Journal of Econometrics and Statistics
Vol. 5, Issue 2, 2025, pp. 215-235

Multi-Output Gaussian Process for Volatility Modelling and Option Pricing in the Heston Framework

Akash Sedai[a] and Francesca Medda[a]

[a] *Institute of Finance & Technology, University College London, UK*

ABSTRACT
This paper presents an approach to volatility modelling and option pricing using multi-output Gaussian processes (MOGP) within the Heston framework. We employ MOGP to approximate a range of financial quantities, including implied volatility grids, options prices, and portfolio values. Our methodology offers a unified model for simultaneously handling multiple outputs, thereby enhancing computational efficiency and reducing the need for multiple separate models. In the process, we also provide a brief overview of Bayesian techniques applied to options pricing and volatility modelling, highlighting their relevance and advantages in the context of the proposed approach. Extensive experiments demonstrate the accuracy and scalability of the MOGP model, with comparative analysis against the Fast Fourier Transform (FFT) method, showing significant improvements in computational speed and precision for pricing and risk management tasks.

KEYWORDS
Volatility, Bayesian Approximations, Option Pricing, Multi-GP Methodology

1. Introduction

Options are a widely used risk-hedging mechanism. Hedgers use derivatives to mitigate the risk of buying or selling commodities at unfavorable prices. For speculators, options are often more attractive than the underlying asset due to their limited loss potential and higher leverage compared to stock trading, which can amplify gains. Arbitrageurs seek discrepancies in derivative pricing to make riskless profits. As the derivatives market grows, extensive computations are required for model calibration, instrument pricing, hedge position calculations, and risk management. Given the rapid pace at which market conditions change, real-time updates are critical in this fast-moving environment. Therefore, reducing calculation times is essential for operational efficiency.

With the increasing popularity of machine learning across industries, its application in option pricing, volatility modeling, and other high-precision, high-stakes decision-making areas is gaining momentum. However, this shift brings about new risks and challenges, particularly around regulatory concerns and the precision of predictions. Volatility, a key variable in stochastic differential equations (SDEs) for asset pricing,

CONTACT Author[c]. Email: akash.sharma@ucl.ac.uk

Article History
Received : 22 March 2025; Revised : 20 April 2025; Accepted : 05 May 2025; Published : 28 June 2025

To cite this paper

Akash Sedai & Francesca Medda (2025). Multi-Output Gaussian Process for Volatility Modelling and Option Pricing in the Heston Framework. *Journal of Econometrics and Statistics.* 5(2), 215-235.

is central to numerous financial applications. For instance, during financial contagion, volatility can transmit between economies [9]. In this context, explainability and accountability become paramount, but many machine learning models are black boxes, offering limited interpretability of their predictions. This lack of transparency can lead to undesirable consequences in practice. Conversely, reliable statistical methods like Monte Carlo simulations, though proven, are often time-consuming, especially with increasingly complex instruments.

While advanced financial models theoretically capture non-linear features observed in financial markets, their high-dimensional nature prevents closed-form solutions for option pricing. The computational burden often limits their practical applicability. Classical parametric models, though widely used, face challenges related to parametric calibration, unrealistic statistical assumptions, rigid functional forms, and susceptibility to overfitting. For an overview of parametric and semi-parametric methods, see [3, 16]. Bayesian non-parametric approaches, such as Gaussian processes, offer promising alternatives.

In this paper, we first review relevant SDEs used in volatility modeling, followed by a discussion of common numerical methods. We then explore advanced machine learning techniques for pricing options and calculating implied volatilities. Finally, we introduce a multi-Gaussian process approach for improved pricing and volatility modeling.

2. Some Stochastic Volatility Models

2.1. *Heston Model*

Under some probability measure μ_p (often called the physical measure), the Heston model is described as follows:

$$\begin{aligned} dS_t &= \mu S_t\, dt + \sqrt{v_t}\, S_t\, dB_t, \\ dv_t &= k(\theta - v_t)\, dt + \sigma\sqrt{v_t}\, dW_t, \\ \mathbb{E}_p[dB_t\, dW_t] &= \rho\, dt, \end{aligned} \tag{1}$$

where μ is the drift of the stock process, $\theta > 0$ is the mean reversion level of the variance, $k > 0$ is the mean reversion rate for the variance, $\sigma > 0$ is the volatility of the variance, and $\rho \in [-1, 1]$ is the correlation between the two Brownian motions B_t and W_t.

The Heston implied volatility often does not fit the market implied volatility perfectly at shorter maturity, although evidence shows a good fit at longer maturities. The Double Heston model overcomes this problem with short maturity. Under the risk-neutral measure, the system of stochastic equations in the (single) Heston model (which is one of the most popular stochastic volatility (SV) models [14]) becomes:

$$\frac{\partial V}{\partial t} + rS\frac{\partial V}{\partial S} + k(\bar{v} - v)\frac{\partial V}{\partial v} + \rho\gamma S v\, \frac{\partial^2 V}{\partial S \partial v} + \rho^2 v\, \frac{1}{2} S^2 \frac{\partial^2 V}{\partial S^2} - rV = 0, \tag{2}$$

where the commonly observed shapes of implied volatility in the market (e.g. smile or skew) can be reproduced by varying the parameters k, ρ, γ, v_0, and $\bar{v}$. Generally, the parameter γ impacts the kurtosis of the asset return distribution, and ρ controls its asymmetry. The Heston model does not have an analytic solution.

2.2. *CEV Model*

The Constant Elasticity of Variance (CEV) model was introduced in 1976 by John Cox and Stephen Ross. It describes the relationship between volatility and price by introducing the following (local) stochastic volatility:

$$dS_t = \mu S_t\, dt + \sigma S_t^{\gamma}\, dW_t. \tag{3}$$

Conceptually, in some markets where volatility rises as prices rise (e.g. commodities), $\gamma > 1$. In others, where volatility tends to rise as prices fall, we have $\gamma < 1$. Some argue that the CEV model is a local volatility model rather than a true stochastic volatility model, since it does not incorporate its own stochastic process for volatility.

The constant parameters σ and γ satisfy $\sigma \geq 0$ and $\gamma \geq 0$. The parameter γ is the central feature of the model, controlling the relationship between volatility and price. Commonly observed in equity markets, $\gamma < 1$ captures a leverage effect: the volatility of a stock increases as its price falls. Conversely, in commodity markets, we often observe $\gamma > 1$, sometimes called the inverse leverage effect [3, 10], where the price volatility of a commodity increases as its price increases.

2.3. *SABR Model*

Introduced by Hagan et al. [20], the SABR model (Stochastic Alpha, Beta, Rho) describes a single forward F (related to any asset, such as an index, bond, equity, currency, or interest rate) under stochastic volatility σ:

$$\begin{aligned} dF_t &= \sigma_t F_t^{\beta}\, dW_t, \\ d\sigma_t &= \alpha\, \sigma_t\, dZ_t, \end{aligned} \tag{4}$$

where the initial values F_0 and σ_0 are the current forward price and volatility respectively. The constant parameters β and α satisfy $0 \leq \beta \leq 1$, $\alpha \geq 0$. The parameter α is a volatility-like parameter for the volatility, and ρ is the instantaneous correlation between the underlying asset and its volatility. Hence, α controls the height of the at-the-money (ATM) implied volatility level, ρ controls the slope of the implied skew, and β controls its curvature. This is simply a stochastic version of the CEV model with a skewness parameter β, which reduces to the CEV model when $\alpha = 0$.

2.4. *GARCH Model*

Although not a purely stochastic model, GARCH is another common approach to estimating stochastic volatility. It assumes that the randomness of the variance process varies with the variance (rather than with the square root of the variance, as in the Heston model). The conditional variance is a deterministic function of the model parameters and historical data (unlike in SV models, where volatility is a random variable). The standard GARCH(1,1) model has the variance differential form:

$$\nu_t = \theta(\omega - \nu_t)\, dt + \xi\, \nu_t\, dB_t. \tag{5}$$

Several variants of GARCH have emerged over time, including non-linear GARCH (NGARCH), Integrated GARCH, LGARCH, TGARCH, GJR-GARCH, and

EGARCH, among others. Strictly speaking, the conditional volatilities from GARCH models are not stochastic since at any given time t, the volatility is deterministic given previous values. One advantage of GARCH models is that they have a computable likelihood function, making their estimation easier [10]. A class of discrete versions of stochastic volatility called DSARV models has also been proposed using Markov chain methods [10], allowing a low-dimensional state space for the volatility. Such models consider a bounded and tractable likelihood function, which significantly improves computational tractability.

3. Literature Review

This section provides a literature survey of Bayesian methods in volatility modelling and options pricing that may assist risk managers and researchers with representation and model validation. Stochastic volatility models are often solved numerically, as they generally do not have analytic solutions. These numerical methods typically fall under three categories: finite differences, numerical integration, and Monte Carlo methods. Finite differences for partial differential equations (PDEs) are frequently used for free boundary problems when valuing American options [4, 50], or for certain exotic options (e.g., barrier, Bermudan, Asian, and lookback options). They can also be used to accurately compute the derivatives of option prices (the Greeks). Monte Carlo methods are often used for valuation of path-dependent options: the process is to generate paths of the stock price using a random walk model under the risk-neutral probability, compute the payoff, and repeat the process many times over the time horizon. Then the present value of the expected payoff is used as an approximation of the option price. The numerical integration method is often used by first transforming the problem into the Fourier domain. An efficient technique for this approach is the COS method, which utilizes Fourier-cosine series expansions to approximate the asset price's probability density function [25]

Financial derivative pricing involves calibrating open parameters of the model to market data, i.e. matching the market prices of heavily traded options to the option prices under the risk-neutral probability measure. Thousands of option prices must be determined in order to fit these asset parameters. However, some high-quality asset models remain unused due to the requirement of efficient numerical computation in financial risk management [25].

Many machine learning (ML) models are "black boxes" whose outputs are not explained in ways interpretable to humans, leading to low-quality inference. On the other hand, although more interpretable models mitigate inferential issues, they are often constrained in model form. They may be useful or conform to structural knowledge of the domain — such as monotonicity, causality, structural (generative) constraints, additivity, or physical constraints coming from domain knowledge [34] — but they also tend to be computationally expensive. Numerous studies have been conducted on ML stochastic volatility (SV) modelling and pricing over the past two decades, and more than a hundred of these have been on neural networks alone (see [35] for a more comprehensive listing of neural network literature, [31] for reviews on stochastic volatility models, and [30] for reviews on major contributions to Bayesian analysis of SV models). In terms of ML applications, the motivation often lies in achieving fast pricing, given that the algorithm can be trained offline.

In this section, we focus on a brief review of literature that uses Bayesian approaches in pricing and volatility modelling. For readers interested in pricing and volatility using

a Bayesian framework, we list a selection of peer-reviewed papers below:

- **Karolyi (1993)** [**24**] is one of the first papers to use a Bayesian approach. Karolyi produced estimates of stock return volatility for a given stock by drawing on prior information from the cross-sectional pattern in the return volatilities for a group of stocks. He suggests that the posterior density of the option price can be derived as a nonlinear transformation of that of the stock return volatility. Using certain stock option pricing models with the Bayesian estimates of volatility, [17] computed call option prices that match CBOE market prices more closely than when implied volatility or standard historical volatility is used.
- **Bauwens (2000)** [**5**] computes option prices from a Bayesian inference viewpoint using a GARCH model for the volatility dynamics. The predictive expectation of the payoff function is used to evaluate the option price under the risk-neutral probability measure.
- **Guidolin & Timmermann (2001)** [**17**] derived closed-form pricing formulas for European options in the context of an equilibrium model where dividend news evolve on a binomial lattice with unknown but recursively updated probabilities. Their Bayesian learning effects model can explain the empirical biases of the Black–Scholes model. They showed that learning generates asymmetric skews in the implied volatility surface and systematic patterns in the term structure of option prices.
- **Darsinos & Satchell (2001)** [**13**] presented an option price predictor by providing a Bayesian analysis of the Black–Scholes (BS) price. They combined a prior density for price and volatility with the likelihood of volatility to obtain the posterior density of both. It is an extension of [24] in that it incorporates the volatility in the posterior density. The posterior density of the option price then follows after dividing by the marginal density of the asset price and applying a nonlinear transformation, accounting for randomness both in the price process and in volatility (unlike the BS price, which neglects the randomness in volatility).
- **Gupta & Reisinger (2011)** [**18**] introduced a Bayesian framework for calibrating the parameters of financial models to market prices. With a practical method for formulating the prior and likelihood functions necessary for the Bayes procedure, they apply it to a local volatility model. Compared to common maximum a posteriori methods, their numerical examples exhibit improved pricing under the Bayesian procedure, highlighting its robustness against inaccuracies in the model and prior, as well as mispricing in the data.
- **Ho, Lee & Marsden (2011)** [**22**] presented Bayesian volatility estimates based on Gamma and mixture priors. Their results provided a more accurate estimate of implied volatility compared to the historical estimate, noting also that the output is sensitive to the choice of prior.
- **Rombouts & Stentoft (2014)** [**36**] used mixed-normal heteroscedasticity models under an appropriate risk-neutral dynamics. They carried out inference in a Bayesian framework, allowing straightforward computation of predictive price densities that account for parameter uncertainty.
- **Uhl (2014)** [**42**] used a regressor selection algorithm under the Bayesian nonparametric regression proposed by [37] to estimate implied volatility surfaces.
- **Wang, Choy & Wong (2016)** [**45**] proposed an SV model that enables simultaneous Bayesian inference and transformation into risk-neutral dynamics, building on the GARCH pricing approach of [14]. They addressed the challenge of

transforming the model into its risk-neutral measure from the estimated physical dynamic. This approach avoids having to transform the model post-estimation with high-dimensional integration under the SV likelihood function.

- **Wang & Wong (2017)** [**46**] derived the risk-neutral dynamic of the DSARV model, enabling direct application to option pricing and VIX forecasting. The authors also derived a closed-form expression for the VIX with the continuous SV model. Although the DSARV model performs rather unsatisfactorily (even compared to GARCH), the maximum likelihood estimation (MLE) method avoids subjectivity and may reduce computational time, though the computation of VIX remains tedious.
- **De Spiegeleer et al. (2018)** [**39**] proposed offline learning of a derivative pricing function via Gaussian process (GP) regression. Specifically, the authors configure the training set over a grid and then use GPs to interpolate at test points. They demonstrate a speed-up of GPs relative to Monte Carlo methods, with a tolerable loss of accuracy when applied to pricing and Greek estimation under the Heston model, as well as for approximating implied volatility surfaces. However, the applications in [39] are limited to single-output GPs and thus require many runs to obtain a range of outputs, which does not demonstrate the full potential of GPs.
- **Jang & Lee (2019)** [**23**] used a generative Bayesian neural network model that leverages prior information about financial derivative market structures. They followed a two-step calibration approach, previously employed by [1, 49], in which the structural parameters of the underlying asset are first approximated, and then the remaining pricing-model parameters are calibrated. They priced options using least-squares Monte Carlo after parameter calibration. Their approach ensures the no-arbitrage condition via a generative Bayesian neural network.
- **Mu et al. (2020)** [**32**] presented a GP approach to pricing American options with Monte Carlo simulations, using control variates to further improve estimation of the continuation value. They showed reduced standard errors of the estimation and improved stability with the GP approach.
- **Tegnér & Roberts (2019)** [**40**] introduced the Gaussian process framework for the local volatility calibration problem within a Bayesian context, inferring a conditional distribution over functions given data. The motivation was twofold: (1) to encode prior domain knowledge easily (e.g. smoothness properties) and (2) to let Bayesian inference automatically tune the complexity of the non-parametric model. They extended this by adding another layer of inference for model selection, building on the literature on Gaussian processes to generate efficient numerical algorithms for sampling the posterior distribution over local volatility.
- **Yin & Mondal (2022)** [**48**] cast the calibration of local volatility models into a Bayesian framework, enabling uncertainty quantification via the posterior distribution. By incorporating prior information and observed option prices, their approach yields a probabilistic characterisation of the local volatility surface. To mitigate the high computational cost associated with Bayesian inference, they apply dimensionality reduction techniques, significantly improving efficiency while maintaining accuracy in volatility estimation.
- **Liu, Wang, Tran & Kohn (2023)** [**27**] introduced a Bayesian framework for volatility modelling by integrating deep learning with realized GARCH. Their approach employs Long Short-Term Memory (LSTM) networks to cap-

ture complex dependencies in high-frequency financial data, enhancing volatility forecasting. Bayesian inference is conducted via the Sequential Monte Carlo method, ensuring robust statistical estimation and uncertainty quantification. Their model outperforms traditional GARCH-based methods in predictive accuracy and adaptability to volatility dynamics.

- **Brignone, Gonzato & Lutkebohmert (2023) [6]** proposed a quasi-Bayesian estimation technique for affine option pricing models using risk-neutral cumulants. Their method, based on Laplace-Type Estimation (LTE) and Sequential Monte Carlo (SMC) techniques, provides an efficient and accurate approach to parameter estimation. The authors applied this approach to models with two stochastic volatility factors, co-jumps between price and volatility, and stochastic jump intensity. Their results show the algorithm's effectiveness in both simulated and real data, demonstrating its superiority over traditional methods in estimating option pricing models.

Other notable papers in the Bayesian literature include [8, 11, 12, 15, 19, 28, 33, 44]. In the next section, we provide a brief overview of Gaussian processes and then present a multi-GP-based methodology to compute multiple prices or volatilities in each optimization.

4. Gaussian Processes (GPs)

Gaussian Processes (GPs) offer a non-parametric Bayesian framework for regression and function estimation. Unlike parametric models that restrict functions to a specific class, GPs place a prior directly over functions, using a covariance (kernel) function to encode assumptions about smoothness, periodicity, or other domain-specific properties. By adapting to the observed data, GPs can capture complex behaviour without imposing a rigid form. This section introduces key ideas of GPs in the single-output case, extends them to multi-output scenarios (MOGPs), and then reviews sparse approximations aimed at mitigating the computational challenges that arise when dealing with large datasets or multiple outputs.

4.1. *Preliminaries*

A Gaussian Process (GP) is an infinite collection of random variables such that any finite subset follows a joint Gaussian distribution. Let

$$\mathcal{D} = \{(\mathbf{x}_i, y_i) \,|\, i = 1, \ldots, N\}$$

be the set of training pairs, where each input $\mathbf{x}_i \in \Omega^d$ (e.g. a d-dimensional feature vector) is mapped to a scalar output $y_i \in \mathbb{R}$. Suppose an unknown latent function $f(\mathbf{x})$ generates the outputs via

$$y_i = f(\mathbf{x}_i) + \varepsilon_i, \quad \text{where} \quad \varepsilon_i \sim \mathcal{N}(0, \sigma_n^2).$$

We place a GP prior on $f(\mathbf{x})$, fully specified by a mean function $m(\mathbf{x})$ and a covariance (kernel) function $k(\mathbf{x}, \mathbf{x}')$:

$$f(\mathbf{x}) \sim \mathcal{GP}\big(m(\mathbf{x}),\, k(\mathbf{x}, \mathbf{x}')\big).$$

Often, one takes $m(\mathbf{x}) = 0$ by appropriate centring or for simplicity, but a non-zero mean may be used if domain knowledge is available. The kernel $k(\mathbf{x}, \mathbf{x}')$ encodes assumptions about smoothness and correlation across input space.

Given N observations, the joint distribution of the corresponding latent function values $\mathbf{f} = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N)]^\top$ is

$$\mathbf{f} \sim \mathcal{N}\big(\mathbf{0},\, K(\mathbf{X}, \mathbf{X})\big),$$

where $K(\mathbf{X}, \mathbf{X}) \in \mathbb{R}^{N \times N}$ is the covariance matrix with entries $\big[K(\mathbf{X}, \mathbf{X})\big]_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. Accounting for i.i.d. Gaussian noise, $\mathbf{y} = \mathbf{f} + \boldsymbol{\varepsilon}$, the marginal likelihood is

$$p(\mathbf{y} \mid \mathbf{X}) = \int p(\mathbf{y} \mid \mathbf{f})\, p(\mathbf{f} \mid \mathbf{X})\, d\mathbf{f} = \mathcal{N}\big(\mathbf{0},\, K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I\big).$$

4.1.1. Posterior Prediction

For a new input $\mathbf{x}_*$, we denote the corresponding latent function value by f_*. Then,

$$\begin{pmatrix} \mathbf{y} \\ f_* \end{pmatrix} \sim \mathcal{N}\Bigg(\begin{pmatrix} \mathbf{0} \\ 0 \end{pmatrix}, \begin{pmatrix} K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I & \mathbf{k}_* \\ \mathbf{k}_*^\top & k(\mathbf{x}_*, \mathbf{x}_*) \end{pmatrix}\Bigg),$$

where $\mathbf{k}_* = k(\mathbf{X}, \mathbf{x}_*)$. Conditioning on $\mathbf{y}$ yields the posterior predictive distribution:

$$p\big(f_* \mid \mathbf{x}_*, \mathbf{X}, \mathbf{y}\big) = \mathcal{N}\big(\bar{f}_*,\, \mathrm{cov}(f_*)\big),$$

whose mean and variance are

$$\bar{f}_* = \mathbf{k}_*^\top \big[K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I\big]^{-1} \mathbf{y}, \quad \mathrm{cov}(f_*) = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^\top \big[K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I\big]^{-1} \mathbf{k}_*.$$

4.1.2. Kernel Hyperparameters

In GP regression, the kernel function $k(\mathbf{x}, \mathbf{x}')$ plays a crucial role in determining the smoothness and structure of the underlying function. The kernel typically depends on hyperparameters such as the length scale, variance, or noise variance, which control properties like how quickly the function varies or how noisy the observations are. These hyperparameters are typically learned from the training data by maximizing the marginal likelihood $p(\mathbf{y} \mid \mathbf{X})$, ensuring that the GP model is tailored to the data at hand. Optimizing these hyperparameters allows the model to capture the appropriate patterns in the data, balancing fit and generalization.

4.1.3. Non-parametric Nature of Gaussian Processes

A key feature of Gaussian Processes is that they provide a non-parametric model. Unlike parametric models, which assume a fixed number of parameters (such as the coefficients in a linear regression model), a GP's complexity grows with the amount of data. This means that a GP can model complex, high-dimensional, and nonlinear relationships without needing a fixed, predefined number of parameters. This flexibility makes GPs especially powerful in scenarios where the underlying function is unknown and needs to be inferred from the data. However, this also implies that GPs can become computationally expensive as the number of data points increases.

4.1.4. Computational Considerations

One of the main challenges in GPs is its computational complexity. The inversion of the covariance matrix $K(\mathbf{X}, \mathbf{X})$, which is required for both the marginal likelihood and the posterior predictive distribution, scales cubically with the number of data points, $O(N^3)$. This can become prohibitive for large datasets. Various techniques have been proposed to mitigate this, such as sparse Gaussian Processes, which approximate the covariance matrix using a subset of the data, reducing computational costs while retaining the flexibility of the GP model.

4.2. *Multi-Output Gaussian Processes (MOGPs)*

In many real-world applications, multiple correlated outputs must be predicted simultaneously. For example, in finance, one might wish to jointly model the prices of several assets or volatility measures. A Multi-Output Gaussian Process (MOGP) extends the single-output GP framework to handle vector-valued functions, where each output is modeled as a function of the same input $\mathbf{x}$. Specifically, a multi-output function is given by:

$$\mathbf{f}(\mathbf{x}) = \big(f_1(\mathbf{x}), \ldots, f_P(\mathbf{x})\big)^\top \in \mathbb{R}^P,$$

where P is the number of outputs. The MOGP is governed by a matrix-valued kernel function $\mathbf{K}_M(\mathbf{x}, \mathbf{x}') \in \mathbb{R}^{P \times P}$, which models both the correlation between inputs and the dependencies between the different outputs. This kernel allows for the incorporation of cross-output correlations, improving the predictive performance by leveraging the relationships between the outputs. Thus, we have:

$$\mathbf{f}(\mathbf{x}) \sim \mathcal{GP}\big(\mathbf{0}, \mathbf{K}_M(\mathbf{x}, \mathbf{x}')\big).$$

By jointly modeling the multiple outputs, MOGPs can improve the overall prediction accuracy by exploiting correlations between the outputs, which is particularly useful when outputs share common latent structures. For instance, predicting related quantities together often yields more robust models and mitigates overfitting to individual outputs.

However, the primary challenge in MOGPs is the increased computational burden. When dealing with N training points and P outputs, the covariance matrix becomes an $(NP) \times (NP)$ matrix, making both its construction and inversion computationally expensive. Specifically, the time complexity for inverting this large matrix scales as $\mathcal{O}(N^3P^3)$, which can become prohibitive for large datasets and multiple outputs. To address this, various techniques, such as sparse MOGPs, have been proposed to reduce computational costs while still capturing the inter-output correlations.

4.2.1. Sparse Variational Gaussian Processes (SVGP)

Gaussian Processes (GPs) are a powerful tool for non-parametric regression, but their computational complexity grows cubically with the number of data points, making them impractical for large datasets. To address this, Sparse Variational Gaussian Processes (SVGPs) provide an efficient approximation to the GP by using a sparse set of "inducing points" that summarize the information in the data. The concept was first introduced by [38] and later extended in subsequent work.

In SVGPs, the full GP prior is replaced by an approximate posterior distribution that is based on a set of inducing variables, $\mathbf{u} = [u_1, \ldots, u_M]^\top$, where $M \ll N$ is the number of inducing points. These inducing points are typically selected to lie in the input space, and their locations are either fixed or optimized as part of the learning process. The key idea is that the inducing points capture the essential characteristics of the GP, allowing the model to approximate the full covariance matrix without requiring its explicit inversion.

The approximate posterior is derived using variational inference, where a variational distribution is introduced over the inducing variables. The variational distribution is typically chosen to be Gaussian, leading to an approximation of the true posterior. This method of variational learning for inducing variables was formalized by [41]. The resulting sparse GP model has a reduced computational complexity, with time complexity scaling as $O(M^2N)$, which is much more manageable than the original $O(N^3)$.

Sparse GPs are particularly useful when working with large datasets, as they significantly reduce the computational cost while maintaining the flexibility and expressiveness of GPs. Moreover, they allow for scalable predictions, making GPs applicable to real-world problems with large-scale data [21].

4.2.2. SVGP for Multi-Output GPs

For multi-output problems, where multiple correlated outputs must be predicted simultaneously, one approach to alleviate the computational complexity is to use sparse variational Gaussian processes (SVGPs). In the context of Multi-Output GPs (MOGPs), the main challenge lies in the high computational cost, which scales as $\mathcal{O}(N^3P^3)$ for N training points and P outputs. To address this, SVGPs approximate the full covariance matrix by utilizing a smaller set of inducing points, M, where $M \ll N$. This approximation significantly reduces the computational burden, bringing the cost down to $\mathcal{O}(M^3P)$, which is much more practical for large datasets.

For multi-output settings, one may maintain separate sets of inducing points for each output, or alternatively, a shared set of inducing points that captures the cross-output correlations. The latter approach is often more efficient, as it leverages the correlations between different outputs to reduce the number of inducing points needed to approximate the joint GP. The shared inducing points model the inter-output dependencies, allowing for improved predictive performance without increasing computational costs substantially.

The use of sparse variational methods in multi-output settings was formalized in earlier work by [2], where they introduced variational inducing kernels for MOGPs, enabling efficient approximations to the joint covariance matrix. Further advancements were made by [43], who extended this framework to more general inter-domain and multi-output Gaussian Processes, providing a comprehensive approach for handling complex, correlated output structures.

These techniques allow for scalable multi-output predictions while maintaining the flexibility of GPs, making them applicable to real-world problems that involve large-scale multi-output data.

4.2.3. Practical Advantages

The Sparse Variational GP (SVGP) framework offers several important benefits in real-world applications:

- **Scalability:** The costly $\mathcal{O}(N^3)$ dependence shifts to $\mathcal{O}(M^3)$ when only $M \ll N$ inducing points are used, significantly reducing computational load for large datasets.
- **Flexibility:** Variational inference techniques can accommodate diverse kernel families, including cross-covariance kernels for multi-output problems. This allows practitioners to capture intricate correlations between different outputs without sacrificing scalability.
- **Broad applications:** Many high-impact domains, such as finance, routinely involve large-scale data (e.g., high-frequency trading) and multiple outputs (e.g., correlated volatility measures). SVGP-based multi-output GPs can handle these challenges efficiently, making them particularly well-suited to such settings.

5. Multi-GP Methodology in Volatility Modelling and Options Pricing

Volatility is widely used in numerous financial applications: it is a building block for structured products, serves for Greek calculations, and is crucial when calibrating advanced pricing models. A common method for pricing under advanced settings, such as the Heston model, is the Fast Fourier Transform (FFT) algorithm [7]. We employ the FFT method to generate our data for experiments and make use of the Heston model pricing library by [47]. In this section, we propose a multi-output Gaussian Process (MOGP) approach to approximate (i) the FFT-based implied volatility grid, (ii) options prices, and (iii) strangle and straddle portfolio prices. All experiments are performed under the Heston framework.

5.1. *Motivation and Data Generation*

The modeling procedure begins by constructing a training set. The input matrix $\mathbf{X}$ consists of N combinations of Heston model parameters:

$$\big(r,\, \rho,\, \kappa,\, \nu,\, \gamma,\, \nu_0\big),$$

where r is the risk-free rate, ρ is the correlation, κ is the mean reversion, ν is the long-run average variance, γ is the volatility of volatility, and ν_0 is the initial variance. The respective ranges of these parameters are provided in Table 1. The output matrix can consist of different combinations of strike prices K and maturities T.

The methodology includes two key components:

(i) *Data generation*: This step uses the FFT-based pricing method (or a suitable equivalent) to produce training data.
(ii) *MOGP predictor*: This component maps from the input matrix $\mathbf{X}$ to multiple outputs simultaneously, capturing dependencies across the outputs.

The outputs may include a temporal dimension (e.g., multiple maturities), a combination of model prices and volatilities, or even model parameters (e.g., strangle strikes). By combining these outputs into a single model, we can achieve parsimony, reducing the need for multiple separate models. This approach enables efficient backtesting of derivatives strategies, where all open positions can be simultaneously "marked-to-market" at regular intervals.

We demonstrate how the multi-output Gaussian process (MOGP) approach can

approximate functions for any underlying asset across different maturities, handling volatility, prices, and portfolio strategies.

Table 1. MOGP Input/Output and Ranges

Inputs	**Range**
Risk-free rate (r)	[1.5%–2.5%]
Correlation (ρ)	[−0.85–−0.45]
Mean reversion (κ)	[1.4–2.6]
Long-run average variance (ν)	[0.01–0.16]
Volatility of volatility (γ)	[0.35–0.75]
Instantaneous variance (ν_0)	[0.01–0.1]

Outputs	**Range**
Strike (K)	[0.5–1.5]
Maturity (M_1)	[0.1–0.5]
Maturity (M_2)	[0.6–1.2]
Maturity (M_3)	[1.4–2.0]

5.2. *Implementation and Results of the Strike-Maturity Grid*

Table 1 presents a typical strike-maturity grid for implied-volatility outputs. To achieve higher accuracy, different sub-models can be trained by varying the density of the parameter ranges as well as the specific maturities and strikes (moneyness).

A representative experimental setup could include:

- Normalizing all input parameters and outputs for stable training.
- Splitting the dataset into Train set = 619 data points, Test set = 110 data points, and selecting Inducing inputs = 124 for sparse GP approximation.
- Using an output dimension of $T = 20$, corresponding to 5×4 strike/maturity pairs in the implied volatility grid.
- Selecting strikes [0.5, 0.75, 1.0, 1.25, 1.5] and several maturities, for instance $T \in \{0.0, 1.0, 1.5\}$ or other sets of interest.

Figure 1 shows the plots obtained from the FFT analytical method compared to the multi-GP-based estimates. Figure 2 presents the relative errors of the approximation. The out-of-sample predictions closely match the analytical results, even when trained on a model with just 619 data points.

Figure 1. Comparison of errors of the Joint GP approximation against the FFT output in Heston volatility for varied strike-maturity pairs.

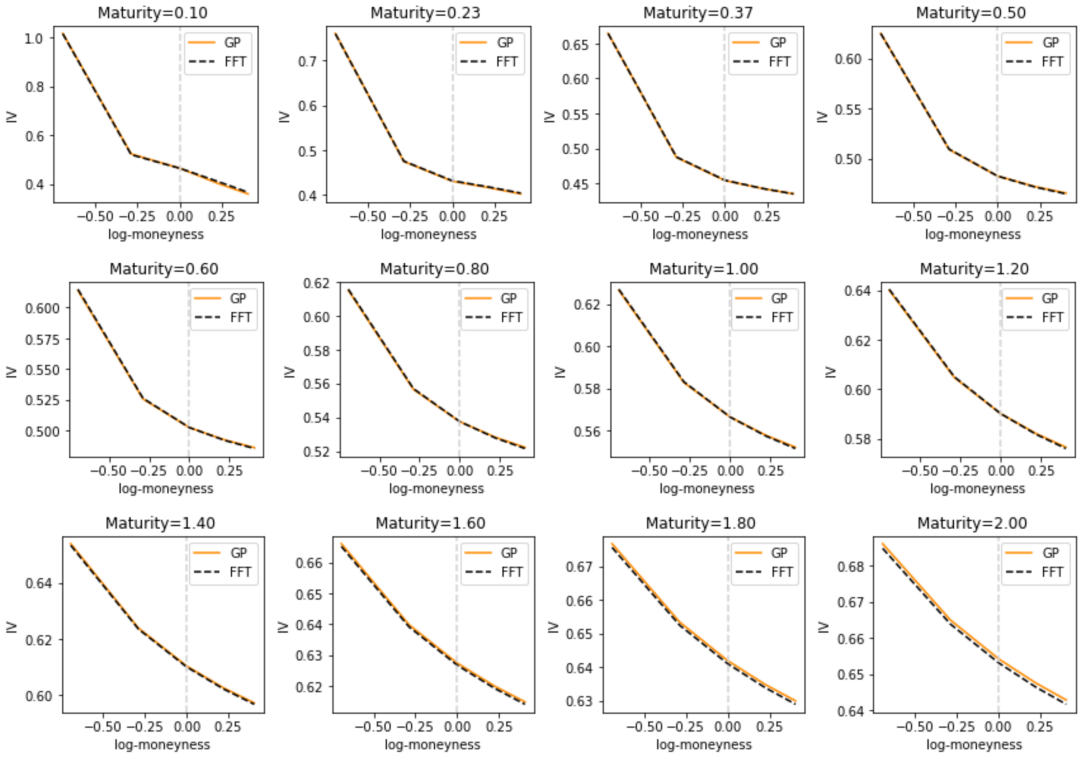
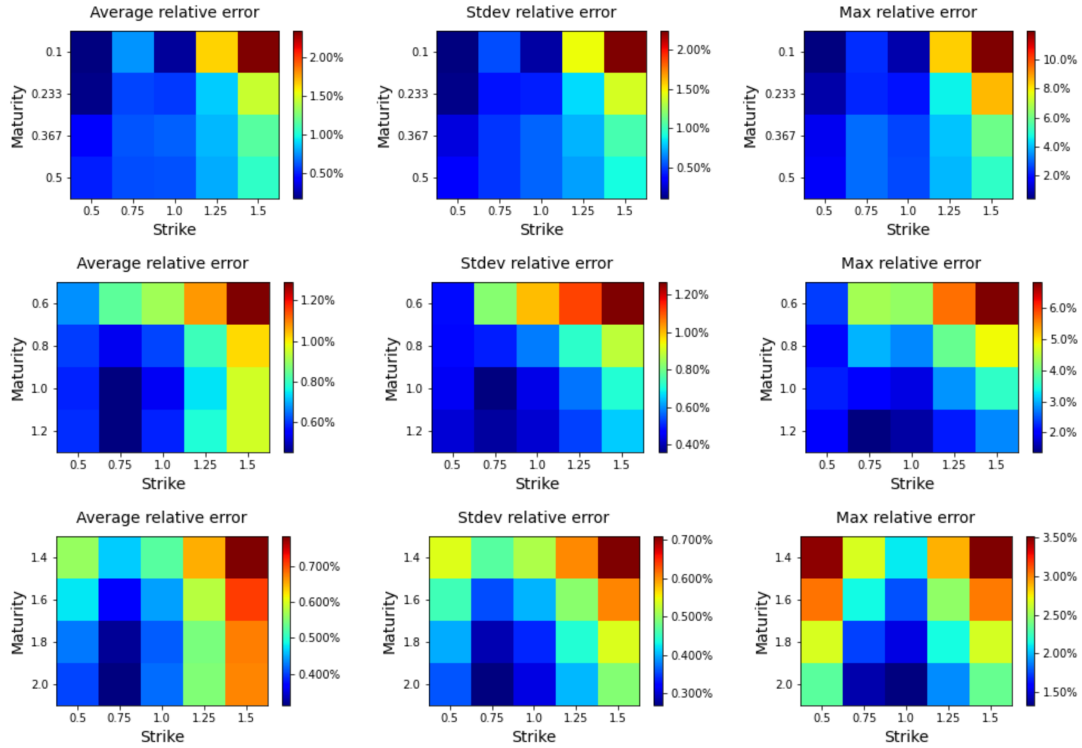


Figure 2. Comparison of surface errors of the GP approximation against the FFT in the Heston SV model.



5.3. *Call, Put, and Volatility in One GP Optimization*

In the previous section, we demonstrated a representative example of multi-GP, where volatilities (and equivalently, prices) can be computed for a range of combinations of strikes and maturities within a single model. In this section, we extend the model to simultaneously compute call and put prices, as well as call and put volatilities for a range of strikes at a given maturity in one optimization. Although we present the results for a single maturity, this approach can easily be extended to include multiple maturities.

Due to the scalability of the GP framework with variational inference and inducing points, we can incorporate multiple maturities without significant computational overhead. However, we observed that the performance near the boundaries of the strikes was suboptimal, particularly for shorter maturities. To address this, we restricted the training range to strikes between 0.3 and 0.17 for the model with $T = 0.5$, and between 0.5 and 0.15 for the models with $T = 1$ and $T = 2$.

The following configuration was used:

- Normalized input parameters and outputs.
- Train set: $N = 619$, Inducing inputs: $M = 124$, Test set: 110.
- Output dimension (Train): $T \in [30, 42]$, with strikes $[10, 16]$ for each of call, put, and volatility.
- Output dimension (Test): $T = 24$, with 8 strikes for each of call, put, and volatility.
- Strikes: $[0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.4, 1.5, 1.6, 1.7]$.
- Maturities in each model: $[0.5, 1, 2]$.

Figure 3. Comparison of errors of the Joint GP approximation against the FFT for Heston prices and volatility.

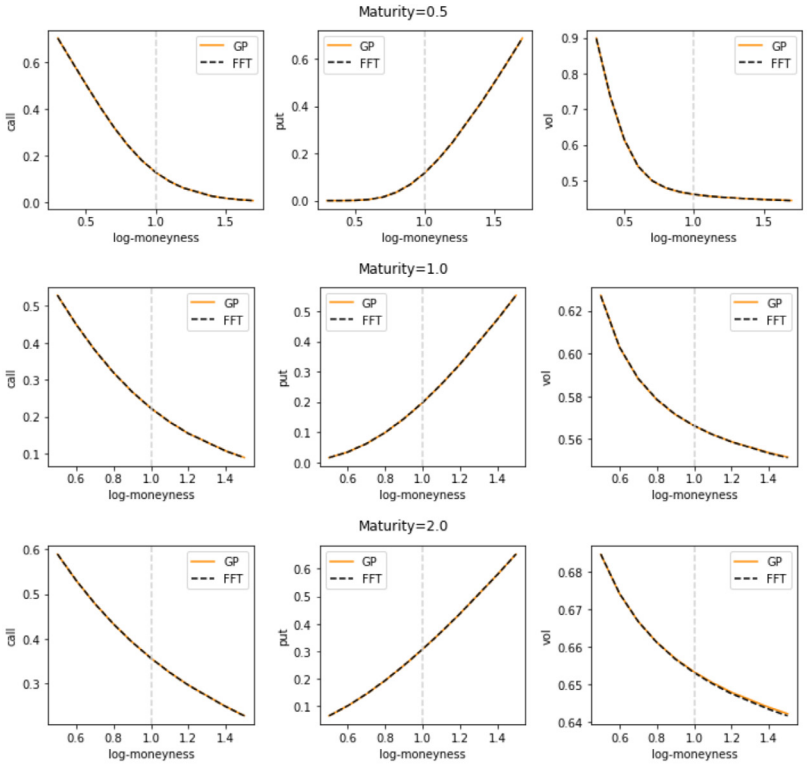
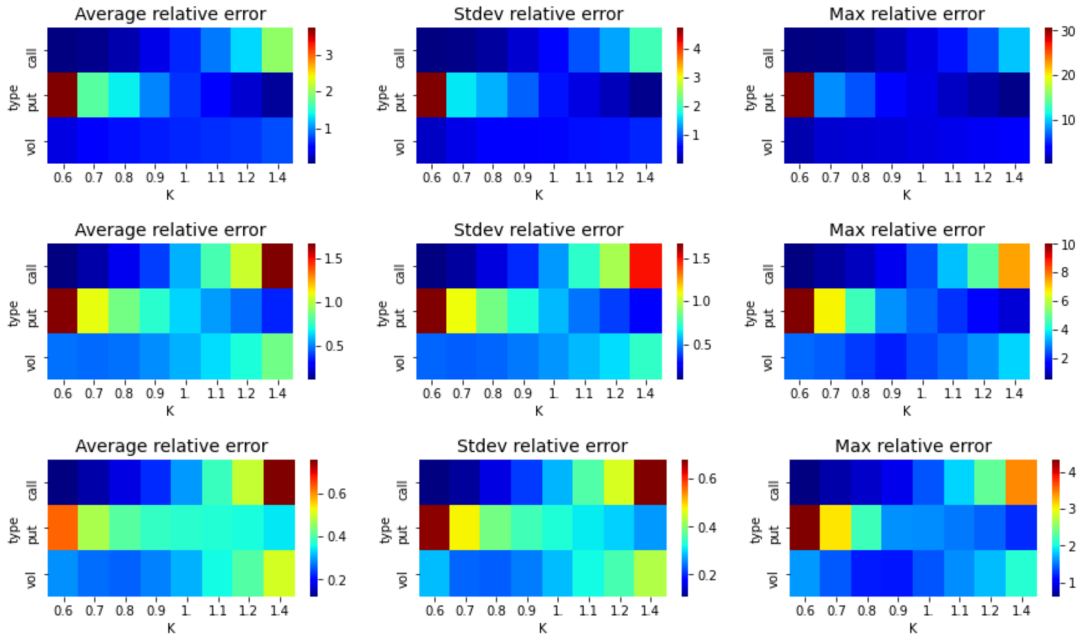


Figure 4. Call, put, and volatility results from the model.



5.4. *Straddle*

A long straddle is an options strategy where an investor buys both a call and a put option on the same underlying asset, with both options having the same strike price K and the same expiration date T. This strategy profits from significant price movements in either direction, as the value of the portfolio is given by the sum of the payoffs from the call and put options: $\pi = C + P$, where C is the payoff from the call option, and P is the payoff from the put option. This strategy is typically employed when an investor expects significant volatility but is uncertain about the direction of the price movement.

We considered three multi-asset models, each corresponding to a different set of maturities. The outputs for the straddle strategy are derived from a grid of call and put option prices for 24 different combinations of strike prices and maturities. These combinations form a comprehensive dataset that enables us to assess the performance of multi-output Gaussian Process (MOGP) models in approximating option prices and volatility across different maturities and strikes.

The following setup was used for training and testing:

- Normalized input parameters and outputs.
- Train set: $N = 619$, Inducing inputs: $M = 124$, Test set: 110.
- Output dimension: $T = 24$, 5×4 strike/maturity pairs of portfolio prices.
- Strikes: Eight equally spaced set between $[0.5, 1.5]$.
- Maturities in each model: Three equally spaced sets between $[0.1, 0.5]$, $[0.6, 1.2]$, and $[1.4, 2.0]$.

In the figures below, the rows represent the three different maturities: 6 months, 1 year, and 2 years, from top to bottom. During the data generation process, both the call and put option prices were computed using the same set of Heston model parameters as were used for volatility approximation in earlier sections. The model's

predictive performance on these option prices, specifically for the straddle portfolio, can be visualized in the figures below, showing how well the Gaussian Process model approximates the true prices and the associated error.

Figure 5. Straddle

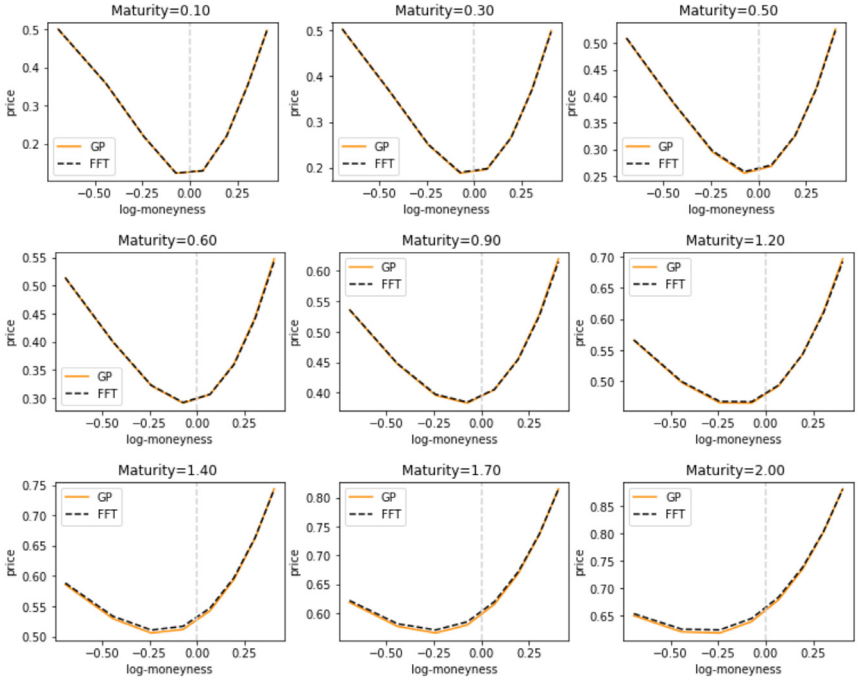
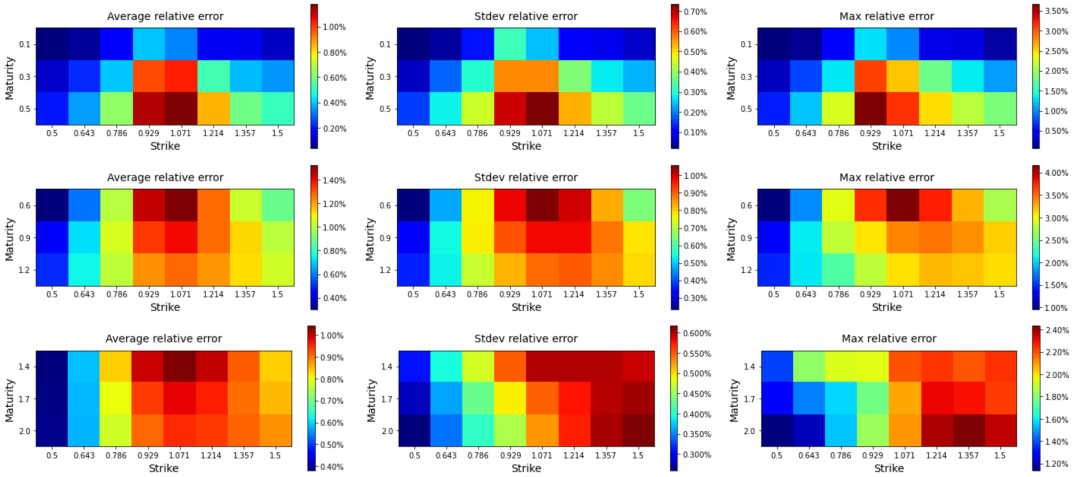


Figure 6. Comparison of errors of the GP approximation against the FFT in Hes- ton prices (Straddle portfolios)



5.5. *Strangle*

A long strangle is an options strategy in which an investor simultaneously buys a call option and a put option on the same underlying asset, but with different strike prices. Specifically, the call option has a strike price K_c, and the put option has a lower strike price K_p, where $K_p < K_c$. Both options share the same maturity date T. The strategy aims to profit from significant price movement in either direction, as the investor expects the underlying asset's price to move substantially beyond one of the strike prices. The total portfolio value at expiration is the sum of the payoffs from both options: $\pi = C + P$, where C represents the payoff from the call option, and P is the payoff from the put option.

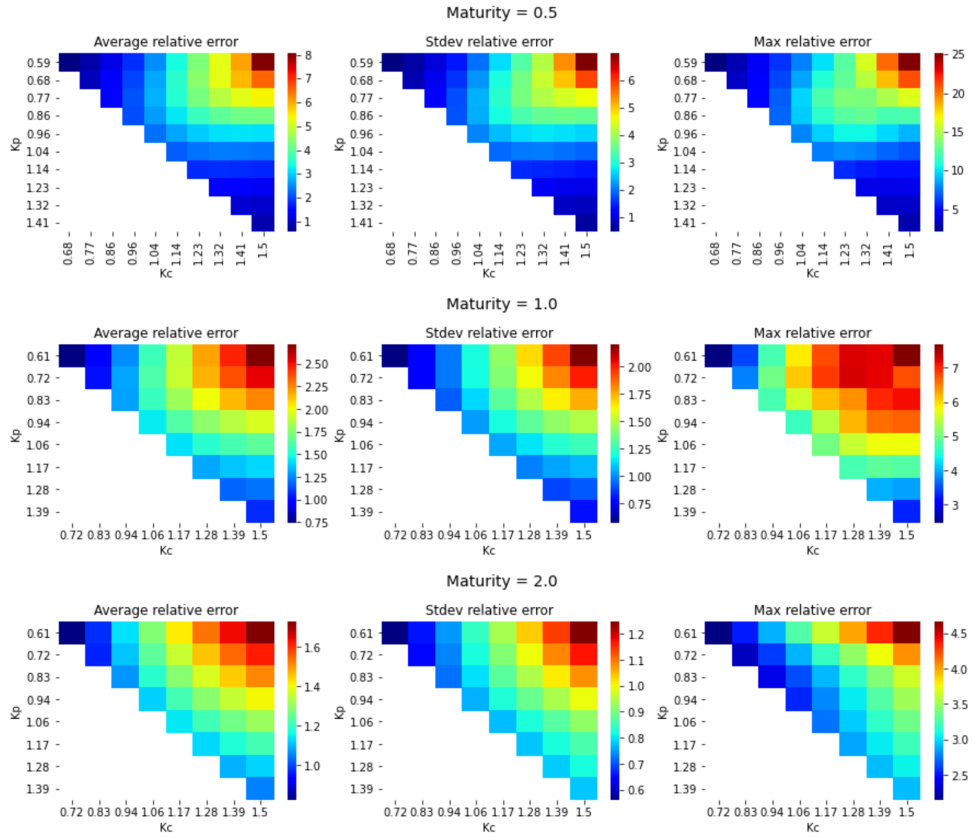
For the long strangle strategy, the key task is to compute the combined prices of the call and put options for various combinations of K_p and K_c. This involves computing pairs of option prices for different strike prices K_p and K_c, and then summing the individual payoffs. Using Gaussian Processes (GP) with variational inference, we can efficiently compute the total portfolio price for these combinations simultaneously.

We considered three multi-asset models, each with a different set of maturities. In the context of strangles, the outputs consist of pairs of call and put option prices for combinations where $K_p < K_c$. Given this constraint, there are a total of 21 unique combinations of strike prices to consider.

In the figures below, the rows correspond to maturities of 6 months, 1 year, and 2 years, from top to bottom. For the data generation, the call and put prices were computed using the same set of Heston model parameters as used for the volatility approximation. This ensures consistency across the models, allowing for a direct evaluation of the model's performance in predicting both option prices and volatilities in the multi-output framework.

- Normalized input parameters and outputs.
- Train set: $N = 619$, Inducing inputs: $M = 124$, Test set: 110.
- Outputs: $T = 21$ call and put strike pairs each with $K_p < K_c$.
- Strikes: Ten equally spaced set between $[0.5, 1.5]$.
- Maturity in each model: $[0.5, 1.0, 1.5]$.

Figure 7. Comparison of errors of the GP approximation against the FFT in Hes- ton prices (Strangle portfolios)



In Figures 1, 3, and 5, we can see that the mean across all strike-maturity combinations between multi-GP and FFT, and the relative error in Figures 2, 4, 6, and 7. The figures show that the average difference for most combinations is lower when considering portfolio values with shorter maturity. Both the mean and standard deviations are off by a small percentage for most of the estimates. The third image in each row represents the maximum error.

For those combinations of strikes and maturities where the errors are higher than the others, these can be improved by simply increasing the sample size. Generating asymmetrically higher amounts of input data for these points could be a good approach. Alternatively, concentrating the output more on those strike-maturity pairs will also improve the output. The tradeoff is between training size and either computation time per model or the number of models to compute.

6. Conclusion

In this paper, we presented an application of the Multi-GP methodology for option pricing and volatility modeling. By leveraging sparsity in the approximation and utilizing a multi-output framework, our approach significantly reduces the computational time and number of models required. We demonstrated how this methodology can be effectively used to model large sets of outputs, such as various strike-maturity combinations, in a Bayesian non-parametric framework. Additionally, the sparsity of the Gaussian Process (GP) approximation enables scalability, while the multi-output de-

sign captures correlations across different outputs, thus enhancing the consistency of our predictions.

Through the use of synthetic data generated from a range of Heston model parameters, we were able to approximate volatility and option portfolio pricing with fewer input variables and reduced matrix sizes. Our results show that the Multi-GP method provides accurate approximations, as evidenced by the comparison with FFT outputs for various strike-maturity pairs and option portfolios (see Figures 1 to 7).

The Multi-GP approach offers a significant advantage in terms of model efficiency, especially in scenarios where multiple outputs are required, such as different strike prices, maturities, or payoff structures. This not only reduces the computational burden but also ensures more consistent results across the various outputs. Moreover, the method streamlines the calibration and risk management tasks when multiple instruments share an underlying model, offering both practical and theoretical advancements in volatility modeling and option pricing.

The results of our experiments suggest that the Multi-GP approach is a valuable tool for improving the efficiency and accuracy of option pricing models, particularly when dealing with large-scale and complex portfolios. Future work could explore further enhancements to this methodology extending its applicability to a broader range of financial instruments.

References

[1] Aitsahlia, F., Goswami, M., Guha, S., 2010. American option pricing under stochastic volatility: An empirical evaluation. Computational Management Science 7, 189–206.

[2] Alvarez, M., Luengo, D., Titsias, M.K., Lawrence, N.D., 2010. Efficient multioutput Gaussian processes through variational inducing kernels, in: Journal of Machine Learning Research - Proceedings Track, 9, 25–32.

[3] Andersen, T., Bollerslev, T., Diebold, F., 2002. Parametric and nonparametric volatility measurement, in: Handbook of Financial Econometrics.

[4] Barone-Adesi, G., 2005. The saga of the American put. Journal of Banking & Finance 29, 2909–2918.

[5] Bauwens, L., 2000. Bayesian option pricing using asymmetric GARCH models. Journal of Empirical Finance 9, 321–342.

[6] Brignone, R., Gonzato, L., Lutkebohmert, E., 2023. Efficient Quasi-Bayesian Estimation of Affine Option Pricing Models Using Risk-Neutral Cumulants. Journal of Banking and Finance 148, 106745. 10.1016/j.jbankfin.2022.106745. ScienceDirect.

[7] Carr, P., Stanley, M., Madan, D., 2001. Option valuation using the fast Fourier transform. Journal of Computational Finance 2.

[8] Chapados, N., Dorion, C., 2010. Gaussian processes and non-parametric volatility forecasting.

[9] Chen, C.T., Chiang, S.M., Huang, T.C., 2020. The contagion effects of volatility indices across the U.S. and Europe. The North American Journal of Economics and Finance 54, 101234.

[10] Cordis, A.S., Kirby, C., 2014. Discrete stochastic autoregressive volatility. Journal of Banking & Finance 43, 160–178.

[11] Crépey, S., Dixon, M., 2020. Gaussian process regression for derivative portfolio modeling and application to credit valuation adjustment computations. The Journal of Computational Finance 24.

[12] Cuervo, E., Achcar, J., Barossi-Filho, M., 2014. New volatility models under a Bayesian perspective: a case study. Economia Aplicada 18, 179–197.

[13] Darsinos, T., Satchell, S., 2001. Bayesian analysis of the Black-Scholes option price, in:

Forecasting Expected Returns in the Financial Markets.
[14] Duan, J.C., 1995. The GARCH option pricing model. Mathematical Finance 5, 1, 13–32.
[15] Fernandes, M., Medeiros, M., Scharth, M., 2014. Modeling and predicting the CBOE market volatility index. Journal of Banking & Finance 40, 1–10.
[16] Granger, C., Poon, S.H., 2003. Forecasting volatility in financial markets: A review. Journal of Economic Literature 41, 478–539.
[17] Guidolin, M., Timmermann, A., 2001. Option prices under Bayesian learning: Implied volatility dynamics and predictive densities. Journal of Economic Dynamics and Control 27, 717–769.
[18] Gupta, A., Reisinger, C., 2011. Robust calibration of financial models using Bayesian estimators. Journal of Computational Finance 17.
[19] Gzyl, H., Ter Horst, E., Malone, S., 2006. Towards a Bayesian framework for option pricing. arXiv preprint arXiv:quant-ph/0610246.
[20] Hagan, P., Kumar, D., Lesniewski, A., Woodward, D., 2002. Managing Smile Risk. Wilmott Magazine, 2002, 84–108. SABR model introduced.
[21] Hensman, J., Fusi, N., Lawrence, N., 2013. Gaussian processes for big data, in: Uncertainty in Artificial Intelligence (UAI) - Proceedings of the 29th Conference.
[22] Ho, S.M., Lee, A., Marsden, A., 2011. Use of Bayesian estimates to determine the volatility parameter input in the Black-Scholes and binomial option pricing models. Journal of Risk & Financial Management 4, 74–96.
[23] Jang, H., Lee, J., 2019. Generative Bayesian neural network model for risk-neutral pricing of American index options. Quantitative Finance 19, 4, 587–603.
[24] Karolyi, G.A., 1993. A Bayesian approach to modeling stock return volatility for option valuation. Journal of Financial and Quantitative Analysis 28, 579–594.
[25] Liu, S., Oosterlee, C.W., Bohte, S.M., 2019. Pricing options and computing implied volatilities using neural networks. Risks 7, 1, 16.
[26] Lázaro-Gredilla, M., Figueiras-Vidal, A., 2009. Inter-domain Gaussian processes for sparse inference using inducing features, in: 1087–1095.
[27] Liu, C., Wang, C., Tran, M.-N., Kohn, R., 2023. Deep Learning Enhanced Realized GARCH. arXiv preprint arXiv:2302.08002.
[28] Liu, B., Kiskin, I., Roberts, S., 2020. An overview of Gaussian process regression for volatility forecasting. 681–686.
[29] Liu, H., Cai, J., Ong, Y.S., 2017. Remarks on multi-output Gaussian process regression. Knowledge-Based Systems 144.
[30] Lopes, H.F., Polson, N.G., 2010. Bayesian inference for stochastic volatility modeling, in: Rethinking Risk Measurement and Reporting: Uncertainty, Bayesian Analysis and Expert Judgement, pp. 515–551.
[31] Mitra, S., 2009. A review of volatility and option pricing.
[32] Mu, G., Godina, T., Maffia, A., Sun, Y., 2018. Supervised machine learning with control variates for American option pricing. Foundations of Computing and Decision Sciences 43, 207–217.
[33] Rizvi, S., Roberts, S., Osborne, M., Nyikosa, F., 2017. A novel approach to forecasting financial volatility with Gaussian process envelopes.
[34] Rudin, C., 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence 1, 206–215.
[35] Ruf, J., Wang, W., 2020. Neural networks for option pricing and hedging: A literature review.
[36] Rombouts, J.V., Stentoft, L., 2014. Bayesian option pricing using mixed normal heteroskedasticity models. Computational Statistics and Data Analytics 76, 588–605.
[37] Smith, M., Kohn, R., 1997. A Bayesian approach to nonparametric bivariate regression. Journal of the American Statistical Association 92, 440, 1522–1535.
[38] Snelson, E., Ghahramani, Z., 2006. Sparse Gaussian process using pseudo-inputs, in: Advances in Neural Information Processing Systems, 18, 1257–1264.
[39] Spiegeleer, J.D., Madan, D.B., Reyners, S., Schoutens, W., 2018. Machine learning for

quantitative finance: fast derivative pricing, hedging and fitting. Quantitative Finance 18, 10, 1635–1643.
[40] Tegnér, M., Roberts, S., 2019. A probabilistic approach to nonparametric local volatility.
[41] Titsias, M., 2009. Variational learning of inducing variables in sparse Gaussian processes, in: Journal of Machine Learning Research - Proceedings Track, 5, 567–574.
[42] Uhl, B., 2014. Flexible Bayesian modelling of implied volatility surfaces. International Journal of Financial Engineering and Risk Management 1, 355.
[43] van der Wilk, M., Dutordoir, V., John, S., Artemev, A., Adam, V., Hensman, J., 2020. A framework for interdomain and multioutput Gaussian processes.
[44] Venegas-Martínez, F., 2005. Bayesian inference, prior information on volatility, and option pricing: A maximum entropy approach. International Journal of Theoretical and Applied Finance 8, 1–12.
[45] Wang, Y., Choy, S.T.B., Wong, H.Y., 2016. Bayesian option pricing framework with stochastic volatility for FX data. Risks 4, 1–12.
[46] Wang, Y., Wong, H.Y., 2017. VIX forecast under different volatility specifications. Asia-Pacific Financial Markets 24, 1–18.
[47] Xu, J., 2016. Heston model pricing library. `https://github.com/junyanxu/Python-Heston-Option-Pricer`.
[48] Yin, K., Mondal, A., 2021. Bayesian Uncertainty Quantification of Local Volatility Model. arXiv preprint arXiv:2105.10210.
[49] Zhang, J., Shu, J., 2003. Pricing S&P 500 index options with Heston's model. 85–92.
[50] Zhao, J., Davison, M., Corless, R., 2007. Compact finite difference method for American option pricing. Journal of Computational & Applied Mathematics 206, 306–321.