + \frac{1}{\pi} \int_0^{+\infty} \Re \left(\frac{e^{-iuk}}{iu} \phi_X(u) \right) du \quad (2.6)$$

Regarding the first term on the RHS, the situation is more involved; for this reason it is necessary to introduce another result.

Theorem 2.1.5 (Radon–Nikodym Theorem). *Let $(\Omega, \mathcal{F}, \mathbb{F}, \mathbb{P})$ a filtered probability space. If the probability measure $\mathbb{Q} \ll_{\mathcal{F}_t} \mathbb{P}$, then there exists a unique, $\mathbb{P}$ -a.s., random variable $D_t : \Omega \rightarrow \mathbb{R}_+$ such that:*

- D_t is $m\mathcal{F}_t$
- $D_t \in L^1(\mathbb{P})$

$$\bullet \quad \mathbb{Q}(A) = \int_A D_t d\mathbb{P} \quad \forall A \in \mathcal{F}_t$$

$D_t = \frac{d\mathbb{Q}}{d\mathbb{P}} \Big|_{\mathcal{F}_t}$ is called density of $\mathbb{Q}$ with respect to $\mathbb{P}$ on $\mathcal{F}_t$ or Radon–Nikodym derivative.

Proposition 2.1.3. Let $\mathbb{P}$ and $\mathbb{Q}$ be two probability measures on the filtered probability space $(\Omega, \mathcal{F}, \mathbb{F})$, such that $\mathbb{Q} \ll_{\mathcal{F}_t} \mathbb{P}$.

Let $X_t : \Omega \rightarrow \mathbb{R}$ be $\mathcal{F}_t$ –measurable with $X_t \in L^1(\mathbb{Q})$ and $D_t = \frac{d\mathbb{Q}}{d\mathbb{P}} \Big|_{\mathcal{F}_t}$, then the product $X_t D_t \in L^1(\mathbb{P})$ and it holds that

$$\mathbb{E}^{\mathbb{Q}}(X_t) = \mathbb{E}(X_t D_t)$$

My aim is, using Proposition 2.1.3, to transform the first term on the RHS into a probability, similarly to what was done for the second term. The difficulty arises from the presence of e^{X_T} inside the expectation.

I proceed defining

$$\Pi_1 := e^{-rT} \mathbb{E}^{\mathbb{Q}}(e^{X_T} Y), \quad Y = \mathbb{1}_{\{X_T > k\}}(x_T)$$

To obtain only Y inside the expectation, I introduce the probability measure $\tilde{\mathbb{Q}}$ such that

$$D_T = \frac{d\tilde{\mathbb{Q}}}{d\mathbb{Q}} \Big|_{\mathcal{F}_T} = \frac{e^{X_T}}{\mathbb{E}^{\mathbb{Q}}(e^{X_T})}$$

In this way, applying the Proposition 2.1.3, it holds that

$$\mathbb{E}^{\tilde{\mathbb{Q}}}(Y) = \mathbb{E}^{\mathbb{Q}}(D_T Y) = \mathbb{E}^{\mathbb{Q}}\left(\frac{e^{X_T}}{\mathbb{E}^{\mathbb{Q}}(e^{X_T})} Y\right) = \frac{1}{\mathbb{E}^{\mathbb{Q}}(e^{X_T})} \mathbb{E}^{\mathbb{Q}}(e^{X_T} Y) = \frac{\Pi_1 e^{rT}}{\mathbb{E}^{\mathbb{Q}}(e^{X_T})}$$

then

$$\Pi_1 := e^{-rT} \mathbb{E}^{\mathbb{Q}}(e^{X_T}) \mathbb{E}^{\tilde{\mathbb{Q}}}(Y)$$

Using the definition of the indicator function, I obtain

$$\Pi_1 := e^{-rT} \mathbb{E}^{\mathbb{Q}}(e^{X_T}) \tilde{\mathbb{Q}}(X_T > k)$$

Since

$$\tilde{P}_1 := \tilde{\mathbb{Q}}(X_T > k) = 1 - F_{X_T}(k)$$

and $e^{-rT} \mathbb{E}^{\mathbb{Q}}(e^{X_T}) = \mathbb{E}^{\mathbb{Q}}(\tilde{S}_T) = S_0$, I can write (2.5) as

$$C_0 = S_0 \tilde{P}_1 - K e^{-rT} P_2 \tag{2.7}$$

where

$$\tilde{P}_1 := \frac{1}{2} + \frac{1}{\pi} \int_0^{+\infty} \Re \left(\frac{e^{-iuk}}{iu} \tilde{\phi}_X(u) \right) du$$

Applying the Proposition 2.1.3, it holds that

$$\tilde{\phi}_X(u) = \mathbb{E}^{\mathbb{Q}}(e^{iuX_T}) = \mathbb{E}^{\mathbb{Q}}(e^{iuX_T} D_T)$$

Rearranging a bit the expression

$$\tilde{\phi}_X(u) = \mathbb{E}^{\mathbb{Q}}(e^{iuX_T} D_T) = \frac{\mathbb{E}^{\mathbb{Q}}(e^{(iu+1)X_T})}{\mathbb{E}^{\mathbb{Q}}(e^{X_T})}$$

At this point, I can collect i at the numerator and in the denominator, obtaining

$$\tilde{\phi}_X(u) = \frac{\mathbb{E}^{\mathbb{Q}}(e^{(iu+1)X_T})}{\mathbb{E}^{\mathbb{Q}}(e^{X_T})} = \frac{\phi_X(u-i)}{\phi_X(-i)}$$

From this result, I can express $\tilde{P}_1$ in terms of $\phi_X(u)$, recovering

$$P_1 := \frac{1}{2} + \frac{1}{\pi} \int_0^{+\infty} \Re \left(\frac{e^{-iuk} \phi_X(u-i)}{iu\phi_X(-i)} \right) du \quad (2.8)$$

Then (2.5) become

$$C_0 = S_0 P_1 - K e^{-rT} P_2 \quad (2.9)$$

As mentioned above, the previous formula can be readily generalized to any t :

$$C_t = S_t P_1 - K e^{-r(T-t)} P_2$$

This means that the Black–Scholes model is a special case of (2.9) for which a convenient closed–form solution exists, however, when departing from its assumptions, P_1 and P_2 must be computed numerically.

2.2 Heston Stochastic Volatility Model

Heston (1993) was a groundbreaking paper because it introduced a new tractable stochastic volatility model for which an analytical expression for the CHF of $X_t = \log(S_t)$ was available.

He described a procedure to follow to derive the probabilities P_1 , P_2 and the corresponding CHF, through a system of ODEs. In the following sections, I will analyze the model's framework and provide the analytical derivation of the characteristic function.

2.2.1 Model Dynamics

The Heston model is a continuous–time model defined on the filtered probability space $(\Omega, \mathcal{F}, \mathbb{F}, \mathbb{P})$, which satisfies the usual assumptions.

In this framework, there are two price processes, respectively:

- *Non-risky asset*: defined as in (2.1)
- *Underlying asset*: described by the following dynamics

$$dS_t = S_t \left(\mu dt + \sqrt{v_t} dW_t^S \right), \quad S_0 \in (0, \infty)$$

The parameters governing the dynamics of $\{S_t\}_{t \geq 0}$ remain the same as those in the Black–Scholes model, with the exception of v_t , which represents a random variable of the corresponding variance process.

The fundamental question that needed to be addressed was: which process should be used to model the variance and obtain the volatility? The answer is straightforward and was partially provided in Section 2.1.2; indeed, it is necessary to identify the stylized facts of volatility. I can summarize them as follows:

1. *Volatility clustering*: periods of high (low) volatility tend to be followed by periods with high (low) volatility
2. *Leverage effect*: variations in the underlying asset's price are negatively related to variations in the corresponding volatility
3. *Mean-reverting behavior*: a process that has deviated from its long-term mean tends to revert toward it.
4. *Positivity*: volatility must be strictly positive, excluding the degenerated case in which it is equal to zero

The process $\{v_t\}_{t \geq 0}$, in this model, characterizes the variance of the underlying asset. Under an assumption that will be discussed in Section 2.4, $\{v_t\}_{t \geq 0}$ follows a CIR (Cox–Ingersoll–Ross) process.

The CIR process was introduced as a variant of the Vasicek model to characterize the behavior of the risk-free rate, with the additional requirement that it remains strictly positive. Therefore, the dynamics of $\{v_t\}_{t \geq 0}$ is represented by

$$dv_t = \kappa (\bar{v} - v_t) dt + \sigma \sqrt{v_t} dW_t^v, \quad \kappa, \bar{v}, \sigma, v_0 \in [0, \infty) \quad (2.10)$$

However, in order to use it to model the variance, the parameters must be defined in the domains:

$$\kappa, \bar{v}, \sigma, v_0 > 0$$

They have the following interpretations:

- κ represents the speed of mean reversion toward the long-term variance
It measures how quickly the process $\{v_t\}_{t \geq 0}$ tends to revert toward its long-term variance. A higher value of κ indicates a faster mean reversion.
- $\bar{v}$ is the long-term variance
It represents the mean value toward which the process $\{v_t\}_{t \geq 0}$ tends.
- σ is the volatility of the variance process (*vol-of-vol*)
It measures the intensity of the stochastic fluctuations in $\{v_t\}_{t \geq 0}$. A higher value of σ indicates larger unpredictable variations.

- v_0 is the starting value of the variance process $\{v_t\}_{t \geq 0}$

However, under some conditions, the CIR process may take value equal to zero. Without going into unnecessary detail, I report the following result.

Proposition 2.2.1 (Positivity). *Let $v_t^{v_0}$ be the solution of the CIR process (2.8) with $v_0 > 0$ as initial value and τ be a stopping time defined by*

$$\tau = \inf\{t \geq 0 : v_t^{v_0} = 0\}$$

- If $2\kappa\bar{v} \geq \sigma^2$, then $\mathbb{P}(\tau = \infty) = 1$, $\forall v_0 > 0$
- If $0 \leq 2\kappa\bar{v} < \sigma^2 \wedge \kappa \geq 0$, then $\mathbb{P}(\tau \leq \infty) = 1$, $\forall v_0 > 0$
- If $0 \leq 2\kappa\bar{v} < \sigma^2 \wedge \kappa < 0$, then $0 < \mathbb{P}(\tau \leq \infty) < 1$, $\forall v_0 > 0$

Proposition 2.2.1 provides conditions that the parameters must satisfy to ensure that the process remains strictly positive or simply non-negative. Therefore, to characterize the variance of the underlying asset, the parameters must satisfy the so-called *Feller Condition*:

$$2\kappa\bar{v} \geq \sigma^2$$

During the calibration, I did not impose this condition because, as reported by Cui et al. (2017), it is probably one of the most unrealistic assumptions to make about financial markets.

To conclude the model dynamics, I introduce

$$dW_t^S dW_t^v = \rho dt, \quad \rho \in [-1, 1]$$

The parameter ρ governs the strength of the *leverage effect*.

Now, it is necessary investigate whether the Heston model is arbitrage-free. To do this, I can proceed similarly to what I have done for BS model, with the difference that, since there are two Brownian Motions that must be expressed under the risk-neutralized probability measure $\mathbb{Q}$, I need to define the couple $(\lambda_t^S, \lambda_t^v)$. Therefore, I obtain

$$\begin{aligned} dW_t^{S,\mathbb{Q}} &= dW_t^S + \lambda_t^S dt \\ dW_t^{v,\mathbb{Q}} &= dW_t^v + \lambda_t^v dt \end{aligned}$$

I can express the dynamics of $\{\tilde{S}_t\}_{t \geq 0}$ under $\mathbb{Q}$ as follows

$$d\tilde{S}_t = \tilde{S}_t \left[(\mu - r) dt + \sqrt{v_t} dW_t^{S,\mathbb{Q}} \right] = \tilde{S}_t \left[(\mu - r) dt + \sqrt{v_t} (dW_t^S - \lambda_t^S dt) \right]$$

Similarly to the BS case, imposing $\tilde{S}_t (\mu - r - \sqrt{v_t} \lambda_t^S) = 0$, then

$$\lambda_t^S = \frac{\mu - r}{\sqrt{v_t}}$$

Now, since the process $\{\tilde{S}_t\}_{t \geq 0}$ is given by

$$d\tilde{S}_t = \tilde{S}_t \sqrt{v_t} dW_t^{S, \mathbb{Q}}$$

it is a $\mathbb{Q}$ -martingale and it holds that

$$dS_t = S_t \left(r dt + \sqrt{v_t} dW_t^{S, \mathbb{Q}} \right)$$

therefore, the Heston model is arbitrage-free.

The dynamics of the variance process can be written as

$$dv_t = \kappa (\bar{v} - v_t) dt + \sigma \sqrt{v_t} dW_t^v = \kappa (\bar{v} - v_t) dt + \sigma \sqrt{v_t} (dW_t^{v, \mathbb{Q}} - \lambda_t^v dt)$$

By rearranging a bit the expression, I obtain

$$dv_t = [\kappa (\bar{v} - v_t) - \sigma \sqrt{v_t} \lambda_t^v] dt + \sigma \sqrt{v_t} dW_t^{v, \mathbb{Q}}$$

From this result, I can state that the Heston market is not complete because the Theorem 2.1.1 does not impose any condition on the term λ_t^v , so the measure $\mathbb{Q}$ is not unique. Indeed, I can define $\tilde{\mathbb{P}} \sim \mathbb{P}$ and the corresponding couple

$$(\tilde{\lambda}_t^S, \tilde{\lambda}_t^v) : \tilde{\lambda}_t^S = \lambda_t^S, \tilde{\lambda}_t^v \neq \lambda_t^v$$

obtaining that $\{\tilde{S}_t\}_{t \geq 0}$ is a $\tilde{\mathbb{P}}$ -martingale; then the choice of λ_t^v is arbitrary. Accordingly to Heston (1993), I use the expression

$$\lambda_t^v = \frac{\lambda}{\sigma} \sqrt{v_t}, \quad \lambda \in \mathbb{R} \tag{2.11}$$

where λ is determined using the market data. Imposing (2.11), I obtain

$$dv_t = [\kappa (\bar{v} - v_t) - v_t \lambda] dt + \sigma \sqrt{v_t} dW_t^{v, \mathbb{Q}}$$

and expanding the product

$$dv_t = (\kappa \bar{v} - \kappa v_t - v_t \lambda) dt + \sigma \sqrt{v_t} dW_t^{v, \mathbb{Q}}$$

To maintain the structure of the CIR process, I define

$$\kappa^{\mathbb{Q}} = \kappa + \lambda, \quad \bar{v}^{\mathbb{Q}} = \frac{\kappa \bar{v}}{\kappa + \lambda}$$

Finally, I obtain the model dynamics under the probability measure $\mathbb{Q}$

$$\begin{aligned} dS_t &= S_t \left(r dt + \sqrt{v_t} dW_t^{S,\mathbb{Q}} \right) \\ dv_t &= \kappa^{\mathbb{Q}} \left(\bar{v}^{\mathbb{Q}} - v_t \right) dt + \sigma \sqrt{v_t} dW_t^{v,\mathbb{Q}} \end{aligned}$$

with

$$dW_t^{S,\mathbb{Q}} dW_t^{v,\mathbb{Q}} = \rho dt$$

Moreover, the process $\{\mathbf{Y}_t\}_{t \geq 0}$, where $\mathbf{Y}_t = (S_t, v_t)$, is Markovian, similarly to what happens for $\{S_t\}_{t \geq 0}$ in the BS model.

2.2.2 Characteristic Function

In this section, I will introduce the Heston PDE and show that the characteristic function of $X_t = \log(S_t)$ admits a closed-form expression.

Theorem 2.2.1 (Feymann–Kac Representation). *Let $f : \mathbb{R}^d \rightarrow \mathbb{R}$, $g : [0, T] \times \mathbb{R}^d \rightarrow \mathbb{R}$ and $r : [0, T] \times \mathbb{R}^d \rightarrow [0, \infty)$ be continuous functions for which there exists a constant $L > 0$ such that*

$$\begin{aligned} |f(\mathbf{x})| &\leq L (1 + \|\mathbf{x}\|^2) \\ |g(t, \mathbf{x})| &\leq L (1 + \|\mathbf{x}\|^2) \end{aligned}$$

Assume that there exists $U \in C^{1,2}([0, T] \times \mathbb{R}^d) \cap C([0, T] \times \mathbb{R}^d)$ which solves:

$$\begin{cases} \frac{\partial U}{\partial t} + rU = \mathcal{A}U + g, & \text{on } [0, T] \times \mathbb{R}^d, \\ U(T, \mathbf{x}) = f(\mathbf{x}). \end{cases} \quad (2.12)$$

and:

$$\exists M, p > 0 : |U(t, \mathbf{x})| \leq M (1 + \|\mathbf{x}\|^p)$$

Then the solution U of (2.12) has the following expression:

$$U(t, x) = \mathbb{E} \left(f(X_T) e^{-\int_t^T r(v, X_v) dv} + \int_t^T g(w, X_w) e^{-\int_t^w r(z, X_z) dz} dw \right)$$

Since $\{\mathbf{Y}_t\}_{t \geq 0}$ is a Markov process, the value of the European call option V_t is represented by a deterministic function of the current state $\mathbf{y} = (S, v)$. Therefore, under the regularity assumptions required from the Theorem 2.2.1, there exists a function $U : [0, T] \times \mathbb{R}^2 \rightarrow \mathbb{R}$ with $U \in C^{1,2}([0, T] \times \mathbb{R}^2) \cap C([0, T] \times \mathbb{R}^2)$ such that

$$V_t = U(t, S_t, v_t)$$

I apply the multidimensional Itô formula with correlated Brownian Motions, obtaining

$$dV_t = \frac{\partial U}{\partial t} dt + \frac{\partial U}{\partial S} dS_t + \frac{\partial U}{\partial v} dv_t + \frac{1}{2} \frac{\partial^2 U}{\partial S^2} (dS_t)^2 + \frac{1}{2} \frac{\partial^2 U}{\partial v^2} (dv_t)^2 + \frac{1}{2} \frac{\partial^2 U}{\partial S \partial v} dS_t dv_t$$

It holds that the quadratic variations are represented by

$$(dS_t)^2 = v_t S_t^2 dt, \quad (dv_t)^2 = \sigma^2 v_t dt, \quad dS_t dv_t = \rho \sigma v_t S_t dt$$

Knowing dS_t and dv_t , I substitute their expression, achieving the following

$$\begin{aligned} dV_t = & \left[\frac{\partial U}{\partial t} + r S_t \frac{\partial U}{\partial S} + \kappa^{\mathbb{Q}} (\bar{v}^{\mathbb{Q}} - v_t) \frac{\partial U}{\partial v} + \frac{1}{2} v_t S_t^2 \frac{\partial^2 U}{\partial S^2} + \frac{1}{2} \sigma^2 v_t \frac{\partial^2 U}{\partial v^2} + \rho \sigma v_t S_t \frac{\partial^2 U}{\partial S \partial v} \right] dt \\ & + \sqrt{v_t} S_t \frac{\partial U}{\partial S} dW_t^{S, \mathbb{Q}} + \sigma \sqrt{v_t} \frac{\partial U}{\partial v} dW_t^{v, \mathbb{Q}} \end{aligned}$$

Now, it is necessary introduce the discounted process $\{\tilde{V}_t\}_{t \geq 0}$, defined as $\tilde{V}_t = e^{-rt} V_t$, which has the following differential:

$$d\tilde{V}_t = -r e^{-rt} V_t dt + e^{-rt} dV_t$$

Therefore

$$\begin{aligned} d\tilde{V}_t = & e^{-rt} \left[\frac{\partial U}{\partial t} + r S_t \frac{\partial U}{\partial S} + \kappa^{\mathbb{Q}} (\bar{v}^{\mathbb{Q}} - v_t) \frac{\partial U}{\partial v} + \frac{1}{2} v_t S_t^2 \frac{\partial^2 U}{\partial S^2} + \frac{1}{2} \sigma^2 v_t \frac{\partial^2 U}{\partial v^2} \right. \\ & \left. + \rho \sigma v_t S_t \frac{\partial^2 U}{\partial S \partial v} - r U_t \right] dt + e^{-rt} \sqrt{v_t} S_t \frac{\partial U}{\partial S} dW_t^{S, \mathbb{Q}} + e^{-rt} \sigma \sqrt{v_t} \frac{\partial U}{\partial v} dW_t^{v, \mathbb{Q}} \end{aligned}$$

Recalling that, under the risk-neutralized probability measure $\mathbb{Q}$ also $\{\tilde{V}_t\}_{t \geq 0}$ is a $\mathbb{Q}$ -martingale, it holds that

$$\frac{\partial U}{\partial t} + r S \frac{\partial U}{\partial S} + \kappa^{\mathbb{Q}} (\bar{v}^{\mathbb{Q}} - v) \frac{\partial U}{\partial v} + \frac{1}{2} v S^2 \frac{\partial^2 U}{\partial S^2} + \frac{1}{2} \sigma^2 v \frac{\partial^2 U}{\partial v^2} + \rho \sigma v S \frac{\partial^2 U}{\partial S \partial v} - r U = 0 \quad (2.13)$$

The (2.13) is the PDE from which Heston derived the probabilities P_1 and P_2 as described by (2.8) and (2.6). It has the following boundary conditions

$$\begin{aligned} U(T, S, v) &= (S - K)_+ \\ U(t, 0, v) &= 0 \\ \frac{\partial U}{\partial S}(t, \infty, v) &= 1 \\ \frac{\partial U}{\partial t}(t, S, 0) + r S \frac{\partial U}{\partial S}(t, S, 0) + \kappa \bar{v} \frac{\partial U}{\partial v}(t, S, 0) - r U(t, S, 0) &= 0 \\ U(t, S, \infty) &= S \end{aligned}$$

In order, they have the following meanings:

1. This is a terminal condition, analogous to the one arising in replicating portfolios, which must have the same value as the option at maturity T
2. This expresses the fact that if the underlying asset has value equal to zero at any time t , then the European call option cannot be exercised profitably and therefore its value is zero
3. This is a condition on the option's delta: if the value of the underlying asset tends to ∞ , then its proportion in the total cost of the strategy approaches 100%, i.e., $\Delta = 1$
4. This expresses the PDE when $v = 0$
5. As $v \rightarrow \infty$, the distribution of the underlying asset becomes increasingly dispersed, and the option's value tends to the one of the underlying asset

Applying the Theorem 2.2.1 and considering

$$g(t, \mathbf{y}) = 0, \quad r(t, \mathbf{y}) = r$$

and that the infinitesimal generator under the risk-neutral measure $\mathbb{Q}$ is given by

$$\mathcal{A}U = rS \frac{\partial U}{\partial S} + \kappa^{\mathbb{Q}} (\bar{v}^{\mathbb{Q}} - v) \frac{\partial U}{\partial v} + \frac{1}{2} v S^2 \frac{\partial^2 U}{\partial S^2} + \frac{1}{2} \sigma^2 v \frac{\partial^2 U}{\partial v^2} + \rho \sigma v S \frac{\partial^2 U}{\partial S \partial v}$$

Then, (2.13) can be written as

$$\begin{cases} \frac{\partial U}{\partial t} + \mathcal{A}U - rU = 0 & \text{on } [0, T) \times \mathbb{R}_+^2, \\ U(T, S, v) = (S - K)_+ \end{cases}$$

identifying the solution

$$U(t, S, v) = \mathbb{E}^{\mathbb{Q}} \left[e^{-r(T-t)} (S_T - K)_+ \mid S_t = S, v_t = v \right] \quad (2.14)$$

This result provides the risk-neutral representation of the option price and since I seek a solution of the form

$$C_t = S_t P_1 - K e^{-r(T-t)} P_2$$

P_1 and P_2 are represented, respectively, by (2.8) and (2.6).

To do that, I made the following implicit assumption: there exists an analytical expression for the CHF of $X_t = \log(S_t)$. $\{X_t\}_{t \geq 0}$ under the measure $\mathbb{Q}$, can be described with the following dynamics

$$dX_t = \left(r - \frac{1}{2} v_t \right) dt + \sqrt{v_t} dW_t^{S, \mathbb{Q}} \quad (2.15)$$

For notational simplicity, let $k = \log(K)$. To find the CHF, I define

$$\phi(t, x, v) = \mathbb{E}^{\mathbb{Q}} [g(X_T, v_T) \mid X_t = x, v_t = v]$$

where $\phi \in C^{1,2}([0, T] \times \mathbb{R} \times \mathbb{R}_+)$. Furthermore, let $\{M_t\}_{t \geq 0}$ a stochastic process such that

$$M_t = \phi(t, X_t, v_t) \quad \forall t \in [0, T]$$

Applying the Itô formula, the differential can be expressed as

$$dM_t = \frac{\partial \phi}{\partial t} + \frac{\partial \phi}{\partial x} dX_t + \frac{\partial \phi}{\partial v} dv_t + \frac{1}{2} \frac{\partial^2 \phi}{\partial x^2} (dX_t)^2 + \frac{1}{2} \frac{\partial^2 \phi}{\partial v^2} (dv_t)^2 + \frac{\partial^2 \phi}{\partial x \partial v} dX_t dv_t$$

Substituting the dynamics of $\{X_t\}_{t \geq 0}$ and $\{v_t\}_{t \geq 0}$ and rearranging a bit the expression, I obtain

$$dM_t = \left[\frac{\partial \phi}{\partial t} + \left(r - \frac{1}{2} v_t \right) \frac{\partial \phi}{\partial x} + \kappa^{\mathbb{Q}} (\bar{v}^{\mathbb{Q}} - v_t) \frac{\partial \phi}{\partial v} + \frac{1}{2} v_t \frac{\partial^2 \phi}{\partial x^2} + \frac{1}{2} \sigma^2 v_t \frac{\partial^2 \phi}{\partial v^2} + \rho \sigma v_t \frac{\partial^2 \phi}{\partial x \partial v} \right] dt + \sqrt{v_t} \frac{\partial \phi}{\partial x} dW_t^{S, \mathbb{Q}} + \sigma \sqrt{v_t} \frac{\partial \phi}{\partial v} dW_t^{v, \mathbb{Q}}$$

By construction M_t is a martingale, therefore

$$\frac{\partial \phi}{\partial t} + \left(r - \frac{1}{2} v \right) \frac{\partial \phi}{\partial x} + \kappa^{\mathbb{Q}} (\bar{v}^{\mathbb{Q}} - v) \frac{\partial \phi}{\partial v} + \frac{1}{2} v \frac{\partial^2 \phi}{\partial x^2} + \frac{1}{2} \sigma^2 v \frac{\partial^2 \phi}{\partial v^2} + \rho \sigma v \frac{\partial^2 \phi}{\partial x \partial v} = 0$$

I can proceed similarly to what I have done to derive the expression (2.14), identifying the following infinitesimal generator of the process

$$\mathcal{A}\phi = \left(r - \frac{1}{2} v \right) \frac{\partial \phi}{\partial x} + \kappa^{\mathbb{Q}} (\bar{v}^{\mathbb{Q}} - v) \frac{\partial \phi}{\partial v} + \frac{1}{2} v \frac{\partial^2 \phi}{\partial x^2} + \frac{1}{2} \sigma^2 v \frac{\partial^2 \phi}{\partial v^2} + \rho \sigma v \frac{\partial^2 \phi}{\partial x \partial v}$$

Moreover, since I want to achieve the CHF, the PDE has the following terminal condition

$$\phi(T, x, v) = g(x, v) = e^{iux}$$

In this case, I obtained a Kolmogorov backward PDE :

$$\begin{cases} \frac{\partial \phi}{\partial t} + \mathcal{A}\phi = 0 & \text{on } [0, T] \times \mathbb{R} \times \mathbb{R}_+, \\ \phi(T, x, v; u) = e^{iux} \end{cases}$$

Following Heston (1993), I provide the following expression for the solution

$$\phi(t, x, v; u) = \exp [A(T - t; u) + B(T - t; u) v + iux] \quad (2.16)$$

From this point onward, let $\tau = T - t$. Now, I compute the partial derivatives of (2.16)

$$\begin{aligned} \frac{\partial \phi}{\partial t} &= - \left(\frac{\partial A}{\partial \tau} + \frac{\partial B}{\partial \tau} v \right) \phi, \quad \frac{\partial \phi}{\partial x} = iu\phi, \quad \frac{\partial^2 \phi}{\partial x^2} = -u^2 \phi, \\ \frac{\partial \phi}{\partial v} &= B\phi, \quad \frac{\partial^2 \phi}{\partial v^2} = B^2 \phi, \quad \frac{\partial^2 \phi}{\partial x \partial v} = iuB\phi \end{aligned}$$

Substituting them into the PDE and dividing by ϕ , which is non-zero since it is an exponential function, I have

$$-\left(\frac{\partial A}{\partial \tau} + \frac{\partial B}{\partial \tau}v\right) + \left(r - \frac{1}{2}v\right)iu + \kappa^{\mathbb{Q}}(\bar{v}^{\mathbb{Q}} - v)B - \frac{1}{2}vu^2 + \frac{1}{2}\sigma^2vB^2 + \rho\sigma viuB = 0$$

subject to $A(0; u) = 0$ and $B(0; u) = 0$.

Collecting the terms in which appears v and defining

$$D = -\frac{\partial A}{\partial \tau} + riu + \kappa^{\mathbb{Q}}\bar{v}^{\mathbb{Q}}B, \quad E = -\frac{\partial B}{\partial \tau} - \frac{1}{2}iu - \kappa^{\mathbb{Q}}B - \frac{1}{2}u^2 + \frac{1}{2}\sigma^2B^2 + \rho\sigma viuB$$

I can rewrite the PDE in the form

$$D + Ev = 0 \tag{2.17}$$

Since (2.17) must hold for any v , then the following conditions must be satisfied

$$D = 0, \quad E = 0$$

In particular, the first term is equal to zero if and only if

$$\frac{\partial A}{\partial \tau} = riu + \kappa^{\mathbb{Q}}\bar{v}^{\mathbb{Q}}B \tag{2.18}$$

while the second one is equal to zero when

$$\frac{\partial B}{\partial \tau} = -\frac{1}{2}(u^2 + iu) + (\rho\sigma iu - \kappa^{\mathbb{Q}})B + \frac{1}{2}\sigma^2B^2 \tag{2.19}$$

This is Heston's system of ODEs. The expression (2.19) is a Riccati equation, while (2.18) is a linear ODE. Therefore, once $B(\tau; u)$ has been obtained, $A(\tau; u)$ can be derived.

To solve (2.19), I first compute the root of the discriminant

$$d(u) = \sqrt{(\rho\sigma iu - \kappa^{\mathbb{Q}})^2 + (u^2 + iu)\sigma^2}$$

Therefore, the two solutions are represented by

$$B_1 = \frac{(\kappa^{\mathbb{Q}} - \rho\sigma iu) - d(u)}{\sigma^2}, \quad B_2 = \frac{(\kappa^{\mathbb{Q}} - \rho\sigma iu) + d(u)}{\sigma^2}$$

Then, I can factor the polynomial in the RHS of the starting equation, achieving

$$\frac{\partial B}{\partial \tau} = \frac{1}{2}\sigma^2(B - B_2)(B - B_1)$$

Consider from now on $\frac{\partial B}{\partial \tau} = \frac{dB}{d\tau}$.

Using a shortcut, it is possible to isolate the two variables appearing on the LHS; in particular, I temporarily assume that $(B - B_2)(B - B_1) \neq 0$. I divide and multiply both sides by $(B - B_2)(B - B_1)$ and $d\tau$, achieving

$$\frac{dB}{(B - B_2)(B - B_1)} = \frac{1}{2}\sigma^2 d\tau$$

Integrating both sides, I obtain

$$\int_0^{B(\tau;u)} \frac{1}{(z - B_2)(z - B_1)} dz = \frac{1}{2}\sigma^2 \int_0^\tau ds$$

I use on the LHS the partial fraction decomposition, then as a result

$$\frac{1}{B_2 - B_1} \int_0^{B(\tau;u)} \left(\frac{1}{(z - B_2)} - \frac{1}{(z - B_1)} \right) dz = \frac{1}{2}\sigma^2 \int_0^\tau ds$$

Solving both integrals, I achieve the following expression

$$\frac{1}{B_2 - B_1} \log \left(\frac{B - B_2}{B - B_1} \frac{B_1}{B_2} \right) = \frac{1}{2}\sigma^2 \tau$$

Knowing that $B_2 - B_1 = \frac{2d(u)}{\sigma^2}$, then

$$\log \left(\frac{B - B_2}{B - B_1} \frac{B_1}{B_2} \right) = d(u) \tau$$

For notational simplicity, let $g = \frac{B_2}{B_1}$, therefore

$$\frac{B - B_2}{B - B_1} \frac{1}{g} = e^{d(u)\tau}$$

Finally, solving for B , I obtain the desired solution

$$B(\tau; u) = B_2 \frac{1 - e^{d(u)\tau}}{1 - ge^{d(u)\tau}} = \frac{\kappa^{\mathbb{Q}} - \rho\sigma iu + d(u)}{\sigma^2} \frac{1 - e^{d(u)\tau}}{1 - ge^{d(u)\tau}}$$

To simplify the notation in the subsequent computations, let $d(u) = \delta$.

To find $A(\tau; u)$, I integrate both sides, achieving

$$A(\tau; u) = riu\tau + \kappa^{\mathbb{Q}} \bar{v}^{\mathbb{Q}} \int_0^\tau B(s; u) ds = riu\tau + \kappa^{\mathbb{Q}} \bar{v}^{\mathbb{Q}} \int_0^\tau \frac{\kappa^{\mathbb{Q}} - \rho\sigma iu + \delta}{\sigma^2} \frac{1 - e^{\delta s}}{1 - ge^{\delta s}} ds$$

Applying the following change of variable $z = e^{\delta s}$, which gives $\frac{dz}{ds} = \delta e^{\delta s} = \delta z$, I can rewrite the expression as

$$\int_0^\tau B_2 \frac{1 - e^{\delta s}}{1 - ge^{\delta s}} ds = \frac{B_2}{\delta} \int_1^{e^{\delta\tau}} \frac{1 - z}{z(1 - gz)} dz = \frac{B_2}{\delta} \left[\log(z) - \frac{g-1}{g} \log(1 - gz) \right]_1^{e^{\delta\tau}}$$

Evaluating the resulting integral, I obtain

$$A(\tau; u) = riu\tau + \kappa^{\mathbb{Q}} \bar{v}^{\mathbb{Q}} B_2 \left[\tau - \frac{g-1}{gd(u)} \log \left(\frac{1 - ge^{d(u)\tau}}{1-g} \right) \right]$$

Rearranging a bit the expression, I can recover the one proposed by Heston (1993)

$$A(\tau; u) = riu\tau + \frac{\kappa^{\mathbb{Q}} \bar{v}^{\mathbb{Q}}}{\sigma^2} \left[\left(\kappa^{\mathbb{Q}} - \rho\sigma iu + d(u) \right) \tau - 2 \log \left(\frac{1 - ge^{d(u)\tau}}{1-g} \right) \right]$$

Finally, all the components required to define Heston's characteristic function for the log-price of the underlying asset have been derived, therefore, to price European options under the Heston model, I can use the formula (2.9).

2.3 Ornstein–Uhlenbeck driven Stochastic Volatility Model

The Ornstein–Uhlenbeck driven stochastic volatility model (or Ornstein–Uhlenbeck model) was developed some years before the Heston one, but it took longer to be studied extensively in the literature.

Scott (1987) and Wiggins (1987) developed this model for the first time. The former, assumed that the volatility process follows a mean-reverting Ornstein–Uhlenbeck (OU) process. Instead, the latter introduce this process to characterize the log-volatility. Following the model definition proposed by Scott (1987), the subsequent literature includes the following contributions:

- Stein and Stein (1991), who, assuming that the underlying asset price process and its volatility were uncorrelated, derived the probability density function of the log-returns
- Schöbel and Zhu (1999), who derived an analytical expression for the characteristic function of the log-returns

2.3.1 Model Dynamics

The OU model is a continuous-time model defined on the filtered probability space $(\Omega, \mathcal{F}, \mathbb{F}, \mathbb{P})$, which satisfies the usual assumptions.

In this framework, there are two price processes:

- *Non-risky asset*: defined as in (2.1)
- *Underlying asset* described by the following dynamics

$$dS_t = S_t \left(\mu dt + v_t dW_t^S \right), \quad S_0 \in (0, \infty)$$

The parameters governing the underlying dynamics are the same of those related to the BS model, except for v_t , which represents a random variable of the corresponding

volatility process.

In the OU model, the process $\{v_t\}_{t \geq 0}$ represents the volatility of the underlying asset and it is defined as a mean-reverting OU process. Before examining its definition, it is useful to introduce a simplified version, in this way, its properties can be studied more easily.

Definition 2.3.1 (OU process). *Let $\{X_t\}_{t \geq 0}$ be a stochastic process and let the following SDE:*

$$\begin{cases} dX_t = -\mu X_t dt + \sigma dW_t \\ X_0 = x \end{cases} \quad (2.20)$$

The unique solution of (2.20) is called OU process and it is represented by

$$X_t = xe^{-\mu t} + \sigma e^{-\mu t} \int_0^t e^{\mu s} dW_s$$

Proposition 2.3.1 (Normality property). *Let $\{X_t\}_{t \geq 0}$ be a OU process, then $\{X_t\}_{t \geq 0}$ conditioned to its starting value $X_0 = x \in \mathbb{R}$ is a gaussian process such that*

$$X_t | X_0 = x \sim \mathcal{N} \left(xe^{-\mu t}, \frac{\sigma^2}{2\mu} (1 - e^{-2\mu t}) \right) \quad \forall t \in (0, T]$$

However, this process is not able to characterize the volatility because it fails to capture its *mean-reverting* behavior. For this reason, I introduce the Definition 2.3.2.

Definition 2.3.2 (Mean-reverting OU process). *Let $\{v_t\}_{t \geq 0}$ be a stochastic process and let the following SDE:*

$$\begin{cases} dv_t = \kappa (\bar{v} - v_t) dt + \sigma dW_t \\ v_0 = v \end{cases} \quad (2.21)$$

The unique solution of (2.21) is called mean-reverting OU process and it is represented by

$$v_t = \bar{v} + (v - \bar{v}) e^{-\kappa t} + \sigma e^{-\kappa t} \int_0^t e^{\kappa s} dW_s$$

Under the Definition 2.3.2, the Proposition 2.3.1 can be generalized, indeed

$$v_t | v_0 = v \sim \mathcal{N} \left(\bar{v} + (v - \bar{v}) e^{-\kappa t}, \frac{\sigma^2}{2\kappa} (1 - e^{-2\kappa t}) \right) \quad \forall t \in (0, T]$$

The parameters governing the dynamics of the mean-reverting OU process have the same meaning as those defined for the Heston model, with the exception of v_0 , which, somewhat imprecisely, could be interpreted as the initial volatility of the process. To explain why this interpretation is somewhat imprecise, it is necessary to examine whether the process is able to capture all four stylized facts presented in Section 2.1.2. Observing the structure of the process and recalling that also in this model

$$dW_t^S dW_t^v = \rho dt$$

the first three facts are satisfied, but it is still necessary to verify the last one.

I need to compute $\mathbb{P}(v_T \leq 0 \mid v_t = v)$, using the Proposition 2.3.1, as a result

$$\mathbb{P}(v_T \leq 0 \mid v_t = v) = N \left(\frac{-\sqrt{2\kappa} [\bar{v} + (v - \bar{v}) e^{-\kappa(T-t)}]}{\sigma \sqrt{1 - e^{-2\kappa(T-t)}}} \right)$$

Then I can report the following result.

Proposition 2.3.2. *Let $\{v_t\}_{t \geq 0}$ be a mean-reverting OU process, then*

$$\lim_{T-t \rightarrow +\infty} \mathbb{P}(v_T \leq 0 \mid v_t = v) = N \left(-\frac{\bar{v} \sqrt{2\kappa}}{\sigma} \right)$$

If

$$\frac{\bar{v} \sqrt{2\kappa}}{\sigma} \longrightarrow +\infty, \quad (2.22)$$

then

$$\lim_{T-t \rightarrow +\infty} \mathbb{P}(v_T \leq 0 \mid v_t = v) = 0$$

As can be seen, this theorem imposes a condition that is not satisfied by all parameter sets, indeed, as discussed by Shöbel and Zhu (1999), it is possible to consider only the parameterizations such that $\mathbb{P}(v_T \leq 0 \mid v_t = v)$ tends to zero. However, as reported by Brignone and Sgarra (2026), during the calibration, some optimal solutions could not verify (2.22), therefore, in general, it holds that

$$\mathbb{P}(v_T \leq 0 \mid v_t = v) \neq 0$$

It is possible to interpret the mean-reverting OU process as the dynamics of a latent variable, which drives the true volatility process, defined by

$$v_t^{true} = \sqrt{V_t}$$

where $\{V_t\}_{t \geq 0}$ is the variance process characterized by $V_t = v_t^2$. In other words, the process $\{v_t^{true}\}_{t \geq 0}$ and $\{v_t\}_{t \geq 0}$ take the same absolute values but the former still always positive.

To find the corresponding variance process, I apply the Itô formula defining as state variable $z = v_t$:

$$\frac{\partial V_t}{\partial t} = 0, \quad \frac{\partial V_t}{\partial z} = 2z, \quad \frac{\partial^2 V_t}{\partial z^2} = 2$$

as a result

$$dV_t = 2v_t dv_t + (dv_t)^2 = 2\kappa \left(\frac{\sigma^2}{2\kappa} + \bar{v} \sqrt{V_t} - V_t \right) dt + 2\sigma \sqrt{V_t} dW_t^v$$

Finally, the dynamics governing the OU model are

$$\begin{cases} dS_t = S_t (\mu dt + v_t dW_t^S) \\ dv_t = \kappa (\bar{v} - v_t) dt + \sigma dW_t^v \\ dV_t = 2\kappa \left(\frac{\sigma^2}{2\kappa} + \bar{v}\sqrt{V_t} - V_t \right) dt + 2\sigma\sqrt{V_t} dW_t^v \\ dW_t^S dW_t^v = \rho dt \end{cases}$$

Examining the results obtained for $\{v_t\}_{t \geq 0}$, in order to correctly characterize the model, I impose the following domains

$$\kappa, \bar{v}, \sigma > 0, \quad \rho \in [-1, 1], \quad v_0 \in \mathbb{R}$$

Similarly to the Heston model, it can be shown that the OU market is arbitrage-free but incomplete.

Considering again

$$\lambda_t^S = \frac{\mu - r}{v_t}, \quad \lambda_t^v = \frac{\lambda}{\sigma} v_t$$

under the probability measure $\mathbb{Q}$, the OU model takes the following structure

$$\begin{cases} dS_t = S_t (r dt + v_t dW_t^S) \\ dv_t = \kappa^{\mathbb{Q}} (\bar{v}^{\mathbb{Q}} - v_t) dt + \sigma dW_t^v \\ dV_t = 2\kappa^{\mathbb{Q}} \left(\frac{\sigma^2}{2\kappa^{\mathbb{Q}}} + \bar{v}^{\mathbb{Q}}\sqrt{V_t} - V_t \right) dt + 2\sigma\sqrt{V_t} dW_t^v \\ dW_t^{S,\mathbb{Q}} dW_t^{v,\mathbb{Q}} = \rho dt \end{cases}$$

where $\kappa^{\mathbb{Q}}$ and $\bar{v}^{\mathbb{Q}}$ are defined as in the Heston model. Moreover, the process $\{\mathbf{Y}_t\}_{t \geq 0}$, where $\mathbf{Y}_t = (S_t, v_t)$, is a Markov process.

2.3.2 Characteristic Function

Similarly to the Heston model, it is possible to derive a PDE involving the processes $\{S_t\}_{t \geq 0}$ and $\{v_t\}_{t \geq 0}$ and, applying the Theorem 2.2.1, to show that there exists a deterministic function U such that

$$U(t, S, v) = \mathbb{E}^{\mathbb{Q}} \left[e^{-r(T-t)} (S_T - K)_+ \mid S_t = S, v_t = v \right]$$

This means that, also in this model, I can price the European call options applying the formula (2.9) and, consequently, P_1 and P_2 are represented, respectively, by (2.8) and (2.6). It remains to show that there exists an analytical expression for the CHF of

$X_t = \log(S_t)$. The process $\{X_t\}_{t \geq 0}$ is described by

$$dX_t = \left(r - \frac{1}{2}v_t^2\right) dt + v_t dW_t^{S, \mathbb{Q}}$$

I want to emphasize that, under this model, the log-returns have a different structure compared to the one defined under the Heston model. This is due to the presence of the quadratic term v_t^2 .

Defining the same objects characterized in the Heston model, I obtain the following PDE

$$\frac{\partial \phi}{\partial t} + \left(r - \frac{1}{2}v^2\right) \frac{\partial \phi}{\partial x} + \kappa^{\mathbb{Q}} (\bar{v}^{\mathbb{Q}} - v) \frac{\partial \phi}{\partial v} + \frac{1}{2}v^2 \frac{\partial^2 \phi}{\partial x^2} + \frac{1}{2}\sigma^2 \frac{\partial^2 \phi}{\partial v^2} + \rho\sigma v \frac{\partial^2 \phi}{\partial x \partial v} = 0$$

By directly modeling volatility with a mean-reverting OU process, the PDE has a quadratic dependence on v term, which represents the main difference with the Heston model. For this reason, following Schöbel and Zhu (1999), I propose the following ansatz to find the analytical expression of the CHF

$$\phi(t, x, v; u) = \exp \left(A(T - t; u) + B(T - t; u) v + C(T - t; u) v^2 + iux \right)$$

Following the steps explained in Section 2.2.2, it is possible to derive the CHF of the OU model. Given the particularly cumbersome nature of the resulting ODEs, although they are not conceptually complex, I have chosen to report only the definitions of the various components of the characteristic function derived by Schöbel and Zhu (1999).

They define the following auxiliary quantities

$$\begin{aligned} \tau &= T - t, \quad s_1 = -\frac{1}{2}(1 + iu)^2(1 - \rho^2) + \frac{1}{2}(1 + iu) \left(1 - \frac{2\kappa^{\mathbb{Q}}\rho}{\sigma}\right), \\ s_2 &= \frac{(1 + iu)\kappa^{\mathbb{Q}}\bar{v}^{\mathbb{Q}}\rho}{\sigma}, \quad s_3 = \frac{(1 + iu)\rho}{2\sigma} \\ \gamma_1 &= \sqrt{(\kappa^{\mathbb{Q}})^2 + 2\sigma^2 s_1}, \quad \gamma_2 = \frac{1}{\gamma_1} (\kappa^{\mathbb{Q}} - 2\sigma^2 s_3), \quad \gamma_3 = (\kappa^{\mathbb{Q}})^2 \bar{v}^{\mathbb{Q}} - s_2 \sigma^2 \end{aligned}$$

and, finally, the following terms of the CHF

$$\begin{aligned} A(\tau; u) &= -\frac{1}{2} \log [\cosh(\gamma_1 \tau) + \gamma_2 \sinh(\gamma_1 \tau)] + \frac{1}{2} \kappa^{\mathbb{Q}} \tau + \\ &\quad \frac{\left[(\kappa^{\mathbb{Q}})^2 (\bar{v}^{\mathbb{Q}})^2 \gamma_1^2 - \gamma_3^2 \right]}{2\sigma^2 \gamma_1^3} \left(\frac{\sinh[\gamma_1 \tau]}{\cosh(\gamma_1 \tau) + \gamma_2 \sinh(\gamma_1 \tau)} - \gamma_1 \tau \right) + \\ &\quad \frac{(\kappa^{\mathbb{Q}} \bar{v}^{\mathbb{Q}} \gamma_1 - \gamma_2 \gamma_3) \gamma_3}{\sigma^2 \gamma_1^3} \left(\frac{\cosh(\gamma_1 \tau) - 1}{\cosh(\gamma_1 \tau) + \gamma_2 \sinh(\gamma_1 \tau)} \right) \end{aligned}$$

$$B(\tau; u) = \frac{1}{\sigma^2 \gamma_1} \left\{ \frac{(\kappa^{\mathbb{Q}} \bar{v}^{\mathbb{Q}} \gamma_1 - \gamma_2 \gamma_3) + \gamma_3 [\sinh(\gamma_1 \tau) + \gamma_2 \cosh(\gamma_1 \tau)]}{\cosh(\gamma_1 \tau) + \gamma_2 \sinh(\gamma_1 \tau)} - \kappa^{\mathbb{Q}} \bar{v}^{\mathbb{Q}} \gamma_1 \right\}$$

$$C(\tau; u) = \frac{1}{2\sigma^2} \left[\kappa^{\mathbb{Q}} - \gamma_1 \frac{\sinh(\gamma_1 \tau) + \gamma_2 \cosh(\gamma_1 \tau)}{\cosh(\gamma_1 \tau) + \gamma_2 \sinh(\gamma_1 \tau)} \right]$$

2.4 Comparison between OU and Heston Model

I created this brief section to make a comparison between the two stochastic volatility models introduced before and to show the reasons why I decided to analyze them and in which sense they are linked.

From now on, I consider both models under the probability measure $\mathbb{Q}$ but, for notational simplicity, I omit the corresponding superscript.

The CIR process, which characterizes the variance in the Heston model, is not an arbitrary choice, it arises from an assumption made by Heston (1993). He assumed that the volatility follows an OU process, presenting the results contained in Proposition 2.4.1.

Proposition 2.4.1. *Let $\{v_t\}_{t \geq 0}$ be a volatility process. If $\{v_t\}_{t \geq 0}$ follows an OU process, which dynamics is defined by*

$$dv_t = -\mu v_t dt + \sigma dW_t^v,$$

Then, the corresponding variance $\{V_t\}_{t \geq 0}$ follows the CIR process

$$dV_t = (\sigma^2 - 2\mu V_t) dt + 2\sigma \sqrt{V_t} dW_t^v$$

When volatility is characterized by a mean-reverting OU process, this result is no longer valid, indeed, to compare the two processes, $\bar{v}^{OU}$ must be equal to zero. Under this condition, as described by Shöbel and Zhu (1999) and Brignone and Sgarra (2026), it is possible to relate the variance process used in the OU model to the CIR process defined by Heston, indeed

$$\kappa_H = 2\kappa_{OU}, \quad \bar{v}_H = \frac{\sigma_{OU}^2}{2\kappa_{OU}}, \quad \sigma_H = 2\sigma_{OU}$$

This shows that the parameters defined in both models have the same meaning, except for

- $\bar{v}$: in the Heston model, it represents the long-term variance, whereas in the OU model, it represents the long-term volatility
- v_0 : in the Heston model, it represents the initial variance, whereas in the OU model, it represents the initial value of the latent variable associated with the true volatility process

I have chosen to leave the most significant difference for last, as I believe it represents the core of this work.

Let $\mathcal{M}(\boldsymbol{\theta})$ be a stochastic volatility model with parameter vector $\boldsymbol{\theta} \in \mathbb{R}^m$, in which the state variables are $\mathbf{Y}_t = (X_t, v_t) \in \mathbb{R} \times \mathbb{R}$, where $\{v_t\}_{t \geq 0}$ characterize the variance or the volatility process. Furthermore, let $D = \mathbb{R} \times \mathbb{R}$ and $\mathbf{u} = (u_x, u_v) \in \mathcal{U}$, where $\mathcal{U} = \mathbb{R}^2$.

Duffie (2003) introduced the definition of affine process; I present a simplified version of it.

Definition 2.4.1 (Exponential–affine process). *Let $\mathbf{Y} = \{\mathbf{Y}_t\}_{t \geq 0}$ be a markov process. $\mathbf{Y}$ is called exponential–affine, if $\forall t \in [0, T]$, $\phi(t, x, v; u)$ shows an esponential–affine dependence from $\mathbf{y}$, i.e.*

$$\forall (t, \mathbf{u}) \in [0, T] \times \mathcal{U} \quad \exists a(t, \mathbf{u}) \in \mathbb{C}, b(t, \mathbf{u}) = (b^x(t, \mathbf{u}), b^v(t, \mathbf{u})) \in \mathbb{C} \times \mathbb{C} :$$

$$\phi(t, x, v; u) = \mathbb{E} \left[e^{i \langle \mathbf{u}, \mathbf{Y}_t \rangle} \mid \mathbf{Y}_0 = \mathbf{y} \right] = \exp [a(t, \mathbf{u}) + b^x(t, \mathbf{u})x + b^v(t, \mathbf{u})v] \quad \forall \mathbf{y} \in D$$

Since $\mathcal{M}(\boldsymbol{\theta})$ is defined by the state process $\mathbf{Y}$, I will use the terms *affine model* and *affine process* interchangeably.

Definition 2.4.1 shows that, according to Duffie (2003), a model is affine if it has an exponential–affine dependence on its state variables.

Definition 2.4.2 (Affine–dependence). *Let $f : D \rightarrow \mathbb{C}$. f is called affine in the state $\mathbf{y} = (x, v)$ if there exist $a \in \mathbb{C}$ and $b = (b^x, b^v) \in \mathbb{C}^2$, independent from the state $\mathbf{y}$, such that*

$$f(\mathbf{y}) = a + \langle b, \mathbf{y} \rangle \quad \forall \mathbf{y} \in D$$

Considering the following CHF of log–returns

$$\phi(T - t, x, v; u) = \mathbb{E} \left[e^{iuX_T} \mid \mathbf{Y}_t = \mathbf{y} \right]$$

I can further simplify the statement in Definition 2.4.1, indeed $\mathbf{u} = (u, 0)$, which means that an affine model must have a characteristic function of the following form:

$$\phi(T - t, x, v; u) = \exp [a(t, u) + b^x(t, u)x + b^v(t, u)v] \quad \forall (x, v) \in D$$

From the form of the CHF, it can be seen that the Heston model is affine, whereas the OU model is not. To emphasize this fact, Brignone and Sgarra (2026) refer to the OU model as *quadratic*, due to its dependence on the term v^2 . This represents a significant difference between the models under consideration, indeed, characterizing volatility instead of the variance results in the loss of the affine property.

As reported by Brignone and Sgarra (2026) there is evidence that *non–affine models* are able to fit market data better than their affine counterparts. One of the aims of this work is to investigate whether such evidence actually exists.

Chapter 3

Calibration Methodology

In this chapter, I introduce all the methods used to calibrate the Heston and the OU models, using as a reference the procedure proposed by Cui et al. (2017). I want to emphasize that, in some cases, to address the problem, I adopt different tools from those presented in the original procedure.

Moreover, from now on, I consider the vectors defined below as column vectors.

3.1 Calibration Problem

In Chapter 1, I have already described what it means calibrate a model, however, to be more rigorous, I propose the following definition.

Definition 3.1.1 (Calibration). *Let $\mathcal{M}(\boldsymbol{\theta})$ with $\boldsymbol{\theta} \in \boldsymbol{\Theta} \subseteq \mathbb{R}^m$ a model. Let also $\mathbf{O}, \mathbf{M}(\boldsymbol{\theta}) \in \mathbb{R}^n$, respectively, the observed and the theoretical values given by $\mathcal{M}(\boldsymbol{\theta})$. Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be the objective function and $\mathbf{r}(\boldsymbol{\theta}) = \mathbf{M}(\boldsymbol{\theta}) - \mathbf{O} \in \mathbb{R}^n$ the residual vector. The calibration of $\mathcal{M}(\boldsymbol{\theta})$ is the procedure that leads to determining a parameter vector $\hat{\boldsymbol{\theta}} \in \boldsymbol{\Theta}$ such that*

$$\hat{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} f(\mathbf{r}(\boldsymbol{\theta}))$$

In the context of stochastic volatility models, in particular those previously mentioned, market prices and implied volatilities represent the observed values and, consequently, the theoretical ones.

To clarify the meaning of the expression *implied volatility*, I consider again the BS model; indeed, in it, the volatility σ is not directly observable, however, if I focus on the case in which S, K, T and r are given, then the BS pricing formula can be seen as a function depending only on σ , i.e.

$$\sigma \mapsto C^{BS}(S, K, T, r, \sigma)$$

Such a function is strictly increasing since

$$\frac{\partial C^{BS}}{\partial \sigma} = SN(d_1) \sqrt{T}$$

and it establishes a one-to-one relationship between price and volatility. Therefore, if $C^{BS} \in (0, S)$ is also available, I can recover the volatility that generated it; this value is called *implied volatility*.

The components of $\mathbf{O}$ are represented by the market data related to the quoted European options, with different strikes and maturities. In particular, the options in question are those written on indexes.

Following the procedure described by Cui et al. (2017), $\mathbf{O}$ is composed of $C^*(K_i, T_i)$, which represents the market price of the European call option with strike K_i and maturity T_i .

For notational simplicity, let

$$\mathbf{O} = \mathbf{C}^*(\mathbf{K}, \mathbf{T}) \in \mathbb{R}^n$$

Respectively, the elements of $\mathbf{M}(\boldsymbol{\theta})$ are the theoretical prices, which are denoted as $C(\boldsymbol{\theta}; K_i, T_i)$, then

$$\mathbf{M}(\boldsymbol{\theta}) = \mathbf{C}(\boldsymbol{\theta}; \mathbf{K}, \mathbf{T}) \in \mathbb{R}^n$$

As a consequence, I define the the residual vector as

$$\mathbf{r}(\boldsymbol{\theta}) = \mathbf{C}(\boldsymbol{\theta}; \mathbf{K}, \mathbf{T}) - \mathbf{C}^*(\mathbf{K}, \mathbf{T})$$

Using the notation introduced in Chapter 2, for both models the parameter vector is $\boldsymbol{\theta} = (\kappa, \bar{v}, \sigma, \rho, v_0)^\top$ and its components are those defined under the risk-neutral probability measure $\mathbb{Q}$.

It is easy to see that $m = 5$ and that the problem is overdetermined, since, typically, $n \gg m$. Cui et al. (2017) treat the calibration as an inverse problem in the nonlinear least squares form, thereby defining

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^5} f(\boldsymbol{\theta}) \tag{3.1}$$

with the following objective function

$$f(\boldsymbol{\theta}) = \frac{1}{2} \|\mathbf{r}(\boldsymbol{\theta})\|^2 = \frac{1}{2} \mathbf{r}(\boldsymbol{\theta})^\top \mathbf{r}(\boldsymbol{\theta}) \tag{3.2}$$

The procedure involves obtaining $\mathbf{C}^*(\mathbf{K}, \mathbf{T})$ from the corresponding implied volatilities, which compose the so-called *volatility surface*.

Definition 3.1.2 (Volatility surface). *Let $\sigma_{\text{imp}} : \mathcal{D} \subseteq \mathbb{R}_+^2 \rightarrow \mathbb{R}_+$ be the implied volatility function. The volatility surface is given by*

$$\mathcal{S} = \{(K, T, \sigma) \in \mathbb{R}_+^3 : (K, T) \in \mathcal{D}, \sigma = \sigma_{\text{imp}}(K, T)\}$$

To calibrate a model, in addition to market prices, I need $\mathbf{S}, \mathbf{K}, \mathbf{T}$ and r . It is

possible to fix

$$S_i = 1, \quad M_i = \frac{K_i}{S_i} \quad \forall i = 1, \dots, n$$

where M represents the moneyness of the option and has the following meanings

- $M > 1$: the option is in-the-money (ITM)
- $M = 1$: the option is at-the-money (ATM)
- $M < 1$: the option is out-the-money (OTM)

To explain why it is possible to apply this normalization, I present Proposition 3.1.1.

Proposition 3.1.1 (Black–Scholes Price Homogeneity). *Let $C^{BS}(S, K, T, r, \sigma)$ be the price of a European option computed under the BS model. Then*

$$C^{BS}(S, K, T, r, \sigma) = S C^{BS}\left(1, \frac{K}{S}, T, r, \sigma\right)$$

Finally, to compute the theoretical prices, I use the formula (2.9), but considering the following quantites

$$X_t = \log\left(\frac{S_t}{S_0}\right), \quad k = \log\left(\frac{K}{S_0}\right)$$

Moreover, defining $F := S_0 e^{rT}$, it is possible to notice that $\phi(\boldsymbol{\theta}; -i, T) = \frac{F}{S_0}$, obtaining

$$\begin{aligned} C(\boldsymbol{\theta}; K, T) = & \frac{1}{2} \left(S_0 - K e^{-rT} \right) + \frac{e^{-rT}}{\pi} \left[S_0 \int_0^\infty \Re \left(\frac{e^{-iuk}}{iu} \phi(\boldsymbol{\theta}; u - i, T) \right) du \right. \\ & \left. - K \int_0^\infty \Re \left(\frac{e^{-iuk}}{iu} \phi(\boldsymbol{\theta}; u, T) \right) du \right] \end{aligned} \quad (3.3)$$

3.2 Branch Switching of the CPF

Under the definition of $\{X_t\}_{t \geq 0}$ presented in the previous section, I must take into account that the terms appearing in $\phi(\boldsymbol{\theta}; u, T)$ also change.

Considering the Heston model, the CHF becomes

$$\begin{aligned} \phi(\boldsymbol{\theta}; u, T) = & \mathbb{E} \left[e^{iuX_T} \mid v_0 \right] = \\ & \exp \left\{ iu \log \left(\frac{F}{S_0} \right) + \frac{k\bar{v}}{\sigma^2} \left[(\xi + d)T - 2 \log \left(\frac{1 - g_1 e^{dT}}{1 - g_1} \right) \right] + \frac{v_0}{\sigma^2} (\xi + d) \frac{1 - e^{dT}}{1 - g_1 e^{dT}} \right\} \end{aligned} \quad (3.4)$$

where

$$\xi = \kappa - \sigma \rho i u, \quad d = \sqrt{\xi^2 + \sigma^2 (u^2 + iu)}, \quad g_1 = \frac{\xi + d}{\xi - d} \quad (3.5)$$

While, for the OU model, I recover the formulation proposed by Brignone and Sgarra (2026):

$$\phi(\boldsymbol{\theta}; u, T) = \exp \left[iurT + L(\boldsymbol{\theta}; T)v_0^2 + G(\boldsymbol{\theta}; T)v_0 + H(\boldsymbol{\theta}; T) + Q(\boldsymbol{\theta}; T) \right] \quad (3.6)$$

where

$$\begin{aligned}
 \gamma_1 &:= \sqrt{2\sigma^2 \left(\frac{u^2}{2} (1 - \rho^2) + \frac{i u}{2} \left(1 - \frac{2\kappa\rho}{\sigma} \right) \right) + \kappa^2}, \quad \gamma_2 := \frac{1}{\gamma_1} (\kappa - \sigma i u \rho), \\
 \gamma_3 &:= \kappa^2 \bar{v} - \sigma i u \kappa \bar{v} \rho, \quad \zeta_1 := \cosh(\gamma_1 T), \quad \zeta_2 := \sinh(\gamma_1 T) \\
 L(\boldsymbol{\theta}; T) &:= \frac{1}{2\sigma^2} \left(\kappa - \frac{\gamma_1 \zeta_2 + \gamma_1 \gamma_2 \zeta_1}{\zeta_1 + \gamma_2 \zeta_2} \right) - \frac{\rho i u}{2\sigma}, \quad Q(\boldsymbol{\theta}; T) := -\frac{1}{2} \log(\zeta_1 + \gamma_2 \zeta_2), \\
 G(\boldsymbol{\theta}; T) &:= \frac{1}{\sigma^2 \gamma_1} \left(-\bar{v} \kappa \gamma_1 + \frac{\kappa \bar{v} \gamma_1 - \gamma_2 \gamma_3 + \gamma_3 \zeta_2 + \gamma_2 \gamma_3 \zeta_1}{\zeta_1 + \gamma_2 \zeta_2} \right), \\
 H(\boldsymbol{\theta}; T) &:= -\frac{1}{2} \rho i u \sigma T + \frac{\kappa T}{2} + \frac{\kappa^2 \bar{v}^2 \gamma_1^2 - \gamma_2^2}{2\sigma^2 \gamma_1^3} \left(\frac{\zeta_2}{\zeta_1 + \gamma_2 \zeta_2} - \gamma_1 T \right) \\
 &\quad + \frac{(\kappa \bar{v} \gamma_1 - \gamma_2 \gamma_3) \gamma_3}{\sigma^2 \gamma_1^3} \left(\frac{\zeta_1 - 1}{\zeta_1 + \gamma_2 \zeta_2} \right)
 \end{aligned}$$

Of the two, the one presenting the more tractable expression is certainly (3.4), for this reason, I consider it first in the following analysis.

As shown by Kahl and Jackel (2005), the (3.4) is not particularly suitable for calibration algorithms, since it could be subject to numerical instability when it is needed to value the options with moderate and long maturities. This problem is due to the presence of the complex logarithm, for which more than one solution exists. This feature is known in the literature as the *choice of the branch of the multivalued complex logarithm* or the *Little Heston Trap*. To understand how to solve it, I introduce the following definition.

Definition 3.2.1 (Complex number and equivalent representations). *Let $z \in \mathbb{C}$. Then the following representations are equivalent:*

- $z = a + ib$
- $z = R(\cos t + i \sin t)$
- $z = Re^{it}$

where $R = \sqrt{a^2 + b^2}$ represents the modulus of z and $t = \arg(z)$ its phase (or argument).

Considering the first and third representations, I present Proposition 3.2.1.

Proposition 3.2.1 (Non-uniqueness of the phase). *Let $z \in \mathbb{C}$. Then*

$$z = a + ib = Re^{it} = Re^{i(t+2\pi n)}, \quad t \in [-\pi, \pi], n \in \mathbb{Z}$$

This proposition shows that the phase of a complex number is not uniquely determined, indeed its periodicity implies that the complex logarithm of z , defined as

$$\log(z) = \log(R) + i(t + 2\pi n), \quad t \in [-\pi, \pi], n \in \mathbb{Z}$$

is multivalued.

Returning to (3.4), by defining

$$A(T, u) = \alpha (\xi + d) T, \quad B(T, u) = \frac{1}{\sigma^2} (\xi + d) \frac{1 - e^{dT}}{1 - g_1 e^{dT}}$$

$$G(T, u) = \frac{1 - g_1 e^{dT}}{1 - g_1}, \quad \alpha = \frac{k\bar{v}}{\sigma^2}$$

I can rewrite it, obtaining

$$\phi(\boldsymbol{\theta}; u, T) = G(T, u)^{-2\alpha} \exp \left\{ iu \log \frac{F}{S_0} + A(T, u) + B(T, u) v_0 \right\}$$

In this way, the logarithm no longer appears in the formula and, therefore, there is no reason to compute it in the implementations. At least apparently, the source of discontinuity disappears. However, the problem now is the evaluation of $G(T, u)^\alpha$, since it is a *complex power function* (CPF) and it is defined as follows

$$G(T, u)^\alpha = \exp \{ \alpha \log [G(T, u)] \}$$

The problem has simply become less evident; when $G(T, u)$, which represents a spiral on the complex plane, crosses the negative real axis, then the sign of its phase changes from $-\pi$ to π , therefore the one of $G(T, u)^\alpha$ changes from $-\alpha\pi$ to $\alpha\pi$. As explained by Kahl and Jackel (2005), this change in phase creates a discontinuity since

$$e^{i\pi} = e^{-i\pi} \rightarrow \begin{cases} e^{i\alpha\pi} \neq e^{-i\alpha\pi}, & \text{if } \alpha \notin \mathbb{Z} \\ e^{i\alpha\pi} = e^{-i\alpha\pi}, & \text{otherwise} \end{cases}$$

Imposing that α can take only integer values would be excessively restrictive, because it depends from the model's parameters, which are in turn determined by the calibration procedure; however, they solved the problem by ensuring that $G(T, u)^\alpha$ does not cross the negative real axis.

As shown by Albrecher et al.(2007), $G(T, u)^\alpha$, in the implemetations, is discontinuous when the principal value of the complex root d is used, whereas this phenomenon does not occur if the other value is selected instead. Indeed, using it, they obtained

$$\phi(\boldsymbol{\theta}; u, T) = \exp \left\{ iu \log \left(\frac{F}{S_0} \right) + \frac{k\bar{v}}{\sigma^2} \left[(\xi - d) T - 2 \log \left(\frac{1 - g_2 e^{-dT}}{1 - g_2} \right) \right] \right. \\ \left. + \frac{v_0}{\sigma^2} (\xi - d) \frac{1 - e^{-dT}}{1 - g_2 e^{-dT}} \right\}$$

where

$$g_2 = \frac{\xi - d}{\xi + d} = \frac{1}{g_1}$$

However, to perform the calibration, Cui et al. (2017) developed the following expression

for the Heston's CHF, which is also easily differentiable:

$$\phi(\boldsymbol{\theta}; u, T) = \exp \left(iu \log \left(\frac{F}{S_0} \right) - \frac{\kappa \bar{v} \rho i u T}{\sigma} - A + \frac{2\kappa \bar{v}}{\sigma^2} D \right)$$

where ξ and d are defined as in (3.5), while the other terms are

$$\begin{aligned} A &:= \frac{A_1}{A_2}, \\ A_1 &:= \left(u^2 + iu \right) \sinh \left(\frac{dT}{2} \right), \\ A_2 &:= \frac{d}{v_0} \cosh \left(\frac{dT}{2} \right) + \frac{\xi}{v_0} \sinh \left(\frac{dT}{2} \right), \\ B &:= \frac{d \exp \left(\frac{\kappa T}{2} \right)}{v_0 A_2}, \\ D &:= \log(B) \end{aligned}$$

Even for the OU model, it is possible to show that the CHF derived by Shöbel and Zhu (1999) is discontinuous, in particular, as reported by Brignone and Sgarra (2026), the source is the $Q(\boldsymbol{\theta}; T)$ term. To solve the problem while maintaining ease of differentiation, they proposed the following algebraically equivalent expression:

$$Q(\boldsymbol{\theta}; T) := -\frac{1}{2} \gamma_1 T - \frac{1}{2} \log \left(\frac{1}{2} \left(1 + \gamma_2 + e^{-\gamma_1 T} (1 - \gamma_2) \right) \right)$$

As can be seen, there is no longer dependence from $e^{\gamma_1 T}$, indeed, similarly to what was done for Heston model, this term has been replaced with $e^{-\gamma_1 T}$, which does not have the same discontinuity problem.

Furthermore, they showed that the complex root γ_1 can be evaluated using its principal branch (i.e. the positive solution), indeed

$$u^2 \sigma^2 (1 - \rho^2) + iu (\sigma^2 - 2\kappa \rho \sigma) = R e^{it}$$

and, since its real part is always positive

$$u^2 \sigma^2 (1 - \rho^2) > 0$$

then $t \in (-\pi, \pi)$. In other words, the phase does not take the critical values $\{-\pi, \pi\}$, so the CHF is continuous for $u \in [0, \infty)$.

3.3 Analytical Gradient

In the previous section, I showed how to solve the discontinuity problem and how to find easily differentiable expression for the CHFs.

The reason why Cui et al. (2017) require an easily differentiable function is related to the distinguish feature of their calibration method; indeed, unlike calibration procedures based entirely on numerical methods, their approach focuses on directly providing the analytical gradient of the CHF to the algorithm, in particular, their aim is to achieve a higher level of accuracy and reduce the computational time.

According to the definitions presented in Section 3.1, let $\mathbf{J}(\boldsymbol{\theta}) = \nabla \mathbf{r}^\top \in \mathbb{R}^{m \times n}$ the Jacobian matrix, with elements

$$J_{ji} = \frac{\partial r_i(\boldsymbol{\theta})}{\partial \theta_j} = \frac{\partial C(\boldsymbol{\theta}; K_i, T_i)}{\partial \theta_j}$$

and, considering the results obtained by Brignone and Sgarra (2026), I present an extended version of the theorem derived by Cui et al. (2017).

Theorem 3.3.1 (Analytical Price Gradient). *Let $\{S_t\}_{t \geq 0}$ be a price process following the underlying dynamics defined in the Heston or OU model. Then, the analytical gradient of the European call option price with respect to the parameter vector $\boldsymbol{\theta}$ is given by*

$$\begin{aligned} \nabla C(\boldsymbol{\theta}; K, T) = \frac{e^{-rT}}{\pi} & \left[S_0 \int_0^\infty \Re \left(\frac{e^{-iuk}}{iu} \nabla \phi(\boldsymbol{\theta}; u - i, T) \right) du \right. \\ & \left. - K \int_0^\infty \Re \left(\frac{e^{-iuk}}{iu} \nabla \phi(\boldsymbol{\theta}; u, T) \right) du \right] \end{aligned} \quad (3.7)$$

Furthermore, let $\mathbf{h}(u) \in \mathbb{R}^m$ defined as

$$\mathbf{h}(u) = \frac{\partial \log [\phi(\boldsymbol{\theta}; u, T)]}{\partial \boldsymbol{\theta}}$$

Then, it holds that

$$\nabla \phi(\boldsymbol{\theta}; u, T) = \phi(\boldsymbol{\theta}; u, T) \mathbf{h}(u)$$

I want to emphasize the implications of Theorem 3.3.1 on $f(\boldsymbol{\theta})$, proposing the following result.

Proposition 3.3.1 (Objective function derivatives). *Let the assumptions of the Theorem 3.3.1 be satisfied and $f(\boldsymbol{\theta})$ be the objective function defined as (3.2). Then*

$$\nabla f = \mathbf{J} \mathbf{r}(\boldsymbol{\theta})$$

$$\nabla^2 f = \mathbf{J} \mathbf{J}^\top + \sum_{i=1}^n r_i(\boldsymbol{\theta}) \mathbf{H}(r_i)$$

where

$$\mathbf{H}(r_i) = \nabla^2 r_i(\boldsymbol{\theta}) \in \mathbb{R}^{m \times m}$$

It should be noted that the second term of $\nabla^2 f$ become negligible when

- $f(\boldsymbol{\theta})$ is almost linear, therefore $\mathbf{H}(r_i) \rightarrow 0 \quad i = 1, \dots, n$

- the problem under consideration is a *small residual problem*, meaning that, in the neighborhood of the optimal solution θ^* , it holds that $r_i(\theta) \rightarrow 0 \quad i = 1, \dots, n$

Since the derivatives involved in CHF's presented above are not conceptually difficult to derive but their computation involves tedious steps, I computed them using the MATLAB®'s Symbolic Math Toolbox.

3.4 Integration Scheme

Observing $C(\theta; K, T)$ and $\nabla C(\theta; K, T)$, it can be seen that, in total, it is necessary to compute four integrals defined on the $[0, \infty)$; this operation could require a substantial amount of time, indeed it is important to carefully choose which quadrature rule to use and how to apply it. Cui et al. (2017) use the following:

- Trapezoidal rule (TR)
- Gauss–Legendre rule (GL)

Without going into too much detail, as this goes beyond the scope of this discussion; they demonstrate that, to achieve an accuracy level of 10^{-8} for all option prices, TR rule requires almost twice as many nodes N at which evaluate the function as GL one. Therefore, they suggest to use the GL rule fixing $N = 64$.

However, this method is native for definite integrals over an interval $[a, b]$ with $a, b \in \mathbb{R}$, so this leads the authors to truncate the integral by identifying an upper bound $u_{\max}$ before applying the quadrature. Studying European options, especially those with shorter maturities, they found that, as T_i increases, the probability density function of S_T tends to spread out, while its Fourier transform tends to become more concentrated, then they fix $u_{\max} = 200$. It follows that

$$\int_0^{+\infty} f(u) du \approx \int_0^{200} f(u) du$$

Subsequently, this quadrature rescales the integration domain to $[-1, 1]$ and selects nodes $\{u_i\}_{i=1}^N$ that are symmetric around the origin and, simultaneously, it determines the corresponding weights $\{w_i\}_{i=1}^N$. Since the nodes are symmetric, the sequence $\{w_i\}_{i=1}^N$ has $\frac{N}{2}$ different values. Applying this quadrature, they obtain

$$\int_{-1}^1 f(u) du \approx \sum_{i=1}^N f(u_i) w_i$$

In this framework, $f(\cdot)$ represents the integrand functions in (3.3) and (3.7).

This is certainly a valid choice, but the quadrature that I want to propose in this work is the Gauss–Laguerre rule (GLA). I believe that GLA is more suitable because is specifically designed for integrals over semi-infinite intervals of the form

$$\int_0^{+\infty} e^{-u} \psi(u) du$$

As before, defining the number of nodes N , the quadrature identifies $\{u_i\}_{i=1}^N$, in this case, this sequence is composed by the roots of the orthogonal Laguerre polynomial of degree N . Simultaneously, for all u_i , it computes the corresponding w_i , but, unlike GL rule, the sequence $\{w_i\}_{i=1}^N$ is composed by N different values. However, it does not seem to have a significant effect on the computational time. Using this quadrature, the integrals of interest should be more accurate, since the truncation is not necessary.

At this point, I want to show how GLA can be applied to the problem under consideration.

Let $f(\cdot)$ be (3.3) or (3.7) and $\psi(u) = e^u f(u)$, then it holds that

$$\int_0^{+\infty} f(u) du = \int_0^{+\infty} \psi(u) e^{-u} du$$

Therefore, applying this quadrature, I obtain

$$\int_0^{+\infty} f(u) du \approx \sum_{i=1}^N e^{u_i} f(u_i) w_i$$

More precisely, the weights $\{w_i\}_{i=1}^N$ are built taking into account e^{-u} term, which represents the peculiarity of this quadrature.

Moreover, as reported by Brignone and Sgarra (2026), I observed that less time is required if the sequences $\{u_i\}_{i=1}^N$ and $\{w_i\}_{i=1}^N$ are determined before starting the optimization procedure.

Following Cui et al. (2017), to speed up the integration, the GLA quadrature has been vectorized.

3.5 Optimization Algorithms

In this section, I introduce the optimization algorithms used to find the optimal parameter vector. In particular, I describe the following topics:

1. Problem implementations
2. Optimization method

3.5.1 Problem Implementations

In the experiments conducted, I considered two different implementations of the problem (3.1), which proved useful several aspects of the analysis, but, most importantly, they allowed me to examine how the computational time varied.

I first present an equivalent formulation proposed by Brignone and Sgarra (2026), who use the built-in MATLAB function `fsolve`, which solves nonlinear systems in the form

$$\mathbf{r}(\boldsymbol{\theta}) = \mathbf{0}$$

defining an initial guess $\boldsymbol{\theta}_0 \in \Theta$.

The other implemetation that I want to propose is the built-in MATLAB function `lsqnonlin`, which solves nonlinear least squares problems specifying an initial parameter vector $\boldsymbol{\theta}_0 \in \boldsymbol{\Theta}$ and constructing an objective function as defined in (3.2).

For both formulations presented, following the procedure defined by Cui et al.(2017), I provided the analytical gradient (3.7). To observe the advantages of doing so, I created a third algorithm, in which the analytical gradient was not supplied. Considering that, whether I use `fsolve` or `lsqnonlin`, the obtained results are essentially the same, the only slight difference concerns the computational time which varies by an order of magnitude of 10^{-3} ; then, I built the third algorithm using only `fsolve`.

3.5.2 Optimization Method

Following Cui et al. (2017), I used the Levenberg–Marquardt (LM) method, which is particularly suitable because, given any function $\boldsymbol{\Phi} : \mathbb{R}^m \rightarrow \mathbb{R}^n$ and letting $f : \mathbb{R}^m \rightarrow \mathbb{R}$ be the objective function, the resulting optimization problem is

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^5} f(\boldsymbol{\theta}) \quad \text{subject to} \quad f(\boldsymbol{\theta}) = \frac{1}{2} \|\boldsymbol{\Phi}(\boldsymbol{\theta})\|^2 = \frac{1}{2} \boldsymbol{\Phi}(\boldsymbol{\theta})^\top \boldsymbol{\Phi}(\boldsymbol{\theta})$$

It is easy to notice that this is the same formulation of the calibration problem (3.1), indeed even this structure is in nonlinear least squares form.

This method, at each iteration, determines $\boldsymbol{\theta}_k$ and defines

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \Delta\boldsymbol{\theta}$$

It considers the first order Taylor approximation of $\boldsymbol{\Phi}$ in the neighborhood of $\boldsymbol{\theta}_k$, obtaining

$$\boldsymbol{\Phi}(\boldsymbol{\theta}_{k+1}) = \boldsymbol{\Phi}(\boldsymbol{\theta}_k + \Delta\boldsymbol{\theta}) \approx \boldsymbol{\Phi}(\boldsymbol{\theta}_k) + \mathbf{J}_k^\top \Delta\boldsymbol{\theta}$$

Therefore, locally, the objective function is approximated as follows

$$f(\boldsymbol{\theta}_{k+1}) \approx \frac{1}{2} \|\boldsymbol{\Phi}_k + \mathbf{J}_k^\top \Delta\boldsymbol{\theta}\|^2 = \frac{1}{2} (\boldsymbol{\Phi}_k + \mathbf{J}_k^\top \Delta\boldsymbol{\theta})^\top (\boldsymbol{\Phi}_k + \mathbf{J}_k^\top \Delta\boldsymbol{\theta}) \quad (3.8)$$

Differentiating (3.8) with respect to the step search $\Delta\boldsymbol{\theta}$ and setting the resulting expression ugal to zero, it is possible to obtain the so-called *Gauss–Newton normal equations*

$$\mathbf{J}_k \mathbf{J}_k^\top \Delta\boldsymbol{\theta} = -\mathbf{J}_k \boldsymbol{\Phi}_k \quad (3.9)$$

Solving (3.9) for $\Delta\boldsymbol{\theta}$, the corresponding Gauss–Newton (GN) step search is

$$\Delta\boldsymbol{\theta} = -(\mathbf{J}_k \mathbf{J}_k^\top)^{-1} \mathbf{J}_k \boldsymbol{\Phi}_k$$

This is the point where the distinctive feature of the LM method comes into play: the introduction of a damping factor ν_k , whose purpose is to rescale the approximated

Hessian matrix $\mathbf{J}\mathbf{J}^\top$, yielding the following relation

$$\Delta\boldsymbol{\theta} = -\left(\mathbf{J}_k\mathbf{J}_k^\top + \nu_k\mathbf{I}\right)^{-1}\mathbf{J}_k\boldsymbol{\Phi}_k$$

The presence of $\nu_k\mathbf{I}$ term induces an endogenous change in the algorithm, which varies between:

- Steepest Descent: when ν_k takes high values:

$$\nabla^2 f(\boldsymbol{\theta}_k) \approx \nu_k\mathbf{I}$$

- GN: when ν_k takes small values:

$$\nabla^2 f(\boldsymbol{\theta}_k) \approx \mathbf{J}_k\mathbf{J}_k^\top$$

The use of $\mathbf{J}_k\mathbf{J}_k^\top$ is justified by the fact that computing the Hessian matrix at each iteration is expensive and leads to a significant increase in the optimizer's execution time. Furthermore, the calibration of the Heston and OU models, as conjectured, respectively, by Cui et al. (2017) and Brignone and Sgarra (2026), can be considered *small residual problems*.

Finally, as stopping criteria, I set:

- `FunctionTolerance` = 10^{-10}
- `StepTolerance` = 10^{-10}
- `OptimalityTolerance` = 10^{-10}
- `MaxFunctionEvaluations` = 5000
- `MaxIterations` = 5000

3.6 Condition Number and Standard Deviations

Unlike Cui et al. (2017), since the Heston and OU models are also tested on real-world data, it is useful to introduce the concepts of condition number and standard deviations.

Definition 3.6.1 (Condition Number). *Let $\mathbf{A} \in \mathbb{R}^{m \times m}$ be invertible. The condition number of $\mathbf{A}$ is represented by the quantity*

$$\delta_p(\mathbf{A}) = \|\mathbf{A}\|_p \|\mathbf{A}^{-1}\|_p$$

$\delta_p(\mathbf{A})$ measures the sensitivity of a linear system of the form $\mathbf{A}\mathbf{x} = \mathbf{b}$ to perturbations in the problem data. In this framework, I consider $p = 2$ and $\mathbf{A} = \nabla^2 f \approx \mathbf{J}\mathbf{J}^\top$, which is the approximated Hessian matrix of the objective function evaluated at $\hat{\boldsymbol{\theta}}$. Then, the condition number is

$$\delta_2(\mathbf{A}) = \|\mathbf{A}\|_2 \|\mathbf{A}^{-1}\|_2 = \frac{\lambda_{\max}(\mathbf{A})}{\lambda_{\min}(\mathbf{A})}$$

where $\lambda_{\max}$ and $\lambda_{\min}$ are, respectively, the largest and smallest eigenvalues of $\mathbf{A}$.

$\mathbf{J}\mathbf{J}^\top$ measures the curvature of the objective function in the neighborhood of $\hat{\boldsymbol{\theta}}$ with respect to variations in the model parameters and its eigenvalues quantify the curvature along the corresponding principal directions in the parameter space; therefore, a large value of $\delta_2(\mathbf{A})$ indicates a great disparity between the maximum and minimum curvatures, suggesting that there could be some directions along which the objective function varies only slightly, while along others, it is more sensitive to variation. This leads to the conclusion that there may exist different sets of parameters, that are able to reproduce the market data equally well or better than the identified optimal solution, showing that it may not be unique. In other words, the aim of $\delta_2(\mathbf{A})$ is to assess the stability of the solution $\hat{\boldsymbol{\theta}}$ and to determine whether the calibration problem under the specified model is ill-conditioned.

Ill-conditioning refers to the situation in which small perturbations in the market data and in the objective function could lead to large variations in the solution obtained.

Example 3.6.1. Let $\delta_2(\mathbf{A}) = 10^2$. Suppose that the market data are subject to a relative perturbation of the order 10^{-4} , then, in the worst case, the corresponding relative perturbation in $\hat{\boldsymbol{\theta}}$ is of order

$$10^2 10^{-4} = 10^{-2}$$

Cui et al. (2017) consider the calibration problem (3.1) ill-conditioned when

$$\delta_2(\mathbf{A}) > 3.98 \times 10^6$$

Since this threshold was proposed for the simulated data and chosen arbitrary, I consider appropriate to require an higher order when dealing with real-world data; therefore, in this work, I consider the calibration problem ill-conditioned when

$$\delta_2(\mathbf{A}) \geq 10^7 \tag{3.10}$$

While presenting the results, I refer to (3.10) as *Fails in $\delta_2(\mathbf{A})$* .

Other tools which are particularly useful when the calibration is performed on real data are the standard deviations on the parameters. The aim is to assess the degree of uncertainty associated with the parameters found by the optimizer.

Let

$$\hat{\boldsymbol{\sigma}}_t = \left(\hat{\sigma}_t^\kappa, \hat{\sigma}_t^{\bar{v}}, \hat{\sigma}_t^\sigma, \hat{\sigma}_t^\rho, \hat{\sigma}_t^{v_0} \right)^\top \quad \forall t$$

the estimated standard deviation vector associated with the parameters of $\mathcal{M}(\hat{\boldsymbol{\theta}})$. To recover $\hat{\boldsymbol{\sigma}}_t$ it is possible to use, under certain assumptions, the matrix $\mathbf{A}_t = \mathbf{J}_t \mathbf{J}_t^\top$, which represents the approximation of $\nabla^2 f$ at time t . Following the procedure proposed by Davidson (2000), I consider the following econometric model applied at time t :

$$\mathbf{C}_t^*(\mathbf{K}, \mathbf{T}) = \mathbf{C}_t(\boldsymbol{\theta}; \mathbf{K}, \mathbf{T}) + \boldsymbol{\epsilon}_t, \quad \boldsymbol{\epsilon}_t \sim i.i.d. \mathcal{N}_n(\mathbf{0}, (\sigma_t^\epsilon)^2 \mathbf{I}) \tag{3.11}$$

where

$$\text{Cov}(\epsilon_{i,t}, \epsilon_{j,t}) = 0, \quad i, j = 1, \dots, n, \quad i \neq j$$

Using the first order Taylor approximation in the neighborhood of the true parameter vector $\boldsymbol{\theta}_t^*$, I obtain

$$\mathbf{C}_t(\boldsymbol{\theta}; \mathbf{K}, \mathbf{T}) \approx \mathbf{C}_t(\boldsymbol{\theta}^*; \mathbf{K}, \mathbf{T}) + \mathbf{J}_t^\top (\boldsymbol{\theta} - \boldsymbol{\theta}^*) \quad (3.12)$$

Therefore, using (3.11) and (3.12), the corresponding residual vector is given by

$$\mathbf{r}_t(\boldsymbol{\theta}) = \mathbf{C}_t(\boldsymbol{\theta}; \mathbf{K}, \mathbf{T}) - \mathbf{C}_t^*(\mathbf{K}, \mathbf{T}) \approx \mathbf{J}_t^\top (\boldsymbol{\theta} - \boldsymbol{\theta}^*) - \boldsymbol{\epsilon}_t$$

The objective function (3.2), in the neighborhood of $\hat{\boldsymbol{\theta}}$, can thus be approximated by

$$f(\boldsymbol{\theta}) \approx \frac{1}{2} \|\mathbf{J}_t^\top (\boldsymbol{\theta} - \boldsymbol{\theta}^*) - \boldsymbol{\epsilon}_t\|^2$$

To determine $\hat{\boldsymbol{\theta}}_t$, I compute the gradient of $f(\boldsymbol{\theta})$ and set it equal to zero, obtaining

$$\mathbf{J}_t \left[\mathbf{J}_t^\top (\boldsymbol{\theta} - \boldsymbol{\theta}_t^*) - \boldsymbol{\epsilon}_t \right] = \mathbf{0}$$

This means that $\hat{\boldsymbol{\theta}}_t$ verifies the following condition

$$\mathbf{A}_t (\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_t^*) = \mathbf{J}_t \boldsymbol{\epsilon}_t$$

therefore

$$\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_t^* = \mathbf{A}_t^{-1} \mathbf{J}_t \boldsymbol{\epsilon}_t$$

Under the assumptions of this model, it holds that

$$\hat{\boldsymbol{\theta}}_t \sim \mathcal{N}_m \left(\boldsymbol{\theta}_t^*, (\sigma_t^\epsilon)^2 \mathbf{A}_t^{-1} \mathbf{J}_t \mathbf{I}_n \mathbf{J}_t^\top \mathbf{A}_t^{-1} \right)$$

By definition of $\mathbf{A}_t$, I can simplify this expression as follows

$$\hat{\boldsymbol{\theta}}_t \sim \mathcal{N}_m \left(\boldsymbol{\theta}_t^*, (\sigma_t^\epsilon)^2 \mathbf{A}_t^{-1} \right)$$

However, the covariance matrix is not available because the coefficient $(\sigma_t^\epsilon)^2$ is not directly observable; therefore it is necessary to propose a consistent estimator of it. In this case, I consider

$$(\hat{\sigma}_t^\epsilon)^2 = \frac{\|\mathbf{r}_t(\hat{\boldsymbol{\theta}}_t)\|^2}{n - m}$$

It follows that

$$\hat{\boldsymbol{\sigma}}_t = \hat{\sigma}_t^\epsilon \sqrt{\text{diag}(\mathbf{A}_t^{-1})}$$

Moreover, since $\hat{\boldsymbol{\theta}}_t$ is normally distributed, it is possible to compute the confidence

interval, with fixed significance level $\alpha = 0.05$, for each parameter obtained by the optimizer. Such interval is defined by

$$CI_{t,95\%}^i = \left[\hat{\theta}_{i,t} - 1.96 \frac{\hat{\sigma}_{i,t}^{\theta_{i,t}}}{\sqrt{n}}, \hat{\theta}_{i,t} + 1.96 \frac{\hat{\sigma}_{i,t}^{\theta_{i,t}}}{\sqrt{n}} \right], \quad \forall i = 1, \dots, 5$$

Chapter 4

Reparameterized Algorithms

In this chapter, I introduce the different techniques investigated to ensure that the parameters, obtained by the algorithm, for the Heston and OU models, belong to their respective admissible regions.

Cui et al. (2017), considering the problem only on simulated data, did not provide many details regarding a procedure to be followed in the event that the optimizer returns an infeasible solution. This leaves space to propose different ideas, in particular, I introduce two different characterizations:

- constrained calibration problem
- reparameterized calibration problem

Without going into the detail, that I will discuss in Chapter 5, I want to clarify that Section 4.2 focuses only on the Heston model, which, based on the results obtained, did not exhibit high performance.

4.1 Constrained Problem

Both models require constraints when they are tested on real-world data, because there are days in which $\bar{v} < 0$, $\sigma < 0$ and $\rho \notin [-1, 1]$. For this reason, I define the following constrained calibration problem for the Heston model

$$\min_{\boldsymbol{\theta} \in \boldsymbol{\Theta}^H} f(\boldsymbol{\theta}) \quad (4.1)$$

where $\boldsymbol{\Theta}^H \in \mathbb{R}^m$ is the admissible region defined as

$$\boldsymbol{\Theta}^H = \left\{ \boldsymbol{\theta} \in \mathbb{R}^5 : \kappa, \bar{v}, \sigma, v_0 > 0, \rho \in [-1, 1] \right\}$$

The (4.1) can be defined for the OU model as well, considering as admissible region

$$\boldsymbol{\Theta}^{OU} = \left\{ \boldsymbol{\theta} \in \mathbb{R}^5 : \kappa, \bar{v}, \sigma > 0, \rho \in [-1, 1], v_0 \in \mathbb{R} \right\}$$

This characterization of the problem is very fast to implement, but it cannot be applied on all the problem formulations introduced in the Section 3.5.1, indeed, constraints only can be imposed when using `lsqnonlin`. Moreover, I observed an increase in MSE on both simulated and real-world data, with respect to the unconstrained case.

4.2 Proposed Reparameterizations

By imposing the constraints on the calibration problems, as described in Section 4.1, the optimizer is prevented from selecting solutions outside the feasible regions Θ^H and Θ^{OU} , thereby restricting the search space. Instead, here I propose a method based on the introduction of a new parameter vector to be optimized.

Definition 4.2.1 (Reparameterized parameter vector). *Let $\Theta \subset \mathbb{R}^5$ be a feasible region. Furthermore, let $g : \mathbb{R}^5 \rightarrow \Theta$ be an invertible mapping and $\tilde{\theta} \in \mathbb{R}^5$. The vector $\tilde{\theta}$ is called reparameterized parameter vector if*

$$\theta = g(\tilde{\theta}), \quad \theta \in \Theta$$

Definition 4.2.2 (Reparameterized Calibration Problem). *Let $\tilde{\theta} \in \mathbb{R}^5$ be a reparameterized parameter vector, then the following*

$$\min_{\tilde{\theta} \in \mathbb{R}^5} \tilde{f}(\tilde{\theta}), \quad \text{where } \tilde{f}(\tilde{\theta}) = f[g(\tilde{\theta})] \quad (4.2)$$

is called reparameterized calibration problem.

This characterization does not introduce any explicit constraint, keeping the optimizer's search space equal to $\mathbb{R}^5$; however, this is not the original parameter space, indeed, this approach introduces a new geometry on the original space $\mathbb{R}^5$.

It is worth noting that applying this method does not make $\hat{\theta} \notin \Theta$ to be feasible, but instead, introduces a new coordinates system in the optimizer's search space.

Proposition 4.2.1. *Let $g : \mathbb{R}^5 \rightarrow \Theta$ be a twice continuously differentiable mapping and $\tilde{\theta} \in \mathbb{R}^5$ be the corresponding reparameterized parameter vector. Furthermore, let*

$$\mathbf{J}_{\tilde{\theta}} = \frac{\partial \theta}{\partial \tilde{\theta}^T} \in \mathbb{R}^{5 \times 5}, \quad \nabla_{\tilde{\theta}}^2 \theta_j = \frac{\partial^2 \theta_j}{\partial \tilde{\theta} \partial \tilde{\theta}^T} \in \mathbb{R}^{5 \times 5} \quad \forall j = 1, \dots, 5$$

Then, the Theorem 3.3.1 holds and

$$\nabla_{\tilde{\theta}} \tilde{f} = \mathbf{J}_{\tilde{\theta}}^T \nabla f \quad (4.3)$$

$$\nabla_{\tilde{\theta}}^2 \tilde{f} = \mathbf{J}_{\tilde{\theta}}^T \nabla^2 f \mathbf{J}_{\tilde{\theta}} + \sum_{j=1}^5 \frac{\partial f}{\partial \theta_j} \nabla_{\tilde{\theta}}^2 \theta_j \quad (4.4)$$

Proof. Considering the reparameterized calibration problem, $\tilde{f}(\tilde{\theta})$ is represented by

$$\tilde{f}(\tilde{\theta}) = f(\theta) = f[g(\tilde{\theta})]$$

It is simply a composite function of the form $f \circ g$, then under the assumptions of Proposition 4.2.1, those of the Theorem 3.3.1 are satisfied as well, therefore I can apply the multivariate chain rule to differentiate $\tilde{f}(\tilde{\theta})$, obtaining

$$\frac{\partial \tilde{f}}{\partial \tilde{\theta}_j} = \sum_{w=1}^5 \frac{\partial f}{\partial \theta_w} \frac{\partial \theta_w}{\partial \tilde{\theta}_j}, \quad \forall j = 1, \dots, 5 \quad (4.5)$$

I can express (4.5) in the equivalent matrix–vector product

$$\nabla_{\tilde{\theta}} \tilde{f} = \left(\frac{\partial \theta}{\partial \tilde{\theta}^\top} \right)^\top \nabla f \quad (4.6)$$

Considering the convention adopted for vectors in these chapters, I have

$$\mathbf{J}_{\tilde{\theta}} = \frac{\partial \theta}{\partial \tilde{\theta}^\top} = \begin{pmatrix} \frac{\partial \theta_1}{\partial \tilde{\theta}_1} & \dots & \frac{\partial \theta_1}{\partial \tilde{\theta}_5} \\ \vdots & \ddots & \vdots \\ \frac{\partial \theta_5}{\partial \tilde{\theta}_1} & \dots & \frac{\partial \theta_5}{\partial \tilde{\theta}_5} \end{pmatrix}$$

To achieve (4.4) it is necessary differentiate again the expression (4.3)

$$\frac{\partial^2 \tilde{f}}{\partial \tilde{\theta}_l \partial \tilde{\theta}_j} = \sum_{w=1}^5 \left[\frac{\partial}{\partial \tilde{\theta}_l} \left(\frac{\partial f}{\partial \theta_w} \right) \frac{\partial \theta_w}{\partial \tilde{\theta}_j} + \frac{\partial f}{\partial \theta_w} \frac{\partial^2 \theta_w}{\partial \tilde{\theta}_l \partial \tilde{\theta}_j} \right] \quad (4.7)$$

Using again the chain rule I recover that

$$\frac{\partial}{\partial \tilde{\theta}_l} \left(\frac{\partial f}{\partial \theta_w} \right) = \sum_{z=1}^5 \frac{\partial^2 f}{\partial \theta_w \partial \theta_z} \frac{\partial \theta_z}{\partial \tilde{\theta}_l} \quad (4.8)$$

Substituting (4.8) into the expression (4.7), I obtain

$$\frac{\partial^2 \tilde{f}}{\partial \tilde{\theta}_l \partial \tilde{\theta}_j} = \sum_{w=1}^5 \sum_{z=1}^5 \frac{\partial^2 f}{\partial \theta_w \partial \theta_z} \frac{\partial \theta_z}{\partial \tilde{\theta}_l} \frac{\partial \theta_w}{\partial \tilde{\theta}_j} + \sum_{w=1}^5 \frac{\partial f}{\partial \theta_w} \frac{\partial^2 \theta_w}{\partial \tilde{\theta}_l \partial \tilde{\theta}_j}$$

This expression can be written in matrix form as

$$\nabla_{\tilde{\theta}}^2 \tilde{f} = \mathbf{J}_{\tilde{\theta}}^\top \nabla^2 f \mathbf{J}_{\tilde{\theta}} + \sum_{j=1}^5 \frac{\partial f}{\partial \theta_j} \nabla_{\tilde{\theta}}^2 \theta_j$$

□

Therefore, an optimal solution $\boldsymbol{\theta}^*$ that may be unreachable under the original parameterization, for example due to the presence of local minima in its neighborhood, may become reachable through this technique. Moreover, it may improve the scaling of $\nabla_{\boldsymbol{\theta}}^2 \tilde{f}$ and, consequently, reduce the condition number of the problem.

In the neighborhood of $\boldsymbol{\theta}^*$, it holds that $\nabla f \approx 0$ and, conjecturing that the reparameterized calibration problem is still a small residual problem, I obtain that

$$\nabla_{\boldsymbol{\theta}}^2 \tilde{f} \approx \mathbf{J}_{\boldsymbol{\theta}}^T \mathbf{J} \mathbf{J}^T \mathbf{J}_{\tilde{\boldsymbol{\theta}}}$$

Therefore, the matrix on which I will compute the condition number is

$$\mathbf{A} = \mathbf{J}_{\boldsymbol{\theta}}^T \mathbf{J} \mathbf{J}^T \mathbf{J}_{\tilde{\boldsymbol{\theta}}} \quad (4.9)$$

Defining $\tilde{\boldsymbol{\theta}} = (\tilde{k}, \tilde{v}, \tilde{\sigma}, \tilde{\rho}, \tilde{v}_0)^T \in \mathbb{R}^5$ for the Heston model, I propose the following reparameterization:

$$g(\tilde{\boldsymbol{\theta}}) = \left(e^{\tilde{\kappa}}, e^{\tilde{v}}, e^{\tilde{\sigma}}, \frac{2}{1 + e^{-\tilde{\rho}}} - 1, e^{\tilde{v}_0} \right)^T \quad (4.10)$$

It can be noticed that there is a one-to-one correspondence between $\tilde{\boldsymbol{\theta}} \in \mathbb{R}^5$ and $\boldsymbol{\theta} \in \boldsymbol{\Theta}$. Moreover, since each component of $g(\tilde{\boldsymbol{\theta}})$ depends only on one element of $\tilde{\boldsymbol{\theta}}$, it is straightforward to obtain

$$\mathbf{J}_{\tilde{\boldsymbol{\theta}}} = \begin{pmatrix} \kappa & 0 & 0 & 0 & 0 \\ 0 & \bar{v} & 0 & 0 & 0 \\ 0 & 0 & \sigma & 0 & 0 \\ 0 & 0 & 0 & \frac{1-\rho^2}{2} & 0 \\ 0 & 0 & 0 & 0 & v_0 \end{pmatrix}$$

From $\mathbf{A}$, I can compute the vector $\hat{\boldsymbol{\sigma}}_t$ and the confidence intervals introduced in Section 3.6. However, $\hat{\boldsymbol{\sigma}}_t$ refers to the reparameterized parameter vector, then to recover the confidence intervals associated with $\hat{\boldsymbol{\theta}}$, I apply the following transformation:

$$CI_{t,95\%}^i = \left[g \left(\hat{\theta}_{i,t} - 1.96 \frac{\hat{\sigma}_{i,t}}{\sqrt{n}} \right), g \left(\hat{\theta}_{i,t} + 1.96 \frac{\hat{\sigma}_{i,t}}{\sqrt{n}} \right) \right], \quad \forall i = 1, \dots, 5$$

The results obtained from its application were satisfactory and I will discuss them in Chapter 5.

I considered other reparameterizations to improve the scaling of the matrix $\mathbf{A}$, for example using

$$g(\tilde{\boldsymbol{\theta}}) = \left(\log(1 + e^{\tilde{\kappa}}), \log(1 + e^{\tilde{v}}), \log(1 + e^{\tilde{\sigma}}), \frac{2}{1 + e^{-\tilde{\rho}}} - 1, e^{\tilde{v}_0} \right)^T \quad (4.11)$$

However, it required too much computational time to be used and the different combinations of the parameter expressions in (4.10) and (4.11) were not able to improve

the model's performance as (4.10) did.

Regarding the OU model, the reparameterized approach only worsened its performance on real-world data compared to that achieved by the constrained one.

4.3 Summary

This section aims to illustrate which methods and optimizers I will test on simulated data.

I test three different characterizations of the problem:

1. Calibration problem defined by Cui et al. (2017)

I apply it on both Heston and OU models, using the following optimizers:

- i. `fsolve` computing the numerical gradient with the finite differences method (F-FD)
 - ii. `fsolve` with the analytical gradient provided (F-AG)
 - iii. `lsqnonlin` unconstrained and with the analytical gradient provided (NLS)
2. Constrained calibration problem as defined in (4.1)

I apply it on both Heston and OU models, using `lsqnonlin` constrained and with the analytical gradient provided (NLS-CON)

3. Reparameterized calibration problem as defined in (4.2)

I apply it only on the Heston model, using the following optimizers:

- i. F-FD-REP
- ii. F-AG-REP
- iii. NLS-REP

Chapter 5

Numerical Results

In this chapter, I present the results obtained following the methodologies described in Chapter 3 and 4 both on simulated and real-world data. First, I report those related to the simulated data and, subsequently, I show the behavior of the proposed models when applied on real-world data. I consider the second part to be the most important because, unlike what Cui et al. (2017) showed, for each day, I used the implied volatility surface generated by the European call options written on S&P500 as reference for recovering the corresponding market prices.

For both simulated and real data, I apply the normalization $S_0 = 1$, therefore the moneyness of the option coincides with its strike price K .

To perform the experiments, I used MATLAB[®] (Version R2025b) in Microsoft Windows 11[®], running on a machine equipped with AMD Ryzen 5 9600X, CPU @3.9GHz and 32 GB of RAM.

5.1 Simulated Data

The expression *simulated data* refers to the case in which, prior to performing the calibration, I define an optimal parameter set θ^* and use it together with the model under consideration to generate the market price vector $\mathbf{C}^*(\mathbf{K}, \mathbf{T})$.

To do that, it is also necessary to set the different maturities and deltas related to the options. In other words, I need to specify which kind of option is being observed on the market, clarifying the corresponding maturity and using the delta to assess whether it is ITM, ATM or OTM. For the proposed experiments, I consider the following vectors:

$$\mathbf{T} = (30, 60, 90, 120, 150, 180, 252, 360)^\top$$

$$\mathbf{\Delta} = (-0.10, -0.25, 0.5, 0.25, 0.10)^\top$$

I normalize $\mathbf{T}$, as it represents the time to maturity of the option expressed in years.

To illustrate what I mean with Δ , I state that

$$\Delta = \frac{\partial C(\boldsymbol{\theta}; K_i, T_i)}{\partial S} \quad \forall t \in [0, T]$$

and represents the quantity of underlying asset that should be purchased to replicate the option, indeed if it is

- $|\Delta| > 0.5$, the option is ITM
- $|\Delta| = 0.5$, the option is ATM
- $|\Delta| < 0.5$, the option is OTM

Furthermore, it is worth noting that, depending on the kind of option considered, Δ is greater or lower than zero:

- call option, then $\Delta > 0$
- put option, then $\Delta < 0$

Proposition 5.1.1 (Put–Call delta relation). *Let*

$$\Delta = \frac{\partial C(\boldsymbol{\theta}; K_i, T_i)}{\partial S} \quad \forall t \in [0, T]$$

Furthermore, let h_c be a European call option and h_p be the corresponding put. Then

$$\Delta_p = \Delta_c - 1$$

So, when a European option call is ITM, the corresponding put is OTM and vice versa. Once set $\boldsymbol{\theta}^*$, $\mathbf{T}$ and Δ , I apply the formula (3.3) to recover the simulated market prices.

The procedure is very simple, but, to do so, I also need to determine $\mathbf{K}$. Given a starting $C^*(\boldsymbol{\theta}; K_i, T_i)$ and its $\sigma_{\text{imp}}^* = \sigma_{\text{imp}}(C^*)$, the appropriate K_i can be determined performing an optimization; indeed, it is required that

$$K_i^* : \Delta = \Delta^{BS}(C^*, \sigma_{\text{imp}}^*)$$

In other words, the optimal K_i must be such that the delta which I initially set in Δ , is equal to the Black–Scholes delta.

5.1.1 Random Starting Point

The only element that remains to be defined is the starting point of the optimization, namely, the initial parameter set $\boldsymbol{\theta}_0 = (\kappa_0, \bar{v}_0, \sigma_0, \rho_0, (v_0)_0)^\top$ at which I evaluate the theoretical prices $\mathbf{C}(\boldsymbol{\theta}; \mathbf{K}, \mathbf{T})$.

The procedure proposed by Cui et al. (2017) was developed to demonstrate the absence of local minima in the objective function of the Heston model; to this end, they propose randomly sampling the components of $\boldsymbol{\theta}_0$. In particular, they define a range of plausible values for each initial parameter, assuming that it is independent from the others and follows a uniform distribution:

- $\theta_i \perp\!\!\!\perp \theta_j \quad i \neq j$
- $\theta_i \sim U[a_i, b_i], \quad i = 1, \dots, 5$

where a_i and b_i represent, respectively, the lower and upper bounds of the proposed interval. To further investigate the issue of local minima, I also adopt this procedure. Table 5.1 reports the ranges identified by Cui et al. (2017) for the Heston model.

Parameter	a_i	b_i
κ	0.05	5.00
$\bar{v}$	0.05	0.95
σ	0.05	0.95
ρ	-0.90	-0.10
v_0	0.05	0.95

Table 5.1: Parameter ranges reported by Cui et al. (2017) for the Heston model

Therefore, when the calibration problem is reparameterized, I consider the reparameterized intervals contained in Table 5.2.

Parameter	$g^{-1}(a_i)$	$g^{-1}(b_i)$
κ	$\log(0.05)$	$\log(5.00)$
$\bar{v}$	$\log(0.05)$	$\log(0.95)$
σ	$\log(0.05)$	$\log(0.95)$
ρ	$\log\left(\frac{1-0.90}{1+0.90}\right)$	$\log\left(\frac{1-0.10}{1+0.10}\right)$
v_0	$\log(0.05)$	$\log(0.95)$

Table 5.2: Parameter ranges for the Heston model under the reparameterized calibration problem

Similarly, Brignone and Sgarra (2026) identified, for each parameter of the OU model, intervals of plausible values while maintaining the independence assumption and the one concerning the uniform distribution; therefore, I report these intervals in Table 5.3.

Parameter	a_i	b_i
κ	0.50	8.00
$\bar{v}$	0.05	0.35
σ	0.05	0.5
ρ	-0.95	0.25
v_0	0.05	0.35

Table 5.3: Parameter ranges reported by Brignone and Sgarra (2026) for the OU model

5.1.2 Validation Procedure

This brief section aims to explain the procedure used to validate the models.

Validate a model means assessing its performance for different optimal parameter sets θ^* and different initial values of θ_0 .

To perform the validation, I need to generate, using the procedure described in Section 5.1.1, an optimal parameter set θ^* and the corresponding θ_0 . In particular, I used a validation procedure consisting of 10^4 cases, which are structured as follows:

1. I generate one hundred θ^*
2. For each θ^* , I generate one hundred θ_0
3. For each θ_0 , I verify whether the optimizer converges to the corresponding θ^*

Since the market data are generated using the model under validation, I require that it is able to recover θ^* with a high degree of accuracy, therefore, I will consider unsuccessful the tests that violate the following condition

$$\frac{1}{n} \sum_{i=1}^n r_i^2(\hat{\theta}) \leq 10^{-10} \quad (5.1)$$

While presenting the results, I refer to (5.1) as *Fails in MSE*.

I provide a schematic representation of the proposed validation procedure in Algorithm 5.1.1.

Algorithm 5.1.1 Validation procedure

- 1: **for** $k = 1, \dots, 100$ **do**
 - 2: Generate the optimal parameter set θ^*
 - 3: Recover $C^*(\theta; K, T)$ and K
 - 4: **for** $j = 1, \dots, 100$ **do**
 - 5: Generate the initial guess θ_0
 - 6: Use the optimizer to find $\hat{\theta}$
 - 7: **end for**
 - 8: **end for**
-

5.1.3 Performance

In Table 5.4, I present the results obtained during the validation of the Heston model under the characterization proposed by Cui et al. (2017) and under the constrained approach.

Considering the number of *Fails in MSE* the results are highly encouraging, as, in fact, providing the analytical gradient improves the identification of the optimal solution by approximately 10% compared with the value achieved by F-FD. Furthermore, it reduces both the computational time and the number of evaluations of $f(\theta)$ by approximately 80%.

	F-FD	F-AG	NLS	NLS-CON
$ \kappa^* - \hat{\kappa} $	6.18×10^{-2}	6.20×10^{-2}	6.20×10^{-2}	1.75×10^{-2}
$ \bar{v}^* - \hat{\bar{v}} $	1.626×10^{-1}	5.4589	5.4589	2.7×10^{-3}
$ \sigma^* - \hat{\sigma} $	3.0×10^{-3}	2.7×10^{-3}	2.7×10^{-3}	3.3×10^{-3}
$ \rho^* - \hat{\rho} $	6.5×10^{-3}	4.5×10^{-3}	4.5×10^{-3}	3.1×10^{-3}
$ v_0^* - \hat{v}_0 $	1.1×10^{-3}	1.1×10^{-3}	1.1×10^{-3}	1.1×10^{-3}
MSE	1.50×10^{-7}	1.44×10^{-7}	1.44×10^{-7}	8.095×10^{-6}
Time	0.50	0.12	0.12	0.12
Iter.	16.1422	16.9566	16.9566	15.4962
Fun. eval.	106.4120	22.1696	22.1696	21.0302
Fails in MSE	219	197	197	95

Table 5.4: Results for the Heston model under the characterization proposed by Cui et al. (2017) and the constrained calibration approach over 10^4 cases

The introduction of constraints improves the optimizer's accuracy, while maintaining the same computational time as in the unconstrained problem; in particular, reducing the number of *Fails in MSE* by approximately 56% compared to F-FD and almost 50% compared to the unconstrained optimizations.

As can be observed, for F-AG and NLS, the *MAE* associated with the parameter $\bar{v}$ is completely outside the range of reasonable values identified in Table 5.1. This means that, in some cases, the optimizer is not able to find the solution; therefore, I examined all such cases to determine how often this severe issue occurs. I identified only two cases out of 10^4 , which are reported in Table 5.5.

Parameters	I			II		
	θ_0	θ^*	$\hat{\theta}$	θ_0	θ^*	$\hat{\theta}$
κ	2.0346	1.7757	-1.4582×10^{-6}	0.6186	1.1285	-4.3862×10^{-7}
$\bar{v}$	0.3960	0.1028	-3.1554×10^4	0.8599	0.0915	2.3002×10^4
σ	0.1340	0.7889	0.5668	0.1151	0.4852	0.3532
ρ	-0.1246	-0.7564	-0.7358	-0.8286	-0.3544	-0.3685
v_0	0.2210	0.0814	0.0756	0.5082	0.1153	0.1117

Table 5.5: Cases with misspecified $\bar{v}$ values using F-AG and NLS

The data clearly indicate an identifiability issue between κ and $\bar{v}$, indeed, there appears to be a relation between the two misspecifications because, in each case, $\kappa \rightarrow 0$ as $\bar{v} \rightarrow +\infty$. This relationship is due to the meanings of the two parameters within the model, however, I cannot conclude that this is the source of the problem, since the same phenomenon does not occur in the OU model.

In this context, given the small number of cases observed, I consider the problem as negligible. In other words, these cases are outliers and I removed them from the results. Table 5.6 show how the corresponding *MAE* values change.

MAE	F-AG	NLS
$ \kappa^* - \hat{\kappa} $	6.17×10^{-2}	6.17×10^{-2}
$ \bar{v}^* - \hat{\bar{v}} $	3.3×10^{-3}	3.3×10^{-3}
$ \sigma^* - \hat{\sigma} $	2.7×10^{-3}	2.7×10^{-3}
$ \rho^* - \hat{\rho} $	4.4×10^{-3}	4.4×10^{-3}
$ v_0^* - \hat{v}_0 $	1.1×10^{-3}	1.1×10^{-3}
MSE	1.4375×10^{-7}	1.4375×10^{-7}

Table 5.6: *MAEs* without outliers for the Heston model over 10^4 cases under the unconstrained calibration approach

Considering the Heston model under (4.2), I report the results of the validation procedure in Table 5.7.

	F-FD-REP	F-AD-REP	NLS-REP
$ \kappa^* - \hat{\kappa} $	7.991×10^{-1}	1.3282	1.3282
$ \bar{v}^* - \hat{\bar{v}} $	3.278×10^{-1}	3.9435×10^{-4}	3.9435×10^{-4}
$ \sigma^* - \hat{\sigma} $	4.1×10^{-3}	3.4×10^{-3}	3.4×10^{-3}
$ \rho^* - \hat{\rho} $	6.5×10^{-3}	2.4×10^{-3}	2.4×10^{-3}
$ v_0^* - \hat{v}_0 $	5.1×10^{-3}	4.0×10^{-3}	4.0×10^{-3}
MSE	8.7102×10^{-8}	1.4773×10^{-7}	1.4773×10^{-7}
Time	0.4802	0.1137	0.1135
Iter.	16.3636	16.8270	16.8270
Fun. eval.	107.7306	21.8161	21.8161
Fails in MSE	84	36	36

Table 5.7: Results for the Heston model over 10^4 cases under the calibration approach (4.2)

The number of *Fails in MSE* decreases by approximately 60% when using F-FD-REP and by 80% for the other optimizers compared with the characterization examined previously. NLS-REP exhibits 60% fewer *Fails in MSE* compared with NLS-CON. This means that NLS-REP converges to the wrong set of parameters in only 0.36% of the cases.

By examining the *MAEs* associated with $\bar{v}$ and κ , I noticed that the former improves by three orders of magnitude, whereas the latter appears to have deteriorated by two orders of magnitude. However, it remains smaller than the width of the range of plausible values identified in Table 5.1. So, I do not consider it as significant issue on simulated data.

In conclusion, I can state that the Heston model under the reparameterized calibration problem, when tested on simulated data, produced results that are either better than or equal to those obtained under the calibration problem defined by Cui et al. (2017), therefore, based on this evidence, I suggest to apply the reparameterization (4.10).

Considering the OU model, the results are reported in Table 5.8.

	F-FD	F-AG	NLS	NLS-CON
$ \kappa^* - \hat{\kappa} $	7.95×10^{-2}	7.65×10^{-2}	7.65×10^{-2}	2.993×10^{-1}
$ \bar{v}^* - \hat{\bar{v}} $	1.8×10^{-3}	1.7×10^{-3}	1.7×10^{-3}	4.2×10^{-3}
$ \sigma^* - \hat{\sigma} $	4.62×10^{-2}	4.62×10^{-2}	4.62×10^{-2}	2.46×10^{-2}
$ \rho^* - \hat{\rho} $	4.11×10^{-2}	3.90×10^{-2}	3.90×10^{-2}	6.07×10^{-2}
$ v_0^* - \hat{v}_0 $	1.6×10^{-3}	1.6×10^{-3}	1.6×10^{-3}	4.4×10^{-3}
MSE	2.4828×10^{-8}	1.3939×10^{-8}	1.3939×10^{-8}	5.1518×10^{-7}
Time	1.2434	0.2774	0.2771	0.3117
Iter.	16.0311	15.9207	15.9207	15.7355
Fun. eval.	105.8648	20.6070	20.6070	21.3464
Fails in MSE	165	131	131	1077

Table 5.8: Results for the OU model over 10^4 cases under the constrained calibration approach

Examining the number of *Fails in MSE*, I observed a deterioration for NLS-CON, however, the average *MAEs* remain very low, indicating that the optimizer is able to converge to the neighborhood of the optimal solution but fails to achieve the level of accuracy imposed by (5.1). It should be recalled that I set that condition arbitrarily; therefore, it is reasonable to expect that increasing the tolerance would result in a considerable reduction of *Fails in MSE*.

Examining the *MAEs*, unlike the Heston model, there is no evidence of misspecification between the parameters κ and $\bar{v}$, this suggests that the problem does not arise from the structure of the CHF, but rather from the different scales on which the parameters are defined in the approximated Hessian matrix $\mathbf{A}$ of the objective function, as I will discuss in Section 5.2.2. However, I noticed a higher computational time compared with the Heston model; it is due to the more complicated structure of the CHF.

Considering the *MSE*, the OU model under the constrained calibration problem, is able to achieve, in almost all optimizers, a higher level of accuracy than the Heston model. This shows that, when applied to the OU model, NLS-CON achieves significantly higher levels of accuracy whenever it is able to converge to the optimal solution.

5.2 Real-world Data

In this section, I present the results obtained by the models on real-world data. For each of them, I selected the optimizer that achieved the best results during the validation procedure, in particular:

- for the Heston model under the constrained approach, I decided to use NLS-CON, whereas NLS-REP is employed under the reparameterized characterization.
- for the OU model, I initially chose NLS, however, during the tests, it was not able to systematically converge to $\hat{\theta} \in \Theta^{OU}$; therefore, I selected NLS-CON instead.

5.2.1 Data

Before presenting the results obtained and the conclusions that I drew from them, I first describe the data used in the analysis.

The dataset contains the daily implied volatilities from 05/12/2007 to 04/03/2021, generated from quoted European call options written on the S&P500. In particular, it contains $n = 54$ options with six different maturities expressed in months and nine different moneyness levels expressed in percentages, on which the models can be calibrated. Once I performed the appropriate transformations, I obtained the following vectors

$$\mathbf{T} = (1, 2, 3, 6, 9, 12)^\top / 12$$

$$\mathbf{M} = (0.80, 0.85, 0.90, 0.95, 1.00, 1.05, 1.10, 1.15, 1.20)^\top$$

I performed the calibration on a weekly basis, using only Wednesdays rather than every day in the dataset, for a total of 692 days. This choice is motivated by the fact that the literature has found options to be more liquid on these days.

For risk free rate time series $\{r_t\}_{t=1}^{692}$, I used the 3 months T-Bill rate downloaded from FRED (Federal Reserve Economic Data).

I used the Black-Scholes formula to recover the market prices $\mathbf{C}^*(\mathbf{K}, \mathbf{T})$, subsequently, I employed a multistart procedure, as described in Section 5.1.1 and 5.1.2, generating only 10 initial guesses $\boldsymbol{\theta}_0$. Then, I used the optimizers to identify the parameter set $\hat{\boldsymbol{\theta}}$.

I provide a schematic representation of the proposed procedure on real-world data in Algorithm 5.2.1.

Algorithm 5.2.1 Calibration on real-world data

```

1: for  $k = 1, \dots, 692$  do
2:   Recover  $\mathbf{C}^*(\mathbf{K}, \mathbf{T})$  through BS formula
3:   for  $j = 1, \dots, 10$  do
4:     Generate the initial guess  $\boldsymbol{\theta}_0$ 
5:     Use the optimizer to find  $\hat{\boldsymbol{\theta}}$ 
6:   end for
7: end for

```

5.2.2 A Case Study

To examine the performance of the models in greater detail, I first consider the following case study. I selected the last day in the dataset, namely 03/03/2021, as no significant market events occurred on that day. First, I present the results obtained for the Heston model under the two selected characterizations followed by those related to the OU model under (4.1).

I considered two experiments on the Heston model. The first one, in which I fixed the initial guess $\boldsymbol{\theta}_0$ and the corresponding reparameterization $\tilde{\boldsymbol{\theta}}_0$, while in the second one I used the multistart optimization procedure.

In the first experiment, I set

$$\boldsymbol{\theta}_0 = (1.2, 0.2, 0.3, -0.6, 0.2)^\top$$

$$\tilde{\boldsymbol{\theta}}_0 = \left(\log(1.2), \log(0.2), \log(0.3), \log\left(\frac{1-0.6}{1+0.6}\right), \log(0.2) \right)^\top$$

and, as risk free rate, $r_{692} = 5.0 \times 10^{-4}$. Under these conditions, both characterizations identify the same optimal solution

$$\hat{\boldsymbol{\theta}} = (2.1058, 0.1054, 1.4464, -0.7532, 0.0615)^\top$$

Moreover, to assess whether the problem is ill-conditioned and to evaluate the accuracy of the model under (4.1) and (4.2), I report the results in Table 5.9.

	NLS-CON	NLS-REP
MSE	7.5672×10^{-7}	7.5672×10^{-7}
Time	0.44	0.16
Iter.	26	13
Fun. eval.	35	14
$\delta(\mathbf{A})$	5.9512×10^5	2.0894×10^3

Table 5.9: Results obtained for the Heston model under (4.1) and (4.2) on 03/03/2021

Since these are real-world data, it is reasonable to obtain $MSE > 10^{-10}$. Analyzing the results, I can immediately observe that, under (4.2), the model required approximately one-third of the computational time required by the one under (4.1), while maintaining the same level of accuracy. This is due to fewer *Iterations* and, most importantly, to the lower number of *Function evaluation* of $f(\boldsymbol{\theta})$. To observe the market price fit, I reported the Figure 5.1.

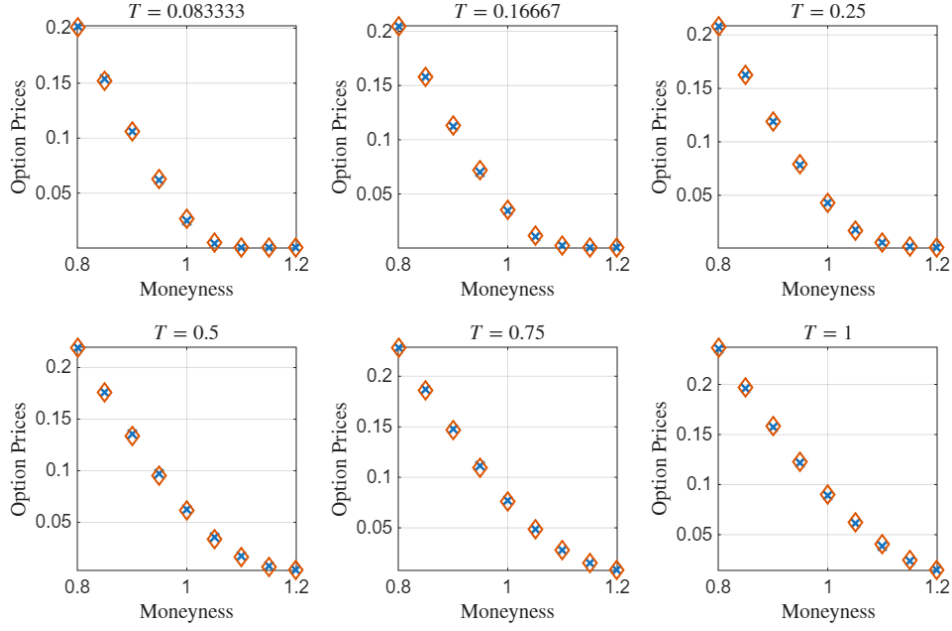


Figure 5.1: $C^*(K, T)$ e $C(\hat{\theta}; K, T)$ obtained using the the Heston model under (4.2) on 03/03/2021. $C^*(K, T)$: plot with blue cross markers. $C(\hat{\theta}; K, T)$: plot with orange diamonds.

The characterization (4.2) achieves greater stability, as it is able to attain a condition number that is two orders of magnitude lower than the one obtained under (4.1). This is likely due to the fact that (4.2) introduces comparable scales among the elements of the matrix $\mathbf{A}$. To highlight these differences, I report the approximated Hessian matrices corresponding to the two different approaches in Tables 5.10 and 5.11.

	$\partial\kappa$	$\partial\bar{v}$	$\partial\sigma$	$\partial\rho$	∂v_0
$\partial\kappa$	1.4×10^{-3}				
$\partial\bar{v}$	5.04×10^{-2}	1.8032			
$\partial\sigma$	-2.7×10^{-3}	-9.28×10^{-2}	5.2×10^{-3}		
$\partial\rho$	4.1×10^{-3}	1.31×10^{-1}	-8.1×10^{-3}	2.12×10^{-2}	
∂v_0	3.99×10^{-2}	1.4484	-7.84×10^{-2}	1.013×10^{-1}	1.6543

Table 5.10: Matrix $\mathbf{A}$ of the Heston model under (4.1)

	$\partial\kappa$	$\partial\bar{v}$	$\partial\sigma$	$\partial\rho$	∂v_0
$\partial\kappa$	6.4×10^{-3}				
$\partial\bar{v}$	1.12×10^{-2}	2.0×10^{-2}			
$\partial\sigma$	-8.2×10^{-3}	-1.41×10^{-2}	1.08×10^{-2}		
$\partial\rho$	1.9×10^{-3}	3.0×10^{-3}	-2.5×10^{-3}	1.0×10^{-3}	
∂v_0	5.2×10^{-3}	9.4×10^{-3}	-7.0×10^{-3}	1.3×10^{-3}	6.3×10^{-3}

Table 5.11: Matrix $\mathbf{A}$ of the Heston model under (4.2)

The main diagonals of the two matrices show that, under (4.2), the objective function exhibits lower curvature with respect to variations in $\tilde{\bar{v}}$ and $\tilde{v}_0$ compared with those obtained for $\bar{v}$ and v_0 . This means that, keeping the other parameters fixed and varying either $\tilde{\bar{v}}$ or $\tilde{v}_0$, $f(\boldsymbol{\theta})$ undergoes relatively smaller changes compared with those resulting from variations in the corresponding non-reparameterized parameters. To take into account also the contribution of the off-diagonal terms, I considered the following eigenvalues spectrums

$$\boldsymbol{\lambda} = (5.3661 \times 10^{-6}, 2 \times 10^{-4}, 1.18 \times 10^{-2}, 2.798 \times 10^{-2}, 3.1935)^T \quad (5.2)$$

$$\boldsymbol{\lambda}^{REP} = (2 \times 10^{-5}, 2 \times 10^{-4}, 8 \times 10^{-4}, 1.7 \times 10^{-3}, 4.18 \times 10^{-2})^T \quad (5.3)$$

I can state that, under (4.1), the spread between the $\lambda_{\min}$ and $\lambda_{\max}$ is substantial, indicating a pronounced disparity of curvature across the associated principal directions. $\lambda_{\min}$ exhibits a much lower value than those associated with the other eigenvalues, indicating a nearly flat valley of objective-function in the neighborhood of $\hat{\boldsymbol{\theta}}$. Instead, under (4.2), this disparity is substantially mitigated, resulting in a considerably more balanced eigenvalue spectrum and, consequently, a more uniform curvature across the principal directions. This means that, under both calibration approaches, there is no evidence of an alternative set of parameters which could be able to fit equally well or better the market prices, however, this evidence is much stronger when using the reparameterized calibration problem, suggesting more stability in the solution found by the optimizer.

Building on this result, in the second experiment, I tested the stability of the solution identified by the two algorithms. The results shows that NLS-REP converged to the same solution $\hat{\boldsymbol{\theta}}$ in all 10 runs, in contrast, NLS-CON converged to $\hat{\boldsymbol{\theta}}$ only 5 times and, in the remaining runs, it became stuck in different local minima with higher associated *MSE*.

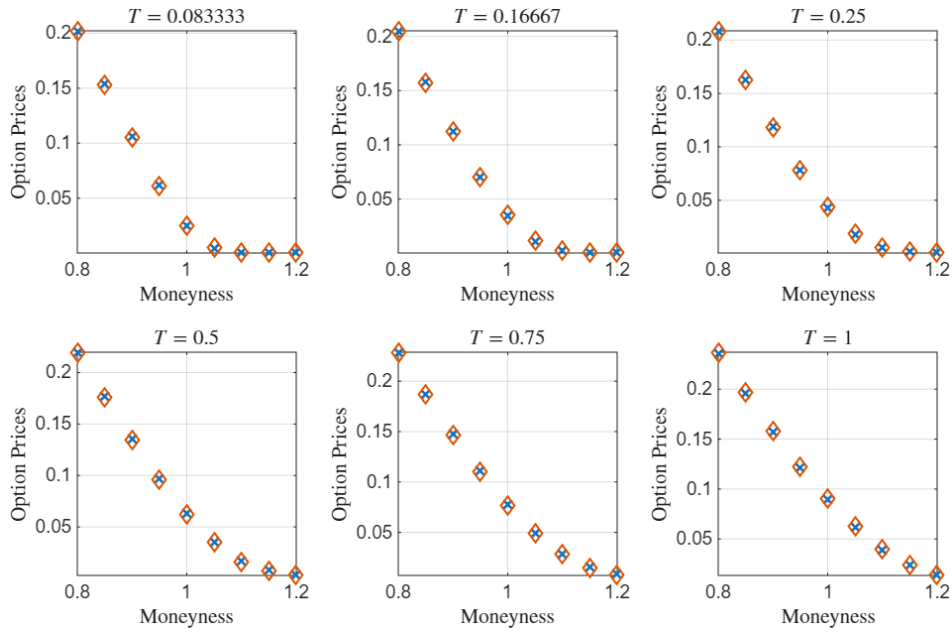
Considering the OU model, I performed directly the multistart optimization procedure, obtaining that NLS-CON converged to the same optimal solution

$$\hat{\boldsymbol{\theta}} = (0.4057, 0, 0.3127, -0.7923, 0.1934)^T$$

in all 10 runs. This suggests that there is no evidence about the presence of multiple local minima in the objective function. Moreover, since $\bar{v} = 0$, on this day the model has the same structure developed by Heston (1993). I reported the corresponding results in Table 5.12.

Even in this case, the condition number is lower than the one obtained by implementing (4.1) on the Heston model. As shown in Figure 5.2, this result supports the evidence that a non-affine model can achieve a better fit on the market prices and higher stability levels, than its affine counterpart.

OU	
MSE	5.0353×10^{-7}
Time	1.8762
Iter.	33
Fun. eval.	49
$\delta(\mathbf{A})$	9.9293×10^3

Table 5.12: Results obtained for the OU model under (4.1) on 03/03/2021**Figure 5.2:** $C^*(K, T)$ e $C(\hat{\theta}; K, T)$ obtained using OU model under (4.1) on 03/03/2021. $C^*(K, T)$: plot with blue cross markers. $C(\hat{\theta}; K, T)$: plot with orange diamonds.

This is due to the fact that the approximation of the Hessian matrix evaluated at $\hat{\theta}$ exhibits a better scaling by default, with no need to reparameterize the problem. I reported the details in Table 5.13.

	$\partial\kappa$	$\partial\bar{v}$	$\partial\sigma$	$\partial\rho$	∂v_0
$\partial\kappa$	2.01×10^{-2}				
$\partial\bar{v}$	-1.66×10^{-2}	1.42×10^{-2}			
$\partial\sigma$	-7.35×10^{-2}	6.11×10^{-2}	2.713×10^{-1}		
$\partial\rho$	-1.27×10^{-2}	1.19×10^{-2}	4.77×10^{-2}	2.16×10^{-2}	
∂v_0	1.149×10^{-1}	9.68×10^{-2}	4.187×10^{-1}	9.03×10^{-2}	7.887×10^{-1}

Table 5.13: Matrix $\mathbf{A}$ of the OU model under (4.1)

The $\mathbf{A}$ matrix shows that there is no evidence of substantial curvature discrepancy in the neighborhood of $\hat{\theta}$ along the orthogonal directions. This fact still valid also looking

the eigenvalues associated with the principal directions, indeed the spectrum

$$\boldsymbol{\lambda} = (10^{-4}, 3 \times 10^{-4}, 1.12 \times 10^{-2}, 4.08 \times 10^{-2}, 1.0634)^T \quad (5.4)$$

shows less discrepancy in the magnitude of $\lambda_{\min}$ and $\lambda_{\max}$ compared with the one observed in the Heston model under (4.1). However, I reported again a higher computational time associated with this model, which represents a disadvantage for trading applications.

5.2.3 Performance

Having highlighted the advantages and disadvantages of using (4.1) and (4.2), as well as the differences between the Heston and OU models, this section aims to present and interpret the results obtained by the optimizers over 692 weeks, specifically over the period from 05/12/2007 to 03/03/2021.

Following the procedure described in Section 5.2.1, I applied the multistart procedure to both models, using the same $\boldsymbol{\theta}_0$ for the Heston model under (4.1) and (4.2). The results obtained are reported in Table 5.14.

	NLS-CON Heston	NLS-REP	NLS-CON OU
MSE	7.1212×10^{-4}	1.997×10^{-6}	2.0311×10^{-6}
Time	2.02	0.21	1.26
Iter.	71.2548	21.5887	36.5186
Fun. eval.	194.9945	29.977	50.7499
Fails in $\delta_t(\mathbf{A})$	45	20	6

Table 5.14: Performance of weekly calibration over the Period 2007–2021

The Heston model, under (4.2) achieves a computational time approximately 10 times lower than that under (4.1), because, as shown during the validation procedure, it requires fewer *Iterations* and *Function Evaluations* to converge. It is able to produce the same level of accuracy of that reported for the OU model, although it shows a higher number of ill-conditioned days.

Observing the *Fails in $\delta_t(\mathbf{A})$* , the OU model is the most versatile, because it is able to converge to a stable solution in almost 99% of the cases; this is a strong evidence of the absence of local minima in the neighborhood of $\hat{\boldsymbol{\theta}}_t^{OU}$. In contrast with the Heston model under both (4.1) and (4.2), this model performed well even during the 2008–2009 global financial crisis and during 2010–2012 European sovereign debt crisis.

Looking both the stability of the solution found $\hat{\boldsymbol{\theta}}_t$ and level of accuracy, the Heston model, under (4.1), reports the worst results. As explained in Section 5.2.2, this is due to the fact that the objective function exhibits local minima as confirmed by the poor scaling of the matrix $\mathbf{A}_t$ and they are the reasons why the *MSE* presents the highest value. Moreover, despite having a less complicated structure, it required almost twice computational time of the OU model.

5.2.4 Condition Number and Parameter Time Series

After applying the multistart procedure, I described the results obtained by both models, however it is also important to examine the resulting times series. In particular, those related to the condition number and the MSE .

To obtain them, I considered the optimal solutions found by the optimizer on each trading day, selecting those associated with the smallest MSE . In other words, I used a selection criterion that prioritizes the fit to market prices, obtaining a single value for each trading day.

Let $\{E_t\}_{t=1}^{692}$ and $\{M_t\}_{t=1}^{692}$ denote, respectively, the condition number and MSE time series. Observing the Figure 5.3, it can be seen that, both models, applying the characterizations (4.1) and (4.2) on the Heston one, produced approximately the same results in terms of MSE , showing only three significant spikes in the following years: 2008, 2012 and 2020.

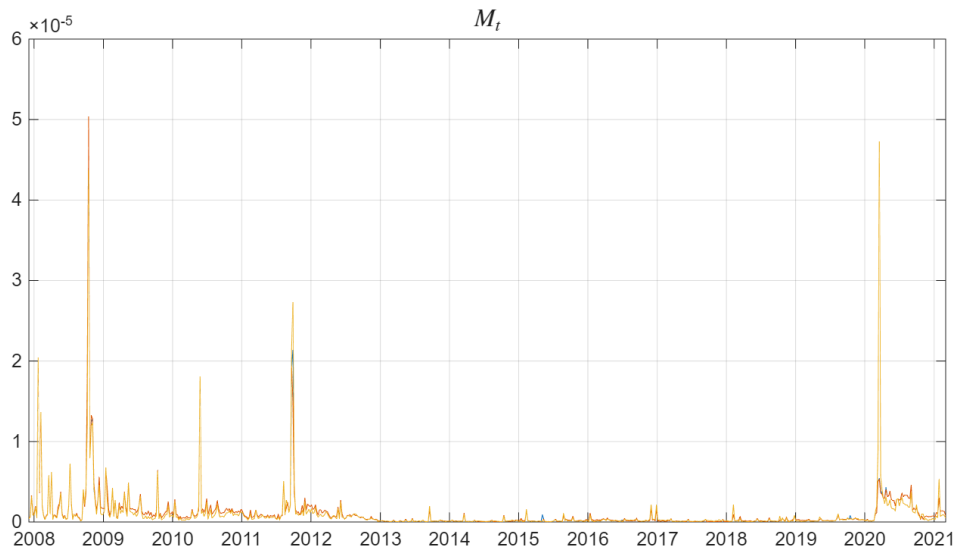


Figure 5.3: Time series of M_t . Blue line: Heston under (4.1); Orange line: Heston under (4.2); Yellow line: OU under (4.1)

I believe that the most informative comparison is the one concerning the condition number, indeed I noted that, in all cases considered, $\{E_t\}_{t=1}^{692}$ occasionally takes very large values; therefore, to properly represent its behavior, I applied a transformation using $\log_{10}(\cdot)$.

As Figure 5.4 shows, the Heston model, under (4.1), exhibits the highest values in almost all trading days, suggesting that the resulting parameter sets may not provide the best fit to market prices. Furthermore, the condition (3.10) is violated 45 times, instead, applying the threshold proposed by Cui et al. (2017), the problem is ill-conditioned approximately in the 68% of the cases considered. This shows the inadequacy on real-world data of the Heston model under the constrained approach.

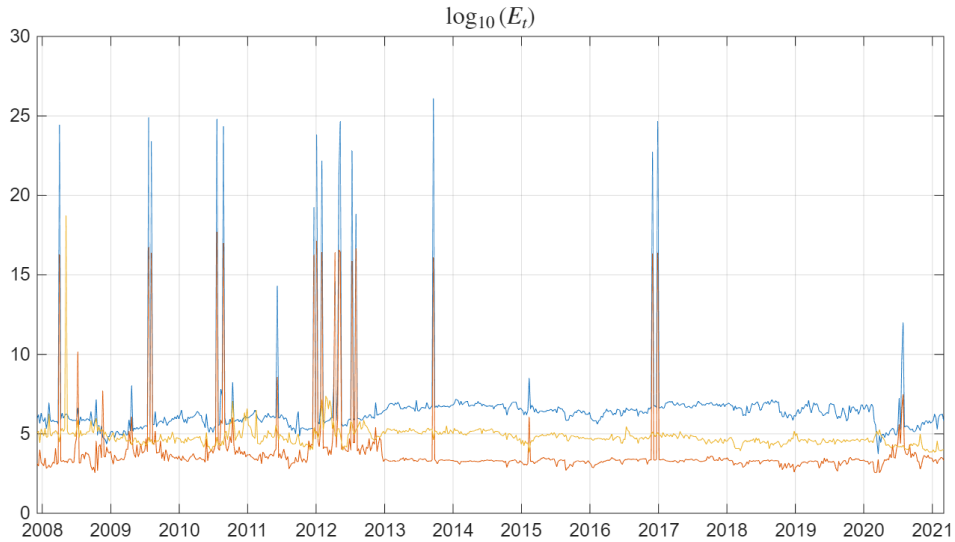


Figure 5.4: Time series of $\log_{10}(E_t)$. Blue line: Heston under (4.1); Orange line: Heston under (4.2); Yellow line: OU under (4.1)

Conversely, the model under (4.2), obtains the lowest values, showing that it is able to identify a highly stable solution in the 97% of the cases. However, I observed the same significant spikes, with smallest values, that seem to be unpredictable. By analyzing the trading days on which these spikes occur, I can conclude that, in some cases, the issue is caused by market data that do not satisfy the no-arbitrage conditions. Figure 5.5 shows an example of strongly ill-conditioned day; in particular, it contains the smiles of the considered options for different maturities on 09/07/2008.

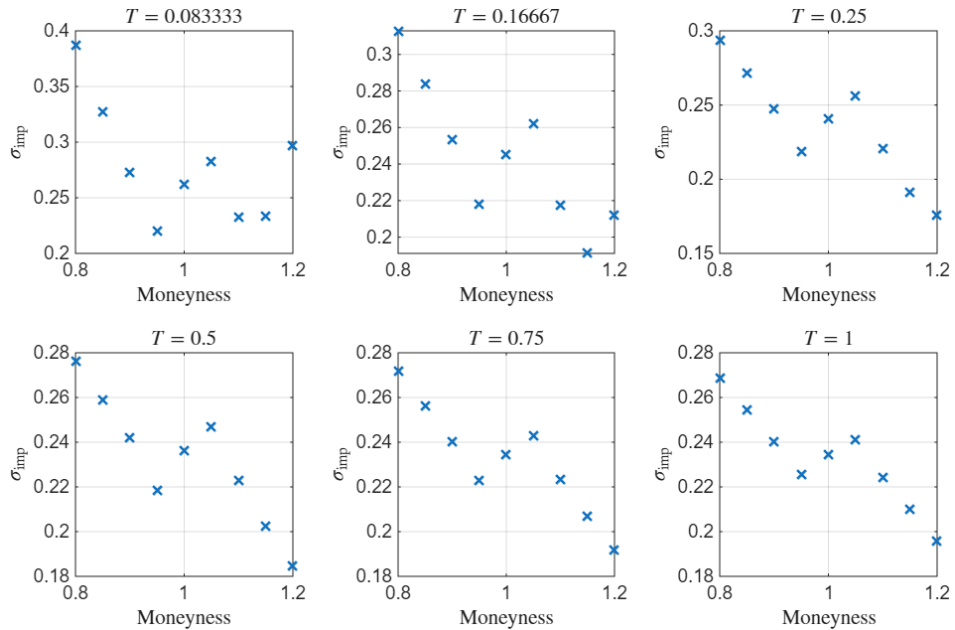


Figure 5.5: Market implied volatility (IV) smiles on 09/07/2008

Regarding the OU model, I found that it exhibits the most stable time series over time, indeed no significant spikes are observed. This confirms the model's ability to fit

market prices, identifying in the 99% of the cases a stable parameter vector. As I have already emphasized, this is due to the non-affine structure of the model, supporting the evidence that it provide a better fit to market data.

From this point onwards, I will comment only the results related to the Heston model under (4.2) and the OU model, since, based on the time series $\{\log_{10}(E_t)\}_{t=1}^{692}$, they obtained the best performance.

By analyzing the days on which the problem is ill-conditioned under both models, I found that this occurs only on 01/02/2012.

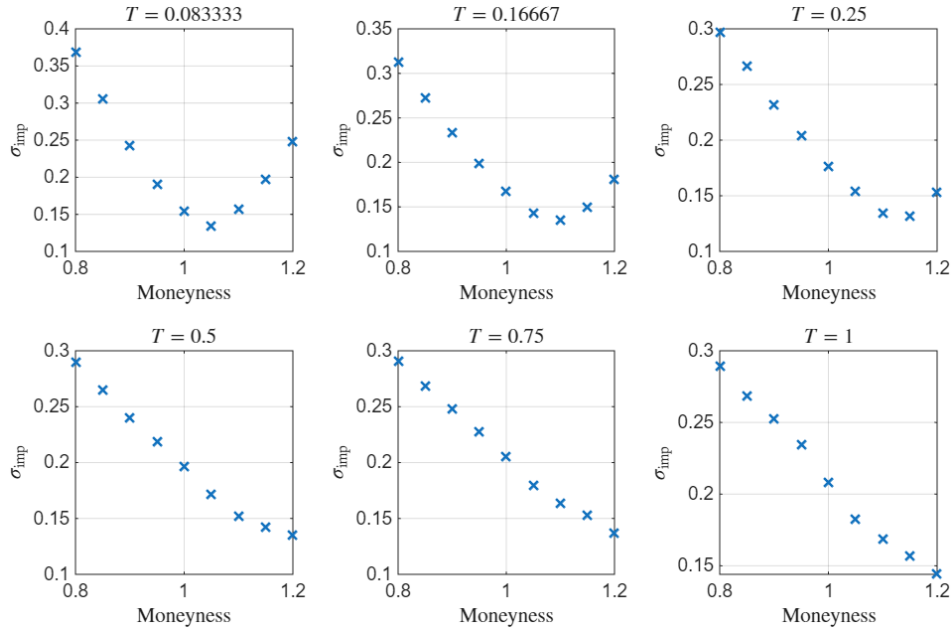


Figure 5.6: Market IV smiles for 01/02/2012

As shown in Figure 5.6, the market data do not violate the arbitrage-free conditions, and both models achieve a high level of accuracy. This situation is caused by a lack of clear relationship between E_t and M_t , meaning that a large condition number is not necessarily associated with a large value of the MSE and vice versa. A large condition number only indicates a strong disparity in the curvature of the objective function along the principal directions associated with $\lambda_{\min}$ and $\lambda_{\max}$, in the neighborhood of the optimal solution found. This suggests that the function may exhibit a flatter valley along the direction associated to $\lambda_{\min}$ than along the one related to $\lambda_{\max}$. In these days, it cannot be ruled out that there exists a better set of parameters for fitting market prices.

Figure 5.7 shows the time series of the parameters obtained by calibrating the Heston and OU models and their corresponding 95% confidence intervals. The Heston model exhibits the same identifiability problem between κ and $\bar{v}$ that arose during the validation procedure. When this phenomenon occurs, the standard errors associated with these parameters explode, becoming extremely large. Therefore, to properly represent the time series, I applied the $\log_{10}(\cdot)$ transformation to $\hat{v}_t$ and did not report the confidence

intervals for these days. Instead, the OU model is able to identify reasonable values for all parameters in every case in the sample, so no transformation was necessary. However, some standard errors fell outside the feasible region Θ^{OU} , so I did not report them.

It can be observed that, except on the ill-conditioned days, all the time series exhibit virtually no uncertainty in the optimal parameters, highlighting the connection between the standard errors and $\{E_t\}_{t=1}^{692}$.

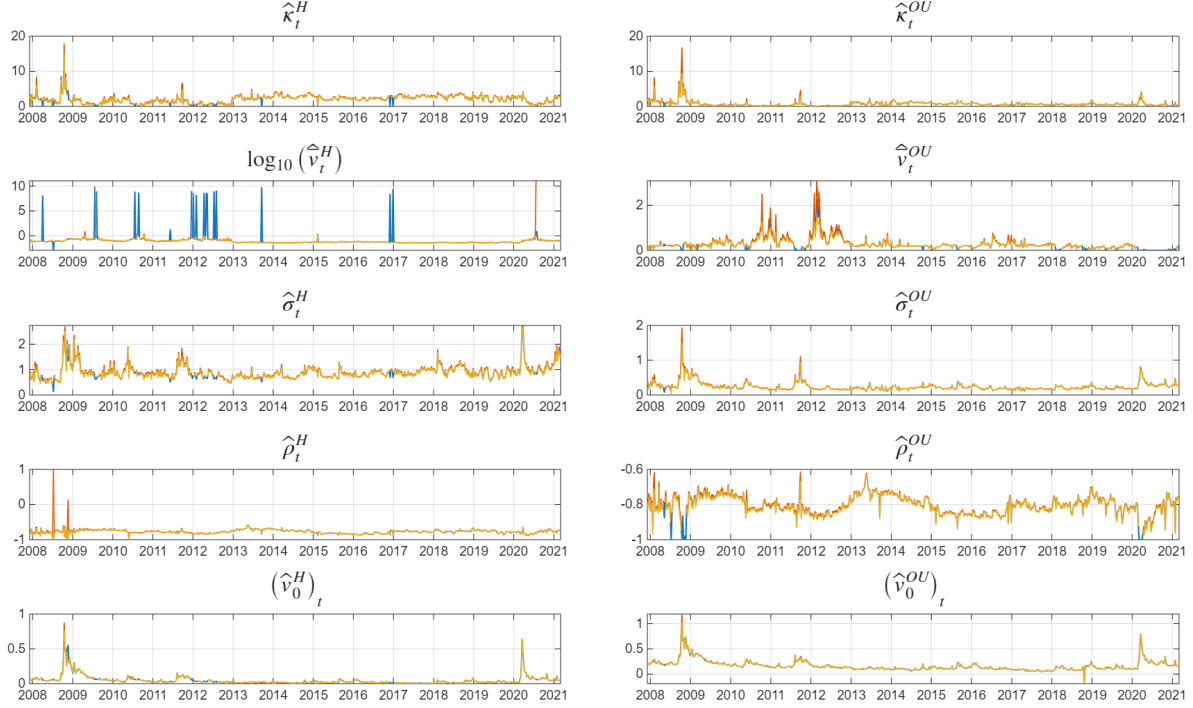


Figure 5.7: Time series of the estimated parameters, together with the upper and lower bounds of the $CI_{95\%}$, represented by the blue, orange, and yellow lines, respectively.

5.2.5 Identifiability Problem in the Heston Model

This Section aims to exploit the identifiability problem related to the parameters κ and $\bar{v}$ in the Heston model. I want to start this discussion by recalling that this kind of problems are already well known in the literature. Indeed, as reported by Cui et al. (2017), it exists a relation between κ , $\bar{v}$ and σ . In particular, they are related by

$$\lim_{t \rightarrow +\infty} \text{Var}(v_t) = \frac{\sigma^2 \bar{v}}{\kappa}$$

This relation shows that: large values of κ and σ can generate similar results to those obtained when they take small values.

Considering this problem, Clark (2011) proposed fixing

$$\kappa = \frac{2.75}{\tau}, \quad \bar{v} = \sigma_{\text{imp}}^{ATM}(\tau)$$

where $\sigma_{\text{imp}}^{ATM}(\tau)$ denotes the implied volatility of ATM options and τ the time to maturity.

This is a heuristic dimension reduction technique, indeed, the resulting calibration problem is defined on a three-dimensional space. Moreover, since κ and $\bar{v}$ are fixed, the identifiability problem is eliminated.

In this work, I also addressed this issue by defining the reparameterized calibration problem, but unfortunately, without success, indeed, modifying the geometry of the parameter space $\mathbb{R}^5$ does not appear to be effective in this regard. Then, I focused my analysis to understand what is the source of the identifiability by studying the matrix $\mathbf{A}$ and its eigenvalues and I investigated whether the OU model exhibits the same issue.

I present the optimal solution obtained by calibrating the Heston model, under the characterization (4.2), on 01/02/2012:

$$\hat{\boldsymbol{\theta}} = \left(5.2098 \times 10^{-10}, 1.4066 \times 10^8, 0.63, -0.8290, -3.4488 \right)^T$$

and the corresponding matrix $\mathbf{A}$ is reported in Table 5.15.

	$\partial\kappa$	$\partial\bar{v}$	$\partial\sigma$	$\partial\rho$	∂v_0
$\partial\tilde{\kappa}$	9.3×10^{-3}	9.3×10^{-3}	-8×10^{-3}	1×10^{-3}	7.9×10^{-3}
$\partial\bar{v}$	9.3×10^{-3}	9.3×10^{-3}	-8×10^{-3}	1×10^{-3}	7.9×10^{-3}
$\partial\sigma$	-8×10^{-3}	-8×10^{-3}	7.7×10^{-3}	-1.2×10^{-3}	-6.7×10^{-3}
$\partial\rho$	1×10^{-3}	1×10^{-3}	-1.2×10^{-3}	3×10^{-4}	7×10^{-4}
∂v_0	7.9×10^{-3}	7.9×10^{-3}	-6.7×10^{-3}	7×10^{-4}	8.2×10^{-3}

Table 5.15: Matrix $\mathbf{A}$ of the Heston model under (4.2) on 01/02/2012

The first and second rows are equal, so the matrix is singular, for this reason, by computing its eigenvalues, I obtained the following vector

$$\boldsymbol{\lambda} = \left(0, 10^{-4}, 6 \times 10^{-4}, 1.4 \times 10^{-3}, 3.28 \times 10^{-2} \right)^T$$

I found that $\lambda_{\min} = 0$ and the corresponding principal direction is proportional to

$$\mathbf{z} = (-1, 1, 0, 0, 0)^T$$

which corresponds to opposite variations of κ and $\bar{v}$; along this direction, the objective function exhibits zero curvature, indeed

$$\mathbf{A}\mathbf{z} = \mathbf{0}$$

This indicates the presence of a flat direction in the neighborhood of $\hat{\boldsymbol{\theta}}$, suggesting a local identifiability issue only between κ and $\bar{v}$.

To further investigate the presence of a flat valley, I created a contour plot with two hundred level curves, by fixing the last elements in the vector of the optimal solution

found and varying

$$\tilde{\theta}_1 \in (-26, -3), \quad \tilde{\theta}_2 \in (-3, 20)$$

The contour plot reported in Figure 5.8 clearly shows the presence of a pronounced flat valley, corresponding to a region of the parameter space in which different combinations of $\tilde{\kappa}$ and $\tilde{v}$ produce very similar values of the objective function. This explains why the algorithm is unable to uniquely identify reasonable values for these two parameters. Moreover, the portion of the parameter space in which $\tilde{\kappa}$ and $\tilde{v}$ are generated lies within the flat valley.

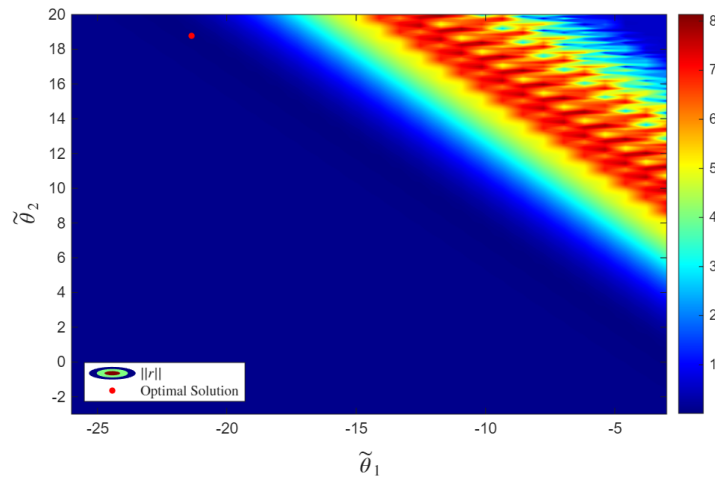


Figure 5.8: Contour plot of $\|r\|$ on 01/02/2012

Considering the OU model, the algorithm identified the following optimal solution

$$\hat{\theta}^{OU} = \left(3.3 \times 10^{-2}, 1.9043, 0.1499, -0.8685, 0.1493\right)^T$$

As can be seen, $\hat{\theta}^{OU}$ contains reasonable values for the model parameters; however, the condition number is equal to 1.3325×10^7 , then it violates the condition (3.10). As in the case of the Heston model, to assess whether the objective function exhibits irregular behavior or a flat valley in the neighborhood of the optimal solution, I present the approximated Hessian matrix, evaluated at $\hat{\theta}^{OU}$, in Table 5.16.

	$\partial\kappa$	$\partial\bar{v}$	$\partial\sigma$	$\partial\rho$	∂v_0
$\partial\tilde{\kappa}$	5.396×10^{-1}				
$\partial\bar{v}$	1.28×10^{-2}	3×10^{-4}			
$\partial\sigma$	2.229×10^{-1}	6.1×10^{-3}	2.227×10^{-1}		
$\partial\rho$	8.38×10^{-2}	2×10^{-3}	2.49×10^{-2}	2.11×10^{-2}	7×10^{-4}
∂v_0	8.526×10^{-1}	2.09×10^{-2}	4.505×10^{-1}	1.415×10^{-1}	1.6041

Table 5.16: Matrix $\mathbf{A}$ of the OU model on 01/02/2012

The matrix $\mathbf{A}$ is better scaled, although the curvature of the objective function is

less sensitive with respect to variations in $\bar{v}$ than to variations in the other parameters. This suggests that, along the direction orthogonal to $\bar{v}$, could exist a flat valley.

To make a comparison with the Heston model, I reported in Figure 5.9 the contour plot with two hundred level curves and

$$\tilde{\kappa} \in (0, 0.1), \quad \tilde{v} \in (0, 2.5)$$

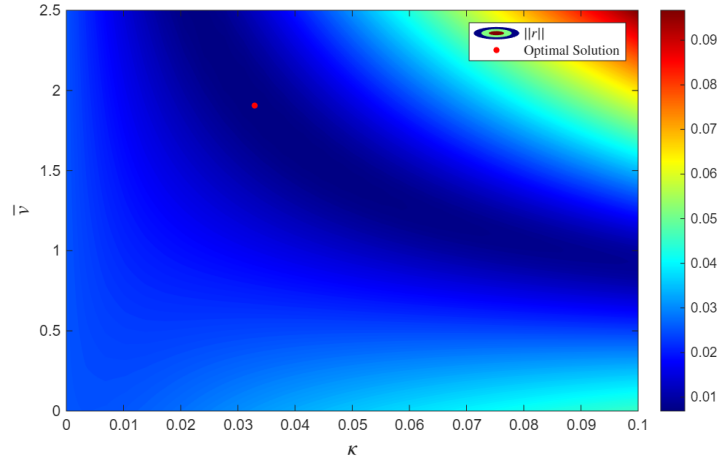


Figure 5.9: Contour plot of $\|r\|$ on 01/02/2012

The contour plot shows that, even for the OU model, there is a flat valley along the directions associated with κ and $\bar{v}$; however, given its scale, this valley is very small, as confirmed by the value of the condition number.

To conclude, I can state that there is no evidence of an identifiability issue regarding the parameters κ and $\bar{v}$ under the OU model.

5.2.6 The η parameter

In the previous section, I presented the identifiability problem related to the parameters κ and $\bar{v}$ in the Heston model, observing that the objective function exhibits a flat valley along the principal direction

$$\mathbf{z} = (-1, 1, 0, 0, 0)^\top$$

Trying to overcome this issue, I introduced a new parameter in the model, defined as

$$\eta = \kappa \bar{v}$$

and the corresponding parameter vector

$$\boldsymbol{\theta}_\eta = (\kappa, \eta, \sigma, \rho, v_0)^\top$$

Since $\bar{v}$ was no longer a component of $\boldsymbol{\theta}_\eta$, I also modified the CHF, obtaining

$$\phi_\eta(\boldsymbol{\theta}; u, T) = \exp\left(iu \log\left(\frac{F}{S_0}\right) - \frac{\eta\rho iuT}{\sigma} - A + \frac{2\eta}{\sigma^2}D\right)$$

where the quantities appearing in the expression are defined as in Section 3.2.

To find $\hat{\boldsymbol{\theta}}_\eta$, I used the constrained optimization described in the Section 4.1, with feasible region

$$\Theta_\eta^H = \left\{ \boldsymbol{\theta} \in \mathbb{R}^5 : \kappa, \eta, \sigma, v_0 > 0, \rho \in [-1, 1] \right\}$$

I inferred on $\hat{\bar{v}}$ using

$$\hat{\bar{v}} = \frac{\hat{\eta}}{\hat{\kappa}}$$

This different characterization of the Heston model solved the identifiability problem, but, as a consequence, led to substantial instability in the behavior of the time series associated with the parameters σ , ρ and v_0 . In particular, v_0 , very frequently, tended to approach zero, suggesting that this parameter had little influence in the model.

The most significant factor that led me to abandon this characterization is that, in almost all cases, I observed an increase in MSE , even though it required, on average, almost 40% less computational time to perform the optimization than the Heston model under the constrained calibration problem.

Chapter 6

Model Selection

After all the experiments, I obtained several results concerning the calibration of each model, therefore, I provide a useful summary of the main findings.

My objective was to study the behavior of both the Heston and the OU models to propose possible approaches for improving their calibration performance on real-world data. Building on the results obtained, I also propose an approach for selecting the most appropriate model to perform the calibration on each trading day in the sample.

6.1 Summary of Discoveries

The calibration procedure proposed by Cui et al. (2017), when applied to the models considered, achieved excellent accuracy on simulated data, however, it appears that, when using real-world data, more structured characterizations of the calibration problem should be adopted, because the time series vector $\{\hat{\boldsymbol{\theta}}_t\}_{t=1}^{692}$ may take values outside the feasible region. For this reason, I proposed both the constrained and the reparameterized calibration approaches.

Applying the second one on the Heston model, the solution found belongs to the feasible region and, moreover, the computational time is almost ten times lower than that achieved by the model under the constrained characterization. It leads to better results in terms of number of *Fails in* $\delta_t(\mathbf{A})$, reducing it by approximately 50%, indeed, the problem is ill-conditioned in only 3% of the cases examined. By analyzing these cases in which the optimizer fails to find the optimal solution $\hat{\boldsymbol{\theta}}$, I found that, in some of those days, an identifiability problem between κ and $\bar{v}$ arose.

Considering the OU model, as defined by Brignone and Sgarra (2026), in contrast with the Heston one, under the constrained calibration problem, it does not exhibit multiple local minima and, consequently, the solution found by the optimizer is stable, even during periods of financial crises. However, $\{E_t^{OU}\}_{t=1}^{692}$, as shown in Figure 6.1, takes highest values compared with those related to the $\{E_t^{REP}\}_{t=1}^{692}$.

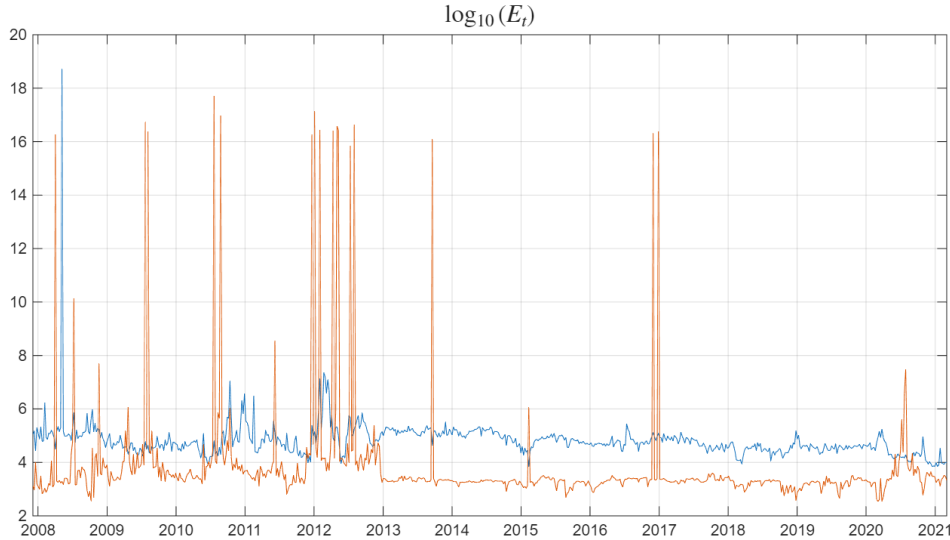


Figure 6.1: Time series of E_t^{OU} and E_t^{REP} . Blue line: OU under (4.1); Orange line: Heston under (4.2).

The only disadvantage I found for this model is its high computational time. This highlights a clear trade-off between the model's ability to fit market prices accurately and the computational time required for calibration.

6.2 Condition Number Approach

In the Chapter 1, I reported that performing a calibration procedure is not straightforward and involves several challenges. The *model risk* and the *calibration risk* are the most relevant; as argued by Ballotta (2024), it is difficult to reduce the risk associated to the selected model and avoiding it requires to being confident that the model is the best one able to fit the market data, either prices or implied volatilities. Even assuming that this is possible, the calibration risk remains to be reduced; indeed, the objective function considered is non-convex and irregular, exhibiting multiple local minima in which the optimizer may become stuck.

Considering the results described in Section 6.1, I believe that, based on the market data available at a given time t , it is possible to select the most appropriate model to calibrate on that day. I propose a model selection procedure based on the meaning of the condition number, identifying as *most appropriate model*, the one with associated the lowest value.

To apply this procedure, I suggest to use models which are, at least in a broad sense, comparable. I propose the Heston model under (4.2) and the OU model under (4.1), because they do not exhibit a condition number higher than 10^7 on the same days. In this sense, they can be considered complementary, indeed when the first is ill-conditioned, the second is well-conditioned and vice versa; there is only one exception: 01/02/2012. However, it represents just one case in a sample spanning 14 years period.

To continue, I introduce the concepts of *main model* and the *auxiliary model*. I

calibrate the former and compute its condition number E_t^M for each day in the sample, whereas the latter is calibrated when

$$E_t^M > 10^7 \quad (6.1)$$

I adopted Heston under (4.2) as the *main model*, because of its computational efficiency; instead, exploiting its convergence to the optimal parameter set, I selected OU as *auxiliary model*. The idea is to perform an ultra-fast calibration and find the optimal parameter set, but, at the same time, if $\hat{\theta}_t^M$ is not enough stable, a more stable model is used, even if it is computationally slower. The algorithm performs two calibrations only on days in which it is violated the (6.1). This is not too computationally demanding if performed using the models considered above, because if I use the multistart procedure described in Section 5.2.1 and calibrate both models on every day in the sample, the algorithm requires only 1.47 seconds on average; just to make a comparison, the Heston model under (4.1) required approximately 2 seconds.

To make the optimization even faster, I suggest adopting the multistart procedure, but generating only one initial guess for both models, since both exhibit strong convergence toward the optimal solution. However, the OU model exhibits the strongest convergence, but, being the slowest to calibrate, it is used only when the Heston model under (4.2) is unable to identify a stable optimal solution.

I provide a schematic representation of the proposed procedure in Algorithm 6.2.1.

Algorithm 6.2.1 Calibration using the condition number approach

```

1: for  $t = 1, \dots, 692$  do
2:   Recover  $\mathbf{C}_t^*(\mathbf{K}, \mathbf{T})$  through BS formula
3:   Generate the initial guess  $\theta_{0,t}^M$ 
4:   Use the optimizer to find  $\hat{\theta}_t^M$ 
5:   Compute  $E_t^M$ 
6:   if  $E_t^M > 10^7$  then
7:     Generate the initial guess  $\theta_{0,t}^A$ 
8:     Use the optimizer to find  $\hat{\theta}_t^A$ 
9:   end if
10: end for

```

I report in Table 6.1 the results obtained by applying this procedure over the period 2007–2021. I want to emphasize that, in this case, I did not measure only the time required to calibrate the *main model*, but also the time employed to compute its condition number and, for days that violated condition (6.1), the one spent to calibrate the *auxiliary model*.

	Cond. Approach
MSE	1.249×10^{-6}
Time	0.29
Fails in E_t	1
Number of switches	20

Table 6.1: Results obtained applying the condition number approach over the period 2007–2021

As can be seen from the average computational time, this procedure is almost seven times faster than the one required to calibrate the Heston model under the constrained characterization. This makes the proposed approach suitable for real-world applications. Although I used randomly generated points that differ from those employed in the experiment described in Section 5.2.3, the number of days in which the algorithm calibrated both models, as reported in the *Number of switches*, remained equal to 20, further confirming their strong convergence towards the optimal solution in the other days.

Finally, as shown in Figure 6.2, I obtained a stable time series of the condition number, reducing the number of *Fails in E_t* to only one day while maintaining the same M_t time series shown in Figure 5.3. This indicates that the proposed approach is also able to achieve a high level of accuracy.

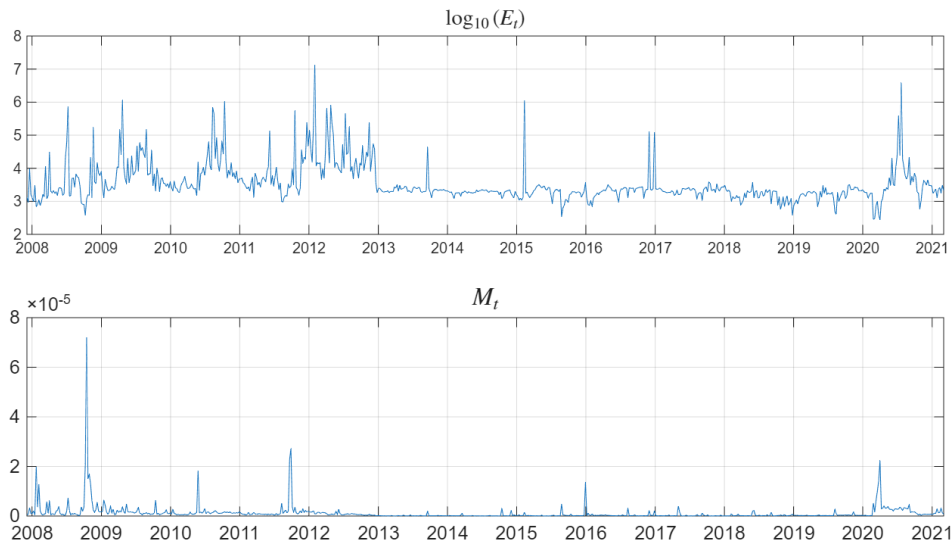


Figure 6.2: Time series of E_t and M_t obtained by applying the condition number approach.

Chapter 7

Conclusions

In this work, I considered both the Heston and Ornstein–Uhlenbeck stochastic volatility models and provided a detailed analysis of their mathematical foundations, showing which are the differences and their implications on real–world data. I followed the procedure proposed by Cui et al. (2017) to calibrate both models on simulated data, obtaining a percentage of *Fails in MSE* within the 2% and 1.3% respectively; however, I found that the calibration problem requires more sophisticated characterizations to be tractable on real–world data, because it provides optimal solutions which lie outside the feasible region identified for both models.

I proposed to build a constrained and a reparameterized calibration problem and showed that their fundamental difference is the following: the former reduced the search space to the feasible region by imposing the constraints, whereas the latter preserves the unconstrained optimization structure by inducing a geometric transformation of the search space.

Testing these on simulated data, I found that, under both characterizations, the Heston model achieves better performance in terms of *Fails in MSE*. In contrast with it, the OU model seems to suffer from the imposition of the constraints, increasing the error rate to 10%.

On real–world data, I applied the Heston model under both the proposed characterizations and the OU model under the constrained one. Firstly, I performed a single day experiment, investigating in details the differences between the performance achieved by both models. I conducted these analysis using the approximated Hessian matrix in the neighborhood of the optimal solution and its eigenvalue spectrum to check whether the objective function’s curvature exhibits different behavior along the orthogonal and principal directions.

I concluded that the Heston model is able to achieve the same level of accuracy obtained by the OU model only under the reparameterized calibration problem, whereas the latter inherently exhibits a structure which allows for better scaling of the approximated Hessian matrix.

Calibrating weekly both models over a time period from 2007–2021, I concluded that the Heston model under the constrained approach is too computationally expensive and

achieves, on average, the lowest level of accuracy, showing that: the optimal solution depends on the initial starting point provided to the optimizer and an identifiability issue arises between the long-term variance parameter and the one which represents the speed of the mean-reversion. Therefore, it is not well suited to real-world applications.

The Heston model, under the proposed reparameterization, instead, obtained better results, in particular the number of ill-conditioned days decreases approximately of 50% and, most importantly, it can be calibrated, on average, in 0.2 seconds, almost 10 time faster than the time obtained by applying the constrained approach. However, the identifiability persisted even applying the reparameterization.

In contrast with it, the OU model recorded the lowest number of ill-conditioned days, showing that it is able to identify a stable optimal solution in the 99% of the cases without exhibiting the same identifiability problem. This suggests that the quadratic structure of the characteristic function allows for the identification of a reasonable set of optimal parameters in almost all the trading days considered. Then, it is possible to conclude that the objective function does not exhibit multiple local minima. However, this model registered, on average, the highest computational time.

Finally, I proposed a calibration approach based on the condition number; in particular, to address the *calibration conundrum*, I developed an algorithm composed by the Heston model under the reparameterized calibration problem and the OU model under the constrained one. The former is calibrated on each trading day and when the optimal solution results to be not enough stable, then the OU model is calibrated. Applying this procedure, I was able to reduce the ill-conditioned days to one, indicating that the algorithm identified a stable parameter set in the 99% of the examined cases, while requiring a computational time of 0.29 seconds. These results show that the algorithm meet the requirements of real-world applications.

There is still room for improvement in this calibration procedure. For example, future developments could include jumps into the characteristic function, as proposed by Bates (1996) for the Heston model, or developing a different reparameterization which is able to solve the identifiability issue.

References

- Albrecher, H., Mayer, P., Schoutens, W., and Tistaert, J. (2007). The little Heston trap. *Wilmott Magazine*, 2007(January), 83–92.
- Baldi, P. (2017). *Stochastic Calculus: An Introduction Through Theory and Exercises*. Universitext. Springer.
- Ballotta, L. 2024. The Calibration Conundrum. *Wilmott*, 2024(134), 18–20.
- Bates, D. S. (1996) Jumps and stochastic volatility: Exchange rate processes implicit in Deutsche Mark options, *The Review of Financial Studies* 9, 69–107.
- Black, F., and Scholes, M. (1973). The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3), 631–659
- Brignone, R. and Sgarra, C. (2026), Full calibration, fast simulation, and exotic option pricing under the Ornstein–Uhlenbeck driven stochastic volatility model. *European Journal of Operational Research*.
- Brignone, R., Gonzato, L., Knaust, S., and Lütkebohmert, E. (2026). Fast Bayesian Calibration of Option Pricing Models Based on Sequential Monte Carlo Methods and Deep Learning. *Journal of Financial Econometrics*, 24(3).
- Brzeźniak, Z. and Zastawniak, T. (1999). *Basic Stochastic Processes: A Course Through Exercises*. Springer–Verlag London.
- Carr, P., and Madan, D., B., (1999). Option valuation using the fast Fourier transform. *Journal of Computational Finance*, 2(4), 61–73.
- Clark, I. J. (2011). *Foreign exchange option pricing: A practitioner’s guide*. Chichester: Wiley.
- Cox, J. C., Ingersoll, J. E., and Ross S. A. (1985). A Theory of the Term Structure of Interest Rates, *Econometrica*, 53, 363–384.
- Cui, Y., del Bano Rollin, S. and Germano, G. (2017) Full and fast calibration of the Heston stochastic volatility model. *European Journal of Operational Research*, 263, 625–638.
- Davidson, J. (2000). *Econometric Theory*. Blackwell Publishers.
- Duffie, D., Filipovic, D. and Schachermayer, W. (2003). Affine processes and applications in finance. *The Annals of Applied Probability*, 13, 984–1053.
- Heston, S. L. (1993). A closed-form solution for options with stochastic volatility and applications to bond and currency options. *Review of Financial Studies*, 6(2), 327–343.
- Hull, J. C. and White, A. (1987) The pricing of options on assets with stochastic volatilities, *Journal of Finance* 42, 281–300.

- Kahl, C., and Jäckel, P. (2005). Not-so-complex logarithms in the Heston model. *Wilmott Magazine*, 2005(September), 94–103.
- Lamberton, D. and Lapeyre, B. (2007). *Introduction to Stochastic Calculus Applied to Finance*, Second Edition. Chapman & Hall/CRC.
- Merton, R. (1976) Option pricing when underlying stock returns are discontinuous. *Journal of Financial Economics*, 3, 125–144.
- Moré, J. J. (1978). The Levenberg-Marquardt algorithm: implementation and theory. In G. A. Watson (Ed.), *Numerical analysis. In Lecture notes in mathematics: Vol. 630*, 105–116. Berlin: Springer.
- Pascucci, A. (2024). *Probability Theory I: Random Variables and Distributions*. Springer, Cham.
- Quarteroni, A., Sacco, R. and Saleri, F. (2007). *Numerical Mathematics*, Second Edition. Springer.
- Schöbel, R. and Zhu, J. (1999) Stochastic volatility with an Ornstein–Uhlenbeck process: An extension. *European Finance Review*, 3(1), 23–46.
- Scott, L. O. (1987). Option pricing when the variance changes randomly: Theory, estimation and an application, *Journal of Financial and Quantitative Analysis* 22(4), 419–438
- Stein, E. M. and Stein, J. C. (1991). Stock price distributions with stochastic volatility: An analytic approach, *The Review of Financial Studies* 4(4), 727–752.
- Wiggins, J. B. (1987). Option values under stochastic volatility: Theory and empirical estimates, *Journal of Financial Economy* 19, 351–372.
- Yeap, C., S. Kwok, and S. Boris Choy (2018). A Flexible Generalized Hyperbolic Option Pricing Model and Its Special Case. *Journal of Financial Econometrics* 16, 425–460.

Appendix: MATLAB Codes

Characteristic Function

Heston CHF

```
1 function [phi] = CHF_Heston_vec(u_k, T, S0, r, k, ...
2     v_bar, sigma, rho, v0)
3 i = 1i;
4 xi = k-sigma*rho*i*u_k;
5 d = sqrt(xi.^2+sigma^2*(u_k.^2+i*u_k));
6 A1 = (u_k.^2+i*u_k).*sinh(d*T/2);
7 A2 = d/v0.*cosh(d*T/2)+xi/v0.*sinh(d*T/2);
8 A = A1./A2;
9 B = (d*exp(k*T/2))./(v0*A2);
10 D = log(B);
11 F = S0*exp(r*T);
12 phi = exp(i*u_k*log(F/S0)-(k*v_bar*rho*T*i*u_k) ...
13     /sigma-A+(2*k*v_bar)/(sigma^2)*D);
14 end
```

Heston CHF under Reparameterized Calibration Problem

```
1 function [phi] = CHF_Heston_vec_tr(u_k, T, S0, r, ...
2     k_tl, v_bar_tl, sigma_tl, rho_tl, v0_tl)
3 i = 1i;
4 xi = exp(k_tl)-exp(sigma_tl)*(2*logsig(rho_tl)-1)*i.*u_k;
5 d = sqrt(xi.^2+exp(sigma_tl)^2*(u_k.^2+i.*u_k));
6 A1 = (u_k.^2+i.*u_k).*sinh(d.*T/2);
7 A2 = d./exp(v0_tl).*cosh(d.*T/2)+xi./exp(v0_tl).*sinh(d.*T/2);
8 A = A1./A2;
9 B = d.*exp(exp(k_tl).*T/2)./(exp(v0_tl).*A2);
10 D = log(B);
11 F = S0*exp(r.*T);
12 phi = exp(i.*u_k.*(log(F/S0))-T.*exp(k_tl).*exp(v_bar_tl)* ...
13     (2*logsig(rho_tl)-1).*i.*u_k./exp(sigma_tl) ...
```

```

14     -A+2.*exp(k_tl).*exp(v_bar_tl)./exp(sigma_tl)^2.*D);
15 end

```

Ornstein–Uhlenbeck CHF

```

1 function [phi, H, L, G, Q2] = CHF_OU_vec(u_k, T, r, k, ...
2     v_bar, sigma, rho, v0)
3 i = 1i;
4 gamma1 = sqrt(2*sigma^2*(u_k.^2/2*(1-rho^2)+ ...
5     i*u_k/2.*(1-2*k*rho/sigma))+k.^2);
6 gamma2 = 1./gamma1.*(k-sigma*i*u_k*rho);
7 gamma3 = k^2*v_bar-sigma*i*u_k.*k*v_bar*rho;
8 zeta1 = cosh(gamma1*T);
9 zeta2 = sinh(gamma1*T);
10 L = 1./(2*sigma^2).*(k-(gamma1.*zeta2+gamma1.*gamma2.* ...
11     zeta1)./(zeta1+gamma2.*zeta2))-rho*i*u_k./(2*sigma);
12 H = -1/2*rho*i*u_k*sigma*T+k*T/2+(k^2*v_bar^2* ...
13     gamma1.^2-gamma3.^2)./(2*sigma.^2*gamma1.^3) ...
14     .*(zeta2./(zeta1+gamma2.*zeta2)-gamma1*T)+((k*v_bar* ...
15     gamma1-gamma2.*gamma3).*gamma3)./ ...
16     (sigma^2*gamma1.^3).*((zeta1-1)./(zeta1+gamma2.*zeta2));
17 G = 1./(sigma^2.*gamma1).*(-v_bar*k*gamma1+(k*v_bar* ...
18     gamma1-gamma2.*gamma3+gamma3.*zeta2+gamma2.* ...
19     gamma3.*zeta1)./(zeta1+gamma2.*zeta2));
20 Q2 = -1/2*gamma1*T-1/2*log(1/2*(1+gamma2+ ...
21     exp(-gamma1*T).*(1-gamma2)));
22 phi = exp(i*u_k*r*T+H+L*v0.^2+G*v0+Q2);
23 end

```

Analytical Gradient

Heston h vector

```

1  clc;clear;close all;
2  syms k v_bar sigma rho v0
3  syms r T u_k S0 real
4  assume(k>0)
5  assume(v_bar>0)
6  assume(sigma>0)
7  assume(rho>-1 & rho<1)
8  assume(v0>0)
9  logcf = log(CHF_Heston_vec(u_k, T, S0, r, k, ...
10     v_bar, sigma, rho, v0));
11 h(1,1) = diff(logcf,k);
12 h(1,2) = diff(logcf,v_bar);
13 h(1,3) = diff(logcf,sigma);
14 h(1,4) = diff(logcf,rho);
15 h(1,5) = diff(logcf,v0);
16 f = matlabFunction(h, ...
17     'File','Vector_h_Heston','Vars',[u_k T S0 r ...
18     k v_bar sigma rho v0],...
19     'Outputs',{'h'});

```

Heston reparameterized h vector

```

1  clc;clear;close all;
2  syms k_tl v_bar_tl sigma_tl rho_tl v0_tl
3  syms r T u_k S0
4  logcf = log(CHF_Heston_vec_tr(u_k, T, S0, r, k_tl, ...
5     v_bar_tl, sigma_tl, rho_tl, v0_tl));
6  h(1,1) = diff(logcf,k_tl);
7  h(1,2) = diff(logcf,v_bar_tl);
8  h(1,3) = diff(logcf,sigma_tl);
9  h(1,4) = diff(logcf,rho_tl);
10 h(1,5) = diff(logcf,v0_tl);
11 f = matlabFunction(h, ...
12     'File','Vector_h_Heston_tr','Vars',[u_k T S0 r ...
13     k_tl v_bar_tl sigma_tl rho_tl v0_tl],...
14     'Outputs',{'h'});

```

Ornstein–Uhlenbeck h vector

```

1  clc;clear;close all;
2  syms k v_bar sigma rho v0
3  syms r T u_k
4  assume(k>0)
5  assume(v_bar>0)
6  assume(sigma>0)
7  assume(rho>-1 & rho<1)
8  assume(v0>0)
9
10 logcf = log(CHF_OU_vec(u_k, T, r, k, ...
11     v_bar, sigma, rho, v0));
12 h(1,1) = diff(logcf,k);
13 h(1,2) = diff(logcf,v_bar);
14 h(1,3) = diff(logcf,sigma);
15 h(1,4) = diff(logcf,rho);
16 h(1,5) = diff(logcf,v0);
17
18 f = matlabFunction(h, ...
19     'File','Vector_h_OU','Vars',[u_k T r ...
20     k v_bar sigma rho v0],...
21     'Outputs',{'h'});

```

Residual vector

BS Pricing Formula

```

1 function [MKT_C] = BLS_price(S0, K, r, T, sigma)
2     d1 = (log(S0/K) + ...
3         ((r+sigma^2/2)*T))/(sigma*sqrt(T));
4     d2 = d1-sigma*sqrt(T);
5     Nd1 = normcdf(d1);
6     Nd2 = normcdf(d2);
7     MKT_C = S0*Nd1 - K*exp(-r*T)*Nd2;
8 end

```

Model Prices and Jacobian Matrix

```

1 function [residual, J] = residual_opt_Heston(theta, ...
2     S0, K, r, T, MKT_C, u_k, w_k)
3 k = theta(1);
4 v_bar = theta(2);
5 sigma = theta(3);
6 rho = theta(4);
7 v0 = theta(5);
8 C = zeros(length(K),1);
9 J = zeros(length(K),length(theta));
10 for n=1:length(T)
11     phi1 = CHF_Heston_vec(u_k-1i,T(n), ...
12         S0,r,k,v_bar,sigma,rho,v0);
13     h1 = Vector_h_Heston(u_k-1i,T(n),S0,r,k,v_bar,sigma,rho,v0);
14     J1 = phi1.*h1;
15     phi2 = CHF_Heston_vec(u_k,T(n),S0,r,k,v_bar,sigma,rho,v0);
16     h2 = Vector_h_Heston(u_k,T(n),S0,r,k,v_bar,sigma,rho,v0);
17     J2 = phi2.*h2;
18     %Compute the model price
19     C(n,1) = 0.5*(S0-K(n)*exp(-r*T(n)))+exp(-r* ...
20         T(n))/pi*(S0*(w_k*(real(exp(-1i*u_k* ...
21         log(K(n)/S0))./(1i.*u_k).*phi1).*exp(u_k))) - ...
22         K(n)*(w_k*(real(exp(-1i*u_k*log(K(n)/S0))./ ...
23         (1i.*u_k).*phi2).*exp(u_k))));
24     %Compute the Jacobian matrix
25     J(n,:) = exp(-r*T(n))/pi*(S0*(w_k*(real(exp(-1i* ...
26         u_k*log(K(n)/S0))./(1i.*u_k).*J1).*exp(u_k))) - ...
27         K(n)*(w_k*(real(exp(-1i*u_k* ...
28         log(K(n)/S0))./(1i.*u_k).*J2).*exp(u_k))));

```

```

29 end
30 residual = (C - MKT_C)';
31 end

```

Simulated and Real-world Data

Recover Strike Price on Simulated Data

```

1 function [K, MKT_C] = find_K_C(CHF, S0, r, T, delta, u_k, w_k)
2 i = 1i;
3 C = @(K) 0.5*(S0-K*exp(-r*T))+exp(-r*T)/pi*(S0*(w_k*( ...
4     real(exp(-i*u_k*log(K/S0))./(i.*u_k).*CHF(u_k-i)).* ...
5     exp(u_k))) - K*(w_k*(real(exp(-i*u_k*log(K/S0))./( ...
6     (i.*u_k).*CHF(u_k)).*exp(u_k))));
7 if delta < 0
8     K = fsolve(@(K) delta - (blsdelta(S0, K, r, T, max(0, ...
9         blsimpv(S0, K, r, T, max(0,C(K)))))-1), S0);
10 else
11     K = fsolve(@(K) delta - blsdelta(S0, K, r, T, max(0, ...
12         blsimpv(S0, K, r, T, max(0,C(K))))), S0);
13 end
14 MKT_C = C(K);
15 end

```

Validation Procedure

```

1 clc; clear all; close all;
2 %Inputs
3 S0 = 1; r = 0.02; N = 64;
4 %Prepare vectorized integration
5 [u_k, w_k] = Gaulagwt(N);
6 %Maturities in years for the synthetic volatility surface
7 T = [30 60 90 120 150 180 252 360]/360;
8 %Deltas for the synthetic volatility surface
9 delta = [-0.1 -0.25 0.5 0.25 0.1];
10 %Define the bound for the uniform sample
11 bounds = [0.5 5; %k
12     0.05 0.95; %v_bar
13     0.05 0.95; %sigma
14     -0.90 -0.1; %rho
15     0.05 0.95]; %v0
16 %Define optimization options

```

```

17 options1 = optimoptions('fsolve', 'Algorithm', ...
18     'levenberg-marquardt', ...
19     'Display', 'off', ...
20     'FunctionTolerance', 1e-10, ...
21     'StepTolerance', 1e-10, ...
22     'OptimalityTolerance', 1e-10, ...
23     'MaxFunctionEvaluations', 5000, ...
24     'MaxIterations', 5000);
25 options2 = optimoptions('fsolve', 'Algorithm', ...
26     'levenberg-marquardt', ...
27     'SpecifyObjectiveGradient', true, 'Display', ...
28     'off', 'FunctionTolerance', 1e-10, ...
29     'StepTolerance', 1e-10, 'OptimalityTolerance', 1e-10, ...
30     'MaxFunctionEvaluations', 5000, ...
31     'MaxIterations', 5000);
32 lower = [0, 0, 0, -1, 0];
33 upper = [inf, inf, inf, 1, inf];
34 options3 = optimoptions('lsqnonlin', 'Algorithm', ...
35     'levenberg-marquardt', ...
36     'SpecifyObjectiveGradient', true, 'Display', 'off', ...
37     'FunctionTolerance', 1e-10, ...
38     'StepTolerance', 1e-10, 'OptimalityTolerance', 1e-10, ...
39     'MaxFunctionEvaluations', 5000, ...
40     'MaxIterations', 5000);
41 %Validation procedure
42 count = 0;
43 for true = 1:100
44     %Generate true parameters
45     theta_true(:,true) = bounds(:,1)+rand(5,1).* ...
46     (bounds(:,2)-bounds(:,1));
47     k = theta_true(1,true);
48     v_bar = theta_true(2,true);
49     sigma = theta_true(3,true);
50     rho = theta_true(4,true);
51     v0 = theta_true(5,true);
52     n = 0;
53     %Recover strikes and the market prices
54     for i = 1:length(T)
55         CHF = @(u_k) CHF_Heston_vec(u_k, T(i), S0, r, ...
56             k, v_bar, sigma, rho, v0);
57         for j = 1:length(delta)
58             n = n + 1;
59             [K(n,1) MKT_C(n,1)] = find_K_C(CHF, S0, ...
60                 r, T(i), delta(j), u_k, w_k);

```

```

61         T_ordered(n,1) = T(i);
62     end
63 end
64 for tempt = 1:100
65     count = count+1;
66     %Generate the starting point
67     theta0(:,count) = bounds(:,1)+rand(5,1).* ...
68     (bounds(:,2)-bounds(:,1));
69     tic
70     [x1(:,count), residual1, exitflag, output1] = ...
71     fsolve(@(theta) residual_opt_Heston(theta, S0, ...
72     K, r, T_ordered, MKT_C, u_k, w_k), ...
73     theta0(:,count), options1);
74     time1(count) = toc;
75     abs_diff_1(:,count) = abs(theta_true(:,true) ...
76     -x1(:,count));
77     MSE1(count) = mean(residual1.^2);
78     iteration1(count) = output1.iterations;
79     fun_eval1(count) = output1.funcCount;
80     tic
81     [x2(:,count), residual2, exitflag, output2] = ...
82     fsolve(@(theta) residual_opt_Heston(theta, S0, ...
83     K, r, T_ordered, MKT_C, u_k, w_k), ...
84     theta0(:,count), options2);
85     time2(count) = toc;
86     abs_diff_2(:,count) = abs(theta_true(:,true) ...
87     -x2(:,count));
88     MSE2(count) = mean(residual2.^2);
89     iteration2(count) = output2.iterations;
90     fun_eval2(count) = output2.funcCount;
91     tic
92     [x3(:,count), resnorm, residual3, exitflag, output3] =
93     lsqnonlin(@(theta) ...
94         residual_opt_Heston(theta, S0, K, r, T_ordered, ...
95         MKT_C, u_k, w_k), theta0(:,count), ...
96         [], [], options3);
97     time3(count) = toc;
98     abs_diff_3(:,count) = abs(theta_true(:,true) ...
99     -x3(:,count));
100    MSE3(count) = mean(residual3.^2);
101    iteration3(count) = output3.iterations;
102    fun_eval3(count) = output3.funcCount;
103    tic
104    [x4(:,count), resnorm, residual4, exitflag, output4] =

```

```

105     lsqnonlin(@(theta) ...
106         residual_opt_Heston(theta, S0, K, r, ...
107         T_ordered, MKT_C, u_k, w_k), theta0(:,count), ...
108         lower, upper, options3);
109     time4(count) = toc;
110     abs_diff_4(:,count) = abs(theta_true(:,true) ...
111         -x4(:,count));
112     MSE4(count) = mean(residual4.^2);
113     iteration4(count) = output4.iterations;
114     fun_eval4(count) = output4.funcCount;
115     disp('Mean absolute difference k')
116     [mean(abs_diff_1(1,:)) mean(abs_diff_2(1,:)) ...
117         mean(abs_diff_3(1,:)) mean(abs_diff_4(1,:))]
118     disp('Mean absolute difference v_bar')
119     [mean(abs_diff_1(2,:)) mean(abs_diff_2(2,:)) ...
120         mean(abs_diff_3(2,:)) mean(abs_diff_4(2,:))]
121     disp('Mean absolute difference sigma')
122     [mean(abs_diff_1(3,:)) mean(abs_diff_2(3,:)) ...
123         mean(abs_diff_3(3,:)) mean(abs_diff_4(3,:))]
124     disp('Mean absolute difference rho')
125     [mean(abs_diff_1(4,:)) mean(abs_diff_2(4,:)) ...
126         mean(abs_diff_3(4,:)) mean(abs_diff_4(4,:))]
127     disp('Mean absolute difference v0')
128     [mean(abs_diff_1(5,:)) mean(abs_diff_2(5,:)) ...
129         mean(abs_diff_3(5,:)) mean(abs_diff_4(5,:))]
130     disp('MSE')
131     [mean(MSE1) mean(MSE2) mean(MSE3) mean(MSE4)]
132     disp('Time')
133     [mean(time1) mean(time2) mean(time3) mean(time4)]
134     disp('Mean iterations')
135     [mean(iteration1) mean(iteration2) ...
136         mean(iteration3) mean(iteration4)]
137     disp('Mean Function evaluations')
138     [mean(fun_eval1) mean(fun_eval2) ...
139         mean(fun_eval3) mean(fun_eval4)]
140     disp("Validation number " + count)
141     disp('Number of total failures in MSE')
142     [sum(MSE1>10^-10) sum(MSE2>10^-10) ...
143         sum(MSE3>10^-10) sum(MSE4>10^-10)]
144     end
145 end

```

Heston Constrained Calibration over 2007–2021

```

1  clc; clear all; close all;
2  %Inputs
3  S0 = 1; N = 64; count = 0;
4  %Prepare numerical integration
5  [u_k, w_k] = Gaulagwt(N);
6  %Load the dataset
7  data = load('data_options.mat');
8  data = data.dataset_preprocessed;
9  risk_free = load('rf_3mths_T_bill.mat');
10 risk_free = risk_free.dataset_preprocessed;
11 %Maturities in years contained in the dataset
12 T = [1 2 3 6 9 12]/12;
13 %Moneyness K/S0 contained in the dataset
14 M = [0.80 0.85 0.9 0.95 1 1.05 1.1 1.15 1.2];
15 %Define the bound for the uniform sample
16 bounds = [0.5 5; %k
17           0.05 0.95; %v_bar
18           0.05 0.95; %sigma
19           -0.90 -0.1; %rho
20           0.05 0.95]; %v0
21 %Set options for the optimization algorithm
22 lower = [0, 0, 0, -1, 0];
23 upper = [inf, inf, inf, 1, inf]; %Case 1
24 options1 = optimoptions('lsqnonlin', 'Algorithm', ...
25                         'levenberg-marquardt', ...
26                         'SpecifyObjectiveGradient', true, 'Display', 'off', ...
27                         'FunctionTolerance', 1e-10, ...
28                         'StepTolerance', 1e-10, 'OptimalityTolerance', 1e-10, ...
29                         'MaxFunctionEvaluations', 5000, ...
30                         'MaxIterations', 5000);
31 %Load the dataset
32 sigma_imp_comp = data{:, :};
33 sigma_imp_comp = sigma_imp_comp';
34 %Start the experiment for any day contained in the dataset
35 for day = 1:692
36     sigma_imp = sigma_imp_comp(:, day);
37     r = risk_free{day, 2};
38     n = 0;
39     %Recover strikes and the market prices
40     for i = 1:length(T)
41         for j = 1:length(M)
42             n = n + 1;

```

```

43         K(n,1) = M(j);
44         MKT_C(n,1) = BLS_price(S0, K(n,1), r, ...
45             T(i), sigma_imp(n,1));
46         T_ordered(n,1) = T(i);
47     end
48 end
49 for tempt = 1:10
50     count = count + 1;
51     %Generate a starting point
52     theta0(:,count) = bounds(:,1) + rand(5,1).* ...
53         (bounds(:,2)-bounds(:,1));
54     %Run optimization algorithm
55     tic
56     [x(:,count), resnorm, residual, exitflag, output, ...
57         lambda, J] = lsqnonlin(@(theta) ...
58         residual_opt_Heston(theta, S0, K, r, ...
59         T_ordered, MKT_C, u_k, w_k), theta0(:,count), ...
60         lower, upper, options1);
61     time(count,1) = toc;
62     MSE(count,1) = mean(residual.^2);
63     iteration1(count,1) = output.iterations;
64     fun_eval(count,1) = output.funcCount;
65     SDT(:, count) = (sum(residual.^2)/ ...
66         (n-length(theta0(:,count))))* ...
67         sqrt(diag(inv((J')*J)));
68     C(count,1) = cond((J')*J);
69     disp("Day number " + day + " tempt " + tempt)
70     disp("Mean MSE " + mean(MSE))
71     disp("Time " + mean(time))
72     disp("Iterations " + mean(iteration))
73     disp("Function evaluations " + mean(fun_eval))
74     disp("Processing " + count)
75     disp("-----")
76 end
77 end

```

Heston Reparameterized Calibration over 2007–2021

```

1  clc; clear all; close all;
2  %Inputs
3  S0 = 1; N = 64; count = 0;
4  %Prepare numerical integration
5  [u_k, w_k] = Gaulagwt(N);
6  %Load the dataset
7  data = load('data_options.mat');
8  data = data.dataset_preprocessed;
9  risk_free = load('rf_3mths_T_bill.mat');
10 risk_free = risk_free.dataset_preprocessed;
11 theta0_t1 = load('theta0_Hes_real_tr.mat');
12 theta0_t1 = theta0_t1.theta0_tr;
13 %Maturities in years contained in the dataset
14 T = [1 2 3 6 9 12]/12;
15 %Moneyness K/S0 contained in the dataset
16 M = [0.80 0.85 0.9 0.95 1 1.05 1.1 1.15 1.2];
17 %Set options for the optimization algorithm
18 options1 = optimoptions('lsqnonlin', 'Algorithm', ...
19     'levenberg-marquardt', ...
20     'SpecifyObjectiveGradient', true, 'Display', 'off', ...
21     'FunctionTolerance', 1e-10, ...
22     'StepTolerance', 1e-10, 'OptimalityTolerance', 1e-10, ...
23     'MaxFunctionEvaluations', 5000, ...
24     'MaxIterations', 5000);
25 %Load the dataset
26 sigma_imp_comp = data{:, :};
27 sigma_imp_comp = sigma_imp_comp';
28 %Start the experiment for any day contained in the dataset
29 for day = 1:692
30     sigma_imp = sigma_imp_comp(:, day);
31     r = risk_free{day, 2};
32     n = 0;
33     %Recover strikes and the market prices
34     for i = 1:length(T)
35         for j = 1:length(M)
36             n = n+1;
37             K(n,1) = M(j);
38             MKT_C(n,1) = BLS_price(S0, K(n,1), r, ...
39                 T(i), sigma_imp(n,1));
40             T_ordered(n,1) = T(i);
41         end
42     end

```

```

43     for tempt = 1:10
44         count = count+1;
45         %Run optimization algorithm
46         tic
47         [x(:,count), resnorm, residual, exitflag, output, ...
48             lambda, J] = lsqnonlin(@(theta) ...
49             residual_opt_Heston_tr(theta, S0, K, r, ...
50             T_ordered, MKT_C, u_k, w_k), theta0_t1(:,count), ...
51             [], [], options1);
52         time1(count,1) = toc;
53         x_final(:,count) = [exp(x1(1,count)) exp(x1(2,count)) ...
54             exp(x1(3,count)) 2*logsig(x1(4,count))-1 ...
55             exp(x1(5,count))];
56         MSE(count,1) = mean(residual.^2);
57         iteration(count,1) = output.iterations;
58         fun_eval(count,1) = output.funcCount;
59         SDT(:,count) = sqrt((sum(residual.^2)/ ...
60             (n-length(theta0_t1(:,count))))) * ...
61         real(sqrt(diag(inv((J')*J))));
62         C(count,1) = cond((J')*J);
63         disp("Day number " + day + " tempt " + tempt)
64         disp("Mean MSE " + mean(MSE))
65         disp("Time " + mean(time))
66         disp("Iterations " + mean(iteration))
67         disp("Function evaluations " + mean(fun_eval))
68         disp("Processing " + count)
69         disp("-----")
70     end
71 end

```

Ornstein–Uhlenbeck Constrained Calibration over 2007–2021

```

1  clc; clear all; close all;
2  %Inputs
3  S0 = 1; N = 64; count = 0;
4  %Prepare numerical integration
5  [u_k, w_k] = Gaulagwt(N);
6  %Load the dataset
7  data = load('data_options.mat');
8  data = data.dataset_preprocessed;
9  risk_free = load('rf_3mths_T_bill.mat');
10 risk_free = risk_free.dataset_preprocessed;
11 %Maturities in years contained in the dataset
12 T = [1 2 3 6 9 12]/12;
13 %Moneyness K/S0 contained in the dataset
14 M = [0.80 0.85 0.9 0.95 1 1.05 1.1 1.15 1.2];
15 %Define the bound for the uniform sample
16 bounds = [0.5 8; %k
17           0.05 0.35; %v_bar
18           0.05 0.5; %sigma
19           -0.95 0.25; %rho
20           0.05 0.35]; %v0
21 %Set options for the optimization algorithm
22 lower = [0, 0, 0, -1, -inf];
23 upper = [inf, inf, inf, 1, inf];
24 options1 = optimoptions('lsqnonlin', 'Algorithm', ...
25                         'levenberg-marquardt', ...
26                         'SpecifyObjectiveGradient', true, 'Display', 'off', ...
27                         'FunctionTolerance', 1e-10, ...
28                         'StepTolerance', 1e-10, 'OptimalityTolerance', 1e-10, ...
29                         'MaxFunctionEvaluations', 5000, ...
30                         'MaxIterations', 5000);
31 %Load the dataset
32 sigma_imp_comp = data{:, :};
33 sigma_imp_comp = sigma_imp_comp';
34 %Start the experiment for any day contained in the dataset
35 for day = 1:692
36     sigma_imp = sigma_imp_comp(:, day);
37     r = risk_free{day, 2};
38     n = 0;
39     %Recover strikes and the market prices
40     for i = 1:length(T)
41         for j = 1:length(M)
42             n = n+1;

```

```

43         K(n,1) = M(j);
44         MKT_C(n,1) = BLS_price(S0, K(n,1), r, ...
45             T(i), sigma_imp(n,1));
46         T_ordered(n,1) = T(i);
47     end
48 end
49 for tempt = 1:10
50     count = count+1;
51     %Define a starting point
52     theta0(:,count) = bounds(:,1)+rand(5,1).* ...
53         (bounds(:,2)-bounds(:,1));
54     %Run optimization algorithm
55     tic
56     [x(:,count), resnorm, residual, exitflag, output, ...
57         lambda, J] = lsqnonlin(@(theta) ...
58         residual_opt_OU(theta, S0, K, r, ...
59         T_ordered, MKT_C, u_k, w_k), theta0(:,count), ...
60         lower, upper, options1);
61     time(count,1) = toc;
62     MSE(count,1) = mean(residual.^2);
63     iteration(count,1) = output.iterations;
64     fun_eval(count,1) = output.funcCount;
65     SDT(:,count) = sqrt((sum(residual.^2)/ ...
66         (n-length(theta0(:,count))))) * ...
67         sqrt(diag(inv((J')*J)));
68     C(count,1) = cond((J')*J);
69     disp("Day number " + day + " tempt " + tempt)
70     disp("Mean MSE " + mean(MSE))
71     disp("Time " + mean(time))
72     disp("Iterations " + mean(iteration))
73     disp("Function evaluations " + mean(fun_eval))
74     disp("Processing " + count)
75     disp("-----")
76 end
77 end

```

Condition Number Approach

```

1  clc; clear all; close all;
2  %Inputs
3  S0 = 1; N = 64; count = 0; g = 0;
4  %Prepare numerical integration
5  [u_k, w_k] = Gaulagwt(N);
6  %Load the dataset
7  data = load('data_options.mat');
8  data = data.dataset_preprocessed;
9  risk_free = load('rf_3mths_T_bill.mat');
10 risk_free = risk_free.dataset_preprocessed(:,:);
11 %Maturities in years contained in the dataset
12 T = [1 2 3 6 9 12]/12;
13 %Moneyness K/S0 contained in the dataset
14 M = [0.80 0.85 0.9 0.95 1 1.05 1.1 1.15 1.2];
15 %Set options for the optimization algorithm
16 options1 = optimoptions('lsqnonlin', 'Algorithm', ...
17     'levenberg-marquardt', ...
18     'SpecifyObjectiveGradient', true, 'Display', 'off', ...
19     'FunctionTolerance', 1e-10, ...
20     'StepTolerance', 1e-10, 'OptimalityTolerance', 1e-10, ...
21     'MaxFunctionEvaluations', 5000, ...
22     'MaxIterations', 5000);
23 %Load the dataset
24 sigma_imp_comp = data{:,:};
25 sigma_imp_comp = sigma_imp_comp';
26 %Define the bound for the uniform sample
27 bounds_H = [log(0.5) log(5); %k_t1
28     log(0.05) log(0.95); %v_bar_t1
29     log(0.05) log(0.95); %sigma_t1
30     log(1-0.90)-log(1+0.90) ...
31     log(1-0.1)-log(1+0.1); %rho_t1
32     log(0.05) log(0.95)]; %v0_t1
33 bounds_OU = [0.5 8; %k
34     0.05 0.35; %v_bar
35     0.05 0.5; %sigma
36     -0.95 0.25; %rho
37     0.05 0.35]; %v0
38 lower = [0, 0, 0, -1, -inf];
39 upper = [inf, inf, inf, 1, inf];
40 %Start the experiment for any day contained in the dataset
41 for day = 1:692
42     sigma_imp = sigma_imp_comp(:,day);

```

```

43     r = risk_free{day,2};
44     n = 0;
45     %Recover strikes and the market prices
46     for i = 1:length(T)
47         for j = 1:length(M)
48             n = n+1;
49             K(n,1) = M(j);
50             MKT_C(n,1) = BLS_price(S0, K(n,1), r, ...
51                 T(i), sigma_imp(n,1));
52             T_ordered(n,1) = T(i);
53         end
54     end
55     count = count+1;
56     theta0(:,count) = bounds_H(:,1)+rand(5,1).* ...
57     (bounds_H(:,2)-bounds_H(:,1));
58     %Run optimization algorithm
59     tic
60     [x(:,count), resnorm, residual, exitflag, output, ...
61         lambda, J] = lsqnonlin(@(theta) ...
62         residual_opt_Heston_tr(theta, S0, K, r, ...
63         T_ordered, MKT_C, u_k, w_k), theta0(:,count), ...
64         [], [], options1);
65     C(count,1) = cond((J')*J);
66     if C(count,1) <= 1e07
67         time1(count,1) = toc;
68         x_final(:,count) = [exp(x1(1,count)) exp(x1(2,count))
69             exp(x1(3,count)) 2*logsig(x1(4,count))-1
70             exp(x1(5,count))];
71         MSE(count,1) = mean(residual.^2);
72         iteration(count,1) = output.iterations;
73         fun_eval(count,1) = output.funcCount;
74         SDT(:,count) = real(sqrt((sum(residual.^2)/ ...
75             (n-length(theta0(:,count))))) * ...
76             sqrt(diag(inv((J')*J))));
77     else
78         g = g + 1;
79         disp("Switch number " + g)
80         theta0(:,count) = bounds_OU(:,1)+rand(5,1).* ...
81         (bounds_OU(:,2)-bounds_OU(:,1));
82         [x(:,count), resnorm, residual, exitflag, output, ...
83             lambda, J] = lsqnonlin(@(theta) ...
84             residual_opt_OU(theta, S0, K, r, ...
85             T_ordered, MKT_C, u_k, w_k), theta0(:,count), ...
86             lower, upper, options1);

```

```
87         time(count,1) = toc;
88         C(count,1) = cond((J')*J);
89         MSE(count,1) = mean(residual.^2);
90         iteration(count,1) = output.iterations;
91         fun_eval(count,1) = output.funcCount;
92         SDT(:,count) = sqrt((sum(residual.^2)/ ...
93             (n-length(theta0(:,count)))))* ...
94             sqrt(diag(inv((J')*J)));
95     end
96     disp("Day number " + day)
97     disp("Number of switches" + g)
98     disp("Mean MSE " + mean(MSE))
99     disp("Time " + mean(time))
100    disp("Processing " + count)
101    disp("-----")
102 end
```