

Faculté des sciences

Pricing European Derivatives via Constrained Gaussian Process Regression

Author: **Arnaud SERVAIS**
Supervisor: **Donatien HAINAUT**
Reader: **Valentin DENDONCKER**
Academic year 2025–2026
Master [120] en sciences actuarielles

Acknowledgements

First and foremost, I would like to thank my supervisor, Professor Hainaut. This work would not have been possible without his guidance, encouragement, and remarkable patience. I am especially grateful for the extensive material and initial direction he provided, which gave this project its essential early momentum and pointed me in the right direction.

Contents

Acknowledgements	1
Introduction	4
1 Derivative Pricing	5
1.1 Jump Model	5
1.2 Levy-Itô for Jump Model	6
1.3 Risk-Neutral Measure	8
1.4 Derivative Properties	9
1.4.1 European Basket Put Option	10
1.4.2 European Basket Call Option	11
1.4.3 Double Knock-Out Barrier Basket Put	11
1.5 Feynman-Kac Equation	12
2 Constrained Gaussian Process Regression	14
2.1 Gaussian Process Methodology	15
2.1.1 Distribution of $\mathcal{L}_x f(x)$	16
2.1.2 Prediction Distribution and Estimate	20
2.2 Fixed-Point Problem	23
2.2.1 Banach Spaces Framework	24
2.2.2 Application to the Feynman-Kac Operator	25
2.2.3 Implementation for the Algorithm	29
2.3 Matrix Conditionning and Numerical Stability	30
2.3.1 The Condition Number and Loss of Precision	31
2.3.2 Sources of Ill-Conditioning	32
2.3.3 Impact on Inversion and Predictions	33
2.3.4 Tikhonov Regularization	36
2.3.5 Cholesky Decomposition	38
2.4 Final Algorithm	42
2.5 Greeks Computation	43

2.6	Hyperparameters Estimation	45
2.6.1	Maximum Likelihood Estimation	45
2.6.2	Likelihood Optimization	46
2.6.3	Domain Normalization	47
3	Numerical Experiment	50
3.1	Heston-Kou Model	50
3.1.1	Kou Jump Dynamics	51
3.2	Single Asset Configuration	52
3.3	Two-Asset Basket Configuration	54
3.4	Domain Discretization with Sobol Sequence	54
3.4.1	Domain Mapping Equation	55
3.4.2	Expectation Evaluation via Gauss-Laguerre	55
3.5	Kernel Choices	57
3.5.1	Matérn 5/2 Kernel	57
3.5.2	Gaussian Kernel	59
3.6	Results	60
3.6.1	Single Asset Model	60
3.6.2	Double Asset Basket Model	69
	Conclusion	77
A	Derivations for the Matérn Kernel	79
A.1	Definition and Distance	79
A.2	First Derivatives	79
A.3	Second Derivatives	80
A.4	Single Application on the FK Operator	80
A.5	Second application on the FK Operator	81
A.6	Construction of the Hessian Matrix	83
A.7	Final Contracted Form	84
B	Derivation for the Gaussian Kernel	86
B.1	Definition and Derivatives	86
B.2	Final Contracted Form	87
	References	88

Introduction

Pricing European derivatives when the underlying asset is governed by a jump process presents significant computational challenges. The presence of jumps transforms the standard pricing partial differential equation into a complex partial integro-differential equation (PIDE). In this thesis, we propose a novel machine learning-based methodology to overcome these challenges by converting the problem of solving the Feynman-Kac (FK) PIDE into a model-constrained regression.

Rather than tackling the PIDE directly, our approach relies on an operator splitting technique. We decouple the pricing problem into a localized PDE, representing the continuous diffusion dynamics without the jump terms, and treat the integral jump term as a lagged source. By sampling state variables from both the interior domain and the boundary, we construct two distinct training sets. The regression function is first trained to fit the derivative's intrinsic value on the boundary sample while being constrained to satisfy the FK PDE on the interior sample. We then recover the full jump model solution by iteratively updating the jump integral via a Fixed-Point method.

By assuming the underlying regression function is a Gaussian process, our framework yields a powerful analytical advantage: it provides closed-form expressions for both the approximated option prices and their corresponding Greeks. To make this approach practically viable, we propose specific techniques to ensure the numerical stability of the algorithm and introduce an innovative hyperparameter estimation procedure.

We validate our proposed methods through comprehensive numerical experiments. The flexibility of the constrained Gaussian process framework is demonstrated by pricing a variety of instruments, ranging from standard Calls and Puts to path-dependent barrier options and high-dimensional basket options. Throughout these experiments, we systematically evaluate the performance of the model by employing both Matérn and Gaussian kernels and comparing it to a Monte-Carlo benchmark.

Chapter 1

Derivative Pricing

This chapter introduces the mathematical framework underlying our derivative pricing methodology. We focus primarily on jump models to govern the dynamics of the underlying assets and their driving factors. To price derivatives robustly, the assumed distribution of asset returns must accurately capture empirical market phenomena, such as volatility smile, return skewness, and excess kurtosis, that reflect the sudden, extreme market movements often missed by purely continuous diffusion models.

Furthermore, this generalized framework remains analytically tractable, allowing us to price complex multi-asset derivatives and integrate stochastic state variables, such as stochastic volatility as modelled in Bates (1993).

The remainder of this chapter contains the core definitions and distributional properties of the chosen jump models. Then, we detail the integration of stochastic volatility into the asset dynamics. Finally, we formalize the formulation of the multi-asset pricing equations that will be utilized in subsequent chapters.

1.1 Jump Model

We consider a financial market driven by a $p - 1$ risk processes vector $\mathbf{y}_t$. We assume those risk processes are following a stochastic jump model which is driven by the following stochastic differential equation:

$$d\mathbf{y}_t = \boldsymbol{\mu}(t, \mathbf{y}_t)dt + \Sigma(t, \mathbf{y}_t)d\mathbf{W}_t + M(t, \mathbf{y}_t)d\mathbf{L}_t \quad (1.1)$$

where $\boldsymbol{\mu}(t, \mathbf{y}_t)$ is a $p - 1$ vector, $\Sigma(t, \mathbf{y}_t)$ and $M(t, \mathbf{y}_t)$ are $p - 1 \times p - 1$ matrices. $\mathbf{W}_t$ is a $p - 1$ vector of independent brownian motions. Note that, even though

$\mathbf{W}_t$ and $\mathbf{L}_t$ contains independant stochastic processes, dependance between risk processes in $\mathbf{y}_t$ can be brought via $\Sigma(t, \mathbf{y}_t)$ and $M(t, \mathbf{y}_t)$ matrices.

Let $\mathbf{L}_t = (L_t^{(1)}, \dots, L_t^{(p-1)})^\top$ denote a $(p-1)$ -dimensional vector of mutually independent compound Poisson processes, where each component $v \in \{1, \dots, p-1\}$ is defined as:

$$L_t^{(v)} = \sum_{k=1}^{N_t^{(v)}} \xi_k^{(v)}$$

where, for each $v \in \{1, \dots, p-1\}$, the following conditions hold:

- $(N_t^{(v)})_{t \geq 0}$ is a homogeneous Poisson process with intensity parameter $\lambda_v \in (0, \infty)$.
- $(\xi_k^{(v)})_{k=1}^\infty$ is a sequence of independent and identically distributed random variables representing the jump sizes, with the same distribution as a generic random variable $\xi^{(v)}$. We assume the jump sizes possess finite moments, such that $\mathbb{E}[|\xi^{(v)}|^m] < \infty$ for $m > 0$.
- The sequence of jump sizes $(\xi_k^{(v)})_{k=1}^\infty$ is strictly independent of the counting process $(N_t^{(v)})_{t \geq 0}$.

Moreover, we specify some conditions bounding the coefficients $\boldsymbol{\mu}$, Σ and M :

$$\begin{aligned} \mathbb{E} \left(\int_0^t \boldsymbol{\mu}(s, \mathbf{y}_s) ds \right) &\leq \infty, \quad \forall t < \infty, \\ \mathbb{E} \left(\int_0^t \Sigma(s, \mathbf{y}_s) \Sigma_{\mathbf{y}}(s, \mathbf{y}_s)^\top ds \right) &\leq \infty, \quad \forall t < \infty, \\ \mathbb{E} \left(\int_0^t M(s, \mathbf{y}_s) M(s, \mathbf{y}_s)^\top ds \right) &\leq \infty, \quad \forall t < \infty, \end{aligned}$$

These specific conditions exist to make sure the financial processes modelled in $\mathbf{y}_t$ are bounded.

1.2 Levy-Itô for Jump Model

In this section, we recall the Levy-Itô Lemma for jump models as introduced in Cont and Tankov 2004. It will allow us to find the dynamics of a function that depends on the time t and the risk processes $\mathbf{y}_t$. Such a function can, in particular, be the price function of a derivative.

Lemma 1 (Levy-Itô Lemma for Jump diffusions). *Let $f : [0, T] \times \mathbb{R}^{p-1} \rightarrow \mathbb{R}$ be a scalar function of class $C^{1,2}([0, T] \times \mathbb{R}^{p-1})$. Suppose the $(p-1)$ -dimensional risk process $\mathbf{y}_t$ is driven by the stochastic differential equation stated in 1.1*

Then, the stochastic differential of $f(t, \mathbf{y}_t)$ is given by:

$$df(t, \mathbf{y}_t) = \left(\frac{\partial f}{\partial t} + \nabla_{\mathbf{y}} f^\top \boldsymbol{\mu} + \frac{1}{2} \text{Tr}(\Sigma \Sigma^\top \nabla_{\mathbf{y}}^2 f) \right) dt + \nabla_{\mathbf{y}} f^\top \Sigma d\mathbf{W}_t + [f(t, \mathbf{y}_{t-} + M \Delta \mathbf{L}_t) - f(t, \mathbf{y}_{t-})] \quad (1.2)$$

where the drift $\boldsymbol{\mu}$, the continuous volatility Σ , the jump matrix M , the gradient $\nabla_{\mathbf{y}} f$ with respect to $\mathbf{y}_t$, and the Hessian $\nabla_{\mathbf{y}}^2 f$ with respect to $\mathbf{y}_t$ are evaluated at $(t, \mathbf{y}_{t-})$. The term $\Delta \mathbf{L}_t = \mathbf{L}_t - \mathbf{L}_{t-}$ denotes the jump vector at time t .

While Equation (1.2) provides the general representation, we can refine this expression by exploiting the specific structure of our jump vector $\mathbf{L}_t$. By definition, the $p-1$ components of $\mathbf{L}_t$ are mutually independent compound Poisson processes. A fundamental property of independent Poisson processes is that the probability of simultaneous jumps is zero almost surely.

Consequently, at any jump time t , exactly one component $N_t^{(v)}$ increments. We can therefore rewrite the differential of the jump vector as a linear combination of individual indicator differentials $dN_t^{(v)}$:

$$d\mathbf{L}_t = \sum_{v=1}^{p-1} \mathbf{e}_v \xi^{(v)} dN_t^{(v)} \quad (1.3)$$

where $\mathbf{e}_v$ is the v -th standard basis vector of $\mathbb{R}^{p-1}$, and $\xi^{(v)}$ is the jump size realized upon the arrival of $dN_t^{(v)}$. Substituting this into our risk process dynamics, the jump matrix multiplication simplifies to:

$$M(t, \mathbf{y}_{t-}) d\mathbf{L}_t = \sum_{v=1}^{p-1} M_v(t, \mathbf{y}_{t-}) \xi^{(v)} dN_t^{(v)} \quad (1.4)$$

where M_v represents the v -th column of the matrix M . Because only one $dN_t^{(v)}$ can be non-zero at any instant t , the jump term in the Itô formula decomposes into a sum over the individual counting processes. The multi-dimensional Itô formula for our specific model thus simplifies to the following explicit representation:

$$df(t, \mathbf{y}_t) = \left(\frac{\partial f}{\partial t} + \nabla_{\mathbf{y}} f^\top \boldsymbol{\mu} + \frac{1}{2} \text{Tr}(\Sigma \Sigma^\top \nabla_{\mathbf{y}}^2 f) \right) dt + \nabla_{\mathbf{y}} f^\top \Sigma d\mathbf{W}_t + \sum_{v=1}^{p-1} [f(t, \mathbf{y}_{t-} + M_v \xi^{(v)}) - f(t, \mathbf{y}_{t-})] dN_t^{(v)} \quad (1.5)$$

1.3 Risk-Neutral Measure

To price derivatives, we must consider a market where no arbitrage is possible. In other words, a market where no self-financing portfolio starting with zero initial wealth can guarantee a riskless positive return. The standard way to enforce this condition is to evaluate assets under the risk-neutral measure $\mathbb{Q}$. Under this probability measure, investors are theoretically assumed to be indifferent to risk. Consequently, market risk premiums are absorbed into the probability distribution itself, meaning that the expected return on any tradable asset is exactly equal to the risk-free interest rate, $r(t, \mathbf{y}_t)$.

Mathematically, this implies that the discounted price process of any non-dividend-paying tradable asset must be a martingale under $\mathbb{Q}$. Assuming that the state vector $\mathbf{y}_t$ defined in Equation (1.1) directly represents the prices of $p - 1$ tradable assets, we can determine the specific functional form of the continuous drift vector $\mu^{\mathbb{Q}}(t, \mathbf{y}_t)$ required to enforce this no-arbitrage condition.

Under the risk-neutral measure, the dynamics of the asset prices are governed by:

$$d\mathbf{y}_t = \mu^{\mathbb{Q}}(t, \mathbf{y}_t)dt + \Sigma(t, \mathbf{y}_t)d\mathbf{W}_t^{\mathbb{Q}} + M(t, \mathbf{y}_t)d\mathbf{L}_t^{\mathbb{Q}}, \quad (1.6)$$

where $\mathbf{W}_t^{\mathbb{Q}}$ and $\mathbf{L}_t^{\mathbb{Q}}$ are respectively the Brownian motion and the compound Poisson processes under $\mathbb{Q}$. It's important to note that such a $\mathbb{Q}$ measure is not necessarily unique and we can specify it so that only the time drift $\mu^{\mathbb{Q}}$ is modified but $\mathbf{W} = \mathbf{W}_t^{\mathbb{Q}}$ and $\mathbf{L} = \mathbf{L}_t^{\mathbb{Q}}$ stay unchanged.

To prevent arbitrage, the total expected instantaneous rate of return of tradable asset $y_t^{(v)}$ must equal $r(t, \mathbf{y}_t)y_t^{(v)}$:

$$\mathbb{E}^{\mathbb{Q}} [dy_t^{(v)} | \mathcal{F}_t] = r(t, \mathbf{y}_t)y_t^{(v)}dt \quad (1.7)$$

We can now set $\mu_v^{\mathbb{Q}}$ to a value that verifies 1.7. For all v such that $y_t^{(v)}$ is a tradable asset:

$$\begin{aligned} r(t, \mathbf{y}_t)y_t^{(v)}dt &= \mathbb{E}^{\mathbb{Q}} [dy_t^{(v)} | \mathcal{F}_t] \\ &= \mathbb{E}^{\mathbb{Q}} [\mu_v^{\mathbb{Q}}(t, \mathbf{y}_t)dt + \Sigma_v(t, \mathbf{y}_t)d\mathbf{W}_t + M_v(t, \mathbf{y}_t)d\mathbf{L}_t | \mathcal{F}_t] \\ &= \mu_v^{\mathbb{Q}}(t, \mathbf{y}_t)dt + \mathbb{E}^{\mathbb{Q}} [M_v(t, \mathbf{y}_t)d\mathbf{L}_t | \mathcal{F}_t] \end{aligned}$$

To compute the expectation of the jump component, let $M_{v,j}(t, \mathbf{y}_t)$ denote the element in the v -th row and j -th column of the matrix $M(t, \mathbf{y}_t)$. The matrix-vector multiplication $M_v(t, \mathbf{y}_t)d\mathbf{L}_t$ can be explicitly written as the sum $\sum_{j=1}^{p-1} M_{v,j}(t, \mathbf{y}_t)dL_t^{(j)}$.

Because $\mathbf{L}_t$ is a vector of independent compound Poisson processes, the expected value of the increment $dL_t^{(j)}$ over an infinitesimal interval dt , conditional on the filtration $\mathcal{F}_t$, depends solely on its jump intensity λ_j and expected jump size $\mathbb{E}^\mathbb{Q}[\xi^{(j)}]$ under the risk-neutral measure:

$$\mathbb{E}^\mathbb{Q} [dL_t^{(j)} | \mathcal{F}_t] = \lambda_j \mathbb{E}^\mathbb{Q} [\xi^{(j)}] dt \quad (1.8)$$

By the linearity of the conditional expectation, substituting this property into the matrix product yields:

$$\begin{aligned} \mathbb{E}^\mathbb{Q} [M_v(t, \mathbf{y}_t) d\mathbf{L}_t | \mathcal{F}_t] &= \mathbb{E}^\mathbb{Q} \left[\sum_{j=1}^{p-1} M_{v,j}(t, \mathbf{y}_t) dL_t^{(j)} \middle| \mathcal{F}_t \right] \\ &= \left(\sum_{j=1}^{p-1} M_{v,j}(t, \mathbf{y}_t) \lambda_j \mathbb{E}^\mathbb{Q} [\xi^{(j)}] \right) dt \end{aligned}$$

Returning to the no-arbitrage condition in Equation (1.7) and substituting the resolved expectation, we have:

$$r(t, \mathbf{y}_t) y_t^{(v)} dt = \mu_v^\mathbb{Q}(t, \mathbf{y}_t) dt + \left(\sum_{j=1}^{p-1} M_{v,j}(t, \mathbf{y}_t) \lambda_j^\mathbb{Q} \mathbb{E}^\mathbb{Q} [\xi^{(j)}] \right) dt \quad (1.9)$$

Dividing both sides by the infinitesimal time step dt and isolating $\mu_v^\mathbb{Q}(t, \mathbf{y}_t)$ provides the explicit risk-neutral drift for the v -th asset:

$$\mu_v^\mathbb{Q}(t, \mathbf{y}_t) = r(t, \mathbf{y}_t) y_t^{(v)} - \sum_{j=1}^{p-1} M_{v,j}(t, \mathbf{y}_t) \lambda_j^\mathbb{Q} \mathbb{E}^\mathbb{Q} [\xi^{(j)}] \quad (1.10)$$

1.4 Derivative Properties

A financial derivative is a contract whose value depends on the trajectory of an underlying state vector, $\mathbf{y}_t$. The terms of this contract are strictly defined by its maturity date, T , and its terminal payoff function, $\mathcal{P}(T, \mathbf{y}_T)$, which determines the exact cash flow exchanged between parties at expiration¹.

Let $f(t, \mathbf{y}_t)$ denote the price of the derivative at time t given the state vector $\mathbf{y}_t$. The domain of this pricing function is defined as $[0, T] \times [y_{\min}^{(1)}, y_{\max}^{(1)}] \times \cdots \times [y_{\min}^{(p-1)}, y_{\max}^{(p-1)}]$. Depending on the dynamics of the underlying risk processes $\mathbf{y}_t$, the theoretical

¹In this work, we only consider derivatives with cash flows exchanged at maturity (e.g., European options).

spatial domain may be unbounded. To facilitate numerical pricing on a finite grid, we truncate the state space by setting the boundaries $y_{\max}^{(v)}$ and $y_{\min}^{(v)}$ to sufficiently extreme values, exploiting the fact that the probability of the processes reaching these limits is negligible.

We decompose this closed domain into two sets:

$$\mathcal{X} = \mathring{\mathcal{X}} \cup \partial\mathcal{X} \quad (1.11)$$

where $\partial\mathcal{X}$ represents the boundaries where the value of f is deterministically known and $\mathring{\mathcal{X}}$ represents the remaining region. What we mean by "deterministically known" is explained with examples in the next sections. We denote the value of the derivative on the boundary region $h(t, \mathbf{y}_t)$:

$$\forall t, \mathbf{y}_t \in \partial\mathcal{X} : f(t, \mathbf{y}_t) = h(t, \mathbf{y}_t) \quad (1.12)$$

1.4.1 European Basket Put Option

To exemplify the formulation of boundary conditions on $\partial\mathcal{X}$, we consider a European basket put option. Let K denote the fixed strike price of the option, and let $\mathbf{w} = (w_1, \dots, w_{p-1})^\top$ represent the vector of positive weights assigned to each of the $p - 1$ risk processes.

The payoff of a European basket put at maturity T depends on the weighted sum of the underlying asset prices. The contract grants the holder the right to sell this basket at the strike price K . Therefore, the terminal payoff function is given by:

$$\mathcal{P}(\mathbf{y}_T) = \max(K - \mathbf{w}^\top \mathbf{y}_T, 0) \quad (1.13)$$

This payoff structure dictates the exact values of the pricing function $f(t, \mathbf{y})$ along the boundary $\partial\mathcal{X}$ of our computational domain. At the exact moment of maturity, the time value of the derivative vanishes. The price of the option across the entire state space must perfectly match the contractual payoff:

$$h(T, \mathbf{y}_T) = \mathcal{P}(\mathbf{y}_T) \quad (1.14)$$

If the prices of all underlying assets in the basket fall to zero, the basket becomes entirely worthless. Indeed, once an asset has a market value of 0 it can't recover from it. In this extreme scenario, the option will undeniably be exercised at maturity, yielding a guaranteed payoff of K . Consequently, the current price of the option on this lower boundary is simply the deterministic present value of the strike price, discounted at the risk-free rate:

$$h(t, \mathbf{y}_t) = Ke^{-r(T-t)} \quad \forall (t, \mathbf{y}_t) \in \partial\mathcal{X} \text{ such that: } \mathbf{y}_t = \mathbf{0} \quad (1.15)$$

1.4.2 European Basket Call Option

Conversely, we can define the boundary conditions for a European basket call option, which grants the holder the right to buy the weighted basket of assets at a fixed strike price K . Using the same strictly positive weight vector $\mathbf{w}$, the terminal payoff function at maturity T is:

$$\mathcal{P}(\mathbf{y}_T) = \max(\mathbf{w}^\top \mathbf{y}_T - K, 0) \quad (1.16)$$

As with the put option, the pricing function $f(t, \mathbf{y})$ must satisfy the terminal condition at maturity:

$$h(T, \mathbf{y}_T) = \mathcal{P}(\mathbf{y}_T) \quad (1.17)$$

Now, if the weighted value of the underlying assets falls to zero ($\mathbf{w}^\top \mathbf{y}_t = 0$), the basket will never recover to surpass the strike price K . Therefore, the call option is deeply out-of-the-money and completely worthless. This yields:

$$h(t, \mathbf{y}_t) = 0 \quad \forall (t, \mathbf{y}_t) \in \partial\mathcal{X} \text{ such that } \mathbf{y}_t = \mathbf{0} \quad (1.18)$$

1.4.3 Double Knock-Out Barrier Basket Put

Path-dependent derivatives, such as barrier options, impose strict, finite boundaries on the computational domain $\mathcal{X}$, which is very convenient for our truncated domain.

Consider a Double Knock-Out barrier basket put option. This contract behaves identically to a standard basket put, provided that the values of the underlying assets never breach a lower barrier $B_L^{(v)}$ or an upper barrier $B_U^{(v)}$ during their lifetime.

If the value of an underlying asset $y_t^{(v)}$ touches or crosses either barrier at any time t , the option is immediately terminated (knocked out) and becomes worthless. Assuming a zero rebate upon knock-out, the terminal payoff function for paths that survive until maturity is:

$$\mathcal{P}(\mathbf{y}_T) = \begin{cases} \max(K - \mathbf{w}^\top \mathbf{y}_T, 0) & \text{if } \forall v : B_L^{(v)} < y_T^{(v)} < B_U^{(v)} \\ 0 & \text{otherwise} \end{cases} \quad (1.19)$$

Because the option price drops to zero the instant a barrier is breached, $\partial\mathcal{X}$ covers a bigger part of the entire boundary of $\mathcal{X}$, which will give more information about the intrinsic value of the option to the regressor. This yields conditions across all risk processes boundary $\partial\mathcal{X}$ for $t \leq T$:

$$h(t, \mathbf{y}_t) = 0 \quad \forall \mathbf{y}_t \in \partial\mathcal{X} \text{ such that } \forall v : y_t^{(v)} \leq B_L^{(v)} \text{ or } y_t^{(v)} \geq B_U^{(v)} \quad (1.20)$$

1.5 Feynman-Kac Equation

In this section, we derive the Partial Integro-Differential equation (PIDE) that drives the dynamics of the derivative price.

Let $f(t, \mathbf{y}_t) \in C^{1,2}([0, T] \times \mathbb{R}^{p-1})$ denote the price of a derivative with a terminal payoff $\mathcal{P}(\mathbf{y}_T)$ at maturity T . We assume that the dynamics in Equation (1.1) are given under a risk-neutral pricing measure $\mathbb{Q}$, and let $r(t, \mathbf{y}_t)$ be the risk-free interest rate. As the option is a tradable asset, its expected return must not exceed the risk free rate:

$$\mathbb{E}^{\mathbb{Q}}[df(t, \mathbf{y}_t)|\mathcal{F}_t] = r(t, \mathbf{y}_t)f(t, \mathbf{y}_t) \quad (1.21)$$

By substituting df with its expression obtained in equation 1.5, we get:

$$\begin{aligned} r(t, \mathbf{y}_t)f(t, \mathbf{y}_t)dt &= \mathbb{E}^{\mathbb{Q}}[df(t, \mathbf{y}_t)|\mathcal{F}_t] \\ \Leftrightarrow rf dt &= \left(\frac{\partial f}{\partial t} + \nabla_{\mathbf{y}} f^{\top} \mu^{\mathbb{Q}} + \frac{1}{2} \text{Tr} \left(\Sigma \Sigma^{\top} \nabla_{\mathbf{y}}^2 f \right) \right) dt \\ &\quad + \sum_{v=1}^{p-1} \lambda_v dt \mathbb{E}^{\mathbb{Q}} \left[f(t, \mathbf{y}_{t-} + M_v \xi^{(v)}) - f(t, \mathbf{y}_{t-}) | \mathcal{F}_t \right] \\ 0 &= \left(\frac{\partial f}{\partial t} + \nabla_{\mathbf{y}} f^{\top} \mu^{\mathbb{Q}} + \frac{1}{2} \text{Tr} \left(\Sigma \Sigma^{\top} \nabla_{\mathbf{y}}^2 f \right) - rf \right) dt \\ &\quad + \sum_{v=1}^{p-1} \lambda_v dt \mathbb{E}^{\mathbb{Q}} \left[f(t, \mathbf{y}_{t-} + M_v \xi^{(v)}) - f(t, \mathbf{y}_{t-}) | \mathcal{F}_t \right] \end{aligned}$$

By removing dt , we get:

$$\begin{aligned} 0 &= \frac{\partial f}{\partial t} - rf + \nabla_{\mathbf{y}} f^{\top} \mu^{\mathbb{Q}} + \frac{1}{2} \text{Tr} \left(\Sigma \Sigma^{\top} \nabla_{\mathbf{y}}^2 f \right) \\ &\quad + \sum_{v=1}^{p-1} \lambda_v \left(\mathbb{E}^{\mathbb{Q}} \left[f(t, \mathbf{y}_{t-} + M_v \xi^{(v)}) | \mathcal{F}_t \right] - f(t, \mathbf{y}_{t-}) \right) \end{aligned} \quad (1.22)$$

The right-hand side of Equation (1.22) constitutes the complete infinitesimal generator of the jump-diffusion process. While the entire operator is linear with respect to f , we decompose it into a local differential component (denoted $\mathcal{L}f$) and a non-local integral component (denoted Δf) for computational tractability. This explicit separation is motivated by the fact that, depending on the underlying probability distribution chosen for the jump sizes $\xi^{(v)}$, the expectation term frequently lacks a closed-form analytical solution. By splitting the operator, we can evaluate the continuous derivatives analytically while isolating the non-local jump compensator,

which often requires dedicated numerical integration schemes.

$$\begin{cases} \mathcal{L}f(t, \mathbf{y}_t) = \frac{\partial f}{\partial t} - rf + \nabla_{\mathbf{y}} f^\top \mu^{\mathbb{Q}} + \frac{1}{2} \text{Tr} \left(\Sigma \Sigma^\top \nabla_{\mathbf{y}}^2 f \right) \\ \Delta f(t, \mathbf{y}_t) = \sum_{v=1}^{p-1} \lambda_v \left(\mathbb{E}^{\mathbb{Q}} \left[f(t, \mathbf{y}_{t^-} + M_v \xi^{(v)}) | \mathcal{F}_t \right] - f(t, \mathbf{y}_{t^-}) \right) \end{cases} \quad (1.23)$$

The last two sections allow to formally write the derivative pricing problem into a regression problem, constrained on the domain by:

$$\begin{cases} \mathcal{L}f(t, \mathbf{y}_t) = -\Delta f(t, \mathbf{y}_t) & \forall (t, \mathbf{y}_t) \in \mathring{\mathcal{X}}, \\ f(t, \mathbf{y}_t) = h(t, \mathbf{y}_t) & \forall (t, \mathbf{y}_t) \in \partial \mathcal{X}, \end{cases} \quad (1.24)$$

Chapter 2

Constrained Gaussian Process Regression

In this chapter, we detail the core methodology of this work: employing Constrained Gaussian Process Regression (CGPR) to approximate the solution to the Feynman-Kac PIDE presented in Equation (1.24).

We begin by outlining the foundations of standard Gaussian processes. Drawing on the framework provided by Rasmussen and Williams 2005, we define the necessary regression mechanics, prior distributions, and covariance functions (kernels) that underpin our model. We then introduce the methodology required to directly constrain the Gaussian process structure with the FK PIDE. To achieve this, we apply the boundary value problem regression techniques explained by Gulian, Frankel, and Swiler 2020, alongside the specific adaptations for European option pricing without jumps, proposed by Hainaut and Vrans 2024.

Because the underlying asset dynamics include jumps, the resulting PIDE features a complex, non-local integral term that complicates direct estimation. To overcome this, we temporarily decouple the problem to solve a relaxed, purely continuous partial differential equation. In the later sections of this chapter, we demonstrate how to reconcile this relaxed continuous solution with the full jump-diffusion dynamics. By treating the jump integral as a lagged source term, we apply a Fixed-Point iteration scheme, updating the methodology of Hainaut 2026, to iteratively converge to the complete option price.

2.1 Gaussian Process Methodology

The aim of the constrained gaussian regression problem is to estimate the function f that is solution to the following boundary value problem:

$$\begin{cases} \mathcal{L}f(x) = z(x), & x \in \mathcal{X}, \\ f(x) = h(x), & x \in \partial\mathcal{X}, \end{cases} \quad (2.1)$$

where $\mathcal{L}$ denotes a linear partial differential operator, $\mathcal{X}$ a domain with boundary $\partial\mathcal{X}$, and z a known function. The method assumes that for any finite set of points $\mathbf{X} = \{x_1, \dots, x_N\} \subset \mathcal{X}$, the corresponding vector of function values $f(\mathbf{X})$ follows a multivariate Gaussian distribution:

$$f(\mathbf{X}) \sim \mathcal{N}(m(\mathbf{X}), k(\mathbf{X}, \mathbf{X})), \quad (2.2)$$

where $m(\mathbf{X})$ is the mean vector, and $k(\mathbf{X}, \mathbf{X})$ is the covariance matrix. The elements of this matrix are defined by a covariance function (or kernel) k :

$$\forall x, x' \in \mathcal{X} : \quad k(x, x') = \mathbb{C}(f(x), f(x')). \quad (2.3)$$

The selection of the kernel function can be chosen arbitrarily and a wide variety of kernel families exist in the literature. Fundamentally, these functions are constructed to encode spatial dependency: for any pair of points $x, x' \in \mathcal{X}$, a smaller distance between them yields a higher covariance between their corresponding function values $f(x)$ and $f(x')$. In other words, nearby points in the domain exhibit similar function values. To illustrate the concept, we present the Gaussian kernel:

$$k(x, x') = a^2 \exp\left(-\sum_{v=1}^p \frac{(x_v - x'_v)^2}{2\eta_v^2}\right) \quad (2.4)$$

In this formulation, the mathematical structure explicitly mirrors the spatial behavior described above. The sum in 2.4 represents the weighted Euclidian distance between the two points within the domain. When the two points are perfectly coincident ($x = x'$), the distance is zero and the exponential term evaluates to one, yielding the maximum covariance. Conversely, as the spatial distance $\|x - x'\|$ increases, the negative exponent causes the kernel value to decay rapidly toward zero, formalizing the assumption that distant points exert negligible statistical influence on one another. The hyperparameters η_v represents the characteristic lengthscales for each dimensions, which dictates the importance of each variables in the covariance computation. Finally, a is a covariance scaling hyperparameter.

In practice and as we go further in this work, we specify a prior mean function of zero, setting $m(\mathbf{X}) = 0$ in equation (2.2). This assumption is not a restriction

on the model, but rather a standard convention that significantly simplifies the mathematical derivations. The reasoning is twofold. First, the covariance kernel $k(\mathbf{X}, \mathbf{X}')$ is flexible enough to capture the variations of f ; consequently, the posterior mean of $f(x)$ knowing $f(\mathbf{X})$ will naturally adapt to the observations and boundary conditions even with a zero prior mean¹. Second, in the specific context of solving partial differential equations via model-constrained regression, applying linear differential operators to a zero-mean Gaussian process eliminates the need to track and differentiate an arbitrary mean function. This allows the physical constraints of the Feynman-Kac equation to be seamlessly and entirely absorbed by the kernel and its derivatives.

2.1.1 Distribution of $\mathcal{L}_x f(x)$

Because $\mathcal{L}_x$ is a linear operator, applying it to a Gaussian Process results in another Gaussian Process. The aim of this section is to prove this statement, particularly for the operator defined in equation 1.23.

Let $x = (t, \mathbf{y})$ denote the state variable, and let $f \sim \mathcal{GP}(0, k(x, x'))$ be a zero-mean Gaussian Process.

Proposition 1. *Let $f, g \in \mathcal{C}^2(\mathcal{X})$ and define the operator $\mathcal{L}_x f(x) = \frac{\partial f(x)}{\partial t} - r f(x) + \nabla_{\mathbf{y}} f(x)^\top \mu^\mathbb{Q} + \frac{1}{2} \text{Tr}(\Sigma \Sigma^\top \nabla_{\mathbf{y}}^2 f(x))$ as in 1.23. This operator is linear:*

$$\forall a, b \in \mathbb{R} : \quad \mathcal{L}_x(a f(x) + b g(x)) = a \mathcal{L}_x f(x) + b \mathcal{L}_x g(x). \quad (2.5)$$

Proof. Applying the operator to the linear combination yields:

$$\begin{aligned} \mathcal{L}_x(a f + b g) &= \frac{\partial(a f + b g)}{\partial t} - r(a f + b g) + \nabla_{\mathbf{y}}(a f + b g)^\top \mu^\mathbb{Q} \\ &\quad + \frac{1}{2} \text{Tr}(\Sigma \Sigma^\top \nabla_{\mathbf{y}}^2(a f + b g)). \end{aligned}$$

As the partial derivative, gradient, Hessian, and Trace operators are linear, we can distribute and regroup the terms:

$$\begin{aligned} \mathcal{L}_x(a f + b g) &= a \left[\frac{\partial f}{\partial t} - r f + \nabla_{\mathbf{y}} f^\top \mu^\mathbb{Q} + \frac{1}{2} \text{Tr}(\Sigma \Sigma^\top \nabla_{\mathbf{y}}^2 f) \right] \\ &\quad + b \left[\frac{\partial g}{\partial t} - r g + \nabla_{\mathbf{y}} g^\top \mu^\mathbb{Q} + \frac{1}{2} \text{Tr}(\Sigma \Sigma^\top \nabla_{\mathbf{y}}^2 g) \right] \\ &= a \mathcal{L}_x f + b \mathcal{L}_x g. \end{aligned}$$

□

¹This sentence seems obscure at first but will be clearer in details, in the light of section 2.1.2.

Proposition 2. Let $f \in \mathcal{C}^2(\mathcal{X})$ and $\mathcal{X}$ be compact. Then:

$$\mathbb{E}[\mathcal{L}_x f(x)] = \mathcal{L}_x \mathbb{E}[f(x)] \quad (2.6)$$

Proof. Writing the expectation explicitly as an integral over the probability space Ω with respect to the risk-neutral measure $\mathbb{Q}$, we have:

$$\mathbb{E}[\mathcal{L}_x f(x)] = \int_{\Omega} \left[\frac{\partial f(x, \omega)}{\partial t} - r f(x, \omega) + \nabla_{\mathbf{y}} f(x, \omega)^{\top} \mu^{\mathbb{Q}} + \frac{1}{2} \text{Tr} \left(\Sigma \Sigma^{\top} \nabla_{\mathbf{y}}^2 f(x, \omega) \right) \right] d\mathbb{Q}(\omega)$$

By the linearity of the integral and $\mathcal{L}_x$ from Proposition 1:

$$\begin{aligned} \mathbb{E}[\mathcal{L}_x f(x)] &= \int_{\Omega} \frac{\partial f(x, \omega)}{\partial t} d\mathbb{Q}(\omega) - r \int_{\Omega} f(x, \omega) d\mathbb{Q}(\omega) \\ &\quad + \left(\int_{\Omega} \nabla_{\mathbf{y}} f(x, \omega) d\mathbb{Q}(\omega) \right)^{\top} \mu^{\mathbb{Q}} + \frac{1}{2} \text{Tr} \left(\Sigma \Sigma^{\top} \int_{\Omega} \nabla_{\mathbf{y}}^2 f(x, \omega) d\mathbb{Q}(\omega) \right) \end{aligned}$$

We can now swap the order of spatial/temporal differentiation and integration over Ω using Leibniz rule:

$$\begin{aligned} \int_{\Omega} \frac{\partial f(x, \omega)}{\partial t} d\mathbb{Q}(\omega) &= \frac{\partial}{\partial t} \int_{\Omega} f(x, \omega) d\mathbb{Q}(\omega) = \frac{\partial}{\partial t} \mathbb{E}[f(x)], \\ \int_{\Omega} \nabla_{\mathbf{y}} f(x, \omega) d\mathbb{Q}(\omega) &= \nabla_{\mathbf{y}} \int_{\Omega} f(x, \omega) d\mathbb{Q}(\omega) = \nabla_{\mathbf{y}} \mathbb{E}[f(x)], \\ \int_{\Omega} \nabla_{\mathbf{y}}^2 f(x, \omega) d\mathbb{Q}(\omega) &= \nabla_{\mathbf{y}}^2 \int_{\Omega} f(x, \omega) d\mathbb{Q}(\omega) = \nabla_{\mathbf{y}}^2 \mathbb{E}[f(x)]. \end{aligned}$$

These last equations yield:

$$\begin{aligned} \mathbb{E}[\mathcal{L}_x f(x)] &= \frac{\partial}{\partial t} \mathbb{E}[f(x)] - r \mathbb{E}[f(x)] + \nabla_{\mathbf{y}} \mathbb{E}[f(x)]^{\top} \mu^{\mathbb{Q}} + \frac{1}{2} \text{Tr} \left(\Sigma \Sigma^{\top} \nabla_{\mathbf{y}}^2 \mathbb{E}[f(x)] \right) \\ &= \mathcal{L}_x \mathbb{E}[f(x)] \end{aligned}$$

□

Proposition 3. Let $k(x, x') = \mathbb{C}(f(x), f(x'))$

$$\mathbb{C}(f(x), \mathcal{L}_{x'} f(x')) = \mathcal{L}_{x'} k(x, x'), \quad (2.7)$$

$$\mathbb{C}(\mathcal{L}_x f(x), f(x')) = \mathcal{L}_x k(x, x'), \quad (2.8)$$

$$\mathbb{C}(\mathcal{L}_x f(x), \mathcal{L}_{x'} f(x')) = \mathcal{L}_x \mathcal{L}_{x'} k(x, x'). \quad (2.9)$$

Proof. We prove Equation (2.7) using propositions 2 and 1:

$$\begin{aligned}
\mathbb{C}(f(x), \mathcal{L}_{x'} f(x')) &= \mathbb{E} \left[\left(f(x) - \mathbb{E}[f(x)] \right) \left(\mathcal{L}_{x'} f(x') - \mathbb{E}[\mathcal{L}_{x'} f(x')] \right) \right] \\
&= \mathbb{E} \left[\left(f(x) - \mathbb{E}[f(x)] \right) \left(\mathcal{L}_{x'} f(x') - \mathcal{L}_{x'} \mathbb{E}[f(x')] \right) \right] \\
&= \mathbb{E} \left[\left(f(x) - \mathbb{E}[f(x)] \right) \mathcal{L}_{x'} \left(f(x') - \mathbb{E}[f(x')] \right) \right].
\end{aligned}$$

Because the operator $\mathcal{L}_{x'}$ acts strictly on the coordinates of x' (treating x as a constant), we can pull the scalar term $(f(x) - \mathbb{E}[f(x)])$ inside the linear operator:

$$\mathbb{C}(f(x), \mathcal{L}_{x'} f(x')) = \mathbb{E} \left[\mathcal{L}_{x'} \left((f(x) - \mathbb{E}[f(x)]) (f(x') - \mathbb{E}[f(x')]) \right) \right].$$

Applying Proposition 2 once more to extract the operator outside the expectation yields:

$$\begin{aligned}
\mathbb{C}(f(x), \mathcal{L}_{x'} f(x')) &= \mathcal{L}_{x'} \mathbb{E} \left[\left(f(x) - \mathbb{E}[f(x)] \right) \left(f(x') - \mathbb{E}[f(x')] \right) \right] \\
&= \mathcal{L}_{x'} \mathbb{C}(f(x), f(x')) \\
&= \mathcal{L}_{x'} k(x, x').
\end{aligned}$$

The proof for Equation (2.8) is symmetric, extracting $\mathcal{L}_x$ in an identical manner.

Finally, to prove Equation (2.9), we apply the same algebraic sequence to both operators:

$$\begin{aligned}
\mathbb{C}(\mathcal{L}_x f(x), \mathcal{L}_{x'} f(x')) &= \mathbb{E} \left[\left(\mathcal{L}_x f(x) - \mathbb{E}[\mathcal{L}_x f(x)] \right) \left(\mathcal{L}_{x'} f(x') - \mathbb{E}[\mathcal{L}_{x'} f(x')] \right) \right] \\
&= \mathbb{E} \left[\mathcal{L}_x \left(f(x) - \mathbb{E}[f(x)] \right) \cdot \mathcal{L}_{x'} \left(f(x') - \mathbb{E}[f(x')] \right) \right].
\end{aligned}$$

Because x and x' reside in independent coordinate spaces, their respective linear operators commute. We group them together and apply Proposition 2 sequentially to pull both operators outside the expectation:

$$\begin{aligned}
\mathbb{C}(\mathcal{L}_x f(x), \mathcal{L}_{x'} f(x')) &= \mathbb{E} \left[\mathcal{L}_x \mathcal{L}_{x'} \left((f(x) - \mathbb{E}[f(x)]) (f(x') - \mathbb{E}[f(x')]) \right) \right] \\
&= \mathcal{L}_x \mathcal{L}_{x'} \mathbb{E} \left[\left(f(x) - \mathbb{E}[f(x)] \right) \left(f(x') - \mathbb{E}[f(x')] \right) \right] \\
&= \mathcal{L}_x \mathcal{L}_{x'} \mathbb{C}(f(x), f(x')) \\
&= \mathcal{L}_x \mathcal{L}_{x'} k(x, x').
\end{aligned}$$

□

Proposition 4. Assume the covariance kernel $k(x, x') \in \mathcal{C}^2(\mathcal{X}^2)$. Then, the process $\mathcal{L}_x f(x)$ remains a Gaussian Process, and for any finite set of points $\mathbf{X} \subseteq \mathcal{X}$, its evaluations follow a multivariate normal distribution:

$$\mathcal{L}_x f(\mathbf{X}) \sim \mathcal{N}(\mathbf{0}, \mathcal{L}_x \mathcal{L}_{x'} k(\mathbf{X}, \mathbf{X})). \quad (2.10)$$

The full proof is given in Hainaut and Vrms 2024 by computing explicitly the probability density function of $\mathcal{L}_x f(\mathbf{X})$ using the low-rank formulation of the CGPR. As we did not covered this formulation in the thesis, we just show an intuitive explanation for the above lemma. The explanation is based on the linearity of the operator. As a linear combination of normally distributed variable is also normal, we infer that $\mathcal{L}_x f(x)$ is normal.

To show it, we apply the linear differential operator $\mathcal{L}_x$ to $f(x)$:

$$\mathcal{L}_x f(x) = \frac{\partial f(x)}{\partial t} - r f(x) + \sum_{i=1}^d \mu_i^{\mathbb{Q}} \frac{\partial f(x)}{\partial y_i} + \frac{1}{2} \sum_{i=1}^d \sum_{j=1}^d (\Sigma \Sigma^\top)_{ij} \frac{\partial^2 f(x)}{\partial y_i \partial y_j} \quad (2.11)$$

By the definition, we write the partial derivatives of $f(x)$ as limits of difference quotients. For the first-order derivatives:

$$\frac{\partial f(x)}{\partial t} = \lim_{h \rightarrow 0} \frac{f(t+h, \mathbf{y}) - f(t, \mathbf{y})}{h} \quad (2.12)$$

$$\frac{\partial f(x)}{\partial y_i} = \lim_{h \rightarrow 0} \frac{f(t, \mathbf{y} + h \mathbf{e}_i) - f(t, \mathbf{y})}{h} \quad (2.13)$$

For the second-order cross derivatives:

$$\frac{\partial^2 f(x)}{\partial y_i \partial y_j} = \lim_{h, k \rightarrow 0} \frac{f(t, \mathbf{y} + h \mathbf{e}_i + k \mathbf{e}_j) - f(t, \mathbf{y} + h \mathbf{e}_i) - f(t, \mathbf{y} + k \mathbf{e}_j) + f(t, \mathbf{y})}{hk} \quad (2.14)$$

where $\mathbf{e}_i$ and $\mathbf{e}_j$ are standard basis vectors in $\mathbb{R}^d$. Because equations 2.12, 2.13 and 2.14, we see that the values of $\frac{\partial f(\mathbf{X})}{\partial t}$, $\frac{\partial f(\mathbf{X})}{\partial y_i}$ and $\frac{\partial^2 f(\mathbf{X})}{\partial y_i \partial y_j}$ are linear combinations of $f(\mathbf{X})$ and are therefore distributed as multivariate gaussian distributions. It follows that, as $\mathcal{L}_x f(\mathbf{X})$ is also a linear combination of $\frac{\partial f(\mathbf{X})}{\partial t}$, $\frac{\partial f(\mathbf{X})}{\partial y_i}$, $\frac{\partial^2 f(\mathbf{X})}{\partial y_i \partial y_j}$ and $f(\mathbf{X})$, it is also distributed as multivariate gaussian distribution.

Finally, we infer, from propositions 2 and 3 that:

$$\mathbb{C}(\mathcal{L}_x f(x), \mathcal{L}_{x'} f(x')) = \mathcal{L}_x \mathcal{L}_{x'} k(x, x') \quad (2.15)$$

and

$$\mathbb{E}[\mathcal{L}_x f(x)] = \mathcal{L}_x \mathbb{E}[f(x)] = \mathbf{0} \quad (2.16)$$

as $\mathbb{E}[f(x)] = \mathbf{0}$

2.1.2 Prediction Distribution and Estimate

To formulate the numerical pricing algorithm, we evaluate the Gaussian Process at finite sets of collocation points. We partition our spatial domain into a set of m interior points $\mathring{\mathbf{X}} \subset \mathring{\mathcal{X}}$, d boundary points $\partial\mathbf{X} \subset \partial\mathcal{X}$, and a prediction point x .

Evaluating the process at these finite locations gives us a joint distribution for the predicted prices, the boundary values, and the interior operator, thanks to equation 2.2 and proposition 4:

$$\begin{bmatrix} f(x) \\ f(\partial\mathbf{X}) \\ \mathcal{L}_x f(\mathring{\mathbf{X}}) \end{bmatrix} \sim \mathcal{N} \left(\mathbf{0}, \begin{bmatrix} k(x, x) & k(x, \partial\mathbf{X}) & \mathcal{L}_{x'} k(x, \mathring{\mathbf{X}}) \\ k(\partial\mathbf{X}, x) & k(\partial\mathbf{X}, \partial\mathbf{X}) & \mathcal{L}_{x'} k(\partial\mathbf{X}, \mathring{\mathbf{X}}) \\ \mathcal{L}_x k(\mathring{\mathbf{X}}, x) & \mathcal{L}_x k(\mathring{\mathbf{X}}, \partial\mathbf{X}) & \mathcal{L}_x \mathcal{L}_{x'} k(\mathring{\mathbf{X}}, \mathring{\mathbf{X}}) \end{bmatrix} \right) \quad (2.17)$$

From equation 2.17, we can deduce the distribution of the predictions $f(x)$ using the conditional distribution property of the multivariate normal distribution in a similarly to what has been done in Holt and Nguyen 2023:

Proposition 5. *Let us define the Gram matrix $C(\mathring{\mathbf{X}}, \partial\mathbf{X})$ and the vector $\beta(x)$:*

$$C(\mathring{\mathbf{X}}, \partial\mathbf{X}) = \begin{bmatrix} k(\partial\mathbf{X}, \partial\mathbf{X}) & \mathcal{L}_{x'} k(\partial\mathbf{X}, \mathring{\mathbf{X}}) \\ \mathcal{L}_x k(\mathring{\mathbf{X}}, \partial\mathbf{X}) & \mathcal{L}_x \mathcal{L}_{x'} k(\mathring{\mathbf{X}}, \mathring{\mathbf{X}}) \end{bmatrix} \quad (2.18)$$

$$\beta(x, \partial\mathbf{X}) = \begin{bmatrix} k(\partial\mathbf{X}, x) \\ \mathcal{L}_x k(\mathring{\mathbf{X}}, x) \end{bmatrix} \quad (2.19)$$

Then, the conditionnal distribution of $f(x)$ is given by:

$$f(x) \mid \begin{bmatrix} h(\partial\mathbf{X}) \\ z(\mathring{\mathbf{X}}) \end{bmatrix} \sim \mathcal{N}(\mu^*, \Sigma^*) \quad (2.20)$$

where:

$$\mu^* = \beta(x, \partial\mathbf{X})^\top C(\mathring{\mathbf{X}}, \partial\mathbf{X})^{-1} \begin{bmatrix} h(\partial\mathbf{X}) \\ z(\mathring{\mathbf{X}}) \end{bmatrix} \quad (2.21)$$

and

$$\Sigma^* = k(x, x) - \beta(x, \partial\mathbf{X})^\top C(\mathring{\mathbf{X}}, \partial\mathbf{X})^{-1} \beta(x, \partial\mathbf{X}) \quad (2.22)$$

Proof. To lighten the developments, we use the following notation:

$$\mathbf{Y} = \begin{bmatrix} h(\partial\mathbf{X}) \\ z(\mathring{\mathbf{X}}) \end{bmatrix}$$

We also denote $C = C(\dot{\mathbf{X}}, \partial \mathbf{X})$ and $\boldsymbol{\beta} = \boldsymbol{\beta}(x, \partial \mathbf{X})$.

As a consequence of Equation 2.17, the joint probability density function of the prediction and the observations is:

$$p(f(x), \mathbf{Y}) = \frac{1}{(2\pi)^{(1+N)/2} |\Sigma|^{1/2}} \exp \left(-\frac{1}{2} \begin{bmatrix} f(x) \\ \mathbf{Y} \end{bmatrix}^\top \Gamma^{-1} \begin{bmatrix} f(x) \\ \mathbf{Y} \end{bmatrix} \right) \quad (2.23)$$

where the joint covariance matrix is given by $\Gamma = \begin{bmatrix} k(x, x) & \boldsymbol{\beta}^\top \\ \boldsymbol{\beta} & C \end{bmatrix}$.

To isolate the conditional distribution, we use the following identity:

$$\begin{bmatrix} 1 & -\boldsymbol{\beta}^\top C^{-1} \\ \mathbf{0} & \mathbf{I}_N \end{bmatrix} \begin{bmatrix} k(x, x) & \boldsymbol{\beta}^\top \\ \boldsymbol{\beta} & C \end{bmatrix} \begin{bmatrix} 1 & \mathbf{0} \\ -C^{-1}\boldsymbol{\beta} & \mathbf{I}_N \end{bmatrix} = \begin{bmatrix} \Sigma^* & \mathbf{0} \\ \mathbf{0} & C \end{bmatrix} \quad (2.24)$$

where $\Sigma^* = k(x, x) - \boldsymbol{\beta}^\top C^{-1} \boldsymbol{\beta}$. Taking the determinant of both sides yields:

$$\begin{aligned} \begin{vmatrix} 1 & -\boldsymbol{\beta}^\top C^{-1} \\ \mathbf{0} & \mathbf{I}_N \end{vmatrix} \begin{vmatrix} k(x, x) & \boldsymbol{\beta}^\top \\ \boldsymbol{\beta} & C \end{vmatrix} \begin{vmatrix} 1 & \mathbf{0} \\ -C^{-1}\boldsymbol{\beta} & \mathbf{I}_N \end{vmatrix} &= \begin{vmatrix} \Sigma^* & \mathbf{0} \\ \mathbf{0} & C \end{vmatrix} \\ \Leftrightarrow |\Sigma^*| |C| &= |\Gamma| \end{aligned} \quad (2.25)$$

By inverting both sides of equation 2.24, we obtain:

$$\begin{aligned} \left(\begin{bmatrix} 1 & -\boldsymbol{\beta}^\top C^{-1} \\ \mathbf{0} & \mathbf{I}_N \end{bmatrix} \Gamma \begin{bmatrix} 1 & \mathbf{0} \\ -C^{-1}\boldsymbol{\beta} & \mathbf{I}_N \end{bmatrix} \right)^{-1} &= \begin{bmatrix} (\Sigma^*)^{-1} & \mathbf{0} \\ \mathbf{0} & C^{-1} \end{bmatrix} \\ \begin{bmatrix} 1 & \mathbf{0} \\ -C^{-1}\boldsymbol{\beta} & \mathbf{I}_N \end{bmatrix}^{-1} \Gamma^{-1} \begin{bmatrix} 1 & -\boldsymbol{\beta}^\top C^{-1} \\ \mathbf{0} & \mathbf{I}_N \end{bmatrix}^{-1} &= \begin{bmatrix} (\Sigma^*)^{-1} & \mathbf{0} \\ \mathbf{0} & C^{-1} \end{bmatrix} \\ \Leftrightarrow \begin{bmatrix} 1 & \mathbf{0} \\ -C^{-1}\boldsymbol{\beta} & \mathbf{I}_N \end{bmatrix} \begin{bmatrix} (\Sigma^*)^{-1} & \mathbf{0} \\ \mathbf{0} & C^{-1} \end{bmatrix} \begin{bmatrix} 1 & -\boldsymbol{\beta}^\top C^{-1} \\ \mathbf{0} & \mathbf{I}_N \end{bmatrix} &= \Gamma^{-1} \end{aligned} \quad (2.26)$$

We substitute this inverse directly into the quadratic form of the exponential within the joint density:

dazdzadza

$$\begin{aligned} \begin{bmatrix} f(x) \\ \mathbf{Y} \end{bmatrix}^\top \Gamma^{-1} \begin{bmatrix} f(x) \\ \mathbf{Y} \end{bmatrix} &= \left(\begin{bmatrix} 1 & -\boldsymbol{\beta}^\top C^{-1} \\ \mathbf{0} & \mathbf{I}_N \end{bmatrix} \begin{bmatrix} f(x) \\ \mathbf{Y} \end{bmatrix} \right)^\top \begin{bmatrix} (\Sigma^*)^{-1} & \mathbf{0} \\ \mathbf{0} & C^{-1} \end{bmatrix} \\ &\quad \cdot \begin{bmatrix} 1 & -\boldsymbol{\beta}^\top C^{-1} \\ \mathbf{0} & \mathbf{I}_N \end{bmatrix} \begin{bmatrix} f(x) \\ \mathbf{Y} \end{bmatrix} \\ &= \begin{bmatrix} f(x) - \boldsymbol{\beta}^\top C^{-1} \mathbf{Y} \\ \mathbf{Y} \end{bmatrix}^\top \begin{bmatrix} (\Sigma^*)^{-1} & \mathbf{0} \\ \mathbf{0} & C^{-1} \end{bmatrix} \begin{bmatrix} f(x) - \boldsymbol{\beta}^\top C^{-1} \mathbf{Y} \\ \mathbf{Y} \end{bmatrix}. \end{aligned}$$

Letting $\mu^* = \beta^\top C^{-1} \mathbf{Y}$, the multiplication with the block-diagonal matrix yields two independant terms:

$$\begin{bmatrix} f(x) \\ \mathbf{Y} \end{bmatrix}^\top \Gamma^{-1} \begin{bmatrix} f(x) \\ \mathbf{Y} \end{bmatrix} = (f(x) - \mu^*)^\top (\Sigma^*)^{-1} (f(x) - \mu^*) + \mathbf{Y}^\top C^{-1} \mathbf{Y}. \quad (2.27)$$

Returning to the joint probability density function and substituting the split determinant and split quadratic form, we factor $p(f(x), \mathbf{Y})$ as follows:

$$\begin{aligned} p(f(x), \mathbf{Y}) &= \left[\frac{1}{\sqrt{2\pi\Sigma^*}} \exp \left(-\frac{1}{2\Sigma^*} (f(x) - \mu^*)^2 \right) \right] \\ &\times \left[\frac{1}{(2\pi)^{N/2} |C|^{1/2}} \exp \left(-\frac{1}{2} \mathbf{Y}^\top C^{-1} \mathbf{Y} \right) \right] \end{aligned}$$

By definition, the second bracketed term is exactly the multivariate marginal density of the observations, $p(\mathbf{Y})$. Because Bayes' rule dictates that $p(f(x), \mathbf{Y}) = p(f(x) | \mathbf{Y})p(\mathbf{Y})$, the conditional probability density $p(f(x) | \mathbf{Y})$ must strictly be the first term. $\square$

As $\mu^* = \mathbb{E} \left[f(x) \mid \begin{bmatrix} h(\partial \mathbf{X}) \\ z(\dot{\mathbf{X}}) \end{bmatrix} \right]$, it follows that μ^* is our best estimate to predict $f(x)$:

$$\hat{f}(x) = \mu^* = \beta(x, \partial \mathbf{X})^\top C(\dot{\mathbf{X}}, \partial \mathbf{X})^{-1} \begin{bmatrix} h(\partial \mathbf{X}) \\ z(\dot{\mathbf{X}}) \end{bmatrix} \quad (2.28)$$

Lastly, we can show that the Gram matrix is symmetric as it will be used in later reasoning:

Lemma 2. *Given $k(x, x') \in \mathcal{C}^2(\mathcal{X}^2)$ is a symmetric kernel: $k(x, x') = k(x', x)$, the Gram matrix $C(\dot{\mathbf{X}}, \partial \mathbf{X})$ defined in Equation 2.18 is also symmetric, meaning $C = C^\top$.*

Proof. Let us decompose the block matrix C into its four constituent sub-matrices:

$$C = \begin{bmatrix} k(\partial \mathbf{X}, \partial \mathbf{X}) & \mathcal{L}_{x'} k(\partial \mathbf{X}, \dot{\mathbf{X}}) \\ \mathcal{L}_x k(\dot{\mathbf{X}}, \partial \mathbf{X}) & \mathcal{L}_x \mathcal{L}_{x'} k(\dot{\mathbf{X}}, \dot{\mathbf{X}}) \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$$

For the full block matrix C to be symmetric, it must satisfy three conditions: the diagonal blocks must be symmetric ($C_{11} = C_{11}^\top$ and $C_{22} = C_{22}^\top$), and the off-diagonal blocks must be transposes of each other ($C_{12} = C_{21}^\top$).

Starting with C_{11} , let $x_i, x_j \in \partial \mathbf{X}$. As the kernel is symmetric, $k(x_i, x_j) = k(x_j, x_i)$. Therefore, C_{11} is also trivially symmetric: $C_{11} = C_{11}^\top$.

Let $x_i \in \partial\mathbf{X}$ and $x'_j \in \mathring{\mathbf{X}}$. The entry (i, j) of block C_{12} is given by applying the operator to the second spatial argument:

$$(C_{12})_{i,j} = \mathcal{L}_{x'} k(x_i, x'_j)$$

The corresponding entry (j, i) of block C_{21} evaluates the operator on the first spatial argument:

$$(C_{21})_{j,i} = \mathcal{L}_x k(x'_j, x_i)$$

Because the base kernel is symmetric, $k(x'_j, x_i) = k(x_i, x'_j)$. Furthermore, the operator $\mathcal{L}_x$ acting on the first argument x'_j is mathematically identical to the operator $\mathcal{L}_{x'}$ acting on the second argument x'_j of the reversed kernel. Consequently, $(C_{21})_{j,i} = (C_{12})_{i,j}$, which proves that $C_{12} = C_{21}^\top$.

The exact same reasoning can be applied for the block C_{22} . Let $x_i, x_j \in \mathring{\mathbf{X}}$. The (i, j) entry is:

$$(C_{22})_{i,j} = \mathcal{L}_x \mathcal{L}_{x'} k(x_i, x_j)$$

Because x and x' are independent, their differential operators commute. Combined with the symmetry of the base kernel $k(x_i, x_j) = k(x_j, x_i)$, we can swap the order of differentiation and the arguments:

$$\mathcal{L}_x \mathcal{L}_{x'} k(x_i, x_j) = \mathcal{L}_{x'} \mathcal{L}_x k(x_j, x_i) = (C_{22})_{j,i}$$

Thus, $C_{22} = C_{22}^\top$.

Since the diagonal blocks are symmetric and the off-diagonal blocks are transposes of one another, the complete Gram matrix $C(\mathring{\mathbf{X}}, \partial\mathbf{X})$ is strictly symmetric. $\square$

2.2 Fixed-Point Problem

In the preceding sections, we successfully constructed a regression with constrained Gaussian processes able to solve local boundary value problems of the form $\mathcal{L}_x f(x) = z(x)$, as seen in Equation 2.1. However, the full Feynman-Kac equation governing jump-diffusion models is an integro-differential equation where the right-hand side also depends on the solution f :

$$\mathcal{L}_x f(x) = -\Delta(f)(x) \tag{2.29}$$

To overcome this, we employ an operator splitting technique inspired from Hainaut 2026. We treat the complex jump integral not as an active operator, but as a lagged, state-dependent source term: $z(x) := -\Delta(f)(x)$. This allows us to decouple the problem. If we guess a source term $z^{(u-1)}$, we can easily solve the local PDE

$\mathcal{L}_x f^{(u)} = z^{(u-1)}$ using our established CGPR framework. We then update our guess by evaluating the jumps on the newly predicted surface: $z^{(u)} = -\Delta(f^{(u)})$.

This predictor-corrector approach naturally forms a recursive loop. The aim of this section is to formalize this iterative loop as a rigorous fixed-point problem, prove its convergence to the true option price, and demonstrate its exact practical implementation within the CGPR framework.

2.2.1 Banach Spaces Framework

To guarantee that our iterative procedure will eventually converge to the correct, unique solution, we rely on the foundational framework formalized by Banach. A summary of this framework is shown in Agarwal, Jleli, and Samet 2018.

Definition 1 (Contraction Mapping). *Let $(X, \|\cdot\|)$ be a Banach space with metric $d(x, y)$ deriving from the norm $\|\cdot\|$. A map $\mathcal{T} : X \rightarrow X$ is called a contraction mapping on X if there exists a nonnegative constant $q < 1$ such that*

$$d(\mathcal{T}(x), \mathcal{T}(y)) \leq q d(x, y) \quad (2.30)$$

for all $x, y \in X$.

Theorem 1 (Banach Fixed-Point Theorem). *Let $(X, \|\cdot\|)$ be a Banach space endowed with a contraction mapping $\mathcal{T} : X \rightarrow X$. Then $\mathcal{T}$ admits a unique fixed point $x^* \in X$, meaning $\mathcal{T}(x^*) = x^*$. Furthermore, this fixed point can be found iteratively: starting from an arbitrary initial element $x_0 \in X$, the sequence defined by $x_u = \mathcal{T}(x_{u-1})$ satisfies:*

$$\lim_{u \rightarrow \infty} x_u = x^*. \quad (2.31)$$

We also introduce the Feynman-Kac theorem (updated for our specific boundary conditions) as it will be used in later proofs. The original Feynman-Kac theorem can be found in the section 5.7 from Karatzas and Shreve 1998.

Theorem 2 (Feynman-Kac Formula for Boundary Value Problems). *Let $\mathcal{X} = [0, T) \times \mathring{\mathcal{Y}}$ be a domain where $\mathring{\mathcal{Y}} \subset \mathbb{R}^p$ is the domain of the process $\mathbf{Y}_t$ such that $d\mathbf{Y}_t = \boldsymbol{\mu}^{\mathbb{Q}}(t, \mathbf{y})dt + \Sigma(t, \mathbf{y})d\mathbf{W}_t$, and let $\partial\mathcal{X}$ denote its boundaries. Define the stopping time τ as the first exit time of $\mathbf{Y}_t$ from $\mathring{\mathcal{Y}}$, and let $\tau^* = \min(\tau, T)$ be the effective stopping time.*

If $u(t, \mathbf{y})$ is a solution to the following PDE:

$$\begin{aligned} & \frac{\partial u}{\partial t}(t, \mathbf{y}) + \boldsymbol{\mu}^{\mathbb{Q}}(t, \mathbf{y})^\top \nabla_{\mathbf{y}} u(t, \mathbf{y}) \\ & + \frac{1}{2} \sum_{i=1}^p \sum_{j=1}^p \left(\Sigma(t, \mathbf{y}) \Sigma(t, \mathbf{y})^\top \right)_{ij} \frac{\partial^2 u}{\partial y_i \partial y_j}(t, \mathbf{y}) \\ & - r(t, \mathbf{y})u(t, \mathbf{y}) + \phi(t, \mathbf{y}) = 0, \quad \forall (t, \mathbf{y}) \in \mathring{\mathcal{X}}, \end{aligned} \quad (2.32)$$

subject to the boundary condition $u(t, \mathbf{y}) = h(t, \mathbf{y})$ on $\partial\mathcal{X}$, then $u(t, \mathbf{y})$ admits the stochastic representation:

$$\begin{aligned} u(t, \mathbf{y}) = & \mathbb{E}_{t, \mathbf{y}}^{\mathbb{Q}} \left[e^{-\int_t^{\tau^*} r(u, \mathbf{Y}_u) du} h(\tau^*, \mathbf{Y}_{\tau^*}) \right] \\ & + \mathbb{E}_{t, \mathbf{y}}^{\mathbb{Q}} \left[\int_t^{\tau^*} e^{-\int_t^s r(u, \mathbf{Y}_u) du} \phi(s, \mathbf{Y}_s) ds \right] \end{aligned} \quad (2.33)$$

2.2.2 Application to the Feynman-Kac Operator

To guarantee strict contraction properties for our fixed-point problem, we must equip our functional space with an appropriate norm. Using the standard supremum norm $\|\cdot\|_\infty$ does not allow us to find a valid contraction. That's why we use the Bielecki renorming technique (as summarized in Kirk and Sims 2001) by defining the following weighted supremum norm for a given weight $\beta > 0$:

$$\|f\|_\beta := \sup_{(t, x) \in \mathcal{X}} e^{-\beta(T-t)} |f(t, x)|. \quad (2.34)$$

We define $\mathcal{B}_\beta = (C_b(\mathcal{X}), \|\cdot\|_\beta)$ as the Banach space of bounded continuous functions on the domain $\mathcal{X}$ endowed with this norm. We impose the following standard assumptions on our financial model:

H1 The drift $\boldsymbol{\mu}_y(\cdot)$, covariance $\Sigma_y(\cdot)$, jump size mapping $M(\cdot)$, and interest rate $r(\cdot)$ are bounded and globally Lipschitz on $\mathcal{X}$.

H2 For any continuous source term $\phi \in \mathcal{B}_\beta$, the purely local boundary value

problem

$$\begin{cases} \mathcal{L}_x f(x) = \phi(x) & x \in \mathring{\mathcal{X}}, \\ f(x) = h(x) & x \in \partial\mathcal{X}, \end{cases} \quad (2.35)$$

admits a unique solution $f = \mathcal{G}(\phi) \in \mathcal{B}_\beta$, where $\mathcal{G}$ acts as the local inverse operator.

We first demonstrate that the non-local jump operator Δ is naturally Lipschitz continuous under the Bielecki norm.

Lemma 3. *Let the total system jump intensity be bounded by $\Lambda = \sum_{v=1}^{p-1} \lambda_v$. For any two option price surfaces $f_1, f_2 \in \mathcal{B}_\beta$, the integro-differential operator Δ satisfies:*

$$\|\Delta(f_1) - \Delta(f_2)\|_\beta \leq 2\Lambda\|f_1 - f_2\|_\beta. \quad (2.36)$$

Proof. For any state $x = (t, \mathbf{y}_t) \in \mathcal{X}$, the jump operator evaluates the expected spatial displacement. By multiplying the absolute difference by the Bielecki weight $e^{-\beta(T-t)}$ and applying the triangle inequality, we obtain:

$$\begin{aligned} e^{-\beta(T-t)}|\Delta(f_1) - \Delta(f_2)| &= |\Delta(f_1)(x) - \Delta(f_2)(x)| \\ &\leq e^{-\beta(T-t)}\Lambda|f_1(t, \mathbf{y}_t) - f_2(t, \mathbf{y}_t)| \\ &\quad + \sum_{v=1}^{p-1} \lambda_v \mathbb{E}_{\xi^{(v)}} \left[e^{-\beta(T-t)} \left| f_1(t, \mathbf{y}_t + M(t, \mathbf{y}_t)\mathbf{e}_v \xi^{(v)}) \right. \right. \\ &\quad \left. \left. - f_2(t, \mathbf{y}_t + M(t, \mathbf{y}_t)\mathbf{e}_v \xi^{(v)}) \right| \right] \end{aligned}$$

Since the exponential weight is deterministic, it passes through the expectation. We can therefore bound both the pre-jump and post-jump terms strictly by their supremum over the entire domain:

$$\begin{aligned} \|\Delta(f_1) - \Delta(f_2)\|_\beta &\leq \Lambda\|f_1 - f_2\|_\beta + \Lambda\|f_1 - f_2\|_\beta \\ &= 2\Lambda\|f_1 - f_2\|_\beta. \end{aligned}$$

□

We now formalize our predictor-corrector intuition. By substituting the local inverse $f = \mathcal{G}(z)$ into the definition of the lagged source term $z = -\Delta(f)$, we obtain an isolated mapping operator strictly in terms of z :

$$\mathcal{T}(z) := -\Delta(\mathcal{G}(z)). \quad (2.37)$$

Using the properties of the Bielecki norm, we can prove that this composite mapping forms a strict contraction.

Theorem 3. *There exists a sufficiently large weight $\beta > 0$ such that the unrelaxed mapping $\mathcal{T}(z) = -\Delta(\mathcal{G}(z))$ is a strict contraction on the Banach space $\mathcal{B}_\beta$.*

Proof. Consider two arbitrary source terms $z_1, z_2 \in \mathcal{B}_\beta$. Let $u = \mathcal{G}(z_1) - \mathcal{G}(z_2)$. By the linearity of the PDE, u is the exact solution to $\mathcal{L}_x u = z_1 - z_2$ subject to a strictly homogeneous boundary condition:

$$\begin{aligned}\mathcal{L}_x u &= \mathcal{L}_x(\mathcal{G}(z_1) - \mathcal{G}(z_2)) \\ &= \mathcal{L}_x \mathcal{G}(z_1) - \mathcal{L}_x \mathcal{G}(z_2)\end{aligned}$$

which implies, on the boundary of the domain $\partial\mathcal{X}$:

$$\begin{aligned}\mathcal{G}(z_1) - \mathcal{G}(z_2) &= h(x) - h(x) \\ u &= 0\end{aligned}$$

Let $\tau^* = \min(\tau, T)$ be the effective exit time from the domain, as defined in 2. By rearranging the PDE to the standard form $\mathcal{L}_x u - (z_1 - z_2) = 0$, our source term becomes $\phi = -(z_1 - z_2)$. Applying theorem 2), the boundary payoff term vanishes, yielding:

$$u(t, y) = \mathbb{E}_{t,y}^{\mathbb{Q}} \left[\int_t^{\tau^*} e^{-\int_t^s r(u, Y_u) du} (z_2(s, Y_s) - z_1(s, Y_s)) ds \right]$$

Let $r_{\min} = \inf_{(t,y) \in \mathcal{X}} r(t, y)$. We can bound the stochastic discount factor strictly by $e^{-r_{\min}(s-t)}$. Taking the exponential weight, the absolute values on both sides and using the identity $e^{-\beta(T-t)} = e^{-\beta(s-t)} e^{-\beta(T-s)}$ yields:

$$\begin{aligned}e^{-\beta(T-t)} |u(t, y)| &= e^{-\beta(T-t)} \left| \mathbb{E}_{t,y}^{\mathbb{Q}} \left[\int_t^{\tau^*} e^{-\int_t^s r(v, Y_v) dv} (z_2(s, Y_s) - z_1(s, Y_s)) ds \right] \right| \\ &\leq \mathbb{E}_{t,y}^{\mathbb{Q}} \left[\int_t^{\tau^*} e^{-(r_{\min} + \beta)(s-t)} \underbrace{(e^{-\beta(T-s)} |z_1(s, Y_s) - z_2(s, Y_s)|)}_{\leq \|z_1 - z_2\|_\beta} ds \right]\end{aligned}$$

We extract the strict β -norm bound from the expectation. Because the remaining integrand is strictly positive and deterministic, extending the upper limit of integration from the random stopping time τ^* to the fixed maturity T strictly preserves the inequality:

$$\begin{aligned}e^{-\beta(T-t)} |u(t, x)| &\leq \|z_1 - z_2\|_\beta \int_t^T e^{-(r_{\min} + \beta)(s-t)} ds \\ &= \|z_1 - z_2\|_\beta \frac{1 - e^{-(r_{\min} + \beta)(T-t)}}{r_{\min} + \beta} \\ &\leq \frac{1}{r_{\min} + \beta} \|z_1 - z_2\|_\beta\end{aligned}$$

Taking the supremum over (t, x) proves that the inverse operator $\mathcal{G}$ is bounded by $C_{\mathcal{G}}^{(\beta)} = \frac{1}{r_{\min} + \beta}$. Combining this with Lemma 3, the Lipschitz constant of the composite mapping $\mathcal{T}$ is $C_{\mathcal{T}}^{(\beta)} = \frac{2\Lambda}{r_{\min} + \beta}$. By choosing a weight $\beta > 2\Lambda - r_{\min}$, we guarantee that $C_{\mathcal{T}}^{(\beta)} < 1$, proving that $\mathcal{T}$ is a strict contraction on $\mathcal{B}_{\beta}$. $\square$

While Theorem 3 proves mathematical convergence, applying $\mathcal{T}$ directly in a numerical algorithm can lead to oscillations/instability in the convergence. To ensure robust convergence, we introduce a relaxation parameter $\theta \in (0, 1]$ and define the damped operator:

$$\mathcal{T}_{\theta}(z) := (1 - \theta)z + \theta\mathcal{T}(z). \quad (2.38)$$

Because $\mathcal{T}$ is a strict contraction under the Bielecki norm, the relaxed operator also inherits this property:

Lemma 4. *The relaxed operator $\mathcal{T}_{\theta}(z)$ is a valid contraction on $\mathcal{B}_{\beta}$ for any $\theta \in (0, 1]$. Namely, there exists a constant $C_{\mathcal{T}_{\theta}}^{(\beta)} \in [0, 1)$ such that:*

$$\|\mathcal{T}_{\theta}(z_1) - \mathcal{T}_{\theta}(z_2)\|_{\beta} \leq C_{\mathcal{T}_{\theta}}^{(\beta)} \|z_1 - z_2\|_{\beta} \quad \forall z_1, z_2 \in \mathcal{B}_{\beta}. \quad (2.39)$$

Proof. For any $z_1, z_2 \in \mathcal{B}_{\beta}$, applying the triangle inequality yields:

$$\begin{aligned} \|\mathcal{T}_{\theta}(z_1) - \mathcal{T}_{\theta}(z_2)\|_{\beta} &\leq (1 - \theta)\|z_1 - z_2\|_{\beta} + \theta\|\mathcal{T}(z_1) - \mathcal{T}(z_2)\|_{\beta} \\ &\leq \left(1 - \theta(1 - C_{\mathcal{T}}^{(\beta)})\right) \|z_1 - z_2\|_{\beta}. \end{aligned}$$

Because $C_{\mathcal{T}}^{(\beta)} < 1$ by Theorem 3 and $\theta > 0$, the scaling factor strictly satisfies $1 - \theta(1 - C_{\mathcal{T}}^{(\beta)}) < 1$. $\square$

Proposition 6. *Under Assumptions **H1** and **H2**, the sequence of option prices $(f^{(u)})_{u \geq 0}$ generated by solving:*

$$\begin{cases} \mathcal{L}_x f^{(u)}(x) = z^{(u-1)}(x) & x \in \mathring{\mathcal{X}}, \\ f^{(u)}(x) = h(x) & x \in \partial\mathcal{X}, \end{cases} \quad (2.40)$$

where the source term is iteratively updated via:

$$z^{(u)}(x) = z^{(u-1)}(x) + \theta \left[-\Delta(f^{(u)})(x) - z^{(u-1)}(x) \right], \quad (2.41)$$

converges uniformly on $\mathcal{X}$ to the unique, exact solution of the Feynman-Kac PIDE.

Proof. Because $\mathcal{B}_\beta$ is complete and $\mathcal{T}_\theta$ is a contraction, the Banach fixed-point theorem guarantees the sequence $z^{(u)} = \mathcal{T}_\theta(z^{(u-1)})$ converges to a unique fixed point z^* where $z^* = \mathcal{T}_\theta(z^*) = -\Delta(\mathcal{G}(z^*))$. Consequently, by defining $f^{(u)} := \mathcal{G}(z^{(u-1)})$, the boundedness of $\mathcal{G}$ ensures that $\|f^{(u)} - f^*\|_\beta \leq C_{\mathcal{G}}^{(\beta)} \|z^{(u-1)} - z^*\|_\beta \rightarrow 0$. $\square$

2.2.3 Implementation for the Algorithm

While Proposition 6 guarantees theoretical convergence in the continuous Banach space $\mathcal{B}_\beta$, our numerical algorithm tracks convergence across a discrete vector space $X = \mathbb{R}^m$ representing the interior collocation points.

Crucially, because the time domain $[0, T]$ is strictly finite, the exponential weight $e^{-\beta(T-t)}$ is bounded between $e^{-\beta T}$ and 1. Consequently, the Bielecki norm and the supremum norm are equivalent:

$$\|f\|_\beta \leq \|f\|_\infty \leq e^{\beta T} \|f\|_\beta \quad (2.42)$$

As a result, we do not need to compute the artificial weight β during algorithm execution. We evaluate our stopping criterion using the unweighted discrete infinity norm ($\|z^{(u)} - z^{(u-1)}\|_\infty < \epsilon$). Because the two norms are equivalent, a converging sequence of $f^{(u)}$ converging with respect to $\|\cdot\|_\infty$ will also converge with $\|\cdot\|_\beta$, as it can be seen in equation 2.42.

In practice, the algorithm works in the following way: at any iteration $u \geq 1$, we construct a combined observation vector from the boundary payoffs and our current jump source guess:

$$\mathbf{Y}^{(u-1)} = \begin{bmatrix} h(\partial\mathbf{X}) \\ z^{(u-1)}(\mathring{\mathbf{X}}) \end{bmatrix}. \quad (2.43)$$

The first step is to evaluate $f^{(u)} = \mathcal{G}(z^{(u-1)})$. As established in our Gaussian Process formulation, solving this PDE maps directly to computing the predictive posterior mean:

$$\hat{f}^{(u)}(x) = \beta(x, \partial \mathbf{X})^\top C(\mathring{\mathbf{X}}, \partial \mathbf{X})^{-1} \mathbf{Y}^{(u-1)}. \quad (2.44)$$

Because the matrix C^{-1} depends strictly on the domain discretization and the local operator $\mathcal{L}_x$, it is computed and inverted exactly once prior to the iterative loop, rendering the evaluation of $\mathcal{G}(z)$ computationally trivial.

The second theoretical step requires evaluating the non-local operator $\Delta(f^{(u)})(x)$. Let $\delta_v(\xi^{(v)})$ denote the spatial displacement caused by a jump of size $\xi^{(v)}$ in the v -th underlying asset. We do so by injecting Equation 2.44 into $\Delta(f^{(u)})(x)$.

Because the Gram matrix inverse and observation vector $\mathbf{Y}^{(u-1)}$ are entirely deterministic with respect to the random jump variable $\xi^{(v)}$, the expectation bypasses them and acts exclusively on the spatially-dependent cross-covariance kernel vector β :

$$\mathbb{E}_{\xi^{(v)}}^{\mathbb{Q}} \left[f^{(u)}(x + \delta_v(\xi^{(v)})) \right] = \left(\mathbb{E}_{\xi^{(v)}}^{\mathbb{Q}} \left[\beta(x + \delta_v(\xi^{(v)}), \partial \mathbf{X})^\top \right] \right) C(\mathring{\mathbf{X}}, \partial \mathbf{X})^{-1} \mathbf{Y}^{(u-1)} \quad (2.45)$$

By computing this expected kernel vector analytically, the jump compensator for iteration u is explicitly constructed via:

$$\Delta(f^{(u)})(x) = \sum_{v=1}^{p-1} \lambda_v \left(\mathbb{E}_{\xi^{(v)}}^{\mathbb{Q}} \left[\beta(x + \delta_v(\xi^{(v)}), \partial \mathbf{X})^\top \right] - \beta(x, \partial \mathbf{X})^\top \right) C(\mathring{\mathbf{X}}, \partial \mathbf{X})^{-1} \mathbf{Y}^{(u-1)} \quad (2.46)$$

Finally, substituting this compensator back into our relaxation formula yields the updated vector $z^{(u)}(\mathring{\mathbf{X}})$ required for the next algorithm iteration:

$$z^{(u)} = (1 - \theta) z^{(u-1)} - \theta \Delta(\hat{f}^{(u)}) \quad (2.47)$$

2.3 Matrix Conditionning and Numerical Stability

While we found the existence and a way to estimate the pricing function f , the numerical implementation relies fundamentally on the inversion of the dense Gram matrix $C(\mathring{\mathbf{X}}, \partial \mathbf{X})$. A challenge that arises is the numerical instability of this inversion, driven by the fact it can be ill-conditioned. This problem is well-known for Gaussian processes and is pointed out in Rasmussen and Williams 2005.

2.3.1 The Condition Number and Loss of Precision

In this section, we define the condition number and summarize what it induces on the stability of a system and its inverse (Chan and Foulser 1988).

The stability of a linear system:

$$C\boldsymbol{\alpha} = \mathbf{y}$$

represents how much the solution $\boldsymbol{\alpha}$ would change if we added an error term $\mathbf{e}$ in the input $\mathbf{y}$. It is quantified by:

$$C\boldsymbol{\alpha}' = \mathbf{y} + \mathbf{e} \quad (2.48)$$

$$\boldsymbol{\alpha}' = C^{-1}\mathbf{y} + C^{-1}\mathbf{e} \quad (2.49)$$

$$\boldsymbol{\alpha}' = \boldsymbol{\alpha} + C^{-1}\mathbf{e} \quad (2.50)$$

Let $\Delta\boldsymbol{\alpha} = \boldsymbol{\alpha}' - \boldsymbol{\alpha} = C^{-1}\mathbf{e}$ represent the absolute error in the solution. By taking a vector norm on both sides, we obtain an upper bound on this error:

$$\|\Delta\boldsymbol{\alpha}\| \leq \|C^{-1}\| \|\mathbf{e}\| \quad (2.51)$$

To understand the relative error, we must relate this to the magnitude of the true solution $\boldsymbol{\alpha}$. From the original system $C\boldsymbol{\alpha} = \mathbf{y}$, we can bound the input vector $\mathbf{y}$:

$$\|\mathbf{y}\| \leq \|C\| \|\boldsymbol{\alpha}\| \implies \frac{1}{\|\boldsymbol{\alpha}\|} \leq \frac{\|C\|}{\|\mathbf{y}\|} \quad (2.52)$$

Multiplying these two inequalities together yields the upper bound for the relative error:

$$\frac{\|\Delta\boldsymbol{\alpha}\|}{\|\boldsymbol{\alpha}\|} \leq (\|C\| \|C^{-1}\|) \frac{\|\mathbf{e}\|}{\|\mathbf{y}\|} \quad (2.53)$$

The scalar multiplier $\|C\| \|C^{-1}\|$ dictates how much the relative error in the input ($\mathbf{e}$) is amplified in the output. This is the condition number of the matrix, denoted as $\kappa(C)$.

When using the induced L_2 -norm, the norm of a matrix is equal to its largest singular value. As C is a symmetric matrix (see 2), its singular values correspond to the absolute values of its eigenvalues. Therefore:

$$\|C\|_2 = \max_i |\lambda_i| = |\lambda_{\max}| \quad (2.54)$$

Because the eigenvalues of an inverse matrix C^{-1} are exactly the reciprocals of the eigenvalues of C ($\frac{1}{\lambda_i}$), the L_2 -norm of the inverse is governed by the smallest

eigenvalue of C :

$$\|C^{-1}\|_2 = \max_i \left| \frac{1}{\lambda_i} \right| = \frac{1}{|\lambda_{\min}|} \quad (2.55)$$

Combining these definitions yields the condition number under the L_2 -norm as the ratio of its largest to smallest eigenvalues:

$$\kappa(C) = \|C\| \|C^{-1}\| = \frac{|\lambda_{\max}|}{|\lambda_{\min}|} \quad (2.56)$$

It shows the condition number dictates how sensitive the solution α is to small perturbations, such as floating-point round-off errors. A rule of the thumb is that if $\kappa(C) \approx 10^k$, the solver will lose approximately k digits of precision (Cheney and Kincaid 2008).

2.3.2 Sources of Ill-Conditioning

A high condition number of the Gram matrix happens when its rows or columns are close to be linear combinations of each others. Indeed, if two or more columns of C are perfectly linearly dependent, the matrix fails to span $\mathbb{R}^N$ and loses full rank (i.e., $\text{rank}(C) < N$). By the Rank-Nullity Theorem (Strang 2016), the matrix must possess a non-trivial null space $\mathcal{N}(C)$. Consequently, there exists a non-zero vector $\mathbf{v}$ such that:

$$C\mathbf{v} = \mathbf{0} = 0 \cdot \mathbf{v}. \quad (2.57)$$

This implies that $\lambda = 0$ is an exact eigenvalue of C . Because of the condition number definition, we obtain $\kappa(C) = \infty$.

There are two main factors that drive C to have a low rank: the choice of hyperparameters $a, \boldsymbol{\eta}$ and the domain discretization sparsity.

Hyperparameter Saturation

A primary source of ill-conditioning arises when the lengthscale hyperparameters are severely misspecified, causing the kernel function $k(x, x')$ to lose its spatial discriminatory power across the domain $\mathcal{X}$. Using the standard Gaussian kernel (Equation 3.23) as an example, we can show this kind of phenomenon.

Assuming a uniform lengthscale $\eta_v = \eta$ across all dimensions on a bounded spatial domain, we first analyze the saturation regime where $\eta \rightarrow \infty$. The exponential decay is entirely suppressed, yielding:

$$\lim_{\eta \rightarrow \infty} k(x, x') = \lim_{\eta \rightarrow \infty} a^2 \exp \left(- \sum_{v=1}^p \frac{(x_v - x'_v)^2}{2\eta^2} \right) = a^2 \quad (2.58)$$

Because the spatial derivatives of the kernel ($\nabla_x k$ and $\nabla_x^2 k$) scale proportionally with strictly negative powers of η , they vanish asymptotically. Consequently, applying the continuous pricing operator $\mathcal{L}_x$ leaves only the zero-order discounting term $-r$:

$$\lim_{\eta \rightarrow \infty} \mathcal{L}_x k(x, x') = -r \quad (2.59)$$

$$\lim_{\eta \rightarrow \infty} \mathcal{L}_{x'} \mathcal{L}_x k(x, x') = r^2 \quad (2.60)$$

In this asymptotic limit, the spatial dependence remains if the risk-free rate is a function of x and x' . Most of the time, however, r still have low variability across the domain. The Gram matrix almost collapses into a rank-1 matrix, driving the condition number $\kappa(C)$ to infinity.

Grid Density

Another primary source of ill-conditioning is the density of the discretization grid. Because the spatial domain $\mathcal{X}$ is strictly bounded, increasing N necessarily forces the distance between nearest neighbors, $\Delta x = \min_{i \neq j} \|x_i - x_j\|$, to shrink toward zero. Two different columns in the matrix C can have very similar values if the kernel is computed on similar points, which increases the condition number of C .

To empirically demonstrate this phenomenon within our specific problem, we evaluate the condition number of the Gram matrix for various lengthscale and grid densities. The results are summarized in Figure 2.1 and Table 2.1.

η	Grid Density		
	Sparse	Medium	Dense
0.50	$2.08e + 10$	$3.03e + 11$	$3.13e + 12$
0.75	$2.32e + 11$	$3.38e + 12$	$3.50e + 13$
1.00	$1.26e + 12$	$1.84e + 13$	$1.91e + 14$
1.25	$4.54e + 12$	$6.63e + 13$	$6.93e + 14$
1.50	$1.26e + 13$	$1.85e + 14$	$1.97e + 15$

Table 2.1: Condition number $\kappa(C)$ of the Gram matrix.

2.3.3 Impact on Inversion and Predictions

In practice, evaluating equations 2.21 and 2.22 requires computing the inverse of the Gram matrix, C^{-1} . Because computers use finite-precision floating-point arithmetic, the matrix we actually invert is never the exact mathematical matrix

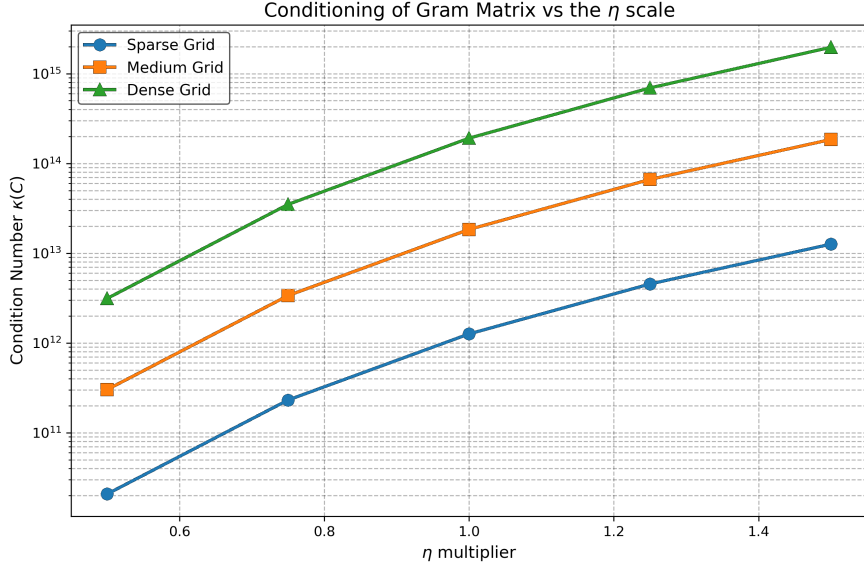


Figure 2.1: Evolution of the Gram matrix condition number $\kappa(C)$ as a function of the lengthscale multiplier η across different grid densities.

C . Instead, it is a perturbed matrix $\tilde{C} = C + E$, where E represents the round-off errors.

We firstly investigate the absolute error in the computed inverse, $\Delta C^{-1} = \tilde{C}^{-1} - C^{-1}$, we begin with the following identity:

$$\tilde{C}^{-1} - C^{-1} = \tilde{C}^{-1}(C - \tilde{C})C^{-1} \quad (2.61)$$

$$\Delta C^{-1} = -\tilde{C}^{-1}EC^{-1} \quad (2.62)$$

Taking the matrix norm on both sides yields:

$$\|\Delta C^{-1}\| \leq \|\tilde{C}^{-1}\| \|E\| \|C^{-1}\| \quad (2.63)$$

To express this bound entirely in terms of the exact matrix C , we must bound the norm of the perturbed inverse, $\|\tilde{C}^{-1}\|$. We factor the perturbed matrix as $\tilde{C} = C(I + C^{-1}E)$. Using the inverse property $(AB)^{-1} = B^{-1}A^{-1}$, we obtain:

$$\|\tilde{C}^{-1}\| = \|(I + C^{-1}E)^{-1}C^{-1}\| \leq \|(I + C^{-1}E)^{-1}\| \|C^{-1}\| \quad (2.64)$$

Assuming the perturbation is small enough that $\|C^{-1}E\| \leq \|C^{-1}\| \|E\| < 1$, we

apply the strict bound derived from the Neumann series:

$$\begin{aligned}
\|\tilde{C}^{-1}\| &\leq \|(I + C^{-1}E)^{-1}\| \|C^{-1}\| \\
&= \left\| \sum_{k=0}^{\infty} (-C^{-1}E)^k \right\| \|C^{-1}\| \\
&\leq \sum_{k=0}^{\infty} \|(-C^{-1}E)^k\| \|C^{-1}\| \\
&\leq \sum_{k=0}^{\infty} \|C^{-1}E\|^k \|C^{-1}\| \\
&= \frac{\|C^{-1}\|}{1 - \|C^{-1}\| \|E\|}
\end{aligned} \tag{2.65}$$

Substituting this strict bound back into Equation 2.63 provides the exact upper bound for the absolute error:

$$\|\Delta C^{-1}\| \leq \frac{\|C^{-1}\|^2 \|E\|}{1 - \|C^{-1}\| \|E\|} \tag{2.66}$$

To find the relative error of the computed inverse, we divide by the norm of the true inverse, $\|C^{-1}\|$. We then multiply the term $\|C^{-1}\| \|E\|$ by $\frac{\|C\|}{\|C\|}$ to introduce the condition number, $\kappa(C) = \|C\| \|C^{-1}\|$:

$$\|C^{-1}\| \|E\| = (\|C\| \|C^{-1}\|) \frac{\|E\|}{\|C\|} = \kappa(C) \frac{\|E\|}{\|C\|} \tag{2.67}$$

Substituting this exact equivalence into both the numerator and denominator yields the final relative error bound:

$$\frac{\|\Delta C^{-1}\|}{\|C^{-1}\|} \leq \frac{\kappa(C) \frac{\|E\|}{\|C\|}}{1 - \kappa(C) \frac{\|E\|}{\|C\|}} \tag{2.68}$$

This exact bound demonstrates that the relative error in the computed inverse is strictly driven by the condition number. Because $\kappa(C)$ appears in both the numerator and the denominator, a massive condition number not only amplifies the machine error $\|E\|$ linearly but accelerates the degradation of the bound as the denominator shrinks away from 1.

When the error becomes too large, the error in the predictive mean obtained in equation 2.21 increases. Moreover, the predictive covariance (Equation 2.22) $\beta^\top \tilde{C}^{-1} \beta$ is no longer guaranteed to be positive definite. This directly causes the subtractive term to exceed the prior variance $k(x, x)$, yielding mathematically invalid, negative predictive variances ($\tilde{\Sigma}^* < 0$).

2.3.4 Tikhonov Regularization

To rescue the Gram matrix from singularity, we must introduce a mechanism that strictly bounds the smallest eigenvalue away from zero. Algebraically, this is achieved through Tikhonov regularization (Rasmussen and Williams 2005), which consists of adding a small, strictly positive constant σ^{*2} to the main diagonal of the Gram matrix C .

Let $\mathbf{I}$ be the $(m + d) \times (m + d)$ identity matrix. We define the regularized Gram matrix as:

$$C_{\text{reg}} = C(\mathring{\mathbf{X}}, \partial\mathbf{X}) + \sigma^{*2}\mathbf{I} \quad (2.69)$$

Adding a scaled identity matrix to C preserves its eigenvectors entirely but shifts its eigenspectrum uniformly upward. If $\mathbf{v}_i$ is an eigenvector of C with eigenvalue λ_i , then:

$$C_{\text{reg}}\mathbf{v}_i = (C + \sigma^{*2}\mathbf{I})\mathbf{v}_i = C\mathbf{v}_i + \sigma^{*2}\mathbf{v}_i = (\lambda_i + \sigma^{*2})\mathbf{v}_i \quad (2.70)$$

Because the eigenvalues shift uniformly from λ_i to $\lambda_i + \sigma^{*2}$, this operation provides a strict, positive mathematical floor for the smallest eigenvalue. Even when spatial degeneracy forces $\lambda_{\min} \rightarrow 0$, the regularized minimum eigenvalue is strictly bounded: $\tilde{\lambda}_{\min} \geq \sigma^{*2}$.

Consequently, the condition number of the regularized linear system is artificially bounded from above:

$$\kappa(C_{\text{reg}}) = \frac{\lambda_{\max} + \sigma^{*2}}{\lambda_{\min} + \sigma^{*2}} \quad (2.71)$$

By choosing a suitably small σ^{*2} , Tikhonov regularization stabilizes the numerical inversion without vastly distorting the solution of the system.

While this shift improves stability, it possesses a statistical interpretation: Tikhonov regularization is equivalent to assuming that our target observations h_i and z_i are corrupted by Gaussian noise.

In the standard Gaussian Process framework, we concatenate our boundary conditions and interior source terms into a single deterministic target vector $\mathbf{y}$:

$$\mathbf{y} = \begin{bmatrix} h(\partial\mathbf{X}) \\ z(\mathring{\mathbf{X}}) \end{bmatrix} \quad (2.72)$$

Adding the identically distributed (i.i.d.) zero-mean Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^{*2}\mathbf{I})$, our observed target vector becomes a random variable:

$$\tilde{\mathbf{y}} = \mathbf{y} + \boldsymbol{\epsilon} \quad (2.73)$$

To compute the new Gram matrix for the predictive distribution, we evaluate the covariance of these noisy observations:

$$\text{Cov}(\tilde{\mathbf{y}}, \tilde{\mathbf{y}}) = \text{Cov}(\mathbf{y} + \boldsymbol{\epsilon}, \mathbf{y} + \boldsymbol{\epsilon}) \quad (2.74)$$

Expanding this expression yields:

$$\text{Cov}(\tilde{\mathbf{y}}, \tilde{\mathbf{y}}) = \text{Cov}(\mathbf{y}, \mathbf{y}) + \text{Cov}(\boldsymbol{\epsilon}, \boldsymbol{\epsilon}) + \text{Cov}(\mathbf{y}, \boldsymbol{\epsilon}) + \text{Cov}(\boldsymbol{\epsilon}, \mathbf{y}) \quad (2.75)$$

Because the artificial observation noise $\boldsymbol{\epsilon}$ is strictly independent of the underlying spatial process $\mathbf{y}$, the cross-covariance terms vanish ($\text{Cov}(\mathbf{y}, \boldsymbol{\epsilon}) = \mathbf{0}$). Substituting the known covariances back into the equation gives:

$$\text{Cov}(\tilde{\mathbf{y}}, \tilde{\mathbf{y}}) = C(\hat{\mathbf{X}}, \partial\mathbf{X}) + \sigma^{*2}\mathbf{I} = C_{\text{reg}} \quad (2.76)$$

Thus, explicitly injecting Gaussian noise of variance σ^{*2} into the targets h and z mathematically reconstructs the exact Tikhonov-regularized Gram matrix.

We test the actual impact of the use of Tikhonov regularization on the condition number $\kappa(C)$. The results are summarized in graph 2.2.

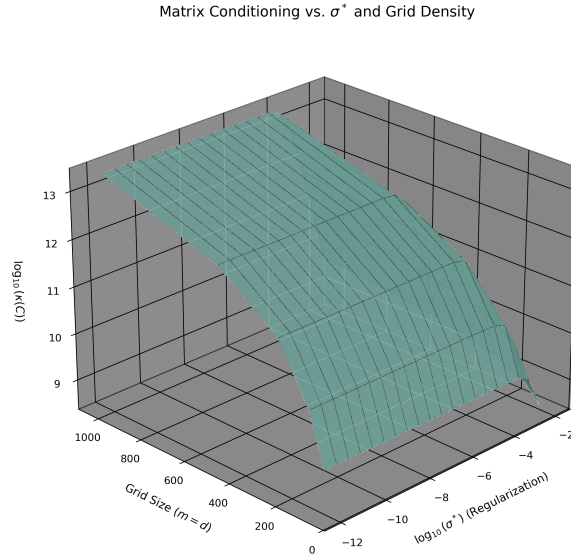


Figure 2.2: Evolution of the Gram matrix condition number $\kappa(C)$ as a function of σ^* across different grid densities.

The graph visually corroborates our previous explanations by showing the use of the regularization of the Gram matrix on all the tested grid lengths.

2.3.5 Cholesky Decomposition

As stated in equation 2.44, our estimate, using the Tikhonov regularized Gram matrix, at step u is:

$$\widehat{f^{(u)}}(x) = \beta(x, \partial \mathbf{X})^\top C_{\text{reg}}(\dot{\mathbf{X}}, \partial \mathbf{X})^{-1} \begin{bmatrix} h(\partial \mathbf{X}) \\ z^{(u-1)}(\dot{\mathbf{X}}) \end{bmatrix} \quad (2.77)$$

Let us define the target vector as $\mathbf{y}^{(u-1)} = [h^\top, z^{(u-1)\top}]^\top$, and the global weight vector as:

$$\boldsymbol{\alpha}^{(u)} = C_{\text{reg}}^{-1} \mathbf{y}^{(u-1)} \quad (2.78)$$

such that the evaluation at any spatial point simplifies to the inner product $\widehat{f^{(u)}}(x) = \beta(x)^\top \boldsymbol{\alpha}^{(u)}$. The major issue with this expression is that it requires explicitly computing the inverse matrix C_{reg}^{-1} , which destroys numerical stability as explained previously.

Instead of computing the inverse, we reformulate the problem as solving the linear system:

$$C_{\text{reg}} \boldsymbol{\alpha}^{(u)} = \mathbf{y}^{(u-1)} \quad (2.79)$$

Because Tikhonov regularization guarantees that C_{reg} is symmetric and strictly positive-definite, we can apply the Cholesky decomposition. This allows us to uniquely factor the Gram matrix into the product of a lower triangular matrix L and its conjugate transpose:

$$C_{\text{reg}} = LL^\top \quad (2.80)$$

Substituting this factorization into our linear system yields:

$$(LL^\top) \boldsymbol{\alpha}^{(u)} = \mathbf{y}^{(u-1)} \quad (2.81)$$

This allows us to find the exact weight vector $\boldsymbol{\alpha}^{(u)}$ without ever computing an inverse by solving two more stable, consecutive triangular systems:

1. Forward Substitution: Solve $L\mathbf{w} = \mathbf{y}^{(u-1)}$ for the intermediate vector $\mathbf{w}$.
2. Backward Substitution: Solve $L^\top \boldsymbol{\alpha}^{(u)} = \mathbf{w}$ for the final vector $\boldsymbol{\alpha}^{(u)}$.
3. Compute the final estimate: $\widehat{f^{(u)}}(x) = \beta(x)^\top \boldsymbol{\alpha}^{(u)}$.

This approach holds two numerical advantages. First, computing the matrix L requires $\mathcal{O}(N^3)$ while solving a triangular system requires only $\mathcal{O}(N^2)$ operations, which does not asymptotically slow the computations.

Finally, it allows to solve system that have reduced condition numbers. When we explicitly compute an inverse or solve a system directly with C_{reg} , the relative error bounds of the solution are strictly governed by the condition number $\kappa(C_{\text{reg}})$. By using the Cholesky factorization $C_{\text{reg}} = LL^\top$, we replace one highly ill-conditioned direct solve with two separate triangular solves. To evaluate the numerical stability of these intermediate systems, we must derive the condition number of the lower triangular matrix L .

To determine the condition number of the lower Cholesky factor L , we first expand its L_2 -norm:

$$\|L\|_2 = \sqrt{\lambda_{\max}(L^\top L)} \quad (2.82)$$

Next, we express this norm in terms of eigenvalue of $LL^\top$:

Lemma 5. *For any strictly positive-definite regularized Gram matrix defined by its Cholesky decomposition $C_{\text{reg}} = LL^\top$, the L_2 norm of L is given by:*

$$\|L\|_2 = \sqrt{\lambda_{\max}(C_{\text{reg}})} \quad (2.83)$$

Proof. Let $\lambda \neq 0$ be a non-zero eigenvalue of $L^\top L$. By definition, there exists a corresponding non-zero eigenvector $v \neq 0$ such that:

$$L^\top L v = \lambda v \quad (2.84)$$

Therefore:

$$\begin{aligned} L(L^\top L v) &= L(\lambda v) \\ (LL^\top)(Lv) &= \lambda(Lv) \end{aligned} \quad (2.85)$$

For Lv to be a valid eigenvector of $LL^\top$, we must rigorously ensure that Lv is not the zero vector. We prove this by contradiction. Assume $Lv = 0$. Substituting this assumption back into the left side of our original eigenvalue equation gives:

$$L^\top(Lv) = L^\top(0) = 0 \implies \lambda v = 0 \quad (2.86)$$

Because we explicitly established that $v \neq 0$, this result forces $\lambda = 0$. This strictly contradicts our initial assumption that $\lambda \neq 0$. Therefore, the assumption is false, guaranteeing that $Lv \neq 0$. The reasoning is valid for $\lambda_{\max}$ in particular, and then:

$$\lambda_{\max}(L^\top L) = \lambda_{\max}(LL^\top) = \lambda_{\max}(C_{\text{reg}}) \quad (2.87)$$

Therefore, it follows that:

$$\|L\|_2 = \sqrt{\lambda_{\max}(LL^\top)} = \sqrt{\lambda_{\max}(C_{\text{reg}})} \quad (2.88)$$

□

Lemma 6. *For any strictly positive-definite regularized Gram matrix defined by its Cholesky decomposition $C_{\text{reg}} = LL^\top$, the L_2 norm of L^{-1} is given by:*

$$\|L^{-1}\|_2 = \frac{1}{\sqrt{\lambda_{\min}(C_{\text{reg}})}} \quad (2.89)$$

Proof. By the definition of the induced L_2 norm, we know that:

$$\|L^{-1}\|_2 = \sqrt{\lambda_{\max}((L^{-1})^\top L^{-1})} \quad (2.90)$$

Using $(L^{-1})^\top = (L^\top)^{-1}$, we can rewrite the matrix product as:

$$(L^{-1})^\top L^{-1} = (L^\top)^{-1} L^{-1} = (LL^\top)^{-1} \quad (2.91)$$

Substituting the Cholesky decomposition $LL^\top = C_{\text{reg}}$ into this relation yields:

$$(L^{-1})^\top L^{-1} = C_{\text{reg}}^{-1} \quad (2.92)$$

Because the eigenvalues of an inverse matrix are exactly the reciprocals of the eigenvalues of the original matrix, the maximum eigenvalue of C_{reg}^{-1} corresponds strictly to the reciprocal of the minimum eigenvalue of C_{reg} :

$$\lambda_{\max}(C_{\text{reg}}^{-1}) = \frac{1}{\lambda_{\min}(C_{\text{reg}})} \quad (2.93)$$

Substituting this result back into our initial norm definition completes the proof:

$$\|L^{-1}\|_2 = \sqrt{\lambda_{\max}(C_{\text{reg}}^{-1})} = \frac{1}{\sqrt{\lambda_{\min}(C_{\text{reg}})}} \quad (2.94)$$

□

Proposition 7. *For any strictly positive-definite regularized Gram matrix defined by its Cholesky decomposition $C_{\text{reg}} = LL^\top$, the condition number of L is given by:*

$$\kappa(L) = \|L\|_2 \|L^{-1}\|_2 = \sqrt{\kappa(C_{\text{reg}})} \quad (2.95)$$

Proof. The condition number of a matrix under the L_2 norm is defined as:

$$\kappa(L) = \|L\|_2 \|L^{-1}\|_2 \quad (2.96)$$

By directly substituting the exact, mathematically proven norms from Lemma 5 and Lemma 6, we obtain:

$$\kappa(L) = \sqrt{\frac{\lambda_{\max}(C_{\text{reg}})}{\lambda_{\min}(C_{\text{reg}})}} = \sqrt{\kappa(C_{\text{reg}})} \quad (2.97)$$

□

Next, we compute the condition number of the backward substitution:

Lemma 7. *For any strictly positive-definite regularized Gram matrix defined by its Cholesky decomposition $C_{\text{reg}} = LL^\top$, the L_2 norm of $L^\top$ is given by:*

$$\|L^\top\|_2 = \sqrt{\lambda_{\max}(C_{\text{reg}})} \quad (2.98)$$

Proof. By the definition of the induced L_2 norm:

$$\begin{aligned} \|L^\top\|_2 &= \sqrt{\lambda_{\max}((L^\top)^\top L^\top)} \\ &= \sqrt{\lambda_{\max}(LL^\top)} \\ &= \sqrt{\lambda_{\max}(C_{\text{reg}})} \end{aligned} \quad (2.99)$$

□

Lemma 8. *For any strictly positive-definite regularized Gram matrix defined by its Cholesky decomposition $C_{\text{reg}} = LL^\top$, the L_2 norm of the inverse factor $(L^\top)^{-1}$ is given by:*

$$\|(L^\top)^{-1}\|_2 = \frac{1}{\sqrt{\lambda_{\min}(C_{\text{reg}})}} \quad (2.100)$$

Proof. Applying the definition of the L_2 norm to the inverse matrix $(L^\top)^{-1}$, we have:

$$\begin{aligned} \|(L^\top)^{-1}\|_2 &= \sqrt{\lambda_{\max}(((L^\top)^{-1})^\top (L^\top)^{-1})} \\ &= \sqrt{\lambda_{\max}(L^{-1}(L^\top)^{-1})} \\ &= \sqrt{\lambda_{\max}((L^\top L)^{-1})} \\ &= \frac{1}{\sqrt{\lambda_{\min}(L^\top L)}} \end{aligned} \quad (2.101)$$

As already shown in the development leading to 2.87, $\lambda(L^\top L) = \lambda(LL^\top)$ and, in particular, $\lambda_{\min}(L^\top L) = \lambda_{\min}(LL^\top)$. Substituting the Cholesky decomposition $LL^\top = C_{\text{reg}}$ into this relation simplifies the expression to:

$$\begin{aligned}\|(L^\top)^{-1}\|_2 &= \frac{1}{\sqrt{\lambda_{\min}(LL^\top)}} \\ &= \frac{1}{\sqrt{\lambda_{\min}(C_{\text{reg}})}}\end{aligned}\tag{2.102}$$

□

Proposition 8. *For any strictly positive-definite regularized Gram matrix defined by its Cholesky decomposition $C_{\text{reg}} = LL^\top$, the condition numbers of the upper and lower triangular factors satisfy:*

$$\kappa(L^\top) = \kappa(L) = \sqrt{\kappa(C_{\text{reg}})}\tag{2.103}$$

Proof. The condition number of the upper triangular Cholesky factor under the L_2 norm is :

$$\kappa(L^\top) = \|L^\top\|_2 \|(L^\top)^{-1}\|_2\tag{2.104}$$

By substituting the explicitly derived norms from Lemma 7 and Lemma 8, we obtain:

$$\kappa(L^\top) = \sqrt{\frac{\lambda_{\max}(C_{\text{reg}})}{\lambda_{\min}(C_{\text{reg}})}} = \sqrt{\kappa(C_{\text{reg}})}\tag{2.105}$$

□

With propositions 7 and 8, we showed that computing $\widehat{f^{(u)}}$ with the Cholesky system reduce the condition number from $\kappa(C_{\text{reg}})$ to $\sqrt{\kappa(C_{\text{reg}})}$ which greatly improves the numerical stability.

2.4 Final Algorithm

Having established the foundational components and mathematical operators in the preceding sections, we now synthesize the complete numerical procedure. The following algorithm outlines the step-by-step computation for approximating the derivative's price.

Algorithm 1 Computation of the approximated price $\widehat{f^{(n)}}(x)$ of a derivative with payoff $h(x)$

1. Sample m points $\mathring{x}_i = (\mathring{t}_i, \mathring{y}_i) \in \mathring{\mathcal{X}}$ and d points $\bar{x}_i \in \partial\mathcal{X}$
2. set $h_i = h(\bar{x}_i) + \epsilon_i$ where $\epsilon_i \sim \mathcal{N}(0, \sigma^*)$.
3. Draw a sample $z_i^{(0)} \sim \mathcal{N}(0, \sigma^*)$ for $i = 1, \dots, m$.
4. Compute the Gram matrix:

$$C(\mathring{X}, \partial X) = \sigma^{*2} I_{m+d} + \begin{pmatrix} k_{\mathcal{L}^2}(\mathring{X}, \mathring{X}) & k_{\mathcal{L}}(\mathring{X}, \partial X) \\ k_{\mathcal{L}}(\mathring{X}, \partial X)^\top & k(\partial X, \partial X) \end{pmatrix}$$

Calculate vectors $\beta(\mathring{x}_i) = \begin{pmatrix} k_{\mathcal{L}}(\mathring{X}, \mathring{x}_i) \\ k(\partial X, \mathring{x}_i) \end{pmatrix}$ for $i = 1, \dots, m$.

Calculate $\mathbb{E}_{\xi^{(v)}} [\beta(\mathring{x}_i + \delta_v)^\top]$ for $v = 1, \dots, p$ and $i = 1, \dots, m$.

5. Loop on $u \in \mathbb{N}$ until convergence of $\hat{f}^{(u)}$ with respect to $\|\cdot\|_\infty$
 - a) $\forall \mathring{x}_i \in \mathring{\mathcal{X}}$, solve the system

$$\alpha(\mathring{x}_i, z^{(u)}) = LL^\top \begin{pmatrix} z^{(u-1)} \\ h \end{pmatrix}$$

and evaluate:

$$\hat{f}^{(u)}(\mathring{x}_i) = \beta(\mathring{x}_i)^\top \alpha(\mathring{x}_i, z^{(u)})$$

- b) Calculate the vector $\Delta^{(u)}$ of $\Delta(\hat{f}^{(u)})(\mathring{x}_i)$ for $i = 1, \dots, m$.
 - c) Update the vector $z^{(u)}$ as follows:

$$z^{(u)} = z^{(u-1)} + \theta \left[-\Delta^{(u)} - z^{(u-1)} \right],$$

6. If convergence after n iterations, return $\widehat{f^{(n)}}(x)$ and $z^{(n-1)}$.
-

2.5 Greeks Computation

A fundamental advantage of Gaussian regression processes is the ability to extract the option's Greeks in closed form. These are the partial derivatives of the option price with respect to its underlying state variable. Unlike traditional Monte Carlo or finite difference PDE solvers that require computationally expensive

and numerically unstable approximation schemes, the CGPR yields an analytical, continuous function for the option price.

Let the state vector be denoted as $\mathbf{x} = (t, y^{(1)}, \dots, y^{(p)})^\top$. The predictive mean of the option price at any arbitrary test point $\mathbf{x}$ is expressed as a linear combination of the precomputed weights $\alpha(h, z^{(n)})$ and the kernel covariance vector $\beta(\mathbf{x})$:

$$\hat{f}(\mathbf{x}) = \beta(\mathbf{x})^T \alpha(h, z^{(n)}) \quad (2.106)$$

where :

$$\alpha(h, z^{(n)}) = C^{-1}(\dot{X}, \partial X) \begin{pmatrix} z^{(n)} \\ h \end{pmatrix}$$

is the static weight vector solved during the model's training phase, and:

$$\beta(\mathbf{x}) = \begin{pmatrix} k_{\mathcal{L}}(\dot{X}, \mathbf{x}) \\ k(\partial X, \mathbf{x}) \end{pmatrix} \quad (2.107)$$

is the covariance elements between the estimation point $\mathbf{x}$ and the inner domain data points $\dot{X}$.

Because the differentiation operator is linear and the weight vector $\alpha(h, z^{(n)})$ is completely independent of $\mathbf{x}$, the spatial derivative operators pass directly onto the basis vector $\beta(\mathbf{x})$. This allows us to formulate exact, analytical propositions for the Greeks.

Lemma 9. *Let $\mathbf{x} = (t, y_1, \dots, y_d)^\top$ be the test point and $\mathbf{x}' = (t', y'_1, \dots, y'_d)^\top \in \dot{X}$ be an inner domain point. Then, by definition of $\beta(\mathbf{x})$ in (2.107), for any state variable $\gamma \in \{t, y_v\}$:*

$$\frac{\partial}{\partial \gamma} \beta(\mathbf{x}) = \begin{pmatrix} \mathcal{L}_{\mathbf{x}'} \left(\frac{\partial k(\mathbf{x}', \mathbf{x})}{\partial \gamma} \right) \\ \frac{\partial k(\partial X, \mathbf{x})}{\partial \gamma} \end{pmatrix} \quad (2.108)$$

where the top block expands explicitly to:

$$\begin{aligned} \mathcal{L}_{\mathbf{x}'} \left(\frac{\partial k(\mathbf{x}', \mathbf{x})}{\partial \gamma} \right) &= \frac{\partial^2 k(\mathbf{x}', \mathbf{x})}{\partial t' \partial \gamma} - r(\mathbf{x}') \frac{\partial k(\mathbf{x}', \mathbf{x})}{\partial \gamma} + \sum_{i=1}^d \mu_i^{\mathbb{Q}}(\mathbf{x}') \frac{\partial^2 k(\mathbf{x}', \mathbf{x})}{\partial y'_i \partial \gamma} \\ &\quad + \frac{1}{2} \sum_{i=1}^d \sum_{j=1}^d (\Sigma(\mathbf{x}') \Sigma^\top(\mathbf{x}'))_{ij}(\mathbf{x}') \frac{\partial^3 k(\mathbf{x}', \mathbf{x})}{\partial y'_i \partial y'_j \partial \gamma} \end{aligned} \quad (2.109)$$

Proof. By definition, the top block of $\beta(\mathbf{x})$ contains the terms $k_{\mathcal{L}}(\mathbf{x}', \mathbf{x}) = \mathcal{L}_{\mathbf{x}'} k(\mathbf{x}', \mathbf{x})$. Substituting the explicit definition of the operator yields:

$$k_{\mathcal{L}}(\mathbf{x}', \mathbf{x}) = \frac{\partial k}{\partial t'} - rk + \sum_{i=1}^d \mu_i^{\mathbb{Q}}(\mathbf{x}') \frac{\partial k}{\partial y'_i} + \frac{1}{2} \sum_{i=1}^d \sum_{j=1}^d (\Sigma \Sigma^\top)_{ij}(\mathbf{x}') \frac{\partial^2 k}{\partial y'_i \partial y'_j} \quad (2.110)$$

We now apply the partial derivative with respect to γ . Because the coefficients $r(\mathbf{x}')$, $\mu_i^{\mathbb{Q}}(\mathbf{x}')$, and $(\Sigma\Sigma^\top)_{ij}(\mathbf{x}')$ are functions strictly of the domain point $\mathbf{x}'$ and hold no dependency on $\mathbf{x}$, they act as constants under $\frac{\partial}{\partial\gamma}$. Furthermore, by the symmetry of mixed partial derivatives, $\frac{\partial}{\partial\gamma}\left(\frac{\partial k}{\partial y'_i}\right) = \frac{\partial^2 k}{\partial y'_i\partial\gamma}$. $\square$

Proposition 9. *The closed-form for the option Theta ($\Theta = \frac{\partial \hat{f}(\mathbf{x})}{\partial t}$) is given by:*

$$\Theta(\mathbf{x}) = \frac{\partial}{\partial t} \beta(\mathbf{x})^\top \alpha(h, z^{(n)}) \quad (2.111)$$

The closed-form for the option Delta ($\Delta^{(v)} = \frac{\partial \hat{f}(\mathbf{x})}{\partial y^{(v)}}$) is given by:

$$\Delta^{(v)}(\mathbf{x}) = \frac{\partial}{\partial y^{(v)}} \beta(\mathbf{x})^\top \alpha(h, z^{(n)}) \quad (2.112)$$

Proof. The estimation of the option price is $\hat{f}(\mathbf{x}) = \beta(\mathbf{x})^\top \alpha(h, z^{(n)})$. Seeing that $\alpha(h, z^{(n)})$ is a constant w.r.t t and $y^{(v)}$, we just use the result of the previous lemma by substituting γ by t or $y^{(v)}$. $\square$

2.6 Hyperparameters Estimation

Prior to testing the method on numerical experiments for derivatives pricing, the model must be properly calibrated. To estimate the hyperparameters, we employ Maximum Likelihood Estimation (MLE), which is the standard approach for Gaussian processes and we incorporate a novelty: domain scaling to enhance the numerical stability of the procedure.

2.6.1 Maximum Likelihood Estimation

Before starting the main algorithm, we must tune the kernel hyperparameters $\theta = [a^2, \eta_t, \eta_1, \dots, \eta_{p-1}]^\top$. To do so we can use the MLE as developed in Rasmussen and Williams 2005.

Let $\partial\mathbf{X}$ denote a subset of d boundary points, and $\mathring{\mathbf{X}}$ denote a subset of m interior points². The target vector for this joint distribution is the concatenated vector $\mathbf{Y} = [\mathbf{z}^\top(\mathring{\mathbf{X}}), \mathbf{h}^\top(\partial\mathbf{X})]^\top \in \mathbb{R}^{m+d}$.

²Note that, for the stability of the optimization, we can reduce the dimension m and d by sampling a limited amount of points in $\mathring{\mathbf{X}}$ and $\partial\mathbf{X}$

For simplicity, we denote the Gram matrix obtained with this discretization as $\mathbf{K}_\theta \in \mathbb{R}^{(m+d) \times (m+d)}$:

$$\mathbf{K}_\theta = \begin{pmatrix} \mathcal{L}_x \mathcal{L}_{x'} k_\theta(\dot{\mathbf{X}}, \dot{\mathbf{X}}) & \mathcal{L}_x k_\theta(\dot{\mathbf{X}}, \partial \mathbf{X}) \\ \mathcal{L}_{x'} k_\theta(\partial \mathbf{X}, \dot{\mathbf{X}}) & k_\theta(\partial \mathbf{X}, \partial \mathbf{X}) \end{pmatrix} \quad (2.113)$$

It follows that the joint likelihood function evaluating the fit of the hyperparameters is:

$$p(\mathbf{Y} \mid \dot{\mathbf{X}}, \partial \mathbf{X}, \boldsymbol{\theta}) = \frac{1}{(2\pi)^{(m+d)/2} |\mathbf{K}_\theta|^{1/2}} \exp \left(-\frac{1}{2} \mathbf{Y}^\top \mathbf{K}_\theta^{-1} \mathbf{Y} \right) \quad (2.114)$$

To find the optimal hyperparameters, we minimize the Negative Log Marginal Likelihood (NLL):

$$\begin{aligned} \text{NLL}(\boldsymbol{\theta}) &= -\log p(\mathbf{Y} \mid \dot{\mathbf{X}}, \partial \mathbf{X}, \boldsymbol{\theta}) = \frac{1}{2} \mathbf{Y}^\top \mathbf{K}_\theta^{-1} \mathbf{Y} + \frac{1}{2} \log |\mathbf{K}_\theta| \\ &\quad + \frac{m+d}{2} \log(2\pi) \end{aligned} \quad (2.115)$$

2.6.2 Likelihood Optimization

To optimize equation (2.115) for the parameters $\boldsymbol{\theta}$, we will use numerical methods like Quasi-Newton methods (e.g., L-BFGS-B). As the NLL is rather simple and the kernels are differentiable, we compute the exact gradient of the NLL with respect to $\boldsymbol{\theta}$ to drastically improve the method's accuracy and speed³.

To compute the partial derivative of the NLL with respect to any generic hyperparameter θ_i , we apply the matrix calculus identities

$$\frac{\partial \mathbf{K}_\theta^{-1}}{\partial \theta_i} = -\mathbf{K}_\theta^{-1} \frac{\partial \mathbf{K}_\theta}{\partial \theta_i} \mathbf{K}_\theta^{-1}$$

and

$$\frac{\partial \log |\mathbf{K}_\theta|}{\partial \theta_i} = \text{Tr} \left(\mathbf{K}_\theta^{-1} \frac{\partial \mathbf{K}_\theta}{\partial \theta_i} \right)$$

obtaining:

$$\begin{aligned} \frac{\partial \text{NLL}}{\partial \theta_i} &= -\frac{1}{2} \mathbf{Y}^\top \mathbf{K}_\theta^{-1} \frac{\partial \mathbf{K}_\theta}{\partial \theta_i} \mathbf{K}_\theta^{-1} \mathbf{Y} \\ &\quad + \frac{1}{2} \text{Tr} \left(\mathbf{K}_\theta^{-1} \frac{\partial \mathbf{K}_\theta}{\partial \theta_i} \right) \end{aligned} \quad (2.116)$$

³While it would also be possible to use standard Newton's method by computing the exact Hessian, it is computationally prohibitive to evaluate and invert at each iteration due to its high-dimensional complexity.

2.6.3 Domain Normalization

A major challenge in Gaussian Process hyperparameter estimation is the anisotropic nature of the domain. Following the nature of the risk processes chosen, $\mathbf{y}_t$ contains variables of vastly different magnitudes. This can slow drastically or even break the convergence of a gradient method because of its implication in the optimization step. Indeed, when the scales of the dimensions are highly different, the gradient algorithm has a tendency to choose a larger optimization step to account for the extent of the bigger dimension scales. That's why we propose to map the original bounded domain $\mathcal{X}$ to a normalized hypercube $\tilde{\mathcal{X}} = [0, 1]^p$. A similar mapping using the means and standard deviations for each dimensions of the discretized grid was proposed in Hainaut and Casas 2024. Here, we propose a different linear normalization that enforces same scales across every dimensions.

Because the normalization is a linear transformation, we can determine the exact structure of the normalized Feynman-Kac operator by directly injecting the normalized state vector $\tilde{\mathbf{x}}$ into the physical operator $\mathcal{L}_{\mathbf{x}}$. Demanding that the mathematical pricing problem remains invariant (such that the option price $f(\mathbf{x}) = \tilde{f}(\tilde{\mathbf{x}})$) allows us to logically deduce the scaled drift, diffusion, and jump evaluation mechanics required to maintain strict equivalence.

We define the following mapping separately for time and space: Let $c_t = t_{\min}$ and $c_i = y_{i,\min}$ be the biases. Let $s_t = t_{\max} - t_{\min}$ and $s_{y_i} = y_{i,\max} - y_{i,\min}$ be the scales for each dimension⁴. The normalized state variables are defined as:

$$\begin{cases} \tilde{t} = \frac{t - c_t}{s_t} \implies t = s_t \tilde{t} + c_t \\ \tilde{y}_i = \frac{y_i - c_{y_i}}{s_{y_i}} \implies y_i = s_{y_i} \tilde{y}_i + c_{y_i} \end{cases} \quad (2.117)$$

In vector notation, defining the scaling matrix $S_{\mathbf{y}} = \text{diag}(s_{y_1}, \dots, s_{y_p})$ and the translation vector $\mathbf{c}_{\mathbf{y}} = (c_{y_1}, \dots, c_{y_{p-1}})^\top$, the mapping is exactly:

$$\mathbf{y} = S_{\mathbf{y}} \tilde{\mathbf{y}} + \mathbf{c}_{\mathbf{y}} \quad (2.118)$$

Demanding that the mathematical pricing function remains strictly invariant, such that $f(t, \mathbf{y}) = \tilde{f}(\tilde{t}, \tilde{\mathbf{y}})$, we logically deduce the scaled drift, diffusion, and jump parameters.

⁴By $t_{\min}$ and $t_{\max}$, we refer to the lower and upper bounds of the temporal domain. By $y_{i,\min}$ and $y_{i,\max}$, we refer to the lower and upper bound of the financial stochastic process y_i .

Proposition 10. *Let $\boldsymbol{\mu}(t, \mathbf{y})$ and $\Sigma(t, \mathbf{y})$ be as defined in 1.1. Under the linear transformations 2.118, preserving the function value $f(t, \mathbf{y}) = \tilde{f}(\tilde{t}, \tilde{\mathbf{y}})$ requires scaling the time derivative by $1/s_t$ in the $\mathcal{L}$ operator, and defining the normalized spatial drift and diffusion matrices as:*

$$\tilde{\boldsymbol{\mu}}(\tilde{t}, \tilde{\mathbf{y}}) = S_{\mathbf{y}}^{-1} \boldsymbol{\mu}(s_t \tilde{t} + c_t, S_{\mathbf{y}} \tilde{\mathbf{y}} + \mathbf{c}_{\mathbf{y}}) \quad (2.119)$$

$$\tilde{\Sigma}(\tilde{t}, \tilde{\mathbf{y}}) = S_{\mathbf{y}}^{-1} \Sigma(s_t \tilde{t} + c_t, S_{\mathbf{y}} \tilde{\mathbf{y}} + \mathbf{c}_{\mathbf{y}}) \quad (2.120)$$

Proof. Applying the chain rule to the equality $f(t, \mathbf{y}) = \tilde{f}(\tilde{t}, \tilde{\mathbf{y}})$, the physical time derivative, spatial gradient, and spatial Hessian transform as follows:

$$\frac{\partial f}{\partial t} = \frac{1}{s_t} \frac{\partial \tilde{f}}{\partial \tilde{t}}, \quad \nabla_{\mathbf{y}} f = S_{\mathbf{y}}^{-1} \nabla_{\tilde{\mathbf{y}}} \tilde{f}, \quad \text{and} \quad \nabla_{\mathbf{y}}^2 f = S_{\mathbf{y}}^{-1} \nabla_{\tilde{\mathbf{y}}}^2 \tilde{f} S_{\mathbf{y}}^{-1} \quad (2.121)$$

We inject these identities directly into the continuous part of the physical FK operator:

$$\begin{aligned} \mathcal{L}f &= \frac{1}{s_t} \frac{\partial \tilde{f}}{\partial \tilde{t}} + \boldsymbol{\mu}(t, \mathbf{y})^\top (S_{\mathbf{y}}^{-1} \nabla_{\tilde{\mathbf{y}}} \tilde{f}) \\ &\quad + \frac{1}{2} \text{Tr} \left(\Sigma(t, \mathbf{y}) \Sigma(t, \mathbf{y})^\top S_{\mathbf{y}}^{-1} \nabla_{\tilde{\mathbf{y}}}^2 \tilde{f} S_{\mathbf{y}}^{-1} \right) - r f(t, \mathbf{y}) \end{aligned} \quad (2.122)$$

Using $\boldsymbol{\mu}^\top S_{\mathbf{y}}^{-1} = (S_{\mathbf{y}}^{-1} \boldsymbol{\mu})^\top$ because $S_{\mathbf{y}}$ is diagonal and the cyclic property of the trace operator $\text{Tr}(ABC) = \text{Tr}(CAB)$, we find:

$$\begin{aligned} \mathcal{L}f &= \frac{1}{s_t} \frac{\partial \tilde{f}}{\partial \tilde{t}} + \left(S_{\mathbf{y}}^{-1} \boldsymbol{\mu}(t, \mathbf{y}) \right)^\top \nabla_{\tilde{\mathbf{y}}} \tilde{f} \\ &\quad + \frac{1}{2} \text{Tr} \left(S_{\mathbf{y}}^{-1} \Sigma(t, \mathbf{y}) \Sigma(t, \mathbf{y})^\top S_{\mathbf{y}}^{-1} \nabla_{\tilde{\mathbf{y}}}^2 \tilde{f} \right) - r \tilde{f}(\tilde{t}, \tilde{\mathbf{y}}) \end{aligned} \quad (2.123)$$

We recognize that $S_{\mathbf{y}}^{-1} \Sigma(t, \mathbf{y}) \Sigma(t, \mathbf{y})^\top S_{\mathbf{y}}^{-1} = \left(S_{\mathbf{y}}^{-1} \Sigma(t, \mathbf{y}) \right) \left(S_{\mathbf{y}}^{-1} \Sigma(t, \mathbf{y}) \right)^\top$.

By explicitly mapping the physical coordinates $(t, \mathbf{y})$ back to their normalized counterparts $(s_t \tilde{t} + c_t, S_{\mathbf{y}} \tilde{\mathbf{y}} + \mathbf{c}_{\mathbf{y}})$, we deduce the new drift:

$$\tilde{\boldsymbol{\mu}}(\tilde{t}, \tilde{\mathbf{y}}) = S_{\mathbf{y}}^{-1} \boldsymbol{\mu}(s_t \tilde{t} + c_t, S_{\mathbf{y}} \tilde{\mathbf{y}} + \mathbf{c}_{\mathbf{y}})$$

and spatial diffusion:

$$\tilde{\Sigma}(\tilde{t}, \tilde{\mathbf{y}}) = S_{\mathbf{y}}^{-1} \Sigma(s_t \tilde{t} + c_t, S_{\mathbf{y}} \tilde{\mathbf{y}} + \mathbf{c}_{\mathbf{y}})$$

□

Proposition 11. *Let the physical jump operator be defined as $\Delta(f(t, \mathbf{y}))$ defined in 1.24. The jump operator under the linear mapping $\tilde{\Delta}(\tilde{f})$ is given by:*

$$\tilde{\Delta}(\tilde{f}(\tilde{t}, \tilde{\mathbf{y}})) = \sum_{v=1}^{p-1} \lambda_v \left(\mathbb{E}^{\mathbb{Q}} \left[\tilde{f}(\tilde{t}, \tilde{\mathbf{y}} + S_{\mathbf{y}}^{-1} M_v \xi^{(v)}) \right] - \tilde{f}(\tilde{t}, \tilde{\mathbf{y}}) \right) \quad (2.124)$$

Proof. The trick we use here is to compute the expectation over the physical distribution of the jump size $\xi^{(v)}$ rather than mapping the probability measure itself. To do so, we again use the price equivalence in both domains $f(t, \mathbf{z}) = \tilde{f}(\tilde{t}, S_{\mathbf{y}}^{-1}(\mathbf{z} - \mathbf{c}_{\mathbf{y}}))$ for any post-jump spatial state $\mathbf{z} = \mathbf{y} + M_v \xi^{(v)}$.

Since the pre-jump state is defined as $\mathbf{y} = S_{\mathbf{y}} \tilde{\mathbf{y}} + \mathbf{c}_{\mathbf{y}}$, the post-jump state becomes $\mathbf{z} = S_{\mathbf{y}} \tilde{\mathbf{y}} + \mathbf{c}_{\mathbf{y}} + M_v \xi^{(v)}$. Mapping this strictly back to the normalized spatial domain yields:

$$S_{\mathbf{y}}^{-1}(\mathbf{z} - \mathbf{c}_{\mathbf{y}}) = S_{\mathbf{y}}^{-1}(S_{\mathbf{y}} \tilde{\mathbf{y}} + M_v \xi^{(v)}) = \tilde{\mathbf{y}} + S_{\mathbf{y}}^{-1} M_v \xi^{(v)} \quad (2.125)$$

Substituting this result back into the trial function provides the evaluation identity:

$$f(t, \mathbf{y} + M_v \xi^{(v)}) = \tilde{f}(\tilde{t}, \tilde{\mathbf{y}} + S_{\mathbf{y}}^{-1} M_v \xi^{(v)}) \quad (2.126)$$

Taking the expectation $\mathbb{E}^{\mathbb{Q}}$ of both sides yields the stated normalized jump operator $\tilde{\Delta} \tilde{f}$. $\square$

By constructing the normalized continuous operator $\tilde{\mathcal{L}}$ using the identified parameters from propositions 10 and the normalized jump operator $\tilde{\Delta}$ from proposition 11:

$$\begin{cases} \tilde{\mathcal{L}} \tilde{f}(\tilde{t}, \tilde{\mathbf{y}}) = -\tilde{\Delta} \tilde{f}(\tilde{t}, \tilde{\mathbf{y}}) \\ \tilde{f}(\tilde{t}, \tilde{\mathbf{y}}) = h(s_t \tilde{t} + c_t, S_{\mathbf{y}} \tilde{\mathbf{y}} + \mathbf{c}_{\mathbf{y}}) \end{cases} \quad (2.127)$$

the differential operators are mathematically equivalent, such that solving $\tilde{\mathcal{L}} \tilde{f} = -\tilde{\Delta} \tilde{f}$ perfectly preserves the relationship $\mathcal{L} f = -\Delta f$. Therefore, finding a function $\tilde{f}$ that satisfies this normalized system on the unit hypercube guarantees that $\tilde{f}(\tilde{t}, \tilde{\mathbf{y}}) = f(t, \mathbf{y})$.

Chapter 3

Numerical Experiment

In this chapter, we show the utility of the method by using it to price options in a Heston model with Kou jumps. We will consider an option with one underlying only and then a basket option with two underlyings to highlight the efficiency of the method in multiple dimensions.

3.1 Heston-Kou Model

We show the variety of the model that can be used with the CGPR by testing the method with a Heston-Kou model. The Heston-Kou model unifies the continuous Heston dynamics with a jump-diffusion process. To better understand the model, we remind the 1D Heston model, In the risk-neutral measure $\mathbb{Q}$:

$$dS_t = (r - \lambda\zeta)S_t dt + \sqrt{\nu_t}S_t dW_t^S + (e^J - 1)S_{t-}dN_t \quad (3.1)$$

$$d\nu_t = \kappa(\bar{\nu} - \nu_t)dt + \epsilon\sqrt{\nu_t}dW_t^\nu \quad (3.2)$$

where S_t is the asset price, ν_t is the instantaneous variance, and the two Brownian motions are can be and are usually correlated. This model allows a mean-reverting stochastic variance successfully captures volatility clustering and the volatility smile over long maturities, while the Poisson jump process (N_t) with random magnitude J generates the heavy tails and skewness observed in short-term returns.

To evaluate this model using our method, we must map these specific dynamics into our generalized risk-neutral SDE for the state vector $\mathbf{y}_t \in \mathbb{R}^{p-1}$, previously defined as:

$$d\mathbf{y}_t = \boldsymbol{\mu}^{\mathbb{Q}}(t, \mathbf{y}_t)dt + \Sigma(t, \mathbf{y}_t)d\mathbf{W}_t + M(t, \mathbf{y}_t)d\mathbf{L}_t \quad (3.3)$$

To formulate the Heston model, the state vector $\mathbf{y}_t$ will encompass both the asset prices S_t and their stochastic variances ν_t . To introduce the Kou component, the

jump differential $M(t, \mathbf{y}_t)d\mathbf{L}_t$ is constructed as a compound Poisson process where the jump sizes follow a double exponential distribution.

3.1.1 Kou Jump Dynamics

In the Kou model, the discontinuities in the asset price are driven by a compound Poisson process with intensity λ_v . Let $J^{(v)}$ denote the random magnitude of the jump in the logreturn space. By definition, if a jump occurs at time t , the log-price experiences an additive shock:

$$\ln S_t - \ln S_{t-} = J^{(v)} \implies S_t = S_{t-}e^{J^{(v)}} \quad (3.4)$$

The absolute change in the asset price is therefore the difference between the post-jump and pre-jump price:

$$\Delta S_t = S_t - S_{t-} = S_{t-}(e^{J^{(v)}} - 1) \quad (3.5)$$

Because our generalized scaling matrix $M(t, \mathbf{y}_t)$ is explicitly defined to extract the pre-jump price S_{t-} , the random variable $\xi^{(v)}$ contained within our abstract jump vector $d\mathbf{L}_t$ must strictly represent the relative percentage jump. Therefore, we mathematically identify $\xi^{(v)} = e^{J^{(v)}} - 1$.

The log-jump $J^{(v)}$ is modeled using an asymmetric double exponential distribution, allowing for distinct probabilities and decay rates for upward and downward jumps. Its probability density function is given by:

$$f_{J^{(v)}}(j) = p_v \rho_v^+ e^{-\rho_v^+ j} \mathbb{1}_{\{j \geq 0\}} + (1 - p_v) \rho_v^- e^{\rho_v^- j} \mathbb{1}_{\{j < 0\}} \quad (3.6)$$

where $p_v \in (0, 1)$ is the probability of an upward jump, $\rho_v^+ > 1$ is the decay parameter for the right tail, and $\rho_v^- > 0$ is the decay parameter for the left tail.

To ensure the discounted asset price remains a martingale under the risk-neutral measure $\mathbb{Q}$, the drift must be compensated by the expected relative jump size $\zeta_v = \mathbb{E}^{\mathbb{Q}}[\xi^{(v)}]$ as discussed in the equation 1.10.

Lemma 10. *For a log-jump $J^{(v)}$ following the Kou double exponential distribution with parameters $(p_v, \rho_v^+, \rho_v^-)$ such that $\rho_v^+ > 1$ and $\rho_v^- > 0$, the expected relative jump size $\zeta_v = \mathbb{E}^{\mathbb{Q}}[\xi^{(v)}] = \mathbb{E}^{\mathbb{Q}}[e^{J^{(v)}} - 1]$ is given by:*

$$\zeta_v = \frac{p_v \rho_v^+}{\rho_v^+ - 1} + \frac{(1 - p_v) \rho_v^-}{\rho_v^- + 1} - 1 \quad (3.7)$$

Proof. By the definition of the expected value:

$$\begin{aligned}
\mathbb{E}^{\mathbb{Q}}[e^{J^{(v)}}] &= \int_{-\infty}^{\infty} e^j f_{J^{(v)}}(j) dj \\
&= \int_0^{\infty} e^j \left(p_v \rho_v^+ e^{-\rho_v^+ j} \right) dj + \int_{-\infty}^0 e^j \left((1 - p_v) \rho_v^- e^{\rho_v^- j} \right) dj \\
&= p_v \rho_v^+ \int_0^{\infty} e^{-(\rho_v^+ - 1)j} dj + (1 - p_v) \rho_v^- \int_{-\infty}^0 e^{(\rho_v^- + 1)j} dj
\end{aligned}$$

Because $\rho_v^+ > 1$ and $\rho_v^- > 0$, both integrals converge. It finally yields:

$$\begin{aligned}
\mathbb{E}^{\mathbb{Q}}[e^{J^{(v)}}] &= p_v \rho_v^+ \left[\frac{-1}{\rho_v^+ - 1} e^{-(\rho_v^+ - 1)y} \right]_0^{\infty} + (1 - p_v) \rho_v^- \left[\frac{1}{\rho_v^- + 1} e^{(\rho_v^- + 1)y} \right]_{-\infty}^0 \\
&= p_v \rho_v^+ \left(0 - \frac{-1}{\rho_v^+ - 1} \right) + (1 - p_v) \rho_v^- \left(\frac{1}{\rho_v^- + 1} - 0 \right) \\
&= \frac{p_v \rho_v^+}{\rho_v^+ - 1} + \frac{(1 - p_v) \rho_v^-}{\rho_v^- + 1}
\end{aligned}$$

substituting into the definition of the relative jump, we obtain:

$$\zeta_v = \mathbb{E}^{\mathbb{Q}}[e^{J^{(v)}} - 1] = \frac{p_v \rho_v^+}{\rho_v^+ - 1} + \frac{(1 - p_v) \rho_v^-}{\rho_v^- + 1} - 1$$

□

3.2 Single Asset Configuration

For a single asset, the state vector is $\mathbf{y}_t = (S_t, \nu_t)^\top$, containing the asset price and its variance. The drift vector $\mu(t, \mathbf{y}_t)$ contains the risk-neutral growth and the mean-reverting variance:

$$\mu(t, \mathbf{y}_t) = \begin{pmatrix} (r - \lambda \zeta) S_t \\ \kappa(\bar{\nu} - \nu_t) \end{pmatrix} \quad (3.8)$$

where r is the risk-free rate, κ is the speed of mean reversion, and $\bar{\nu}$ is the mean variance.

As seen earlier the Heston model uses correlated brownian motions dW_t^S and dW_t^ν . To evaluate the Feynman-Kac PDE within our generalized multi-dimensional framework, the underlying stochastic driver $d\mathbf{W}_t$ must be a vector of mutually independent standard Brownian motions. To bridge this gap, we absorb the correlation structure directly into the spatial diffusion matrix $\Sigma(t, \mathbf{y}_t)$ using the

Cholesky decomposition. Let $V(t, \mathbf{y}_t)$ be the diagonal matrix containing the uncorrelated volatilities of the state variables, and let R be the correlation matrix:

$$V(t, \mathbf{y}_t) = \begin{pmatrix} S_t \sqrt{\nu_t} & 0 \\ 0 & \epsilon \sqrt{\nu_t} \end{pmatrix}, \quad R = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} \quad (3.9)$$

where ϵ is the volatility of variance. By computing the lower-triangular Cholesky factor L such that $LL^\top = R$, the full diffusion matrix is constructed as the product $\Sigma(t, \mathbf{y}_t) = V(t, \mathbf{y}_t)L$:

$$\Sigma(t, \mathbf{y}_t) = \begin{pmatrix} S_t \sqrt{\nu_t} & 0 \\ 0 & \epsilon \sqrt{\nu_t} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ \rho & \sqrt{1 - \rho^2} \end{pmatrix} \quad (3.10)$$

The justification for this matrix construction relies on the following result:

Lemma 11. *Let $\mathbf{W}_t = (W_t^{(1)}, W_t^{(2)})^\top$ be a vector of independent standard Brownian motions, meaning its infinitesimal covariance matrix is:*

$$\mathbb{E}[d\mathbf{W}_t d\mathbf{W}_t^\top] = Idt$$

where I is the identity matrix. Let R be a valid symmetric positive-definite correlation matrix and L be its lower-triangular Cholesky factor ($LL^\top = R$). The transformed vector $d\mathbf{Z}_t = Ld\mathbf{W}_t$ defines a system of correlated Brownian motions with exactly the correlation structure Rdt .

Proof. To prove that $d\mathbf{Z}_t$ exhibits the correct correlation structure, we must compute its infinitesimal covariance matrix, defined as $\text{Cov}(d\mathbf{Z}_t) = \mathbb{E}[d\mathbf{Z}_t d\mathbf{Z}_t^\top]$. Substituting the definition of $d\mathbf{Z}_t$:

$$\begin{aligned} \mathbb{E}[d\mathbf{Z}_t d\mathbf{Z}_t^\top] &= \mathbb{E}[(Ld\mathbf{W}_t)(Ld\mathbf{W}_t)^\top] \\ &= \mathbb{E}[Ld\mathbf{W}_t d\mathbf{W}_t^\top L^\top] \\ &= L \cdot \mathbb{E}[d\mathbf{W}_t d\mathbf{W}_t^\top] \cdot L^\top \end{aligned}$$

Since the original Brownian motions $d\mathbf{W}_t$ are mutually independent, their covariance simplifies to Idt :

$$\begin{aligned} \mathbb{E}[d\mathbf{Z}_t d\mathbf{Z}_t^\top] &= L(Idt)L^\top \\ &= (LL^\top)dt \\ &= Rdt \end{aligned}$$

□

Because jumps only affect the asset price, the scaling matrix $M(t, \mathbf{y}_t)$ isolating the jump differential $d\mathbf{L}_t = (\xi^{(1)} dN_t, 0)^\top$ is strictly diagonal:

$$M(t, \mathbf{y}_t) = \begin{pmatrix} S_{t-} & 0 \\ 0 & 0 \end{pmatrix} \quad (3.11)$$

3.3 Two-Asset Basket Configuration

For a basket option on two assets, the dimension expands to $\mathbf{y}_t = (S_{1,t}, \nu_{1,t}, S_{2,t}, \nu_{2,t})^\top$. The 4-dimensional drift vector immediately scales to:

$$\mu(t, \mathbf{y}_t) = \begin{pmatrix} (r - \lambda_1 \zeta_1) S_{1,t} \\ \kappa_1 (\bar{\nu}_1 - \nu_{1,t}) \\ (r - \lambda_2 \zeta_2) S_{2,t} \\ \kappa_2 (\bar{\nu}_2 - \nu_{2,t}) \end{pmatrix} \quad (3.12)$$

The diffusion matrix $\Sigma(t, \mathbf{y}_t) \in \mathbb{R}^{4 \times 4}$ requires the full Cholesky factor L of the 4×4 correlation matrix R^1 , which encodes the correlation between the assets and their stochastic volatility (ρ_1, ρ_2) and the cross-asset correlation (ρ_{12}) . Defining the uncorrelated volatility matrix V , we have:

$$\begin{aligned} V(t, \mathbf{y}_t) &= \text{diag} \left(S_{1,t} \sqrt{\nu_{1,t}}, \epsilon_1 \sqrt{\nu_{1,t}}, S_{2,t} \sqrt{\nu_{2,t}}, \epsilon_2 \sqrt{\nu_{2,t}} \right) \\ \Sigma(t, \mathbf{y}_t) &= V(t, \mathbf{y}_t) L \end{aligned} \quad (3.13)$$

Finally, we let $\mathbf{L}_t$ be generated by two independent jump processes affecting the respective assets. The jump scaling matrix $M(t, \mathbf{y}_t) \in \mathbb{R}^{4 \times 2}$ maps these shocks strictly to the price coordinates, ensuring variance continuity:

$$M(t, \mathbf{y}_t) = \begin{pmatrix} S_{1,t-} & 0 \\ 0 & 0 \\ 0 & S_{2,t-} \\ 0 & 0 \end{pmatrix} \quad (3.14)$$

Through this exact identification, the multi-dimensional Heston-Kou dynamics can therefore be embedded into the generalized regression algorithm.

3.4 Domain Discretization with Sobol Sequence

To evaluate the continuous Partial Differential Equation numerically, we must discretize the continuous state space into a finite set of interior points $\overset{\circ}{\mathbf{X}}$ and boundary points $\partial \mathbf{X}$.

¹As shown in lemma 11.

The choice of discretization is highly important. If we use a grid of points uniformly at random, we are exposed to points clustering or gaps in the domain. On the other hand, if we use a deterministic cartesian grid, we will need an exponentially increasing number of points when the dimensionality increases.

To achieve a uniform, space-filling distribution of points without the exponential cost of a grid, we use a Sobol sequence (CITE). This sequence is a quasi-Monte Carlo method explicitly designed to minimize gaps and clustering, ensuring that the sampled points cover the high-dimensional domain as evenly as possible.

3.4.1 Domain Mapping Equation

A standard Sobol sequence generator produces a deterministic sequence of N points within a D -dimensional unit hypercube. Let $\mathbf{u}^{(i)} = (u_1^{(i)}, u_2^{(i)}, \dots, u_D^{(i)}) \in [0, 1]^D$ represent the i -th point in this standard sequence, for $i = 1, \dots, N$.

To generate the discretization, these unit points must be linearly mapped to our actual domain. The method is the same for both the boundary conditions and the inner domain: let $\mathbf{x}_{\min} \in \mathbb{R}^D$ and $\mathbf{x}_{\max} \in \mathbb{R}^D$ represent the lower and upper bounds of our state variables (time, asset prices, and variances) for a domain subset.

The i -th grid point $\mathbf{x}^{(i)} \in \mathbb{R}^D$ is obtained via the following transformation:

$$\mathbf{x}^{(i)} = \mathbf{x}_{\min} + \mathbf{u}^{(i)} \odot (\mathbf{x}_{\max} - \mathbf{x}_{\min}) \quad (3.15)$$

where $\odot$ denotes the Hadamard product. By applying this transformation to the desired number of samples, across $\mathcal{X}$ and across the different boundary conditions on $\partial\mathcal{X}$, we successfully construct the subsets $\mathring{\mathbf{X}}$ and $\partial\mathbf{X}$. The details of the sequence generator algorithm is given in Glasserman 2003.

3.4.2 Expectation Evaluation via Gauss-Laguerre

Evaluating $\Delta(f^{(u)})(x)$ necessitates computing $\mathbb{E}_{\xi^{(v)}}^{\mathbb{Q}} [\boldsymbol{\beta}(x + \delta_v(\xi^{(v)}), \partial\mathbf{X})^\top]$. While this expectation represents jumps via the relative magnitude $\xi^{(v)}$, it is mathematically easier to evaluate this expectation using the log-jump magnitude $J^{(v)}$, since $\xi^{(v)} = e^{J^{(v)}} - 1$.

Let $\tilde{\delta}_v(J^{(v)})$ denote the function that maps a log-jump to the corresponding additive shift in the state vector $\mathbf{x}$. The required expectation is thus $\mathbb{E}_{J^{(v)}}^{\mathbb{Q}} [\boldsymbol{\beta}(x + \tilde{\delta}_v(J^{(v)}), \partial\mathbf{X})^\top]$. To evaluate this under Kou's Double Exponential jump model, we compute the integral of the kernel feature vector against the log-jump probability density func-

tion². Remembering the log-jump size $J^{(v)}$ follows Kou's density defined in 3.6, the expectation splits into two integrals over the positive and negative domains:

$$\begin{aligned}\mathbb{E}_{J^{(v)}}^{\mathbb{Q}} \left[\beta(x + \tilde{\delta}_v(J^{(v)})) \right] &= \int_0^\infty \beta(x + \tilde{\delta}_v(y)) p_v \rho_v^+ e^{-\rho_v^+ y} dy \\ &\quad + \int_{-\infty}^0 \beta(x + \tilde{\delta}_v(y)) (1 - p_v) \rho_v^- e^{\rho_v^- y} dy.\end{aligned}\quad (3.16)$$

Because both integrals feature exponentially decaying weights, they can be efficiently solved using Gauss-Laguerre quadrature. The standard Gauss-Laguerre rule approximates integrals of the form

$$\int_0^\infty e^{-\tau} g(\tau) d\tau \approx \sum_{q=1}^{N_Q} \omega_q g(\tau_q) \quad (3.17)$$

where τ_q and ω_q are the quadrature nodes and weights, as explained in Salzer and Zucker 1949.

This numerical integration relies on the properties of Laguerre polynomials, denoted $L_n(\tau)$, which form a sequence of orthogonal polynomials with respect to the exponential weighting function $W(\tau) = e^{-\tau}$. For a chosen quadrature order N_Q , the integration nodes τ_q ($q = 1, \dots, N_Q$) are strictly defined as the roots of the N_Q -th Laguerre polynomial, such that $L_{N_Q}(\tau_q) = 0$. The corresponding weights ω_q are constructed to ensure the integration is exact for all polynomials of degree up to $2N_Q - 1$, and are computed analytically via:

$$\omega_q = \frac{\tau_q}{(N_Q + 1)^2 [L_{N_Q+1}(\tau_q)]^2} \quad (3.18)$$

where the Laguerre polynomial $L_n(x)$ is defined as:

$$\begin{aligned}L_n(x) &= \frac{e^x}{n!} \left[\frac{d^n}{dx^n} (e^{-x} x^n) \right] \\ &= \frac{(-1)^n}{n!} \left(x^n - \frac{n^2}{1!} x^{n-1} + \frac{n^2(n-1)^2}{2!} x^{n-2} - \dots \right)\end{aligned}\quad (3.19)$$

To map our specific financial jump integrals to this standard analytical domain, we apply the change of variables $\tau = \rho_v^+ y$ for the positive tail, and $\tau = -\rho_v^- y$ for the negative tail. Substituting this into our expectations yields the final quadrature

²Note that depending on the jump model and the kernel used, a closed-form formula for the integral may be available.

approximation:

$$\begin{aligned}
\mathbb{E}_{J^{(v)}}^{\mathbb{Q}} \left[\beta(x + \tilde{\delta}_v(J^{(v)})) \right] &= p_v \int_0^\infty \beta \left(x + \tilde{\delta}_v \left(\frac{\tau}{\rho_v^+} \right) \right) e^{-\tau} d\tau \\
&\quad + (1 - p_v) \int_0^\infty \beta \left(x + \tilde{\delta}_v \left(-\frac{\tau}{\rho_v^-} \right) \right) e^{-\tau} d\tau \\
&\approx \sum_{q=1}^{N_Q} \omega_q p_v \beta \left(x + \tilde{\delta}_v \left(\frac{\tau_q}{\rho_v^+} \right) \right) \\
&\quad + \sum_{q=1}^{N_Q} \omega_q (1 - p_v) \beta \left(x + \tilde{\delta}_v \left(-\frac{\tau_q}{\rho_v^-} \right) \right) \tag{3.20}
\end{aligned}$$

3.5 Kernel Choices

The only step left to dive in the numerical experiments is to choose the kernel definition. To ensure the Gaussian Process possesses sufficient smoothness to be operated on by the Feynman-Kac PDE operator, we must choose a covariance function that is at least $\mathcal{C}^2$.

In this work, we explore two different kernel types: the Matérn 5/2 which is exactly twice differentiable and therefore allow less smooth regressions, and the Gaussian kernel which is infinitely differentiable and therefore smoother. Both kernels are widely used and broadly discussed in Rasmussen and Williams 2005.

3.5.1 Matérn 5/2 Kernel

We tune the kernel with a signal variance parameter a^2 , which controls the expected amplitude of the functions, and a set of lengthscales η controlling the correlation decay. For two arguments $x = (t, \mathbf{y})$ and $\tilde{x} = (\tilde{t}, \tilde{\mathbf{y}})$, the kernel is given by:

$$k(\rho) = a^2 \left(1 + \sqrt{5}\rho + \frac{5}{3}\rho^2 \right) \exp(-\sqrt{5}\rho)$$

where the generalized anisotropic distance $\rho(x, \tilde{x})$ is defined as:

$$\rho(x, \tilde{x}) = \left(\frac{(t - \tilde{t})^2}{\eta_t^2} + \sum_{k=1}^p \frac{(y_k - \tilde{y}_k)^2}{\eta_k^2} \right)^{\frac{1}{2}}$$

To train the Gaussian process and enforce the PDE constraints, we must evaluate the action of the Feynman-Kac operator on this kernel. We first define several intermediate variables and matrices to compactly express these high-dimensional derivatives:

- The temporal scalar $d_t = \frac{t-\tilde{t}}{\eta_t^2}$ and the spatial vector $d_y = \left(\frac{y_1-\tilde{y}_1}{\eta_1^2}, \dots, \frac{y_p-\tilde{y}_p}{\eta_p^2} \right)^\top$.
- The spatial covariance matrix $C(x) = \Sigma_y(x)\Sigma_y(x)^\top$.
- The diagonal matrix of spatial lengthscales $D_\eta = \text{diag}\left(\frac{1}{\eta_1^2}, \dots, \frac{1}{\eta_p^2}\right)$.
- Functions capturing the kernel's decay profile, defined recursively as $\phi_{n+1}(\rho) = \frac{\phi'_n(\rho)}{\rho}$, starting from $\phi_1(\rho) = \frac{k'(\rho)}{\rho}$:

$$\begin{aligned}\phi_1(\rho) &= -a^2 \frac{5}{3} (1 + \sqrt{5}\rho) e^{-\sqrt{5}\rho} \\ \phi_2(\rho) &= a^2 \frac{25}{3} e^{-\sqrt{5}\rho} \\ \phi_3(\rho) &= -a^2 \frac{125\sqrt{5}}{3} \frac{e^{-\sqrt{5}\rho}}{\rho} \\ \phi_4(\rho) &= a^2 \frac{625}{3} \frac{e^{-\sqrt{5}\rho}}{\rho^2} + a^2 \frac{125\sqrt{5}}{3} \frac{e^{-\sqrt{5}\rho}}{\rho^3}\end{aligned}$$

Single Operator Application

The Gram matrix requires the cross-covariance between interior PDE points and standard evaluation points, which is computed via the single application of the operator with respect to $\tilde{x}$. We denote this function $F(x, \tilde{x}) = \mathcal{L}_{\tilde{x}}k(x, \tilde{x})$. Using the previously defined symbols, it evaluates to:

$$F(x, \tilde{x}) = -r(\tilde{x})k(\rho) - \phi_1(\rho)\theta + \frac{1}{2}\phi_2(\rho)\Omega + \frac{1}{2}\phi_1(\rho)\Lambda_{\tilde{x}} \quad (3.21)$$

where we have grouped the spatially-dependent scalar terms evaluated at $\tilde{x}$ as:

$$\begin{aligned}\theta &= d_t + \mu_y(\tilde{x})^\top d_y \\ \Omega &= d_y^\top C(\tilde{x})d_y \\ \Lambda_{\tilde{x}} &= \text{tr}\left(C(\tilde{x})D_\eta\right)\end{aligned}$$

Double Operator Application

For the interaction between two interior PDE points in the Gram matrix, we require the double application of the Feynman-Kac operator, $\mathcal{L}_x(\mathcal{L}_{\tilde{x}}k) = \mathcal{L}_x F$. To express this compactly, we define two final intermediate scalar blocks:

$$\begin{aligned}S(x, \tilde{x}) &= -r(\tilde{x})\phi_1(\rho) - \phi_2(\rho)\theta + \frac{1}{2}\phi_3(\rho)\Omega + \frac{1}{2}\phi_2(\rho)\Lambda_{\tilde{x}} \\ R(x, \tilde{x}) &= -r(\tilde{x})\phi_2(\rho) - \phi_3(\rho)\theta + \frac{1}{2}\phi_4(\rho)\Omega + \frac{1}{2}\phi_3(\rho)\Lambda_{\tilde{x}}\end{aligned}$$

Because the differential operator is linear, the final contracted form resolves to:

$$\begin{aligned}
\mathcal{L}_x(\mathcal{L}_{\tilde{x}}k) = & -r(x)F(x, \tilde{x}) - \frac{\phi_1(\rho)}{\eta_t^2} - \phi_1(\rho) \left(\mu_y(x)^\top D_\eta \mu_y(\tilde{x}) \right) + \phi_2(\rho) \left(\mu_y(x)^\top D_\eta C(\tilde{x}) d_y \right) \\
& + S(x, \tilde{x}) \left[d_t + \mu_y(x)^\top d_y + \frac{1}{2} \text{tr} \left(C(x) D_\eta \right) \right] + \frac{1}{2} R(x, \tilde{x}) \left(d_y^\top C(x) d_y \right) \\
& - \phi_2(\rho) \left(d_y^\top C(x) D_\eta \mu_y(\tilde{x}) \right) + \phi_3(\rho) \left(d_y^\top C(x) D_\eta C(\tilde{x}) d_y \right) \\
& + \frac{1}{2} \phi_2(\rho) \text{tr} \left(C(x) D_\eta C(\tilde{x}) D_\eta \right)
\end{aligned} \tag{3.22}$$

The full mathematical derivation of these structures, including the matrix-vector gradient cascades, the Hessian constructions, and the singularity limit as $\rho \rightarrow 0$, is detailed in Appendix A.

3.5.2 Gaussian Kernel

The Gaussian kernel is the standard choice for infinitely differentiable functions $\mathcal{C}^\infty$. While its assumption of perfect smoothness is often considered overly strong for financial jump-diffusion processes, it is worth giving it a try. Using the same signal variance a^2 and the unified anisotropic distance $\rho(x, \tilde{x})$, the kernel is equivalent to:

$$k(\rho) = a^2 \exp \left(-\frac{1}{2} \rho^2 \right) \tag{3.23}$$

The theoretical advantage of our generalized formulation is that the entire matrix calculus derived for the double Feynman-Kac operator depends strictly on the sequence of radial functions $\phi_n(\rho)$. For the Gaussian kernel, this derivative cascade collapses into a remarkably simple alternating sequence:

$$\begin{aligned}
\phi_1(\rho) &= -k(\rho) \\
\phi_2(\rho) &= k(\rho) \\
\phi_3(\rho) &= -k(\rho) \\
\phi_4(\rho) &= k(\rho)
\end{aligned}$$

By substituting these specific radial functions into the generalized single operator $F(x, \tilde{x})$ and the double contracted operator $\mathcal{L}_x(\mathcal{L}_{\tilde{x}}k)$ defined in Equation 3.22, the exact analytical Gram matrix for the Gaussian kernel is immediately obtained without requiring any further matrix differentiation. The brief proof of this alternating cascade is provided in Appendix B.

3.6 Results

In this section, we showcase the results obtained to price different derivatives with the Heston-Kou Model. We benchmark our results with Monte-Carlo simulations.

3.6.1 Single Asset Model

We first evaluate the model on a standard three-dimensional state space encompassing time to maturity t , a single underlying asset price y , and its stochastic variance ν . We will price a call option, a put option and a knock-out put option. The parameters and the convergence metrics are displayed in tables 3.1 and 3.2.

Parameter	Single Asset (y, ν)
<i>Heston Volatility Dynamics</i>	
Mean Volatility $\bar{\nu}$	0.15 ²
Mean Reversion Rate κ	1.8
Vol of Vol ϵ	0.10
Asset-Vol Correlation $\rho_{y,\nu}$	-0.30
<i>Kou Jump Dynamics</i>	
Jump Intensity λ	10.0
Up-Jump Prob. p	0.35
Up-Jump Scale ρ^+	30.0
Down-Jump Scale ρ^-	40.0
<i>Shared Contract Parameters</i>	
Risk-free rate $r = 0.05$	
Maturity $T = 2.0$	
Strike $K = 100.0$	
Lower Knock-Out Barrier $B_L = 50.0$ (Double KO Put only)	
Upper Knock-Out Barrier $B_U = 150.0$ (Double KO Put only)	

Table 3.1: Financial and stochastic parameters for the single-asset models.

Solver Metric	Vanilla Call	Vanilla Put	Double KO Put
Kernel Function	Matérn-5/2	Matérn-5/2	Matérn-5/2
Collocation Points m, d	1024, 1000	1024, 1000	256, 100
Regularization factor σ^*	10^{-5}	10^{-5}	10^{-4}
Variance Scale a^2	8042.65	1959.05	719.30
Time Lengthscale η_t	15.98	17.47	1.79
Asset Lengthscale η_y	0.45	0.44	0.36
Vol Lengthscale η_ν	12.29	14.51	4.78
Iterations n	24	32	16
Terminal Error $\ f^{(n)} - f^{(n-1)}\ _\infty$	1.99e-06	1.27e-06	4.95e-07

Table 3.2: Hyperparameters and numerical convergence results for the single-asset options.

Vanilla Options Pricing

Firstly, to efficiently train the CGPR, we need to define our boundary conditions for the option. For a standard Call option with strike K and risk-free rate $r = r(t, y, \nu)$, the three conditions we use are:

- Terminal Payoff ($t = T$): At expiration, the value of the option is exactly its intrinsic payoff.

$$f(T, y, \nu) = \max(y - K, 0)$$

- Out-of-the-Money Boundary ($y = 0$): If the asset price that reaches zero, the call option can never recover and is permanently worthless.

$$f(t, 0, \nu) = 0$$

- In-the-Money Boundary ($y = y_{\max}$): We truncate the theoretically infinite upper bound of the asset price at a sufficiently large value $y_{\max}$. At this extreme price, the option is almost guaranteed to end in-the-money, and its value converges to the discounted forward price. Even though this condition is not completely accurate, it can help with the convergence. Note that it can be removed when the domain dimension is low.

$$f(t, y_{\max}, \nu) = y_{\max} - Ke^{-r(T-t)}$$

On the other hand for a standard Put option, the boundary logic at the asset extremes is inverted:

- Terminal Payoff ($t = T$): At expiration, the intrinsic payoff is paid.

$$f(T, y, \nu) = \max(K - y, 0)$$

- In-the-Money Boundary ($y = 0$): Because the asset is worthless and cannot recover, the put option is guaranteed to yield the strike price K exactly at maturity. The value of the option today is simply the present value of that guaranteed strike.

$$f(t, 0, \nu) = Ke^{-r(T-t)}$$

- Out-of-the-Money Boundary ($y = y_{max}$): The asset price is astronomically high, meaning the probability of the put ending in-the-money is infinitesimally small. The option is completely worthless. Again, this condition is not completely accurate, but is used to help with the convergence.

$$f(t, y_{max}, \nu) = 0$$

The figure 3.1 helps to visualize the conditions on the domain.

Boundary Conditions visualization for a Put Option

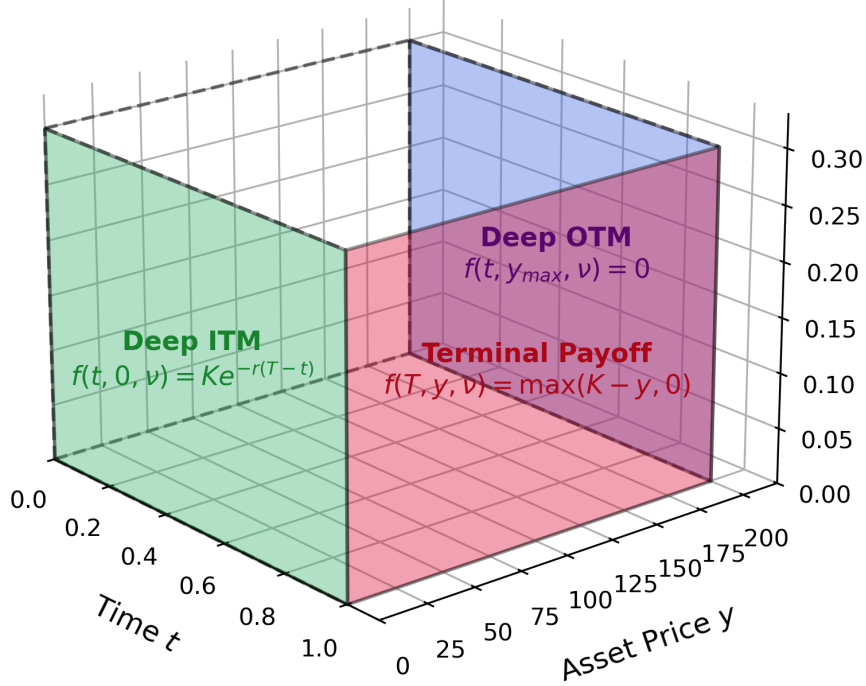


Figure 3.1: Chart of the boundary conditions locations and definitions

As it can be seen, not the whole frontier is defined by boundary conditions. The less the frontier is covered by boundary conditions, the less information the method has to propagate to the grid. This is why we add the deep in-the-money condition.

Another important feature of Gaussian processes comes from its fundamental assumption stated in equation 2.2: the function values are drawn from a known normal distribution. Using equations 2.20, 2.21 and 2.22, we get, for a prediction point $\mathbf{x}$:

$$f(\mathbf{x}) \sim \mathcal{N}(\mu_*, \Sigma_*)$$

For this particular point, we have:

$$\mu^* = \beta(\mathbf{x}, \partial\mathbf{X})^\top C(\mathring{\mathbf{X}}, \partial\mathbf{X})^{-1} \begin{bmatrix} h(\partial\mathbf{X}) \\ z(\mathring{\mathbf{X}}) \end{bmatrix} \quad (3.24)$$

and

$$\Sigma^* = k(\mathbf{x}, \mathbf{x}) - \beta(\mathbf{x}, \partial\mathbf{X})^\top C(\hat{\mathbf{X}}, \partial\mathbf{X})^{-1} \beta(\mathbf{x}, \partial\mathbf{X}) \quad (3.25)$$

Using this exact variance, we construct a 95% confidence interval around our predictions to visualize the model's uncertainty, :

$$P[f(\mathbf{x}) \in [\mu^* \pm 1.96\Sigma^*]] = 0.95 \quad (3.26)$$

To assess the validity of the CGPR model, we visualize the predicted option prices by isolating each physical dimension (t , y , and ν) and comparing them directly against the Monte Carlo baseline.

Since evaluating the price curve along a single dimension requires controlling for the remaining state variables, we compute a marginalized average over a discrete grid of the hidden dimensions. For example, let's say we want to see the impact of the spot price y . We then define $\mathcal{H}$ represent a uniform grid containing N_H discrete configurations of the remaining two variables t and ν . The marginalized average CGPR price, $\bar{f}_*(y)$, is computed mathematically as the arithmetic mean of the predictive posterior across the hidden grid:

$$\bar{f}_*(y) = \frac{1}{N_H} \sum_{t, \nu \in \mathcal{H}} f_*(t, y, \nu)$$

This identical marginalization operator is applied to the Monte Carlo benchmark³ to guarantee a strictly equivalent comparison. The results, demonstrating the model's behavior for both Call and Put options, are presented side-by-side in the figures below.

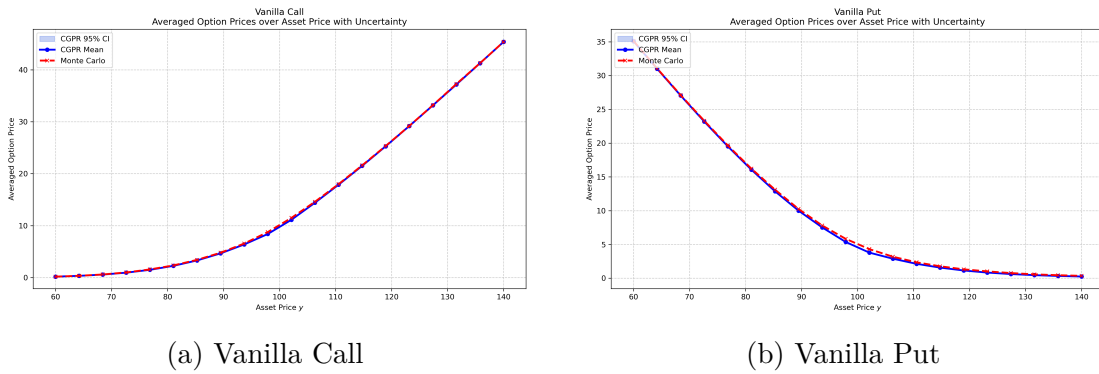


Figure 3.2: Marginalized option price across the asset price domain y .

³Note that we used 25000 simulations of 50 time steps.

The CGPR predictions in figure 3.2 are very close to the Monte Carlo baseline, anchored by a tight 95% confidence interval. It shows how the method is able to price efficiently the options.

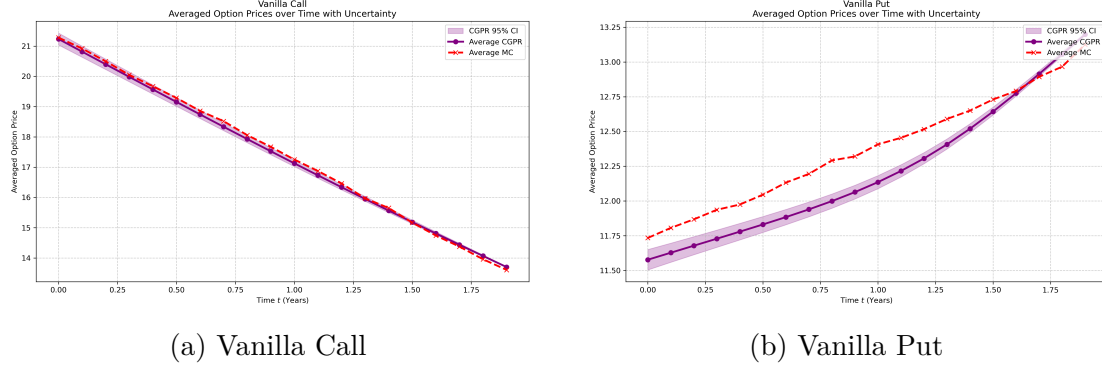


Figure 3.3: Marginalized option price across the time domain t .

Figure 3.3 illustrates the marginalized option prices over the time domain t . For the Vanilla Call option, we observe a standard, monotonically decreasing trend as time advances. This represents the classic loss of time value; as the option approaches maturity, the probability of favorable asset movements diminishes, eroding the premium.

Conversely, the Vanilla Put option exhibits an upward trajectory, gaining value as time passes. Because the graph represents an average across the entire asset domain, including Deep In-The-Money states, the trend is dominated by the ITM region. A Deep ITM Put guarantees a payoff bounded by the strike price K . The present value of this guaranteed payoff is discounted by the risk-free rate: $\bar{f}(t) = Ke^{-r(T-t)}$. For deep ITM Puts, this positive pull from the discount factor outweighs the loss of standard time value, resulting in an upward-sloping marginalized average.

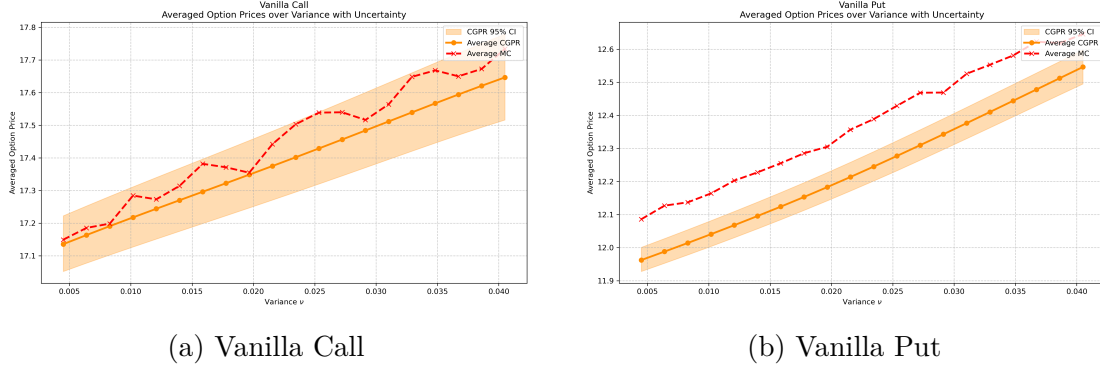


Figure 3.4: Marginalized option price across the stochastic variance domain ν .

Figure 3.4 perfectly shows that an increase in underlying stochastic volatility widens the distribution of terminal asset prices. This increases the probability of deep ITM payoffs, universally inflating the value for both Calls and Puts.

While the CGPR effectively captures the global trends, figures 3.3 and 3.4 reveal instances where the Monte Carlo baseline behaves noisily and occasionally drifts toward the upper boundary of the CGPR’s 95% confidence interval. This divergence can be seen as an advantage of the CGPR method and stems from three distinct numerical phenomena:

1. The CGPR analytically solves the deterministic PDE, whereas the MC relies on finite random sampling. Jump-diffusion models inherently exhibit massive variance in simulated terminal payoffs. The roughness of the MC curve is a direct consequence of this statistical noise.
2. The PDE framework models time continuously. In contrast, Monte Carlo simulations must discretize time. For the MC, we use an explicit Euler scheme to simulate the Heston-Kou differential equations. This discrete time-stepping introduces a systematic bias, which often scales with volatility and jump intensity, pulling the MC average slightly away from the true continuous-time solution estimated by the CGPR.
3. The last reason has the lowest impact and is actually a limitation of the CGPR for calls pricing only. To solve the PDE, the CGPR strictly enforces a boundary condition at $y = y_{max}$. Any probability mass beyond this boundary is effectively capped. The MC simulator, however, simulates unbounded spot prices. Unconstrained simulated paths can occasionally experience massive upward jumps far beyond y_{max} , generating extreme individual payoffs that pull the arithmetic MC average slightly higher than the domain-truncated PDE solution.

Barrier Options

In Vanilla options, the asset price domain is theoretically unbounded. To evaluate the PDE using CGPR, we must artificially truncate the spatial grid at a large value y_{max} and apply an approximate asymptotic boundary condition, as seen previously. While mathematically effective, this truncation inherently introduces a slight approximation error at the edges of the domain. However, this limitation can be overturned when pricing Knock-Out (KO) Barrier options. Because their physical boundaries are mathematically exact and finite, the truncation of the domain can be set at the barriers. Indeed, the option value collapses to strictly zero the exact moment the asset touches the barriers so that we no longer have to approximate an asymptotic boundary at infinity.

Firstly, to efficiently train the CGPR, we need to define our boundary conditions for the option. For a Double KO Barrier Put option with strike K , lower barrier B_L , and upper barrier B_U , the three exact boundary conditions we use are:

- Terminal Payoff ($t = T$): At expiration, the value of the option is exactly its intrinsic payoff, provided the asset price has strictly remained within the barriers.

$$f(T, y, \nu) = \begin{cases} \max(K - y, 0) & \text{if } B_L < y < B_U \\ 0 & \text{otherwise} \end{cases}$$

- Lower Knock-Out Boundary ($y = B_L$): If the asset price falls to or below the lower barrier at any time prior to maturity, the option knocks out and becomes permanently worthless.

$$f(t, B_L, \nu) = 0$$

- Upper Knock-Out Boundary ($y = B_U$): Similarly, if the asset price rallies to or above the upper barrier, the option immediately ceases to exist and holds zero value.

$$f(t, B_U, \nu) = 0$$

Unlike Vanilla options, Barrier options introduce severe discontinuities and extreme curvature into the PDE domain, as the option premium must aggressively collapse to zero at both the lower and upper bounds. The marginalized average prices across the asset, time, and variance domains are presented in the figures below. The barriers are $B_L = 50$ and $B_U = 150$.

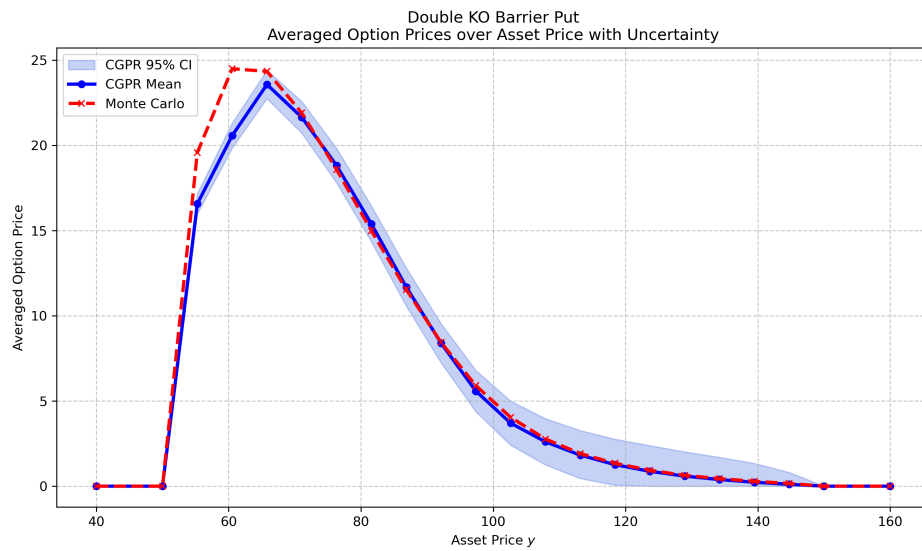


Figure 3.5: Marginalized Double KO Barrier Put price across the asset price domain y .

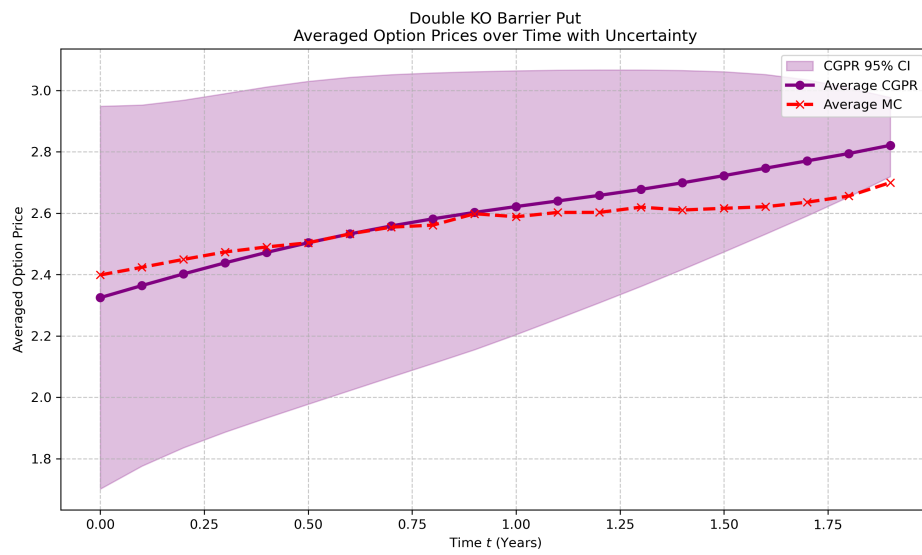


Figure 3.6: Marginalized Barrier Put price over time t .

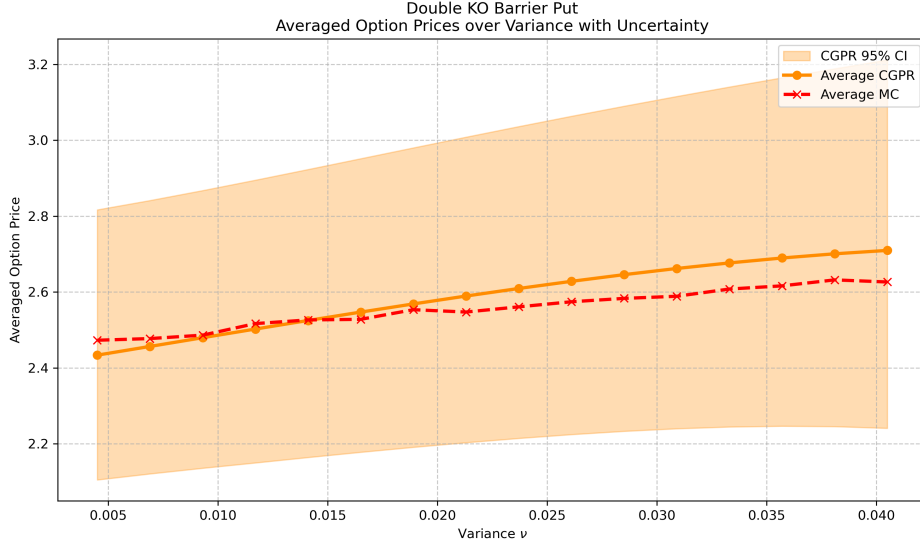


Figure 3.7: Marginalized Barrier Put price across the variance domain ν .

Figure 3.5 confirms that the CGPR is working for the barrier option. In the interval $y \in [55; 70]$ however, the CGPR seems to be slightly off when compared to the MC price. This is because this section is very steep and, even the Matérn kernel is too smooth to be completely accurate in this interval.

The CGPR also seems to correctly capture the price as a function of the time and the price as a function of the stochastic variance. As those parameters don't directly appear in the price and therefore the boundary conditions. Both the method can not be completely accurate. Let's still observe the CGPR is smoother, which is better for the greeks computations.

3.6.2 Double Asset Basket Model

To demonstrate the scalability and flexibility of the CGPR framework in high-dimensional state spaces, we evaluate double asset basket options. The state space consists of time t , two correlated asset prices y_1, y_2 , and their respective stochastic variances ν_1, ν_2 driven by independent Heston-Kou processes.

To evaluate the double asset basket options using the CGPR, we define a set of boundary conditions across the domain. Let the basket price be the equally weighted average of the two assets: $B = \frac{1}{2}(y_1 + y_2)$. The state space is bounded spatially by $y_1 \in [0, y_{1,\max}]$ and $y_2 \in [0, y_{2,\max}]$.

For a Basket Call option with strike K , the boundary conditions enforced on the

collocation points are:

- Terminal Payoff ($t = T$): At expiration, the value of the basket call is exactly its intrinsic payoff.

$$f(T, y_1, y_2, \nu_1, \nu_2) = \max \left(\frac{1}{2}(y_1 + y_2) - K, 0 \right)$$

- Out-of-the-Money Boundary ($y_1 = 0$ and $y_2 = 0$): If both underlying asset prices drop to zero, the basket is completely worthless and can never recover.

$$f(t, 0, 0, \nu_1, \nu_2) = 0$$

- Deep In-the-Money Boundary ($y_1 = y_{1,\max}$ or $y_2 = y_{2,\max}$): If either asset reaches its truncated upper bound, the option is almost guaranteed to end in-the-money. Its value is approximated by the discounted forward price of the basket.

$$f(t, y_1, y_2, \nu_1, \nu_2) = \frac{1}{2}(y_1 + y_2) - Ke^{-r(T-t)}$$

For a Basket Put option, the behavior at the spatial boundaries reverses. The boundary conditions are defined as:

- Terminal Payoff ($t = T$): At expiration, the value of the basket put is exactly its intrinsic payoff.

$$f(T, y_1, y_2, \nu_1, \nu_2) = \max \left(K - \frac{1}{2}(y_1 + y_2), 0 \right)$$

- In-the-Money Boundary ($y_1 = 0$ and $y_2 = 0$): If both underlying assets drop to 0, the put option guarantees the maximum possible payoff, discounted to its present value.

$$f(t, 0, 0, \nu_1, \nu_2) = Ke^{-r(T-t)}$$

- Out-of-the-Money Boundary ($y_1 = y_{1,\max}$ and $y_2 = y_{2,\max}$): If both assets reach the chosen upper bounds, the put option will expire out-of-the-money, carrying zero value, with a high probability. We approximate this behavior with:

$$f(t, y_{1,\max}, y_{2,\max}, \nu_1, \nu_2) = 0$$

To compare kernels, we model a Basket Call using a Gaussian kernel and a Basket Put using a Matérn-5/2 kernel. The financial parameters, jump-diffusion characteristics, and solver hyperparameters are detailed in Table 3.3, and the

convergence metrics of the two distinct kernel approaches are compared in Table 3.4.

Parameter	Asset 1 (y_1, ν_1)	Asset 2 (y_2, ν_2)
<i>Heston Volatility Dynamics</i>		
Mean Volatility $\bar{\nu}$	0.15 ²	0.20 ²
Mean Reversion Rate κ	1.8	2.0
Vol of Vol ϵ	0.10	0.15
Asset-Vol Correlation $\rho_{y,\nu}$	-0.30	-0.40
<i>Kou Jump Dynamics</i>		
Jump Intensity λ	10.0	5.0
Up-Jump Prob. p	0.35	0.40
Up-Jump Scale ρ^+	30.0	25.0
Down-Jump Scale ρ^-	40.0	35.0
<i>Shared Contract Parameters</i>		
Risk-free rate $r = 0.05$		
Maturity $T = 2.0$		
Strike $K = 100.0$		
Asset Correlation $\rho_{12} = 0.5$		

Table 3.3: Financial parameters for the double asset basket Heston-Kou model.

Solver Metric	Basket Call	Basket Put
Kernel Function	Gaussian	Matérn-5/2
Collocation Points m, d	1024, 1000	1024, 1000
Regularization factor σ^*	10^{-3}	10^{-8}
Variance Scale a^2	5132.41	1650.91
Time Lengthscale η_t	6.85	15.54
Asset 1 Lengthscale η_{y_1}	0.59	1.14
Asset 2 Lengthscale η_{y_2}	0.34	0.86
Vol 1 Lengthscale η_{ν_1}	5.70	15.66
Vol 2 Lengthscale η_{ν_2}	12.23	13.10
Iterations n	49	11
Terminal Error $\ f^{(n)} - f^{(n-1)}\ _\infty$	1.31e-04	7.29e-07

Table 3.4: Hyperparameters and convergence results.

Call Option Analysis

For the Basket Call option, the CGPR was configured using an infinitely differentiable Gaussian kernel. As it is particularly smooth, using this kernel leads to a very high condition number of the Gram matrix. Therefore, the regularization factor σ^* had to be increased to 10^{-3} .

Table 3.4 reveals the direct consequence of this numerical compromise: the iterative solver stalled at an error of $1.31e - 04$; which is higher than the general tolerance we have used so far.

We can now compare the price visually of the CGPR and the MC methods with averaged graphs, similarly to what we have done in the single asset section. Note that we performed 25000 MC simulations with 50 time steps each.

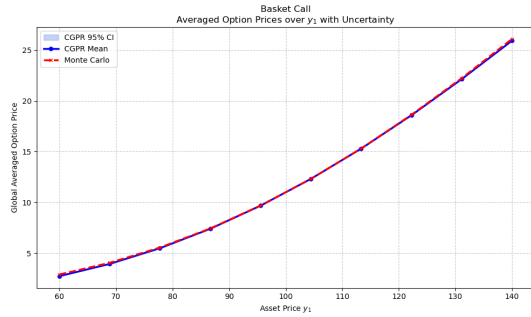


Figure 3.8: Marginalized Call price over y_1 .

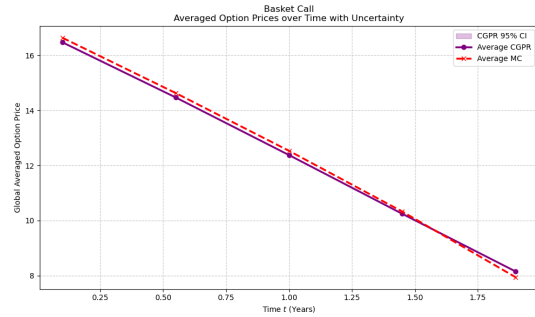


Figure 3.9: Marginalized Call price over Time.

We see the CGPR price curve as a function of the first asset is very close to the MC one which tends to indicate the method work as expected. While the price curve as a function of the time of the MC and the CGPR are very close, they are still not exactly agreeing. This can be due to the MC variance or the deep ITM boundary conditions approximation of the CGPR.

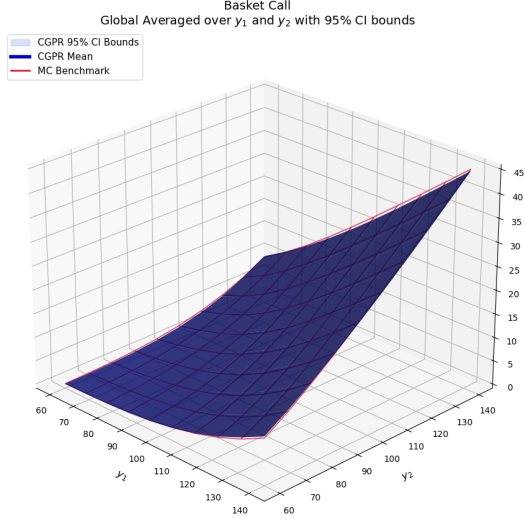


Figure 3.10: 3D Surface over the asset domain y_1, y_2 .

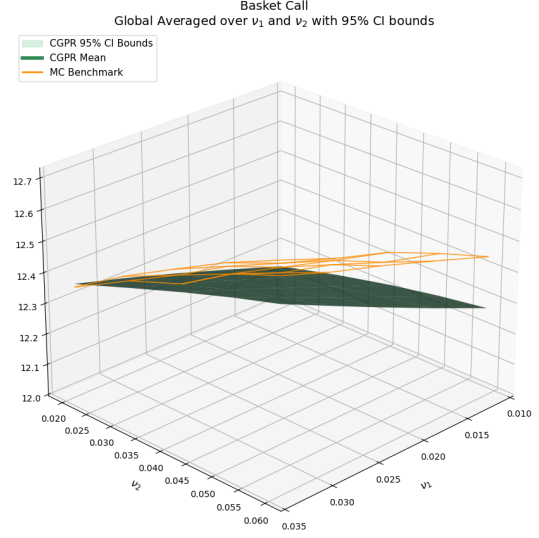


Figure 3.11: 3D Surface over the variance domain ν_1, ν_2 .

Figure 3.10 confirms the previous observations: the marginalized surface over the asset prices (y_1, y_2) demonstrates an alignment between the CGPR and the MC.

Conversely, Figure 3.11 reveals a severe structural decoupling between the two methods across the stochastic variance domain (ν_1, ν_2) . While the Monte Carlo shows upward slopes with respect to the volatilities, the CGPR surface flattens out and systematically underprices the option at higher initial variances. This can be due to the fact there are no boundary conditions on ν_1 or ν_2 , which makes the training harder.

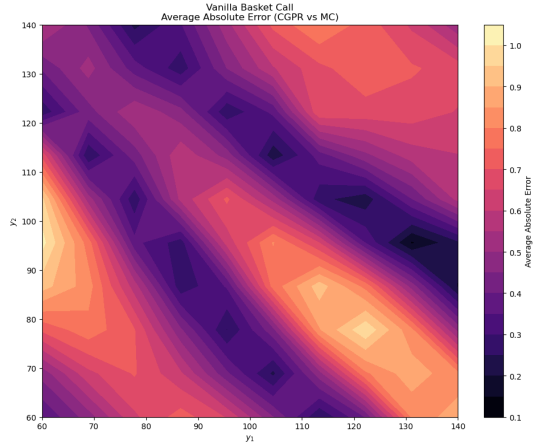


Figure 3.12: Assets Absolute Error Heatmap.

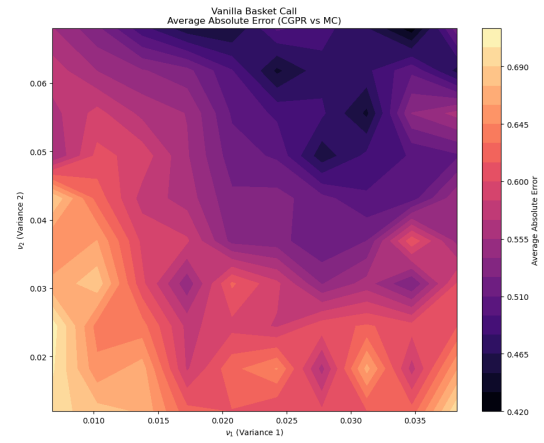


Figure 3.13: Vols Absolute Error Heatmap.

The last heatmaps allows to see where on the domain the CGPR and the MC disagree. For the y_1, y_2 plane, the errors are well-distributed on the whole domain, showing a convincing convergence of the CGPR. For ν_1, ν_2 plane, the disagreement is important for low values of ν_1 and ν_2 .

Put Option Analysis

In contrast, the Basket Put option was evaluated using the Matérn-5/2 kernel. Because this kernel is only twice differentiable, it is less smooth, decreasing the condition number of the Gram matrix. This allows us to drop the regularization factor down to a strict $\sigma^* = 10^{-8}$. Moreover, the iterative solver achieved a final error of convergence of 7.29×10^{-7} in just 11 iterations (Table 3.4).

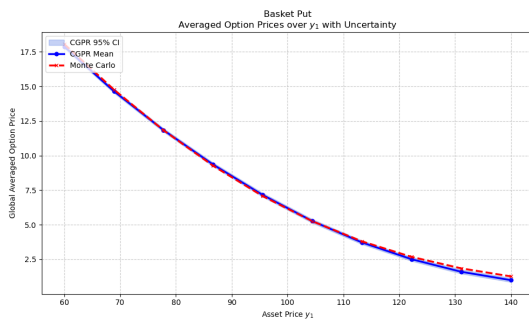


Figure 3.14: Marginalized Put price over y_1 .

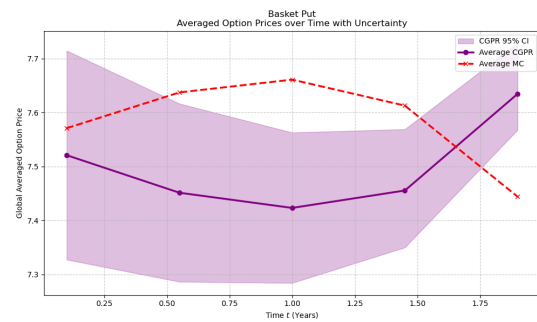


Figure 3.15: Marginalized Put price over Time.

The 1D marginalized plots reflect a great performance of the CGPR compared to the MC. Figure 3.14 generally aggre except when the option is OTM. This can be due to the OTM boundary condition approximation. In the time domain, the CGPR curve seems to be more convincing than the MC. Indeed, the value of the put should be decaying over time except near the maturity where we observe upward slope, reflecting the dominance of the discount factor $e^{-r(T-t)}$ on deep ITM put payoffs as maturity approaches.

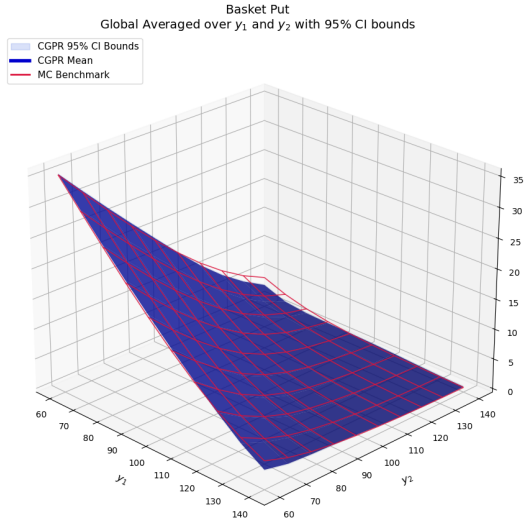


Figure 3.16: Surface over the asset domain y_1, y_2 with 95% CI.

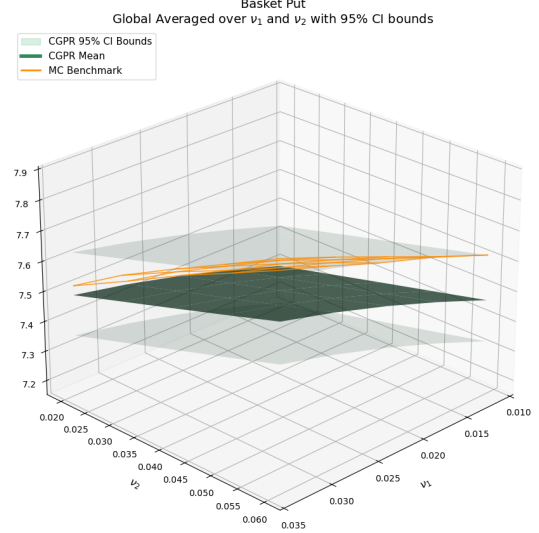


Figure 3.17: Surface over the variance domain ν_1, ν_2 with 95% CI.

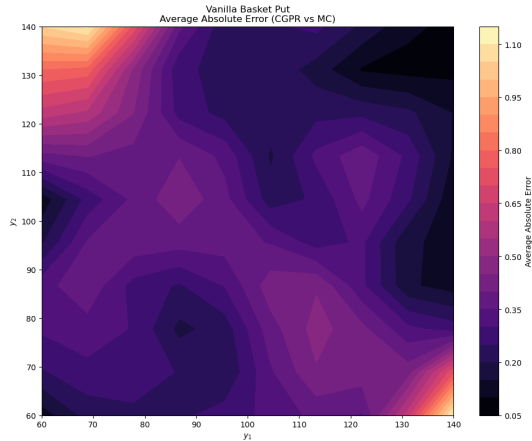


Figure 3.18: 3D Surface over the variance domain ν_1, ν_2 with 95% CI.

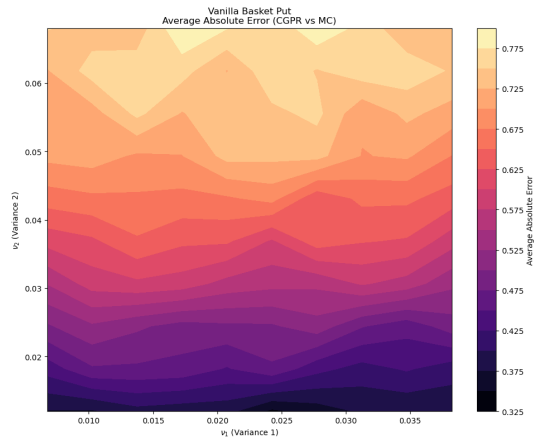


Figure 3.19: Heatmap of Absolute Error (ν_1, ν_2).

The 3D plots and the heatmaps gives visual insights about where the methods disagree: high values of y_1 , y_2 and ν_2 . Note that, because of the positive correlation between y_1 and y_2 and the negative correlation between y_2 and ν_2 , the disagreement could have the same source. As important disagreement occurs for high values of y_1 and y_2 , this is probably an artefact of the OTM boundary condition approximation.

Conclusion

This thesis presented Constrained Gaussian Process Regression to solve the multi-dimensional Feynman-Kac partial integro-differential equation governing multi-asset derivative pricing under jump-diffusion dynamics. By decoupling the continuous parabolic differential operator from the non-local jump integral via an operator-splitting scheme, the continuous boundary value problem was resolved analytically through a linearly constrained Gaussian process prior, while the discontinuous jump expectation was reconciled iteratively through a fixed-point scheme. Crucially, this decoupling method enables the construction of a general framework for any jump model, rather than being restricted to those admitting closed-form expectation.

The numerical results confirm the accuracy and efficiency of the proposed solver when benchmarked against Monte Carlo simulations on one-asset and two-asset basket options. While there is still room for improvement with the hyperparameters estimation, the model accurately recovered pricing surfaces while maintaining low residual errors. Unlike Monte Carlo methods that require independent path generations for every query and suffer from time step errors, the converged surrogate yields a smooth and analytically differentiable pricing function paired with an intrinsic posterior confidence metric, allowing the easy computation of the Greeks.

Nevertheless, the architecture faces practical constraints related to the conditioning of the dense Gram matrix. Evaluating high-order differential kernel operators over dense discretization grids yields ill-conditioned covariance systems, necessitating diagonal regularization and Cholesky factorizations. Additionally, joint negative log-likelihood calibration over asymmetric spatiotemporal lengthscales presents non-convex optimization challenges that become harder to navigate as dimensionality grows.

These challenges outline avenues for future research. Firstly, hyperparameters estimation and computational scalability can be elevated by leveraging the lower rank representation of the kernels via Bochner’s theorem. By decomposing standard radial basis function kernels into sine and cosine basis expansions through Random Fourier Features or generalized spectral mixture kernels, the kernel matrix can be

approximated in randomized linear time. This expansion simplifies the evaluation of differential and integral PIDE operators into linear transformations of trigonometric bases, reframing the non-convex lengthscale search into the spectral estimation of dominant frequency components.

Furthermore, the method could be improved through the development of a boosted Constrained Gaussian Process framework. A boosting scheme would train an ensemble of localized or shallow constrained Gaussian process learners sequentially on the residual errors of preceding stages. This iterative residual fitting would bypass the need to invert a high dimensional, ill-conditioned Gram matrix, thereby reducing computational overhead while enabling adaptive, stage-wise hyperparameters optimization.

Finally, while this work focused on European payoff structures, the framework can be extended to American-style options or, more generally, unit-linked insurance product by re-casting the optimal stopping boundary using smooth penalization methods within the regression constraints.

Appendix A

Derivations for the Matérn Kernel

A.1 Definition and Distance

Let the Matérn 5/2 kernel with signal variance a^2 be defined as:

$$k(\rho) = a^2 \left(1 + \sqrt{5}\rho + \frac{5}{3}\rho^2 \right) \exp(-\sqrt{5}\rho)$$

$$\rho(x, \tilde{x}) = \left(\frac{(t - \tilde{t})^2}{\eta_t^2} + \sum_{k=1}^p \frac{(y_k - \tilde{y}_k)^2}{\eta_k^2} \right)^{\frac{1}{2}}$$

The partial derivatives of the distance metric yield:

$$\begin{aligned} \frac{\partial \rho}{\partial t} &= \frac{1}{\rho} \frac{(t - \tilde{t})}{\eta_t^2} \\ \frac{\partial \rho}{\partial y_i} &= \frac{1}{\rho} \frac{(y_i - \tilde{y}_i)}{\eta_i^2} \end{aligned}$$

A.2 First Derivatives

$$\begin{aligned} k'(\rho) &= a^2 \left(\sqrt{5} + \frac{10}{3}\rho \right) e^{-\sqrt{5}\rho} - a^2 \sqrt{5} \left(1 + \sqrt{5}\rho + \frac{5}{3}\rho^2 \right) e^{-\sqrt{5}\rho} \\ &= -a^2 \frac{5}{3} \rho (1 + \sqrt{5}\rho) e^{-\sqrt{5}\rho} \end{aligned}$$

With respect to the state variables, this gives:

$$\begin{aligned}\frac{\partial k}{\partial t} &= \underbrace{-a^2 \frac{5}{3} (1 + \sqrt{5}\rho) e^{-\sqrt{5}\rho}}_{\phi_1(\rho)} \left(\frac{t - \tilde{t}}{\eta_t^2} \right) \\ \frac{\partial k}{\partial y_i} &= \underbrace{-a^2 \frac{5}{3} (1 + \sqrt{5}\rho) e^{-\sqrt{5}\rho}}_{\phi_1(\rho)} \left(\frac{y_i - \tilde{y}_i}{\eta_i^2} \right)\end{aligned}$$

A.3 Second Derivatives

$$\frac{\partial^2 k}{\partial y_i \partial y_j} = \left(\frac{\partial}{\partial y_j} \phi_1(\rho) \frac{y_i - \tilde{y}_i}{\eta_i^2} \right) + \phi_1(\rho) \frac{\delta_{ij}}{\eta_i^2}$$

Differentiating $\phi_1(\rho)$:

$$\begin{aligned}\phi_1'(\rho) &= -a^2 \frac{5}{3} \left[\sqrt{5} e^{-\sqrt{5}\rho} - \sqrt{5} (1 + \sqrt{5}\rho) e^{-\sqrt{5}\rho} \right] \\ &= a^2 \frac{25}{3} \rho e^{-\sqrt{5}\rho}\end{aligned}$$

Applying the chain rule:

$$\begin{aligned}\frac{\partial}{\partial y_j} \phi_1(\rho) &= \phi_1'(\rho) \frac{\partial \rho}{\partial y_j} \\ &= \underbrace{a^2 \frac{25}{3} e^{-\sqrt{5}\rho}}_{\phi_2(\rho)} \left(\frac{y_j - \tilde{y}_j}{\eta_j^2} \right)\end{aligned}$$

Therefore, the spatial Hessian components are:

$$\frac{\partial^2 k}{\partial y_i \partial y_j} = \phi_2(\rho) \frac{(y_i - \tilde{y}_i)(y_j - \tilde{y}_j)}{\eta_i^2 \eta_j^2} + \phi_1(\rho) \frac{\delta_{ij}}{\eta_i^2} \quad (\text{A.1})$$

A.4 Single Application on the FK Operator

The continuous continuous spatial operator is defined as:

$$\mathcal{L}_x f(x) = \partial_t f - r(x)f + \mu_y(x)^\top \nabla_y f + \frac{1}{2} \text{tr} \left(\Sigma_y(x) \Sigma_y(x)^\top \mathcal{H}_y f \right)$$

Let $d_y \in \mathbb{R}^p$ be the scaled spatial distance vector:

$$d_y = \left(\frac{y_1 - \tilde{y}_1}{\eta_1^2}, \dots, \frac{y_p - \tilde{y}_p}{\eta_p^2} \right)^\top \quad (\text{A.2})$$

The derivatives of the kernel can then be written in compact vector and matrix form:

$$\begin{aligned}\partial_t k &= \phi_1(\rho) \frac{t - \tilde{t}}{\eta_t^2} \\ \nabla_y k &= \phi_1(\rho) d_y \\ \mathcal{H}_y k &= \phi_2(\rho) d_y d_y^\top + \phi_1(\rho) D_\eta\end{aligned}$$

where $D_\eta = \text{diag}\left(\frac{1}{\eta_1^2}, \dots, \frac{1}{\eta_p^2}\right)$. Let $C(x) = \Sigma_y(x) \Sigma_y(x)^\top$ be the covariance matrix. Finally, note that both matrices D_η and $C(x)$ are strictly symmetric. Substituting these into the operator yields:

$$\begin{aligned}\mathcal{L}_x k(x, \tilde{x}) &= -r(x)k(\rho) \\ &+ \phi_1(\rho) \left(\frac{t - \tilde{t}}{\eta_t^2} + \mu_y(x)^\top d_y + \frac{1}{2} \text{tr}(C(x) D_\eta) \right) \\ &+ \frac{1}{2} \phi_2(\rho) \text{tr}(C(x) d_y d_y^\top)\end{aligned}$$

A.5 Second application on the FK Operator

Let the family of radial functions be defined recursively:

$$\phi_1(\rho) = \frac{k'(\rho)}{\rho}, \quad \phi_2(\rho) = \frac{\phi_1'(\rho)}{\rho}, \quad \phi_3(\rho) = \frac{\phi_2'(\rho)}{\rho}, \quad \phi_4(\rho) = \frac{\phi_3'(\rho)}{\rho}$$

Let $d_t = \frac{t - \tilde{t}}{\eta_t^2}$. Taking into account that $\partial_t k = -\phi_1(\rho) d_t$ and $\nabla_y k = -\phi_1(\rho) d_y$, we define the first application $F(x, \tilde{x}) = \mathcal{L}_{\tilde{x}} k(x, \tilde{x})$:

$$F(x, \tilde{x}) = -r(\tilde{x})k(\rho) - \phi_1(\rho)\theta + \frac{1}{2}\phi_2(\rho)\Omega + \frac{1}{2}\phi_1(\rho)\Lambda_{\tilde{x}} \quad (\text{A.3})$$

where we have defined the following scalars, evaluated at $\tilde{x}$:

$$\begin{aligned}\theta &= d_t + \mu_y(\tilde{x})^\top d_y \\ \Omega &= d_y^\top C(\tilde{x}) d_y \\ \Lambda_{\tilde{x}} &= \text{tr}(C(\tilde{x}) D_\eta)\end{aligned}$$

The objective is to evaluate:

$$\mathcal{L}_x \mathcal{L}_{\tilde{x}} k(x, \tilde{x}) = \mathcal{L}_x F = \partial_t F - r(x)F + \mu_y(x)^\top \nabla_y F + \frac{1}{2} \text{tr}(C(x) \mathcal{H}_y F)$$

Since $\nabla_y \rho = \frac{1}{\rho} d_y$, the spatial gradient of any radial function $f(\rho)$ is systematically written as:

$$\nabla_y f(\rho) = f'(\rho) \nabla_y \rho = f'(\rho) \left(\frac{1}{\rho} d_y \right) = \left(\frac{f'(\rho)}{\rho} \right) d_y \quad (\text{A.4})$$

Consequently, we establish the following cascade of gradients:

$$\begin{aligned} \nabla_y k(\rho) &= \phi_1(\rho) d_y \\ \nabla_y \phi_1(\rho) &= \phi_2(\rho) d_y \\ \nabla_y \phi_2(\rho) &= \phi_3(\rho) d_y \\ \nabla_y \phi_3(\rho) &= \phi_4(\rho) d_y \end{aligned}$$

The derivatives of the non-radial, spatially-dependent terms are:

$$\begin{aligned} \nabla_y \theta &= D_\eta \mu_y(\tilde{x}) \\ \nabla_y \Omega &= 2D_\eta C(\tilde{x}) d_y \end{aligned}$$

Let us compute the gradient $\nabla_y F$ by applying the product rule to each term:

$$\begin{aligned} \nabla_y F &= -r(\tilde{x}) [\nabla_y k(\rho)] \\ &\quad - [\nabla_y \phi_1(\rho)] \theta - \phi_1(\rho) [\nabla_y \theta] \\ &\quad + \frac{1}{2} [\nabla_y \phi_2(\rho)] \Omega + \frac{1}{2} \phi_2(\rho) [\nabla_y \Omega] \\ &\quad + \frac{1}{2} \Lambda_{\tilde{x}} [\nabla_y \phi_1(\rho)] \end{aligned}$$

Substituting the radial and non-radial gradients:

$$\begin{aligned} \nabla_y F &= -r(\tilde{x}) [\phi_1(\rho) d_y] \\ &\quad - [\phi_2(\rho) d_y] \theta - \phi_1(\rho) [D_\eta \mu_y(\tilde{x})] \\ &\quad + \frac{1}{2} [\phi_3(\rho) d_y] \Omega + \frac{1}{2} \phi_2(\rho) [2D_\eta C(\tilde{x}) d_y] \\ &\quad + \frac{1}{2} \Lambda_{\tilde{x}} [\phi_2(\rho) d_y] \end{aligned}$$

Factoring out the vector d_y , we define the scalar block $S(x, \tilde{x})$:

$$\begin{aligned} \nabla_y F &= \underbrace{\left[-r(\tilde{x}) \phi_1(\rho) - \phi_2(\rho) \theta + \frac{1}{2} \phi_3(\rho) \Omega + \frac{1}{2} \phi_2(\rho) \Lambda_{\tilde{x}} \right]}_{=S(x, \tilde{x})} d_y \\ &\quad - \phi_1(\rho) D_\eta \mu_y(\tilde{x}) + \phi_2(\rho) D_\eta C(\tilde{x}) d_y \end{aligned}$$

Analogously, for the gradient of $S(x, \tilde{x})$ we define $R(x, \tilde{x})$:

$$\begin{aligned}
\nabla_y S &= -r(\tilde{x})[\nabla_y \phi_1(\rho)] - [\nabla_y \phi_2(\rho)]\theta - \phi_2(\rho)[\nabla_y \theta] \\
&\quad + \frac{1}{2}[\nabla_y \phi_3(\rho)]\Omega + \frac{1}{2}\phi_3(\rho)[\nabla_y \Omega] + \frac{1}{2}\Lambda_{\tilde{x}}[\nabla_y \phi_2(\rho)] \\
&= \underbrace{\left[-r(\tilde{x})\phi_2(\rho) - \phi_3(\rho)\theta + \frac{1}{2}\phi_4(\rho)\Omega + \frac{1}{2}\phi_3(\rho)\Lambda_{\tilde{x}}\right]}_{=R(x, \tilde{x})} d_y \\
&\quad - \phi_2(\rho)D_\eta \mu_y(\tilde{x}) + \phi_3(\rho)D_\eta C(\tilde{x})d_y
\end{aligned}$$

For the time derivative:

$$\begin{aligned}
\partial_t F &= -r(\tilde{x})[\partial_t k(\rho)] - [\partial_t \phi_1(\rho)]\theta - \phi_1(\rho)[\partial_t \theta] \\
&\quad + \frac{1}{2}[\partial_t \phi_2(\rho)]\Omega + \frac{1}{2}\phi_2(\rho)[\partial_t \Omega] + \frac{1}{2}\Lambda_{\tilde{x}}[\partial_t \phi_1(\rho)] \\
&= -r(\tilde{x})[\phi_1(\rho)d_t] - [\phi_2(\rho)d_t]\theta - \frac{\phi_1(\rho)}{\eta_t^2} \\
&\quad + \frac{1}{2}[\phi_3(\rho)d_t]\Omega + \frac{1}{2}\Lambda_{\tilde{x}}[\phi_2(\rho)d_t] \\
&= S(x, \tilde{x})d_t - \frac{\phi_1(\rho)}{\eta_t^2}
\end{aligned}$$

A.6 Construction of the Hessian Matrix

To construct $\mathcal{H}_y F$, we differentiate $\nabla_y F$ with respect to $y^\top$. By the product rule:

$$\frac{\partial}{\partial y^\top}(Sd_y) = S(\nabla_y d_y) + d_y(\nabla_y S)^\top \quad (\text{A.5})$$

Evaluating the outer product term:

$$\begin{aligned}
d_y(\nabla_y S)^\top &= d_y \left(Rd_y - \phi_2(\rho)D_\eta \mu_y(\tilde{x}) + \phi_3(\rho)D_\eta C(\tilde{x})d_y \right)^\top \\
&= d_y \left(Rd_y^\top - \phi_2(\rho)\mu_y(\tilde{x})^\top D_\eta + \phi_3(\rho)d_y^\top C(\tilde{x})D_\eta \right) \\
&= Rd_y d_y^\top - \phi_2(\rho)d_y \mu_y(\tilde{x})^\top D_\eta + \phi_3(\rho)d_y d_y^\top C(\tilde{x})D_\eta
\end{aligned}$$

Applying the operator to the three terms of the spatial gradient gives:

$$\frac{\partial}{\partial y^\top}(Sd_y) = SD_\eta + Rd_y d_y^\top - \phi_2(\rho)d_y \mu_y(\tilde{x})^\top D_\eta + \phi_3(\rho)d_y d_y^\top C(\tilde{x})D_\eta \quad (\text{A.6})$$

$$\begin{aligned}\frac{\partial}{\partial y^\top} \left(-\phi_1(\rho) D_\eta \mu_y(\tilde{x}) \right) &= -\left(D_\eta \mu_y(\tilde{x}) \right) \left(\nabla_y \phi_1(\rho) \right)^\top \\ &= -\phi_2(\rho) D_\eta \mu_y(\tilde{x}) d_y^\top\end{aligned}\tag{A.7}$$

$$\begin{aligned}\frac{\partial}{\partial y^\top} \left(\phi_2(\rho) D_\eta C(\tilde{x}) d_y \right) &= \phi_2(\rho) \nabla_y \left(D_\eta C(\tilde{x}) d_y \right) + \left(D_\eta C(\tilde{x}) d_y \right) \left(\nabla_y \phi_2(\rho) \right)^\top \\ &= \phi_2(\rho) D_\eta C(\tilde{x}) D_\eta + \phi_3(\rho) D_\eta C(\tilde{x}) d_y d_y^\top\end{aligned}\tag{A.8}$$

Assembling the components yields the full spatial Hessian:

$$\begin{aligned}\mathcal{H}_y F &= R d_y d_y^\top + S D_\eta \\ &\quad - \phi_2(\rho) \left[D_\eta \mu_y(\tilde{x}) d_y^\top + d_y \mu_y(\tilde{x})^\top D_\eta \right] \\ &\quad + \phi_3(\rho) \left[D_\eta C(\tilde{x}) d_y d_y^\top + d_y d_y^\top C(\tilde{x}) D_\eta \right] \\ &\quad + \phi_2(\rho) D_\eta C(\tilde{x}) D_\eta\end{aligned}$$

A.7 Final Contracted Form

To summarize the components required for the final evaluation:

$$\begin{aligned}\partial_t F &= S d_t - \frac{\phi_1(\rho)}{\eta_t^2} \\ \nabla_y F &= S d_y - \phi_1(\rho) D_\eta \mu_y(\tilde{x}) + \phi_2(\rho) D_\eta C(\tilde{x}) d_y \\ \mathcal{H}_y F &= R d_y d_y^\top + S D_\eta - \phi_2(\rho) \left[D_\eta \mu_y(\tilde{x}) d_y^\top + d_y \mu_y(\tilde{x})^\top D_\eta \right] \\ &\quad + \phi_3(\rho) \left[D_\eta C(\tilde{x}) d_y d_y^\top + d_y d_y^\top C(\tilde{x}) D_\eta \right] + \phi_2(\rho) D_\eta C(\tilde{x}) D_\eta\end{aligned}$$

Injecting these derivatives into $\mathcal{L}_x F$ and utilizing the cyclic trace property $\text{tr}(A d_y b^\top) = b^\top A d_y$, we obtain the final operator:

$$\begin{aligned}\mathcal{L}_x(\mathcal{L}_{\tilde{x}} k) &= -r(x) F(x, \tilde{x}) - \frac{\phi_1(\rho)}{\eta_t^2} - \phi_1(\rho) \left(\mu_y(x)^\top D_\eta \mu_y(\tilde{x}) \right) + \phi_2(\rho) \left(\mu_y(x)^\top D_\eta C(\tilde{x}) d_y \right) \\ &\quad + S(x, \tilde{x}) \left[d_t + \mu_y(x)^\top d_y + \frac{1}{2} \text{tr} \left(C(x) D_\eta \right) \right] + \frac{1}{2} R(x, \tilde{x}) \left(d_y^\top C(x) d_y \right) \\ &\quad - \phi_2(\rho) \left(d_y^\top C(x) D_\eta \mu_y(\tilde{x}) \right) + \phi_3(\rho) \left(d_y^\top C(x) D_\eta C(\tilde{x}) d_y \right) \\ &\quad + \frac{1}{2} \phi_2(\rho) \text{tr} \left(C(x) D_\eta C(\tilde{x}) D_\eta \right)\end{aligned}$$

On the diagonal, evaluating the apparent singularity at $\rho = 0$:

$$\begin{aligned} \mathcal{L}_x(\mathcal{L}_x k) \Big|_{\rho=0} &= -r(x)F(0) - \frac{\phi_1(0)}{\eta_t^2} - \phi_1(0) \left(\mu_y(x)^\top D_\eta \mu_y(x) \right) \\ &\quad + \frac{1}{2} S(0) \operatorname{tr} \left(C(x) D_\eta \right) + \frac{1}{2} \phi_2(0) \operatorname{tr} \left(C(x) D_\eta C(x) D_\eta \right) \end{aligned}$$

Appendix B

Derivation for the Gaussian Kernel

Because the matrix calculus for the Feynman-Kac operator acting on an anisotropic distance metric was fully generalized in Appendix A, we only need to derive the specific radial cascade $\phi_n(\rho)$ for the Gaussian kernel to complete its analytical operator.

B.1 Definition and Derivatives

Using the unified anisotropic distance $\rho(x, \tilde{x})$, the Gaussian kernel is defined as:

$$k(\rho) = a^2 \exp\left(-\frac{1}{2}\rho^2\right)$$

The first derivative with respect to the radial distance ρ evaluates to:

$$k'(\rho) = -a^2 \rho \exp\left(-\frac{1}{2}\rho^2\right) = -\rho k(\rho)$$

By the definition of the first radial function $\phi_1(\rho)$:

$$\phi_1(\rho) = \frac{k'(\rho)}{\rho} = -a^2 \exp\left(-\frac{1}{2}\rho^2\right) = -k(\rho) \quad (\text{B.1})$$

The subsequent radial functions are defined recursively as $\phi_{n+1}(\rho) = \frac{\phi'_n(\rho)}{\rho}$.

For $\phi_2(\rho)$, we differentiate ϕ_1 :

$$\phi'_1(\rho) = \frac{d}{d\rho} \left[-a^2 \exp\left(-\frac{1}{2}\rho^2\right) \right] = a^2 \rho \exp\left(-\frac{1}{2}\rho^2\right) = \rho k(\rho) \quad (\text{B.2})$$

$$\phi_2(\rho) = \frac{\phi_1'(\rho)}{\rho} = a^2 \exp\left(-\frac{1}{2}\rho^2\right) = k(\rho) \quad (\text{B.3})$$

For $\phi_3(\rho)$, we differentiate ϕ_2 :

$$\phi_2'(\rho) = \frac{d}{d\rho} [k(\rho)] = -\rho k(\rho) \quad (\text{B.4})$$

$$\phi_3(\rho) = \frac{\phi_2'(\rho)}{\rho} = -k(\rho) \quad (\text{B.5})$$

For $\phi_4(\rho)$, we differentiate ϕ_3 :

$$\phi_3'(\rho) = \frac{d}{d\rho} [-k(\rho)] = \rho k(\rho) \quad (\text{B.6})$$

$$\phi_4(\rho) = \frac{\phi_3'(\rho)}{\rho} = k(\rho) \quad (\text{B.7})$$

B.2 Final Contracted Form

The radial cascade for the Gaussian kernel strictly alternates signs, maintaining the original kernel structure scaled by the signal variance a^2 :

$$\phi_1(\rho) = -k(\rho), \quad \phi_2(\rho) = k(\rho), \quad \phi_3(\rho) = -k(\rho), \quad \phi_4(\rho) = k(\rho) \quad (\text{B.8})$$

Because $\phi_n(0) = \pm a^2$ is strictly bounded, the apparent singularity at the diagonal ($\rho = 0$) is perfectly resolved. Replacing the Matérn radial functions with this alternating sequence in the final contracted forms of $\mathcal{L}_{\tilde{x}}k$ and $\mathcal{L}_x(\mathcal{L}_{\tilde{x}}k)$ instantly yields the exact analytical structure required for the Gaussian Process Gram matrix.

References

- Agarwal, Praveen, Mohamed Jleli, and Bessem Samet (2018). “Banach Contraction Principle and Applications”. In: *Fixed Point Theory in Metric Spaces: Recent Advances and Applications*. Singapore: Springer Singapore, pp. 1–23. ISBN: 978-981-13-2913-5. DOI: 10.1007/978-981-13-2913-5_1. URL: https://doi.org/10.1007/978-981-13-2913-5_1.
- Bates, David S. (Dec. 1993). *Jumps and Stochastic Volatility: Exchange Rate Processes Implicit in PHLX Deutschemark Options*. Working Paper 4596. National Bureau of Economic Research. URL: <https://ssrn.com/abstract=253132>.
- Chan, Tony F. and David E. Foulser (1988). “Effectively Well-Conditioned Linear Systems”. In: *SIAM Journal on Scientific and Statistical Computing* 9.6, pp. 963–969. DOI: 10.1137/0909067. eprint: <https://doi.org/10.1137/0909067>. URL: <https://doi.org/10.1137/0909067>.
- Cheney, Ward and David Kincaid (2008). *Numerical Mathematics and Computing*. Cengage Learning. ISBN: 978-0-495-11475-8.
- Cont, Rama and Peter Tankov (2004). *Financial Modelling with Jump Processes*. Chapman and Hall/CRC. ISBN: 9781584884132.
- Glasserman, Paul (2003). *Monte Carlo Methods in Financial Engineering*. Vol. 53. Stochastic Modelling and Applied Probability. Springer. ISBN: 978-0-387-00451-8.
- Gulian, Mamikon, Ari Frankel, and Laura Swiler (2020). *Gaussian Process Regression constrained by Boundary Value Problems*. arXiv: 2012.11857 [cs.LG]. URL: <https://arxiv.org/abs/2012.11857>.
- Hainaut, Donatien (2026). “American option pricing with model constrained Gaussian process regressions”. In: *Applied Mathematics and Computation* 512, p. 129747. ISSN: 0096-3003. DOI: <https://doi.org/10.1016/j.amc.2025>.

129747. URL: <https://www.sciencedirect.com/science/article/pii/S0096300325004722>.
- Hainaut, Donatien and Alex Casas (2024). “Option pricing in the Heston model with physics inspired neural networks”. In: *Annals of Finance* 20.3, pp. 353–376. ISSN: 1614-2454. DOI: 10.1007/s10436-024-00452-7. URL: <https://doi.org/10.1007/s10436-024-00452-7>.
 - Hainaut, Donatien and Frederic Daniel Vrins (Oct. 2024). *European Option Pricing with Model Constrained Gaussian Process Regressions*. Working Paper. SSRN. DOI: 10.2139/ssrn.4982344. URL: <https://ssrn.com/abstract=4982344>.
 - Holt, William and Duy Nguyen (June 2023). *Essential Aspects of Bayesian Data Imputation*. Working Paper. SSRN. DOI: 10.2139/ssrn.4494314. URL: <https://ssrn.com/abstract=4494314>.
 - Karatzas, Ioannis and Steven E. Shreve (1998). *Brownian Motion and Stochastic Calculus*. 2nd ed. Vol. 113. Graduate Texts in Mathematics. Springer. ISBN: 978-1-4612-0949-2.
 - Kirk, William A. and Brailey Sims, eds. (2001). *Handbook of Metric Fixed Point Theory*. Dordrecht: Kluwer Academic Publishers. ISBN: 0-7923-7073-2.
 - Rasmussen, Carl Edward and Christopher K. I. Williams (Nov. 2005). *Gaussian Processes for Machine Learning*. The MIT Press. ISBN: 9780262256834. DOI: 10.7551/mitpress/3206.001.0001. eprint: https://direct.mit.edu/book-pdf/2514321/book_9780262256834.pdf. URL: <https://doi.org/10.7551/mitpress/3206.001.0001>.
 - Salzer, Herbert E. and Ruth Zucker (1949). “Table of the zeros and weight factors of the first fifteen Laguerre polynomials”. In: *Bulletin of the American Mathematical Society* 55, pp. 1004–1012. DOI: 10.1090/S0002-9904-1949-09327-8. URL: <https://doi.org/10.1090/S0002-9904-1949-09327-8>.
 - Strang, Gilbert (2016). *Introduction to Linear Algebra*. 5th ed. Wellesley-Cambridge Press. ISBN: 978-0-9802327-7-6.

