A monotone piecewise constant control integration approach for the two-factor uncertain volatility model

Duy-Minh Dang* Hao Zhou †

February 27, 2024

Abstract

Option contracts written on two underlying assets that follow an uncertain volatility model have their worst-case and best-case prices determined by solution to a two-dimensional Hamilton-Jacobi-Bellman (HJB) partial differential equation (PDE) with cross derivatives terms. Existing numerical techniques for multi-dimensional HJB PDEs primarily utilize finite differences and policy iteration to solve the resulting non-linear algebraic equations at each timestep. This "discretize, then optimize" paradigm demands a rather complex local coordinate rotation of computational stencils for a monotonicity-preserving approximation of cross derivative terms.

This paper presents a novel and more streamlined "decompose and integrate, then optimize" approach to tackle the aforementioned HJB PDE. Within each timestep, our strategy employs a piecewise constant control, breaking down the HJB PDE into independent linear two-dimensional PDEs. Using known closed-form expressions for the Fourier transforms of the Green's functions associated with these PDEs, we determine an explicit formula for these functions. Since the Green's functions are non-negative, the solutions to the PDEs, cast as two-dimensional convolution integrals, can be conveniently approximated using a monotone integration method. Such integration methods, including a composite quadrature rule, are generally available in popular programming languages. To further enhance efficiency, we propose an implementation of this monotone integration scheme via Fast Fourier Transforms, exploiting the Toeplitz matrix structure. Optimal solution/control is subsequently obtained by efficiently synthesizing the solutions of the individual PDEs.

The proposed monotone piecewise constant control method is demonstrated to be both ℓ_∞-stable and consistent in the viscosity sense, ensuring its convergence to the viscosity solution of the HJB equation. Numerical results show remarkable agreement with benchmark solutions obtained by unconditionally monotone finite differences, tree methods, and Monte Carlo simulation, underscoring the robustness and effectiveness of our method.

Keywords: Hamilton-Jacobi-Bellman, viscosity solution, uncertain volatility, piecewise constant control, monotone, numerical integration

AMS Subject Classification: 65D30, 65M12, 90C39, 49L25, 93E20, 91G20

1 Introduction

The uncertain volatility model is an approach in quantitative finance where the instantaneous volatility of a risky asset is allowed to vary within a specified range [1, 15, 20]. This stands in contrast to the more traditional approaches where volatility is often assumed to be either deterministic (as in the Black-Scholes model) or stochastic (as in the Heston model [14] or the SABR model [13]). While stochastic volatility models can deliver a more detailed depiction of volatility's dynamic evolution and

*School of Mathematics and Physics, The University of Queensland, St Lucia, Brisbane 4072, Australia, email: duyminh.dang@uq.edu.au

†School of Mathematics and Physics, The University of Queensland, St Lucia, Brisbane 4072, Australia, email: h.zhou3@uq.net.au

its interaction with asset prices, uncertain volatility models are particularly well-suited for worst-case scenario analysis. Specifically, although the price of a financial contract is no longer unique under an uncertain volatility model, for risk management, especially for sellers, the primary concern often lies in the worst-case scenario, which corresponds to the contract's maximum value. Conversely, for the buyers, the worst-case scenario corresponds to the minimum potential value of a contract. It is worth noting that the worst-case scenario for the seller of a contract is essentially the buyer's best-case scenario, and vice versa. The maximum and minimum value of a contract can be formulated as solution to a Hamilton-Jacobi-Bellman (HJB) equation, which needs to be solved numerically [2, 7, 16, 22, 25, 26, 27, 30].

Provable convergence of numerical methods for (multi-dimensional) HJB equations are typically built upon the framework established by Barles and Souganidis in [3]. This framework requires numerical methods to be (i) ℓ_∞-stable, (ii) consistent, and (iii) monotone (in the viscosity sense), provided that a strong comparison result holds. Among these requirements, monotonicity is often the most challenging to achieve. Non-monotone schemes could produce numerical solutions that fail to converge to viscosity solutions, resulting in a violation of the no-arbitrage principle (see, for example, [23, 26, 35], among many other publications).

To the best of our knowledge, numerical techniques for HJB PDEs are predominantly dominated by finite difference (FD) methods. At each timestep, these methods typically involve discretizing the temporal and spatial partial derivatives respectively using (i) a fully implicit timestepping and (ii) a positive coefficient discretization method [9, 34]. This combination ensures the monotonicity of the numerical schemes. The optimal control is subsequently determined by solving the resulting nonlinear discretized equations, often via variants of policy iteration [4]. During this process, a local optimization problem at each grid point is addressed in every policy iteration. This conventional approach is succinctly termed "discretize, then optimize" [27]. Importantly, the positive coefficient discretization method provides a sufficient condition ensuring the convergence of policy iteration, regardless of the initial iterate. As such, this condition must be satisfied at each policy iteration.

We highlight that multi-dimensional HJB PDEs, including those from two-factor uncertain volatility models [22], pose significant challenges due to cross derivative terms when the correlation between the two underlying risky assets is non-zero. At each policy iteration, construction of a monotone finite difference scheme via a positive coefficient discretization method is often addressed using a local coordinate rotation of the computational stencil. Originally developed for explicit wide stencil schemes in [5, 6], this method was refined in [22] for a fully implicit timestepping, circumventing timestep stability restrictions. However, as noted in [22], this approach adds a significant computational overhead. For further details of numerical techniques for one- and two-dimensional HJB PDEs resulting from one-factor and two-factor uncertain volatility models (with an uncertain correlation between the two underlying risky assets), we refer the reader to [26] and [22], respectively.

In this paper, we present a streamlined approach to tackle the two-dimensional (2D) HJB PDE stemming from two-factor uncertain volatility models. Moving beyond the conventional "discretize, then optimize", we introduce a "decompose and integrate, then optimize" approach. For each timestep, we utilize a piecewise constant control technique [18] to decompose the HJB PDE into independent 2D linear PDEs. By employing closed-form expressions of the Green's functions, we develop an unconditionally monotone integration method to solve these PDEs. Optimization is then efficiently attained by synthesizing solutions of individual PDEs, significantly simplifying the process compared to policy iteration and avoiding aforementioned challenges associated with positive coefficient FD discretization of cross derivative terms. We note a recent study [27] that also utilizes a piecewise constant control technique. However, this work remains anchored in the FD framework and incorporates a switching system, and hence necessitating interpolation in searching for optimal control.

The main contributions of our paper are outlined below.

(i) The maximum and minimum value of an option contract under a two-factor uncertain volatility model with uncertain correlation is presented as an HJB PDE posed on an finite definition domain consisting of an interior and boundary sub-domains with appropriate boundary conditions.

(ii) We develop a monotone piecewise constant control integration scheme for the HJB equation, addressing it through the solution of independent linear 2D PDEs at each timestep. By utilizing the known closed-form Fourier transforms of the Green's functions for these PDEs, we establish explicit formulas for these functions, facilitating direct approximation of solutions via 2D convolution integrals through a monotone integration method. This process enables the effective synthesis of approximations to the HJB equation from the numerical solutions of the individual PDEs.

Our scheme not only simplifies the optimization process compared to policy iteration but also avoids the usual complications with positive coefficient FD discretization of cross derivative terms. The availability of the Green's functions in closed form enables a systematic and quantifiable approach for determining computational domain sizes, marking a significant advantage over the heuristic or trial-and-error methods common in FD and tree techniques. Furthermore, the Green's function's "cancellation property" [10] effectively mitigates the impact of errors in artificial boundary conditions. These combined factors ensure that our method significantly enhances the numerical solution's accuracy and reliability.

(iii Utilizing the Toeplitz matrix structure, we present an efficient implementation of our monotone piecewise constant control integration scheme using FFTs and circulant convolution. The implementation process includes expanding the inner summation's convolution kernel into a circulant matrix, followed by expanding the kernel for the double summation to achieve a circulant block arrangement. This allows the circulant matrix-vector product to be efficiently computed as a circulant convolution using 2D FFTs.

(iv We mathematically demonstrate that the proposed monotone scheme is also ℓ_∞-stable and consistent in the viscosity sense, proving its pointwise convergence to the viscosity solution of the 2D HJB PDE as the discretization parameter approaches zero.

(v) Extensive numerical results show remarkable agreement with benchmark solutions in published test cases obtained through monotone FD and tree-grid methods, and Monte Carlo simulation, underscoring the effectiveness of our approach. Notably, we often observe experimentally first order convergence, significantly exceeding the convergence rate 1/6 proved in [18] using purely probabilistic techniques.

Although our focus is specifically on monotone piecewise constant control integration methods for two-factor uncertain volatility models with uncertain correlation, our comprehensive and systematic approach could serve as a numerical and convergence analysis framework for the development of similar piecewise constant control monotone integration methods for other HJB PDEs arising in finance.

The remainder of the paper is organized as follows. In Section 2, we briefly describe the two-factor uncertain volatility model and present a 2D HJB PDE. We then define a localized problem for this HJB equation, including conditions for boundary sub-domains. A simple and easy-to-implement monotone piecewise constant control integration scheme via a composite 2D quadrature rule is described in Section 3. In Section 4, we mathematically establish convergence the proposed piecewise constant control monotone integration scheme to the viscosity solution of the 2D HJB PDE. Numerical results are given in Section 5. Section 6 concludes the paper and outlines possible future work.

2 Formulation

Let $T > 0$ be a finite investment horizon. For each $t \in [0, T]$, we denote by X_t and Y_t the prices at time t of two distinct underlying assets. In this paper, for brevity, we occasionally employ the subscript/superscript $z \in \{x, y\}$ to indicate that the discussion pertains to quantities related to the respective underlying assets. We assume that the risk-neutral dynamics of the process $\{Z_t\}_{t\in[0,T]}$, where Z_t can be either X_t or Y_t, follow

$$dZ_t = rZ_t dt + \sigma_z Z_t dW_t^z, \quad Z_0 > 0 \text{ given}, \quad Z_t \in \{X_t, Y_t\}, \quad t \in (0, T]. \tag{2.1}$$

Here, $r > 0$ is the risk-free interest rate; $\sigma_z > 0$, $z \in \{x, y\}$, respectively are the instantaneous volatility for the associated underlying asset; $\{W_t^z\}_{t\in[0,T]}$ are correlated Brownian motions, with $dW_t^x dW_t^y = \rho dt$, where $-1 \le \rho \le 1$ is the correlation parameter. In the uncertain volatility model, the instantaneous volatility σ_z, $z \in \{x, y\}$, in (2.1) are uncertain, but are assumed to lie within a known range [21]. That is, $\sigma_z \in [\sigma_{\min}^z, \sigma_{\max}^z]$, $z \in \{x, y\}$, where $0 < \sigma_{\min}^z < \sigma_{\max}^z$ are pre-determined and fixed constants. In addition, the correlation between the two underlying assets is also permitted to be uncertain, lying within a known range, i.e. $\rho \in [\rho_{\min}, \rho_{\max}]$, where $-1 \le \rho_{\min} \le \rho_{\max} \le 1$ are also pre-determined and fixed constants. In this setting, since the instantaneous volatilities σ_z, $z \in \{x, y\}$ and the correlation ρ are uncertain, the price of an option is no longer unique. However, for hedging purposes, we can determine the worst-case prices for the long or short positions. These prices are essentially the hedging costs for the associated positions.

For the underlying asset processes $\{X_t, Y_t\}$, $t \in [0, T]$, defined in (2.1), we let (x', y') be the state of system. We denote by $v'(x', y', t)$ the time-t worst-case price of the short or long position in a European option contract with time-T payoff given by function $p(x', y')$. By dynamic programming, $v'(x', y', t)$ is shown to satisfy the HJB PDEs

$$0 = \begin{cases} \big(-v'_t - \sup\limits_{\alpha\in\mathcal{A}'} \mathcal{L}'_\alpha v' \big), \text{ or } \big(-v'_t - \inf\limits_{\alpha\in\mathcal{A}'} \mathcal{L}'_\alpha v' \big), & (x', y', t) \in \mathbb{R}^+ \times \mathbb{R}^+ \times [0, T), \quad \text{(2.2a)} \\ v'(x', y', t) - p(x', y'), & (x', y', t) \in \mathbb{R}^+ \times \mathbb{R}^+ \times \{T\}. \quad \text{(2.2b)} \end{cases}$$

In (2.2), the $\sup_\alpha$ and $\inf_\alpha$ correspond to the worst-case for the short and for the long positions, respectively; α is the control, where $\alpha = (\sigma_x, \sigma_y, \rho)$; the differential operator $\mathcal{L}'_\alpha(\cdot)$, where the subscript indicates its dependence α, is defined as

$$\mathcal{L}'_\alpha v' = \frac{(\sigma_x)^2 (x')^2}{2} v'_{x'x'} + rx' v'_{x'} + \frac{(\sigma_y)^2 (y')^2}{2} v'_{y'y'} + ry' v'_{y'} + \rho\sigma_x\sigma_y x' y' v'_{x'y'} - rv'. \tag{2.3}$$

The admissible control set, denoted by $\mathcal{A}'$, is given by

$$\mathcal{A}' = \mathcal{A}_x \times \mathcal{A}_y \times \mathcal{A}_\rho, \text{ where } \mathcal{A}_x \equiv [\sigma_{\min}^x, \sigma_{\max}^x], \ \mathcal{A}_y \equiv [\sigma_{\min}^y, \sigma_{\max}^y], \ \mathcal{A}_\rho \equiv [\rho_{\min}, \rho_{\max}], \tag{2.4}$$
$$0 < \sigma_{\min}^z < \sigma_{\max}^z < \infty, \ z \in \{x, y\}, \ -1 \le \rho_{\min} \le \rho_{\max} \le 1.$$

Remark 2.1 (Restriction of control set $\mathcal{A}'$)**.** *The literature highlights that the optimal value for the objective function in* (2.2) *can be accurately determined by considering only the boundary values within the 3D admissible optimal control set $\mathcal{A}'$. See, for example, [22][Proposition 3.1] and [17]. Specifically, it is established that the search for optimal control can be limited to a much smaller set $\mathcal{A}$, defined as:*

$$\mathcal{A} = \big((\{\sigma_{\min}^x, \sigma_{\max}^x\} \times \mathcal{A}_y) \cup (\mathcal{A}_x \times \{\sigma_{\min}^y, \sigma_{\max}^y\}) \big) \times \{\rho_{\min}, \rho_{\max}\}, \tag{2.5}$$

where $\mathcal{A}_x$ and $\mathcal{A}_y$ are defined in (2.4)*. Consequently, we focus our analysis on the boundary set $\mathcal{A}$, enhancing the efficiency of the proposed piecewise constant control scheme by eliminating the need to search across the entire 3D set $\mathcal{A}'$.*

The aforementioned restriction presumes the existence of second-order partial derivatives, which, despite appearing restrictive, is consistent with the viscosity solution framework that utilizes smooth

test functions. Unlike traditional grid-based methods such as FD and tree-grid [17, 22], which require discretizing the differential operator and may not always yield optimal values at $\mathcal{A}$, our approach bypasses differential operator discretization. This not only resolves related issues but also simplifies the optimization process, offering a more direct and user-friendly path to identifying optimal control values, highlighting the method's practicality and ease of implementation.

Let $\tau = T - t$, and we apply the change of variables $x = \ln(x') \in (-\infty, \infty)$ and $y = \ln(y') \in (-\infty, \infty)$. Let $\mathbf{x} = (x, y, \tau)$, and denote by $v(\mathbf{x}) \equiv v(x, y, \tau) = v'(e^x, e^y, T - t)$. With these in mind, formulation (2.2) becomes

$$0 = \begin{cases} \left(v_\tau - \sup\limits_{\alpha \in \mathcal{A}} \mathcal{L}_\alpha v\right), \text{ or } \left(v_\tau - \inf\limits_{\alpha \in \mathcal{A}} \mathcal{L}_\alpha v\right), & \mathbf{x} \in \mathbb{R} \times \mathbb{R} \times (0, T], \quad (2.6a) \\ v(\mathbf{x}) - p\left(e^x, e^y\right), & \mathbf{x} \in \mathbb{R} \times \mathbb{R} \times \{0\}, \quad (2.6b) \end{cases}$$

where $(x, y, \tau) \in \mathbb{R} \times \mathbb{R} \times [0, T]$ and the differential operator $\mathcal{L}_\alpha(\cdot)$ is given by

$$\mathcal{L}_\alpha v = \frac{(\sigma_x)^2}{2} v_{xx} + \left(r - \frac{(\sigma_x)^2}{2}\right) v_x + \frac{(\sigma_y)^2}{2} v_{yy} + \left(r - \frac{(\sigma_y)^2}{2}\right) v_y + \rho \sigma_x \sigma_y v_{xy} - rv. \quad (2.7)$$

Without loss of generality, we only consider the $\sup_\alpha$ problem, i.e. worst-case for the short position, in the following discussion. The theoretical analysis of this paper holds for the $\inf_\alpha$ problem as well.

2.1 Localization and definition

For the problem statement and convergence analysis of numerical schemes, we define a localized two-factor uncertain volatility model pricing problem.

To this end, with $x^\dagger_{\min} < x_{\min} < 0 < x_{\max} < x^\dagger_{\max}$, $y^\dagger_{\min} < y_{\min} < 0 < y_{\max} < y^\dagger_{\max}$, where $|x^\dagger_{\min}|$, $|x_{\min}|$, $|y^\dagger_{\min}|$,$|y_{\min}|$, $x_{\max}$, $x^\dagger_{\max}$, $y_{\max}$ and $y^\dagger_{\max}$ are chosen sufficiently large, we define the following sub-domains:

$$\begin{aligned} \Omega &= [x^\dagger_{\min}, x^\dagger_{\max}] \times [y^\dagger_{\min}, y^\dagger_{\max}] \times [0, T], \\ \Omega_{\tau_0} &= [x^\dagger_{\min}, x^\dagger_{\max}] \times [y^\dagger_{\min}, y^\dagger_{\max}] \times \{0\}, \\ \Omega_{\text{in}} &= (x_{\min}, x_{\max}) \times (y_{\min}, y_{\max}) \times (0, T], \\ \Omega_{\text{out}} &= \Omega \setminus \Omega_{\tau_0} \setminus \Omega_{\text{in}}. \end{aligned} \quad (2.8)$$

An illustration of the sub-domains for the localized problem corresponding to a fixed $\tau \in (0, T]$ is given in Figure 2.1.

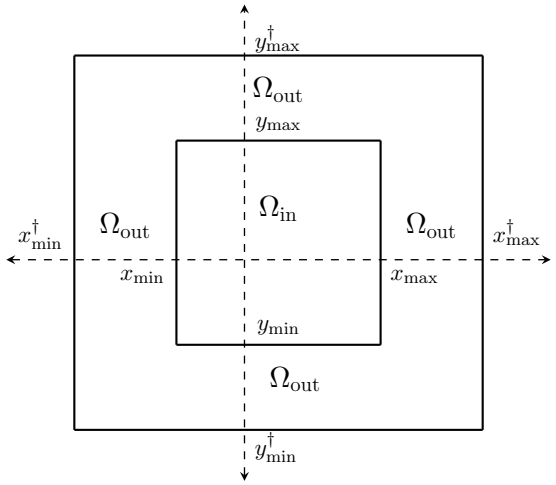

FIGURE 2.1: *Spatial definition sub-domain at each τ.*

We now present equations for sub-domains defined in (2.8).

- For $(x, y, \tau) \in \Omega_{\text{in}}$, we have (2.6).
- For $(x, y, \tau) \in \Omega_{\tau_0}$, we use the initial condition $v(x, y, 0) = p(e^x, e^y)$.
- For the outer boundary sub-domain Ω_{out}, boundary conditions are generally informed by financial reasonings or derived from the asymptotic behavior of the solution. In this study, we implement a straightforward Dirichlet condition based on discounted payoff as follows

$$v(x, y, \tau) = p(e^x, e^y) e^{-r\tau}, \quad (x, y, \tau) \in \Omega_{\text{out}}. \quad (2.9)$$

While more sophisticated boundary conditions might involve the asymptotic properties of the HJB equation (2.6)) as $z \to -\infty$ or $z \to \infty$, where $z \in \{x, y\}$, our observations indicate that these sophisticated boundary conditions do not significantly impact the accuracy of the numerical solution within Ω_{in}. This observation is largely due to the so called "cancellation property" of the Green's function [10], which effectively mitigates the impact of approximation errors in artificial boundary condition behavior on the solution in Ω_{in}. This will be illustrated through numerical experiments in Subsection 5.4.

With $\mathbf{x} = (x, y, \tau)$, we let $Dv(\mathbf{x}) = (v_x, v_y, v_\tau)$ and $D^2v(\mathbf{x}) = (v_{xx}, v_{yy}, v_{xy})$, and define

$$F_\Omega(\mathbf{x}, v) \equiv F_\Omega\left(\mathbf{x}, v(\mathbf{x}), Dv(\mathbf{x}), D^2v(\mathbf{x})\right). \tag{2.10}$$

Here,

$$F_\Omega(\mathbf{x}, v) = \begin{cases} F_{\text{in}}(\mathbf{x}, v) \equiv F_{\text{in}}\left(\mathbf{x}, v(\mathbf{x}), Dv(\mathbf{x}), D^2v(\mathbf{x})\right), & \mathbf{x} \in \Omega_{\text{in}}, \\ F_{\text{out}}(\mathbf{x}, v) \equiv F_{\text{out}}(\mathbf{x}, v(\mathbf{x})), & \mathbf{x} \in \Omega_{\text{out}}, \\ F_{\tau_0}(\mathbf{x}, v) \equiv F_{\tau_0}(\mathbf{x}, v(\mathbf{x})), & \mathbf{x} \in \Omega_{\tau_0}, \end{cases} \tag{2.11}$$

with operators

$$F_{\text{in}}(\mathbf{x}, v) = v_\tau - \sup_{\alpha \in \mathcal{A}} \mathcal{L}_\alpha v, \tag{2.12}$$

$$F_{\text{out}}(\mathbf{x}, v) = v - p(e^x, e^y)e^{-r\tau}, \tag{2.13}$$

$$F_{\tau_0}(\mathbf{x}, v) = v - p(e^x, e^y). \tag{2.14}$$

Definition 2.1 (Two-factor uncertain volatility pricing problem)**.** *The pricing problem for the two-factor uncertain volatility model is defined as*

$$F_\Omega\left(\mathbf{x}, v(\mathbf{x}), Dv(\mathbf{x}), D^2v(\mathbf{x})\right) = 0, \tag{2.15}$$

where the operator $F_\Omega(\cdot)$ is defined in (2.10)*.*

We recall the notions of the upper semicontinuous (u.s.c. in short) and the lower semicontinuous (l.s.c. in short) envelops of a function $u : \mathbb{X} \to \mathbb{R}$, where $\mathbb{X}$ is a closed subset of $\mathbb{R}^n$. They are respectively denoted by $u^*(\cdot)$ (for the u.s.c. envelop) and $u_*(\cdot)$ (for the l.s.c. envelop), and are given by

$$u^*(\hat{\mathbf{x}}) = \limsup_{\substack{\mathbf{x} \to \hat{\mathbf{x}} \\ \mathbf{x}, \hat{\mathbf{x}} \in \mathbb{X}}} u(\mathbf{x}) \quad (\text{resp.} \quad u_*(\hat{\mathbf{x}}) = \liminf_{\substack{\mathbf{x} \to \hat{\mathbf{x}} \\ \mathbf{x}, \hat{\mathbf{x}} \in \mathbb{X}}} u(\mathbf{x})). \tag{2.16}$$

Definition 2.2 (Viscosity solution of equation (2.15))**.** *A locally bounded function $v : \Omega \to \mathbb{R}$ is a viscosity subsolution (resp. supersolution) of* (2.15) *if for all test function $\phi \in \mathcal{C}^\infty(\Omega)$ and for all points $\hat{\mathbf{x}} \in \Omega$ such that $v^* - \phi$ has a* global *maximum on Ω at $\hat{\mathbf{x}}$ and $v^*(\hat{\mathbf{x}}) = \phi(\hat{\mathbf{x}})$ (resp. $v_* - \phi$ has a* global *minimum on Ω^∞ at $\hat{\mathbf{x}}$ and $v_*(\hat{\mathbf{x}}) = \phi(\hat{\mathbf{x}})$), we have*

$$\begin{aligned} &(F_\Omega)_*\left(\hat{\mathbf{x}}, \phi(\hat{\mathbf{x}}), D\phi(\hat{\mathbf{x}}), D^2\phi(\hat{\mathbf{x}})\right) \le 0, \\ (\textit{resp.} \quad &(F_\Omega)^*\left(\hat{\mathbf{x}}, \phi(\hat{\mathbf{x}}), D\phi(\hat{\mathbf{x}}), D^2\phi(\hat{\mathbf{x}})\right) \ge 0,) \end{aligned} \tag{2.17}$$

where the operator $F_\Omega(\cdot)$ is defined in (2.10)*.*

Remark 2.2 (Strong comparison result and convergence region)**.** *As noted in [22], if the payoff function $p(e^x, e^y)$ is continuous with quadratic growth, then the value function of the problem* (2.6) *satisfies a strong comparison principle result [12, 24]. That means, there exists an unique continuous viscosity solution of the problem* (2.6)*. In the context of this paper, our focus is on a finite interior subdomain Ω_{in} (with boundary conditions given in Ω_{out} and Ω_{τ_0}) for the HJB equation* (2.15)*. Therefore, the value function* (2.15) *satisfies a strong comparison result in Ω_{in}. This sub-domain defines the convergence region for our proposed scheme.*

3 Numerical methods

3.1 Piecewise constant control

The key step of our numerical scheme is a piecewise constant control time-stepping method for Ω_{in} that utilizes a convolution integral that involves the Green's function of an associated 2D PDE in x and y. As discussed in Remark 2.1, we search for the optimal control within the boundary set $\mathcal{A}$ given in (2.5). To this end, we first make an observation that the admissible control set $\mathcal{A}$, as defined in (2.4)), is a compact set. Therefore, it can be approximated arbitrarily well by a finite set [28]. Specifically, for any discretization parameter $h > 0$, there exists a finite partition $\mathcal{A}_h$ of $\mathcal{A}$ such that for any $\alpha \in \mathcal{A}$, the distance to its nearest point in $\mathcal{A}_h$ is no greater than h. That is,

$$\max_{\alpha\in\mathcal{A}} \min_{\alpha'\in\mathcal{A}_h} \|\alpha - \alpha'\|_2 \leq h. \tag{3.1}$$

Motivated by (3.1), to address the two-factor uncertain volatility pricing problem in Defn (2.1), we propose an approach that involves approximating $\mathcal{A}$ with $\mathcal{A}_h$. Specifically, for Ω_{in}, instead of solving the HJB equation $v_\tau - \sup_{\alpha\in\mathcal{A}} \mathcal{L}_\alpha v = 0$, we solve $v_\tau - \sup_{\alpha\in\mathcal{A}_h} \mathcal{L}_\alpha v = 0$. In our convergence analysis, we will establish that this solution converges to the viscosity solution of the pricing problem in Defn (2.1) as described in Defn (2.2).

We now elaborate the piecewise constant control for Ω_{in}. We let $\{\tau_m\}$, $m = 0, \ldots, M$, be an equally spaced partition in the τ-dimension, where $\tau_m = m\Delta\tau$ and $\Delta\tau = T/M$. With a fixed $\tau_m > 0$ such that $\tau_{m+1} \leq T$, we consider the HJB equation

$$v_\tau - \sup_{\alpha\in\mathcal{A}_h} \mathcal{L}_\alpha v = 0, \qquad (x, y, \tau) \in \mathbb{R} \times \mathbb{R} \times (\tau_m, \tau_{m+1}], \tag{3.2}$$

where the differential operator $\mathcal{L}_\alpha(\cdot)$ is defined in (2.7). Here, we note that, in (3.2), the admissible control set $\mathcal{A}$ is approximated by the finite discretized control set $\mathcal{A}_h$, with $h > 0$ being the discretization parameter.

For fixed $\alpha \in \mathcal{A}_h$, we denote by $u(\cdot;\alpha) \equiv u(x, y, \tau; \alpha)$ the solution to the linear PDE in (x, y, τ) of the form

$$u_\tau - \mathcal{L}_\alpha u = 0, \qquad (x, y, \tau) \in \mathbb{R} \times \mathbb{R} \times (\tau_m, \tau_{m+1}]. \tag{3.3}$$

where $\mathcal{L}_\alpha(\cdot)$ is defined in (2.7), subject to a generic initial condition at time τ_m given by $\hat{v}(x, y, \tau_m)$. Here,

$$\hat{v}(x, y, \tau_m) = \begin{cases} v(x, y, \tau_m) & (x, y, \tau_{m+1}) \in \Omega_{\text{in}}, \\ v_{bc}(x, y, \tau_m) & (x, y, \tau_{m+1}) \in \Omega \setminus \Omega_{\text{in}}, \end{cases} \tag{3.4}$$

where $v_{bc}(x, y, \tau_m)$ is the boundary conditions at time τ_m satisfying (2.9) in Ω_{out}. respectively.

We denote by $g_\alpha(\cdot) \equiv g_\alpha(x, x', y, y'; \Delta\tau)$, $\alpha \in \mathcal{A}_h$, the Green's function associated with the 2D linear PDE (3.3). It can be shown that $g_\alpha(\cdot)$ has the form $g_\alpha(\cdot; \Delta\tau) = g_\alpha(x - x', y - y'; \Delta\tau)$.[1] By a Green's function argument [8, 10], for fixed $\alpha \in \mathcal{A}_h$, the solution $u(x, y, \tau_{m+1})$ for $(x, y) \in \mathbf{D}$, where

$$\mathbf{D} \equiv (x_{\min}, x_{\max}) \times (y_{\min}, y_{\max}),$$

can be represented as the convolution integral of the Green's function $g_\alpha(\cdot; \Delta\tau)$ and the initial condition $\hat{v}(\cdot, \tau_m)$ as follows

$$u(x, y, \tau_{m+1}; \alpha) = \iint_{\mathbb{R}^2} g_\alpha \left(x - x', y - y'; \Delta\tau\right) \hat{v}(x', y', \tau_m) dx' dy', \qquad (x, y) \in \mathbf{D}, \quad \alpha \in \mathcal{A}_h. \tag{3.5}$$

The solution $u(x, y, \tau_{m+1}; \alpha)$ for $(x, y) \notin \mathbf{D}$ are given by the boundary condition (2.9).

For computational purposes, we truncate the infinite region of integration of (3.5) to $\mathbf{D}^\dagger$, where

$$\mathbf{D}^\dagger \equiv [x^\dagger_{\min}, x^\dagger_{\max}] \times [y^\dagger_{\min}, y^\dagger_{\max}]. \tag{3.6}$$

[1] This is due to the spatial homogeneity of the stochastic system (2.1).

Here, recall that $z \in \{x, y\}$, $z^\dagger_{\min} < z_{\min} < 0 < z_{\max} < z^\dagger_{\max}$ and $|z^\dagger_{\min}|$ and $z^\dagger_{\max}$ are sufficiently large. This results in the approximation

$$u(x, y, \tau_{m+1}; \alpha) \simeq \iint_{\mathbf{D}^\dagger} g_\alpha \left(x - x', y - y'; \Delta\tau\right) \hat{v}(x', y', \tau_m) dx' dy', \qquad (x, y) \in \mathbf{D}, \quad \alpha \in \mathcal{A}_h. \tag{3.7}$$

Finally, an approximation to the solution of the HJB (3.2) for $(x, y, \tau_{m+1}) \in \Omega_{\text{in}}$ is given by

$$v(x, y, \tau_{m+1}) \simeq \max_{\alpha \in \mathcal{A}_h} u(x, y, \tau_{m+1}; \alpha), \qquad (x, y) \in \mathbf{D}. \tag{3.8}$$

We conclude by noting that the errors arising from (i) approximating $\mathcal{A}$ by $\mathcal{A}_h$ and (ii) from truncating the infinite integration domain in (3.5) to a finite one in (3.7) are discussed subsequently.

3.2 A closed-form representation of $g_\alpha(\cdot)$ for Ω_{in}

We now present a closed-form expression for the Green's function $g_\alpha(\cdot)$ of the linear PDE (3.3), where the control $\alpha \equiv (\sigma_x, \sigma_y, \rho) \in \mathcal{A}$ is fixed. To this end, we denote by $G_\alpha(\cdot; \Delta\tau)$ the Fourier transform of $g_\alpha(\cdot; \Delta\tau)$, i.e.

$$\begin{cases} \mathscr{F}|g_\alpha(x, y; \cdot)| & = G_\alpha(\eta, \zeta; \cdot) \quad = \iint_{\mathbb{R}^2} e^{-i(\eta x + \zeta y)} g_\alpha(x, y; \cdot) dx dy, \\ \mathscr{F}^{-1}|G_\alpha(\eta, \zeta; \cdot)| & = g_\alpha(x, y; \cdot) \quad = \frac{1}{(2\pi)^2} \iint_{\mathbb{R}^2} e^{i(\eta x + \zeta y)} G_\alpha(\eta, \zeta; \cdot) d\eta d\zeta. \end{cases} \tag{3.9}$$

A closed-form expression for $G_\alpha(\eta, \zeta; \cdot)$ is given by [29]

$$\begin{aligned} G_\alpha(\eta, \zeta; \cdot) &= \exp(\Psi(\eta, \zeta)\Delta\tau), \\ \text{with } \Psi(\eta, \zeta) &= \left(-\frac{\sigma_x^2 \eta^2}{2} - \frac{\sigma_y^2 \zeta^2}{2} + (r - \frac{\sigma_x^2}{2}) i\eta + (r - \frac{\sigma_y^2}{2}) i\zeta - \rho\sigma_x\sigma_y\eta\zeta - r \right). \end{aligned} \tag{3.10}$$

We now introduce a lemma providing a closed-form expression for the Green's function $g_\alpha(x, y; \Delta\tau)$.

Lemma 3.1. *Let $\Delta\tau > 0$ be fixed, and $g_\alpha(x, y; \Delta\tau)$ and $G_\alpha(\eta, \zeta; \Delta\tau)$ be a Fourier transform pair defined in (3.9), and $G_\alpha(\eta, \zeta; \Delta\tau)$ be given in (3.10). When $|\rho| < 1$, $g_\alpha(x, y; \Delta\tau)$ can be expressed in the form of a "scaled" joint density as follows*

$$g_\alpha(x, y; \Delta\tau) = e^{-r\Delta\tau} f_\alpha(x, y; \Delta\tau), \ \textit{where} \tag{3.11}$$

$$f_\alpha(x, y; \Delta\tau) = \frac{1}{2\pi\kappa_x\kappa_y\sqrt{1-\rho^2}} \exp\left(\frac{-1}{2(1-\rho^2)} \left[(\frac{x-\mu_x}{\kappa_x})^2 - 2\rho(\frac{x-\mu_x}{\kappa_x})(\frac{y-\mu_y}{\kappa_y}) + (\frac{y-\mu_y}{\kappa_y})^2 \right] \right).$$

$$\textit{with} \quad \mu_x = \left(\frac{\sigma_x^2}{2} - r\right)\Delta\tau, \ \kappa_x = \sigma_x\sqrt{\Delta\tau}, \ \mu_y = \left(\frac{\sigma_y^2}{2} - r\right)\Delta\tau, \ \kappa_y = \sigma_y\sqrt{\Delta\tau}, \tag{3.12}$$

When $\rho = \pm 1$, $g_\alpha(x, y; \Delta\tau)$ is given by

$$g_\alpha(x, y; \Delta\tau) = e^{-r\Delta\tau} \frac{1}{\sqrt{2\pi}\kappa_x} \exp\left(-\frac{(x-\mu_x)^2}{2\kappa_x^2} \right) \delta(y - (a + \rho b x)). \tag{3.13}$$

Here, $\delta(\cdot)$ is a Dirac delta function, and $a = \mu_y - \rho b \mu_x$ with $b = \frac{\sigma_y}{\sigma_x}$.

Proof of Lemma 3.1. When $|\rho| < 1$, applying inverse Fourier transform to $G_\alpha(\cdot)$, provided in (3.10), we obtain the expression for the Green's function $g_\alpha(x, y; \Delta\tau)$ given in (3.11). When $\rho = \pm 1$, $f_\alpha(x, y; \Delta\tau)$ can be expressed in the form $f_\alpha(x, y; \Delta\tau) = \frac{1}{\sqrt{2\pi}\kappa_x} \exp\left(-\frac{(x-\mu_x)^2}{2\kappa_x^2}\right) \delta(y - (a + \rho b x))$, where a and b are constants, with $b > 0$ [11]. We then solve for a and b by comparing the Fourier transform of $g_\alpha(x, y; \Delta\tau)$ in this case with the closed-form expression of $G_\alpha(\cdot)$. This gives $a = \mu_y - \rho b \mu_x$ and $b = \frac{\sigma_y}{\sigma_x}$. This completes the proof. □

Remark 3.1 ($\rho = \pm 1$)**.** *In our study, while we acknowledge the theoretical significance of the cases where $\rho = \pm 1$, we have chosen not to explore this scenario in depth. Such extreme correlation values, though mathematically interesting, are rarely encountered in practical applications and financial modeling. Therefore, our focus remains predominantly on scenarios where the correlation coefficient lies strictly between -1 and 1, which are more representative of the conditions commonly observed and of greater relevance to practitioners. However, it's important to note that our piecewise constant control integration scheme can effectively manage the special case of $\rho = \pm 1$. For computational purposes, approximating the Dirac delta function $\delta(y - (a \pm bx))$ in (3.13) by a suitable Gaussian function is necessary (refer to [31][Chapter 10], for example, for more details of such approximations). Essential aspects of our scheme in this case are elaborated in Appendix A.*

We now present a lemma on the boundary truncation error of the Green's function $g_\alpha(\cdot)$ defined in (3.11). for the case $|\rho| < 1$.

Lemma 3.2. *Let $\Delta\tau > 0$ be fixed. Suppose $|x^\dagger_{\min}|$, $x^\dagger_{\max}$, $|y^\dagger_{\min}|$, and $y^\dagger_{\max}$ are chosen such that*

$$\min\left\{|x^\dagger_{\min}|,\ x^\dagger_{\max},\ |y^\dagger_{\min}|,\ y^\dagger_{\max}\right\} > \max\left\{\mu_x \pm \gamma, \mu_y \pm \gamma\right\},$$

where $\gamma \gg 0$ is a fixed constant, $\mu_x = \left(\frac{\sigma_x^2}{2} - r\right)\Delta\tau$, and $\mu_y = \left(\frac{\sigma_y^2}{2} - r\right)\Delta\tau$, as defined in (3.12). Then, for sufficiently small $\Delta\tau$, $g_\alpha(\cdot)$, as defined in (3.11) for a fixed $\alpha \in \mathcal{A}$, satisfies

$$\iint_{\mathbb{R}^2\setminus\mathbf{D}^\dagger} g_\alpha\left(x, y; \Delta\tau\right) dxdy < C\Delta\tau e^{-\frac{1}{2\Delta\tau}}, \quad \mathbf{D}^\dagger \equiv [x^\dagger_{\min}, x^\dagger_{\max}] \times [y^\dagger_{\min}, y^\dagger_{\max}], \tag{3.14}$$

where C is a bounded constant independently of $\Delta\tau$.

Proof of Lemma 3.2. Without loss of generality, we present a proof for the case $0 \le \rho < 1$. For subsequent use, let $\Phi(s) \equiv \int_{-\infty}^{s} \phi(z)dz$ and $\phi(z) \equiv (2\pi)^{-1/2}\exp(-z^2/2)$ respectively be the CDF and the probability density function of standard normal distribution.

For simplicity, we let $w = \min\left\{|x^\dagger_{\min}|,\ x^\dagger_{\max},\ |y^\dagger_{\min}|,\ y^\dagger_{\max}\right\}$. With $z_x = \frac{x-\mu_x}{\kappa_x}$, $z_y = \frac{y-\mu_y}{\kappa_y}$, we define the region $\mathbf{B}$ as follows

$$\mathbf{B} = [-b, b] \times [-b, b], \quad \text{where } b = \min\left\{\Big|\frac{-w-\mu_x}{\kappa_x}\Big|, \frac{w-\mu_x}{\kappa_x}, \Big|\frac{-w-\mu_y}{\kappa_y}\Big|, \frac{w-\mu_y}{\kappa_y}\right\}. \tag{3.15}$$

We have

$$\begin{aligned}
&\iint_{\mathbb{R}^2\setminus\mathbf{D}^\dagger} g_\alpha\left(x, y; \Delta\tau\right) dxdy \le e^{-r\Delta\tau} \iint_{\mathbb{R}^2\setminus\mathbf{B}} \frac{\exp\left(-\frac{1}{2[1-\rho^2]}\left[z_x^2 - 2\rho z_x z_y + z_y^2\right]\right)}{2\pi\sqrt{1-\rho^2}} dz_x dz_y \\
&\le 2e^{-r\Delta\tau} P\big(z_x \ge b,\ z_y \ge b\big) \overset{\text{(i)}}{\le} 2e^{-r\Delta\tau}(1+\rho)\Phi(-b)\Phi\left(\frac{-b(1-\rho)}{\sqrt{1-\rho^2}}\right) \\
&\overset{\text{(ii)}}{\le} \frac{e^{-r\Delta\tau}(1+\rho)^{3/2}}{\pi(1-\rho)^{1/2}} \times \frac{e^{-b^2/2}}{b^2}.
\end{aligned} \tag{3.16}$$

Here, (i) is due to an upper bound for the bivariate normal distribution in [36]; in (ii), we apply the following fact: if $X \sim N(0,1)$, then $P(X > x) \le \frac{1}{x\sqrt{2\pi}}\exp(-x^2/2)$. It is straightforward to see that $\frac{e^{-r\Delta\tau}(1+\rho)^{3/2}}{\pi(1-\rho)^{1/2}} \le \frac{(1+\rho_{\max})^{3/2}}{\pi(1-\rho_{\max})^{1/2}}$. Thus, for sufficiently small $\Delta\tau$, the condition $w > \max\left\{\mu_x \pm \gamma, \mu_y \pm \gamma\right\}$, where $\gamma \gg 0$ is fixed, implies the rhs of (3.16) is bounded by $C\Delta\tau e^{-1/2\Delta\tau}$, where C is a bounded constant independently of $\Delta\tau$. This completes the proof. □

Remark 3.2 (Boundary truncation error)**.** *The boundary truncation error upper bound, as detailed in equation (3.16), serves as a practical tool for selecting an appropriate definition domain, $\mathbf{D}^\dagger$, to ensure*

this truncation error remains below a predefined threshold $\epsilon > 0$. To achieve this, we first identify a value of b satisfying

$$\frac{e^{-r\Delta\tau}(1+\rho)^{3/2}}{\pi(1-\rho)^{1/2}} \times \frac{e^{-b^2/2}}{b^2} \leq \frac{(1+\rho_{\max})^{3/2}}{\pi(1-\rho_{\max})^{1/2}} \frac{e^{-b^2/2}}{b^2} < \epsilon. \tag{3.17}$$

Given b, we then determine w through equation (3.15)) by ensuring the following conditions are met: $b \leq \left|\frac{-w-\mu_x}{\kappa_x}\right|$, $b \leq \frac{w-\mu_x}{\kappa_x}$, $b \leq \left|\frac{-w-\mu_y}{\kappa_y}\right|$ and $b \leq \frac{w-\mu_y}{\kappa_y}$. Subsequently, $\mathbf{D}^\dagger$ is derived via $w = \min\left\{|x^\dagger_{\min}|,\ x^\dagger_{\max},\ |y^\dagger_{\min}|,\ y^\dagger_{\max}\right\}$.

The methodological approach outlined above represents a significant advantage over traditional finite difference methods, which typically depend on heuristic strategies or trial-and-error for determining appropriate domain sizes. Our approach introduces a systematic and quantifiable method for determining domain size, significantly enhancing the accuracy and reliability of numerical solutions. The efficacy of this systematic approach is demonstrated through numerical experiments detailed in Subsection 5.3.

3.3 Discretization

We highlight that, in approximating the 2D convolution integral (3.7) over the finite integration domain $\mathbf{D}^\dagger$, it is necessary to obtain values of the Green's function $g_\alpha(x, y; \cdot)$, at points (x, y) outside $\mathbf{D}^\dagger$. To define these points, we let $z^\ddagger_{\max} = z_{\max} - z^\dagger_{\min}$ and $z^\ddagger_{\min} = z_{\min} - z^\dagger_{\max}$ for $z \in \{x, y\}$. Consequently, we need $g_\alpha(x, y; \cdot)$ at $(x, y) \in \mathbf{D}^\dagger_{\text{out}}$, where

$$\mathbf{D}^\dagger_{\text{out}} = \left([x^\ddagger_{\min}, x^\ddagger_{\max}] \times [y^\ddagger_{\min}, y^\ddagger_{\max}]\right) \setminus \mathbf{D}^\dagger, \qquad z^\ddagger_{\min} = z_{\min} - z^\dagger_{\max} \text{ for } z \in \{x, y\}. \tag{3.18}$$

Although $\mathbf{D}^\dagger_{\text{out}}$ lies outside the pricing problem's definition domain, the availability of a closed-form expression for $g_\alpha(x, y; \cdot)$ ensures no issues for our numerical methods. Moreover, the value functions for $(x, y) \in \mathbf{D}^\dagger_{\text{out}}$ are not required for our convergence analysis. The role of $\mathbf{D}^\dagger_{\text{out}}$ is to ensure the well-definedness of an associated Green's function for the convolution integral, which is crucial for time advancement within Ω_{in}.

Without loss of generality, for convenience, we assume that $|z_{\min}|$ and $z_{\max}$, where $z \in \{x, y\}$, are chosen sufficiently large so that

$$z^\dagger_{\min} = z_{\min} - \frac{z_{\max} - z_{\min}}{2}, \quad \text{and} \quad z^\dagger_{\max} = z_{\max} + \frac{z_{\max} - z_{\min}}{2}. \tag{3.19}$$

With (3.19) in mind, recalling $z^\ddagger_{\min}$ and $z^\ddagger_{\max}$, $z \in \{x, y\}$ as defined in (3.18) gives

$$z^\ddagger_{\min} = z^\dagger_{\min} - z_{\max} = -\frac{3}{2}\left(z_{\max} - z_{\min}\right), \quad \text{and} \quad z^\ddagger_{\max} = z^\dagger_{\max} - z_{\min} = \frac{3}{2}\left(z_{\max} - z_{\min}\right). \tag{3.20}$$

We denote by N (resp. $N^\dagger$ and $N^\ddagger$) the number of intervals of a uniform partition of $[x_{\min}, x_{\max}]$ (resp. $[x^\dagger_{\min}, x^\dagger_{\max}]$ and $[x^\ddagger_{\min}, x^\ddagger_{\max}]$). For convenience, we typically choose $N^\dagger = 2N$ and $N^\ddagger = 3N$ so that only one set of x-coordinates is needed. Also, let $P_x = x_{\max} - x_{\min}$, $P^\dagger_x = x^\dagger_{\max} - x^\dagger_{\min}$, and $P^\ddagger_x = x^\ddagger_{\max} - x^\ddagger_{\min}$. We define $\Delta x = \frac{P_x}{N} = \frac{P^\dagger_x}{N^\dagger} = \frac{P^\ddagger_x}{N^\ddagger}$. We use an equally spaced partition in the x-direction, denoted by $\{x_n\}$, and is defined as follows

$$\begin{aligned} x_n &= \hat{x}_0 + n\Delta x; \quad n = -N^\ddagger/2, \ldots, N^\ddagger/2, \quad \text{where} \\ \Delta x &= P_x/N = P^\dagger_x/N^\dagger = P^\ddagger_x/N^\ddagger, \quad \text{and} \\ \hat{x}_0 &= (x_{\min} + x_{\max})/2 = (x^\dagger_{\min} + x^\dagger_{\max})/2 = (x^\ddagger_{\min} + x^\ddagger_{\max})/2. \end{aligned} \tag{3.21}$$

Similarly, for the y-dimension, with $J^\dagger = 2J$, $J^\ddagger = 3J$, $P_y = y_{\max} - y_{\min}$, $P^\dagger_y = y^\dagger_{\max} - y^\dagger_{\min}$, and $P^\ddagger_y = y^\ddagger_{\max} - y^\ddagger_{\min}$, we denote by $\{y_j\}$, an equally spaced partition in the y-direction defined as follows

$$\begin{aligned} y_j &= \hat{y}_0 + j\Delta y; \quad j = -J^\ddagger/2, \ldots, J^\ddagger/2, \quad \text{where} \\ \Delta y &= P_y/J = P^\dagger_y/J^\dagger = P^\ddagger_y/J^\ddagger, \quad \text{and} \\ \hat{y}_0 &= (y_{\min} + y_{\max})/2 = (y^\dagger_{\min} + y^\dagger_{\max})/2 = (y^\ddagger_{\min} + y^\ddagger_{\max})/2. \end{aligned} \tag{3.22}$$

We use the same previously defined uniform partition $\{\tau_m\}$, $m = 0, \ldots, M$, with $\tau_m = m\Delta\tau$ and $\Delta\tau = T/M$.[2]

Regarding the control set $\mathcal{A}$, defined in (2.5), we let Q_x and Q_y respectively be the number of intervals of a uniform partition of $\mathcal{A}_x = [\sigma^x_{\min}, \sigma^x_{\max}]$ and $\mathcal{A}_y = [\sigma^y_{\min}, \sigma^y_{\max}]$. We denote by $\{\sigma^x_q\}$ and $\{\sigma^y_{q'}\}$ an equally spaced partition for $\mathcal{A}_x$ and $\mathcal{A}_y$, respectively, each with a uniform interval length $\Delta\sigma_z = \frac{\sigma^z_{\max} - \sigma^z_{\min}}{Q_z}$, where $z \in \{x, y\}$. Consequently, the discretized control set $\mathcal{A}_h$ approximating $\mathcal{A}$ is given by

$$\mathcal{A}_h = \left\{ \left(\{\sigma^x_{\min}, \sigma^x_{\max}\} \times \{\sigma^y_{q'}\} \right) \cup \left(\{\sigma^x_q\} \times \{\sigma^y_{\min}, \sigma^y_{\max}\} \right) \right\} \times \{\rho_{\min}, \rho_{\max}\}. \tag{3.23}$$

For subsequent use, we denote by Q the cardinality of the set $\mathcal{A}_h$, assuming that both $\mathcal{A}_x$ and $\mathcal{A}_y$ are discretized using the same number of partitions.

We assume that there is a discretization parameter $h > 0$ such that

$$\Delta x = C_1 h, \quad \Delta y = C_2 h, \quad \Delta\tau = C_3 h, \quad \Delta\sigma_x = C_4 h, \quad \Delta\sigma_y = C_5 h, \tag{3.24}$$

where the positive constants C_1, C_2, C_3, C_4, and C_5 are independent of h.

For convenience, we let $\mathbb{M} = \{0, \ldots M-1\}$ and we also define the following index sets:

$$\begin{aligned} \mathbb{N} &= \{-N/2+1, \ldots N/2-1\}, \quad \mathbb{N}^\dagger = \{-N, \ldots N\}, \quad \mathbb{N}^\ddagger = \{-3N/2+1, \ldots 3N/2-1\}, \\ \mathbb{J} &= \{-J/2+1, \ldots J/2-1\}, \quad \mathbb{J}^\dagger = \{-J, \ldots J\}, \quad \mathbb{J}^\ddagger = \{-3J/2+1, \ldots, 3J/2-1\}. \end{aligned} \tag{3.25}$$

With $n \in \mathbb{N}^\dagger$, $j \in \mathbb{J}^\dagger$, and $m \in \{0, \ldots, M\}$, we denote by $v^m_{n,j}$ (resp. $u^m_{n,j}$) a numerical approximation to the exact solution $v(x_n, y_j, \tau_m)$ (resp. $u(x_n, y_j, \tau_m)$) at the reference node $(x_n, y_j, \tau_m) = \mathbf{x}^m_{n,j}$. We also denote by $(\alpha^*)^m_{n,j} \equiv (\sigma^*_x, \sigma^*_y, \rho^*)^m_{n,j}$ the optimal control obtained by a numerical method for this reference node. For $m \in \mathbb{M}$, nodes $\mathbf{x}^{m+1}_{n,j}$ having (i) $n \in \mathbb{N}$ and $j \in \mathbb{J}$, are in Ω_{in}, (ii) either $n \in \mathbb{N}^\dagger \setminus \mathbb{N}$ and $j \in \mathbb{J}^\dagger$ or $n \in \mathbb{N}^\dagger$ and $j \in \mathbb{J}^\dagger \setminus \mathbb{J}$ are in Ω_{out}. For double summation, unless otherwise noted, we adopt the short-hand notation: $\sum\limits_{d\in\mathbb{D}}^{q\in\mathbb{Q}\,*}(\cdot) := \sum\limits_{q\in\mathbb{Q}}\sum\limits_{d\in\mathbb{D}}(\cdot)$. Lastly, it's important to note that references to indices $n \in \mathbb{N}^\ddagger \setminus \mathbb{N}^\dagger$ or $j \in \mathbb{J}^\ddagger \setminus \mathbb{J}^\dagger$ pertain to points within $\mathbf{D}^\dagger_{\text{out}}$ (as defined in (3.18)). As noted earlier, no numerical solutions are required for these points.

3.4 Numerical schemes

For $(x_n, y_j, \tau_0) \in \Omega_{\tau_0}$, we impose the initial condition (2.14) by

$$v^0_{n,j} = p(e^{x_n}, e^{y_j}), \quad n \in \mathbb{N}^\dagger \text{ and } j \in \mathbb{J}^\dagger. \tag{3.26}$$

For $(x_n, y_j, \tau_{m+1}) \in \Omega_{\text{out}}$, we impose the boundary condition (2.13) as follow

$$v^{m+1}_{n,j} = p(e^{x_n}, e^{y_j}) e^{-r\tau_{m+1}}, \quad n \in \mathbb{N}^\dagger \setminus \mathbb{N} \text{ or } j \in \mathbb{J}^\dagger \setminus \mathbb{J}. \tag{3.27}$$

For $(x_n, y_j, \tau_{m+1}) \in \Omega_{\text{in}}$, let $g^\alpha_{n-l,j-d} \equiv g_\alpha(x_n - x_l, y_j - y_d; \Delta\tau)$ with $n \in \mathbb{N}$, $j \in \mathbb{J}$, $l \in \mathbb{N}^\dagger$ and $d \in \mathbb{J}^\dagger$. Here, $g_\alpha(\cdot)$ is given by the closed-form expression in (3.11) in Lemma 3.1, where $\alpha \in \mathcal{A}_h$ is fixed. We let $u^{m+1,\alpha}_{n,j}$ be an approximation to the double integral (3.7) at $x = x_n$, $y = y_j$ and τ_{m+1}) obtained via a 2D composite quadrature rule. It is computed by

$$u^{m+1,\alpha}_{n,j} = \Delta x \Delta y \sum_{l\in\mathbb{N}^\dagger}^{d\in\mathbb{J}^\dagger\,*} \varphi_{l,d}\, g^\alpha_{n-l,j-d}\, v^m_{l,d}, \quad n \in \mathbb{N} \text{ and } j \in \mathbb{J}. \tag{3.28}$$

[2]While it is straightforward to generalize the numerical method to non-uniform partitioning of the τ-dimension, to prove convergence, uniform partitioning suffices.

Here, the coefficients $\varphi_{l,d}$ in (3.28) are the weights of the composite quadrature rule. Finally, $v_{n,j}^{m+1}$ is computed as follow

$$v_{n,j}^{m+1} = \max_{\alpha\in\mathcal{A}_h} u_{n,j}^{m+1,\alpha} = \max_{\alpha\in\mathcal{A}_h}\left\{\Delta x\Delta y \sum_{l\in\mathbb{N}^\dagger}^{d\in\mathbb{J}^\dagger *} \varphi_{l,d}\; g_{n-l,j-d}^{\alpha}\; v_{l,d}^{m}\right\}, \quad n\in\mathbb{N} \text{ and } j\in\mathbb{J}. \tag{3.29}$$

By solving the optimization problem (3.29), we obtain the optimal control $(\alpha^*)_{n,j}^{m+1} \equiv (\sigma_x^*, \sigma_y^*, \rho^*)_{n,j}^{m+1}$. Finally, unless otherwise stated, two-dimensional composite trapezoidal quadrature rule is used.

3.5 Efficient implementation and algorithms

In this section, we discuss an efficient implementation of the 2D discrete convolution (3.28) using FFT. For convenience, with $N^\dagger = 2N$, $N^\ddagger = 3N$, $J^\dagger = 2J$ and $J^\ddagger = 3J$, we define/recall sets of indices: $\mathbb{N}^\ddagger = \{-N^\ddagger/2+1,\ldots,N^\ddagger/2-1\}$, $\mathbb{N}^\dagger = \{-N^\dagger/2,\ldots,N^\dagger/2\}$, $\mathbb{N} = \{-N/2+1,\ldots,N/2-1\}$, $\mathbb{J}^\ddagger = \{-J^\ddagger/2+1,\ldots,J^\ddagger/2-1\}$, $\mathbb{J}^\dagger = \{-J^\dagger/2,\ldots,J^\dagger/2\}$, $\mathbb{J} = \{-J/2+1,\ldots,J/2-1\}$.

For a fixed m and a fixed α, to write (3.28) for all $n\in\mathbb{N}$ and $j\in\mathbb{J}$ into a matrix-vector multiplication form, we adopt the following notation:

- For a fixed $j\in\mathbb{J}$ and a fixed $\alpha\in\mathcal{A}_h$ and $m\in\{1,\ldots,M\}$, let $\mathbf{u}_j^{m,\alpha}$ be a column vector of length $(N-1)$ defined by $\mathbf{u}_j^{m,\alpha} \equiv \left[u_{-N/2+1,j}^{m,\alpha}, u_{-N/2+2,j}^{m,\alpha},\ldots,u_{N/2-1,j}^{m,\alpha}\right]^{\mathrm{T}}$;
- For a fixed $q\in\mathbb{J}^\ddagger$ and $m\in\{0,\ldots,M-1\}$, let $\mathbf{v}_q^m$ be a column vector of length $(2N+1)$ defined by $\mathbf{v}_q^m \equiv \left[v_{-N^\dagger/2,q}^m\;\varphi_{-N^\dagger/2,q}, v_{-N^\dagger/2+1,q}^m\;\varphi_{-N^\dagger/2+1,q},\ldots,v_{N^\dagger/2,q}^m\;\varphi_{N^\dagger/2,q}^m\right]^{\mathrm{T}}$.
- For a fixed $q\in\mathbb{J}^\ddagger$ and a fixed $\alpha\in\mathcal{A}_h$, let $\mathbf{g}_q^\alpha$ be a (non-square) matrix of size $(N-1)\times(2N+1)$, representing the convolution kernel in the inner summation (over n), defined as follows

$$\mathbf{g}_q^\alpha = \left[g_{n-l,q}^\alpha\right]_{n\in\mathbb{N},l\in\mathbb{N}^\dagger} = \begin{bmatrix} g_{N/2+1,q}^\alpha & g_{N/2,q}^\alpha & \cdots & \cdots & \cdots & g_{-3N/2+1,q}^\alpha \\ g_{N/2+2,q}^\alpha & g_{N/2+1,q}^\alpha & \cdots & \cdots & \cdots & g_{-3N/2+2,q}^\alpha \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ g_{3N/2-1,q}^\alpha & g_{3N/2-2,q}^\alpha & \cdots & g_{N/2+1,q}^\alpha & \cdots & g_{-N/2-1,q}^\alpha \end{bmatrix}. \tag{3.30}$$

In this setup, we can express the 2D discrete convolution (3.28) for all $n\in\mathbb{N}$ and $j\in\mathbb{J}$ into a matrix-vector product form as follows

$$\underbrace{\begin{bmatrix}\mathbf{u}_{-J/2+1}^{m+1,\alpha}\\ \mathbf{u}_{-J/2+2}^{m+1,\alpha}\\ \vdots \\ \mathbf{u}_{J/2-1}^{m+1,\alpha}\end{bmatrix}}_{\mathbf{u}^{m+1,\alpha}} = \Delta x\Delta y \underbrace{\begin{bmatrix} \mathbf{g}_{J/2+1}^\alpha & \mathbf{g}_{J/2}^\alpha & \cdots & \cdots & \cdots & \mathbf{g}_{-3J/2+1}^\alpha \\ \mathbf{g}_{J/2+2}^\alpha & \mathbf{g}_{J/2+1}^\alpha & \cdots & \cdots & \cdots & \mathbf{g}_{-3J/2+2}^\alpha \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{g}_{3J/2-1}^\alpha & \mathbf{g}_{3J/2-2}^\alpha & \cdots & \mathbf{g}_{J/2+1}^\alpha & \cdots & \mathbf{g}_{-J/2-1}^\alpha \end{bmatrix}}_{\left[\mathbf{g}_{j-d}^\alpha\right]_{j\in\mathbb{J},d\in\mathbb{J}^\dagger}} \underbrace{\begin{bmatrix}\mathbf{v}_{-J^\dagger/2}^m\\ \mathbf{v}_{-J^\dagger/2+1}^m\\ \vdots \\ \mathbf{v}_{J^\dagger/2}^m\end{bmatrix}}_{\mathbf{v}^m}. \tag{3.31}$$

Here, $\mathbf{u}^{m+1,\alpha}$ is a column vector of length $(N-1)(J-1)$; the (block) matrix $\left[\mathbf{g}_{j-d}^\alpha\right]_{j\in\mathbb{J},d\in\mathbb{J}^\dagger}$, which represents the convolution kernel for the double summation, is of size $(N-1)(J-1)\times(2N+1)(2J+1)$; $\mathbf{v}^m$ is a column vector of length $(2N+1)(2J+1)$.

It is noteworthy that the non-square matrix $\left[\mathbf{g}_{j-d}^\alpha\right]_{j\in\mathbb{J},d\in\mathbb{J}^\dagger}$ is a Toeplitz matrix [33], enabling efficient computation of (3.31) using FFT and circular convolution. This technique, initially applied to 1D problems in [37], is now adapted to the 2D case given by (3.31). Our goal is to represent (3.31) as a circulant matrix-vector product. This involves expanding $\left[\mathbf{g}_{j-d}^\alpha\right]_{j\in\mathbb{J},d\in\mathbb{J}^\dagger}$ to a 2D ciculant matrix - a block matrix where each block is circulant and the blocks are arranged in a circulant pattern. More

specifically, the process involves (i) expanding each block $\mathbf{g}^{\alpha}_{j-d}$ to a circulant matrix, denoted by $\tilde{\mathbf{g}}^{\alpha}_{j-d}$, and (ii) expanding $\left[\tilde{\mathbf{g}}^{\alpha}_{j-d}\right]_{j\in\mathbb{J},d\in\mathbb{J}^{\dagger}}$ to a 2D circulant matrix. Correspondingly, the vector $\mathbf{v}^m$ ia also expanded to conform with this format. Key steps of this expansion process are outlined below.

- Expansion of blocks: For each matrix $\mathbf{g}^{\alpha}_q = \left[g^{\alpha}_{n-l,q}\right]_{n\in\mathbb{N},l\in\mathbb{N}^{\dagger}}$, $q\in\mathbb{J}^{\ddagger}$, of size $(N-1)\times(2N+1)$, we expand it into a circular matrix $\tilde{\mathbf{g}}^{\alpha}_q$ of size $(3N-1)\times(3N-1)$. This expansion, detailed in [37], results in the matrix

$$\tilde{\mathbf{g}}^{\alpha}_q = \left[\begin{array}{c|c} \tilde{\mathbf{g}}^{q,\alpha}_{-1,0} & \tilde{\mathbf{g}}^{q,\alpha}_{-1,1} \\ \hline \mathbf{g}^{\alpha}_q & \tilde{\mathbf{g}}^{q,\alpha}_{0,1} \\ \hline \tilde{\mathbf{g}}^{q,\alpha}_{1,0} & \tilde{\mathbf{g}}^{q,\alpha}_{1,1} \end{array}\right], \qquad \mathbf{g}^{\alpha}_q = \left[g^{\alpha}_{n-l,q}\right]_{n\in\mathbb{N},l\in\mathbb{N}^{\dagger}}. \tag{3.32}$$

  Here, $\tilde{\mathbf{g}}^{q,\alpha}_{-1,0}$, $\tilde{\mathbf{g}}^{q,\alpha}_{1,0}$, $\tilde{\mathbf{g}}^{q,\alpha}_{-1,1}$, $\tilde{\mathbf{g}}^{q,\alpha}_{0,1}$ and $\tilde{\mathbf{g}}^{q,\alpha}_{1,1}$ are padding matrices of sizes $N\times(2N+1)$, $N\times(2N+1)$, $N\times(N-2)$, $(N-1)\times(N-2)$, and $N\times(N-2)$, respectively. These matrices are appropriately defined to ensure the circulant structure of $\tilde{\mathbf{g}}^{\alpha}_q$. Further details on these padding matrices are provided in Appendix B.

- Expansion of block matrix: We then substitute $\mathbf{g}^{\alpha}_{j-d}$ with circulant block $\tilde{\mathbf{g}}^{\alpha}_{j-d}$ in $\left[\mathbf{g}^{\alpha}_{j-d}\right]_{j\in\mathbb{J},d\in\mathbb{J}^{\dagger}}$. The resulting block matrix $\left[\tilde{\mathbf{g}}^{\alpha}_{j-d}\right]_{j\in\mathbb{J},d\in\mathbb{J}^{\dagger}}$ is then expanded into a circulant matrix of size $(3N-1)(3J-1)\times(3N-1)(3J-1)$, denoted as $\tilde{\mathbf{g}}^{\alpha}$. Specifically, $\tilde{\mathbf{g}}^{\alpha}$ is constructed as follows:

$$\tilde{\mathbf{g}}^{\alpha} = \left[\begin{array}{c|c} \tilde{\mathbf{g}}^{\alpha}_{-1,0} & \tilde{\mathbf{g}}^{\alpha}_{-1,1} \\ \hline \left[\tilde{\mathbf{g}}^{\alpha}_{j-d}\right]_{j\in\mathbb{J},d\in\mathbb{J}^{\dagger}} & \tilde{\mathbf{g}}^{\alpha}_{0,1} \\ \hline \tilde{\mathbf{g}}^{\alpha}_{1,0} & \tilde{\mathbf{g}}^{\alpha}_{1,1} \end{array}\right]. \tag{3.33}$$

  Here, $\tilde{\mathbf{g}}^{\alpha}_{-1,0}$, $\tilde{\mathbf{g}}^{\alpha}_{1,0}$, $\tilde{\mathbf{g}}^{\alpha}_{-1,1}$, $\tilde{\mathbf{g}}^{\alpha}_{0,1}$ and $\tilde{\mathbf{g}}^{\alpha}_{1,1}$ components are (block) matrices with dimensions $(3N-1)J\times(3N-1)(2J+1)$, $(3N-1)J\times(3N-1)(2J+1)$, $(3N-1)J\times(3N-1)(J-2)$, $(3N-1)(J-1)\times(3N-1)(J-2)$ and $(3N-1)J\times(3N-1)(J-1)$, respectively. These matrices are appropriately defined to ensure the circulant structure of $\tilde{\mathbf{g}}^{\alpha}$. Further details on these padding matrices are provided in Appendix B.

- Vector expansion: To conform with the circulant-maxtrix format, for each $q\in\mathbb{J}^{\dagger}$, we construct the augmented column vector $\tilde{\mathbf{v}}^m_q$ of length $(3N-1)$, by appending zeros to the column vector $\mathbf{v}^m_q$. This is defined as follows.

$$\tilde{\mathbf{v}}^m_q = \left[v^m_{-N^{\dagger}/2,q}\ \varphi_{-N^{\dagger}/2,q}, v^m_{-N^{\dagger}/2+1,q}\ \varphi_{-N^{\dagger}/2+1,q}, \ldots, v^m_{N^{\dagger}/2,q}\ \varphi_{N^{\dagger}/2,q}, 0, 0, \ldots, 0\right]^{\mathrm{T}}. \tag{3.34}$$

  Then, we form the vector $\tilde{\mathbf{v}}^m$ of size $(3N-1)(3J-1)$ by appending zeros as follows:

$$\tilde{\mathbf{v}}^m = \left[\tilde{\mathbf{v}}^m_{-J^{\dagger}/2}, \tilde{\mathbf{v}}^m_{-J^{\dagger}/2+1}, \ldots, \tilde{\mathbf{v}}^m_{J^{\dagger}/2}, \mathbf{0}, \mathbf{0}, \ldots, \mathbf{0}\right]^{\mathrm{T}}. \tag{3.35}$$

  where $\mathbf{0}$'s are zero vectors of length $(3N-1)$.

- Circulant matrix-vector product: Utilizing this setup, we express the matrix-vector product (3.31) as a circulant matrix-vector product, which is used to compute an intermediate column vector of discrete solutions. This column vector, denoted by $\tilde{\mathbf{u}}^{m+1,\alpha}$, has a length of $(3N-1)(3J-1)$ and is determined as follows:

$$\tilde{\mathbf{u}}^{m+1,\alpha} = \Delta x\Delta y\ \tilde{\mathbf{g}}^{\alpha}\ \tilde{\mathbf{v}}^m, \qquad \alpha\in\mathcal{A}_h. \tag{3.36}$$

  Here, $\tilde{\mathbf{g}}^{\alpha}$ is the circulant matrix defined in (3.33), $\tilde{\mathbf{v}}^m$ is the (augmented) column vector given by (3.35). We note that discrete solutions $u^{m+1,\alpha}_{n,j}$ for Ω_{in} are obtained by discarding the components in $\tilde{\mathbf{u}}^{m+1,\alpha}$ corresponding to indices $n\in\mathbb{N}^{\ddagger}\setminus\mathbb{N}$ or $j\in\mathbb{J}^{\ddagger}\setminus\mathbb{J}$.

The circulant matrix-vector product in (3.36) can be efficiently computed as a circulant convolution using 2D FFT. To this end, we let $\widehat{\mathbf{g}}_1^\alpha$ be the first column $\tilde{\mathbf{g}}^\alpha$ defined in (3.33) reshaped into a $(3N-1)\times(3J-1)$ matrix as follows

$$\widehat{\mathbf{g}}_1^\alpha = \Big[\ \Big[\tilde{\mathbf{g}}^\alpha_{-J/2+1}\Big]_1 \ \cdots\ \Big[\tilde{\mathbf{g}}^\alpha_{J/2}\Big]_1\ \Big[\tilde{\mathbf{g}}^\alpha_{J/2+1}\Big]_1\ \cdots\ \Big[\tilde{\mathbf{g}}^\alpha_{3J/2-1}\Big]_1\ \Big[\tilde{\mathbf{g}}^\alpha_{-3J/2+1}\Big]_1\ \cdots\ \Big[\tilde{\mathbf{g}}^\alpha_{-J/2}\Big]_1\ \Big] \tag{3.37}$$

Here, $\big[\tilde{\mathbf{g}}^\alpha_q\big]_1$, $q\in\mathbb{J}^\ddagger$, denotes the first column of the matrix $\tilde{\mathbf{g}}^\alpha_q$. We reshape the vector $\tilde{\mathbf{v}}^m$ into a $(3N-1)\times(3J-1)$ matrix, denoted by $[\tilde{\mathbf{v}}^m]$. The circulant matrix-vector product in (3.36) can be expressed as a 2D circular convolution product

$$\big[\tilde{\mathbf{u}}^{m+1,\alpha}\big] = \Delta x\Delta y\ \widehat{\mathbf{g}}_1^\alpha * [\tilde{\mathbf{v}}^m]\,, \qquad \alpha\in\mathcal{A}_h. \tag{3.38}$$

Here, $\big[\tilde{\mathbf{u}}^{m+1,\alpha}\big]$ is a $(3N-1)\times(3J-1)$ matrix, representing the reshaped version of $\tilde{\mathbf{u}}^{m+1,\alpha}$ from (3.36). The circular convolution product (3.38) is computed efficiently using FFT and inverse FFT (iFFT) as follows

$$\big[\tilde{\mathbf{u}}^{m+1,\alpha}\big] = \Delta x\Delta y\ \mathrm{FFT}^{-1}\left\{\mathrm{FFT}\left\{[\tilde{\mathbf{v}}^m]\right\}\circ\mathrm{FFT}\left\{\widehat{\mathbf{g}}_1^\alpha\right\}\right\}, \qquad \alpha\in\mathcal{A}_h. \tag{3.39}$$

Finally, we discard the components in $\big[\tilde{\mathbf{u}}^{m+1,\alpha}\big]$ corresponding to indices $n\in\mathbb{N}^\ddagger\setminus\mathbb{N}$ or $j\in\mathbb{J}^\ddagger\setminus\mathbb{J}$, obtaining discrete solutions $u^{m+1,\alpha}_{n,j}$ for Ω_{in}.

The implementation (3.39) suggests that we compute the weight components of $\widehat{\mathbf{g}}_1^\alpha$ only once for each $\alpha\in\mathcal{A}_h$ through the closed-form expression in (3.11), and reuse them for the computation over all time intervals. Putting everything together, the proposed numerical scheme for the two-factor uncertain volatility model pricing problem is presented in Algorithm 3.1 below.

Algorithm 3.1 A monotone piecewise constant control integration algorithm for a two-factor uncertain volatility model pricing problem defined in Definition (2.1).

1: for each $\alpha\in\mathcal{A}_h$, and for each $j\in\mathbb{N}^\ddagger$, compute weight matrices $\mathbf{g}^\alpha_j = \Big[g^\alpha_{n-l,j}\Big]_{n\in\mathbb{N},l\in\mathbb{N}^\dagger}$ defined in (3.30) using the closed-form expression (3.11);
2: construct weight matrices $\widehat{\mathbf{g}}_1^\alpha$, $\alpha\in\mathcal{A}_h$, using $\Big[g^\alpha_{n-l,j}\Big]_{n\in\mathbb{N},l\in\mathbb{N}^\dagger}$, $j\in\mathbb{N}^\ddagger$, defined in (3.37);
3: initialize $v^0_{n,j} = p(e^{x_n}, e^{y_j})$, $n\in\mathbb{N}^\dagger, j\in\mathbb{J}^\dagger$;
4: **for** $m=0,\ldots,M-1$ **do**
5: **for** $\alpha\in\mathcal{A}_h$ **do**
6: compute matrices of intermediate values $[\tilde{\mathbf{u}}^{m+1,\alpha}]$ using FFT as per (3.39);
7: obtain vector of discrete solutions $\mathbf{u}^{m+1,\alpha} = \Big[u^{m+1,\alpha}_{n,j}\Big]_{n\in\mathbb{N},j\in\mathbb{J}}$ by discarding the components in $[\tilde{\mathbf{u}}^{m+1,\alpha}]$ corresponding to indices $n\in\mathbb{N}^\ddagger\setminus\mathbb{N}$ or $j\in\mathbb{J}^\ddagger\setminus\mathbb{J}$;
8: **end for**
9: set $v^{m+1}_{n,j} = \max_{\alpha\in\mathcal{A}_h} u^{m+1,\alpha}_{n,j}$, $n\in\mathbb{N}$ and $j\in\mathbb{J}$, where $u^{m+1,\alpha}_{n,j}$ are from Line 7; Ω_{in}
10: compute $v^{m+1}_{n,j}$, $n\in\mathbb{N}^\dagger\setminus\mathbb{N}$ or $j\in\mathbb{J}^\dagger\setminus\mathbb{J}$, using (3.27); Ω_{out}
11: **end for**

Remark 3.3 (Complexity)**.** *As noted earlier, the cardinality of $\mathcal{A}_h$, denoted by Q, is $Q=\mathcal{O}(1/h)$. Algorithm 3.1 involves, for $m=0,\ldots,M-1$, the following key steps:*

- *Compute $u^{m+1,\alpha}_{n,j}$, $n\in\mathbb{N}^\dagger$, $j\in\mathbb{J}^\dagger$ for all $\alpha\in\mathcal{A}_h$ via FFT algorithm. The complexity of this step is $\mathcal{O}(QNJ\log(NJ)=\mathcal{O}(1/h^3\cdot\log(1/h))$, where we take into account* (3.24)*.*
- *Finding the optimal control $(\alpha^*)^{m+1}_{n,j}$ for each node $\mathbf{x}^{m+1}_{n,j}$ by comparing $u^{m+1,\alpha}_{n,j}$ for all $\alpha\in\mathcal{A}_h$ requires $\mathcal{O}(1/h)$ complexity. Thus, with a total of $\mathcal{O}(1/h^2)$ nodes, this gives a complexity $\mathcal{O}(1/h^3)$.*
- *Therefore, the major cost of Algorithm 3.1 is determined by the step of FFT Algorithm. With $\mathcal{O}(1/h)$ timesteps, the total complexity is $\mathcal{O}(1/h^4\cdot\log(1/h))$.*

4 Convergence to viscosity solution

In this section, we appeal to a Barles-Souganidis-type analysis [3] to rigorously study the convergence of our scheme in Ω_{in} as $h \to 0$ by verifying three properties: ℓ_∞-stability, monotonicity, and consistency. Our scheme consists of (3.26) (for Ω_{τ_0}), (3.27) (for Ω_{out}), and (3.29) (for Ω_{in}).

For subsequent use, we state several results below. For Ω_{in}, from Lemma 3.1, for a fixed $\alpha \in \mathcal{A}$, we have $\iint_{\mathbb{R}^2} g_\alpha(x, y; \Delta\tau)dxdy = e^{-r\Delta\tau}$, hence $\iint_{\mathbb{D}^\dagger} g_\alpha(x, y; \Delta\tau)dxdy \le e^{-r\Delta\tau} < 1$, where $\mathbb{D}^\dagger$ is defined in (3.6). For $n \in \mathbb{N}$ and $j \in \mathbb{J}$ (i.e. Ω_{in}), we define

$$\epsilon_g = \max_{\alpha,n,j} \epsilon^\alpha_{n,j}, \text{ where } \epsilon^\alpha_{n,j} := \left| \iint_{\mathbb{D}^\dagger} g_\alpha(x_n - x', y_j - y'; \Delta\tau)dx'dy' - \Delta x\Delta y \sum_{l\in\mathbb{N}^\dagger}^{d\in\mathbb{J}^\dagger *} \varphi_{l,d}\, g^\alpha_{n-l,j-d} \right|.$$

Here, at noted earlier, $\varphi_{l,d}$ are the weights of a two-dimensional composite trapezoidal quadrature rule. Since for sufficiently small h, $\epsilon^\alpha_{n,j} = \mathcal{O}(h^2)$, it follows that $\epsilon_g = \mathcal{O}(h^2)$. In addition, because the weights $\varphi_{l,d}$ of the quadrature rule are positive, for any fixed $\alpha \in \mathcal{A}$, we have

$$0 \;\le\; \Delta x\Delta y \sum_{l\in\mathbb{N}^\dagger}^{d\in\mathbb{J}^\dagger *} \varphi_{l,d}\, g^\alpha_{n-l,j-d} \;<\; 1 + \epsilon^\alpha_{n,j} \;\le\; 1 + \epsilon_g < e^{\epsilon_g}. \tag{4.1}$$

4.1 Stability

Lemma 4.1 (ℓ_∞-stability)**.** *Suppose the discretization parameter h satisfies* (3.24)*. Then our scheme, which consists of* (3.26)*,* (3.27)*, and* (3.29)*, satisfies the bound $\sup_{h>0} \|v^m\|_\infty < \infty$ for all $m = 0, \dots, M$, as the discretization parameter $h \to 0$. Here, we have $\|v^m\|_\infty = \max_{n,j} |v^m_{n,j}|$, $n \in \mathbb{N}^\dagger$ and $j \in \mathbb{J}^\dagger$.*

Proof of Lemma 4.1. First, we note that, for any fixed $h > 0$, as given by (3.26), we have $||v^0||_\infty < \infty$, since Ω is a bounded domain. Therefore, we have $\sup_{h>0} ||v^0||_\infty < \infty$. Motivated by this observation, to demonstrate l_∞-stability of our scheme, we will show that, for a fixed $h > 0$, at any (x_n, y_j, τ_m), $m = 0, \dots, M$, we have

$$|v^m_{n,j}| < e^{m\epsilon_g} ||v^0||_\infty, \quad m = 0, 1, \dots, M. \tag{4.2}$$

It is straightforward to show that (3.26) is ℓ_∞-stable, since $\max_{n,j} |v^0_{n,j}| \le ||v^0||_\infty$ for $(n, j) \in \mathbb{N}^\dagger \times \mathbb{J}^\dagger$, clearly satisfying (4.2). Next, for equation (3.27), we note that, since $|v^{m+1}_{n,j}| = |v^m_{n,j} e^{-r\Delta\tau}| < |v^m_{n,j}|$, by induction on m, we have $\max_{n,j} |v^m_{n,j}| \le ||v^0||_\infty$, for either $(n, j) \in (\mathbb{N}^\dagger \setminus \mathbb{N}) \times \mathbb{J}^\dagger$ or $\mathbb{N}^\dagger \times (\mathbb{J}^\dagger \setminus \mathbb{J})$.

Now we focus on the main task, demonstrating ℓ_∞-stability for (3.29) (Ω_{in}) through an induction proof on m. For the base case $m = 1$, with a fixed $\alpha \in \mathcal{A}_h$,

$$u^1_{n,j} = \Delta x\Delta y \sum_{l\in\mathbb{N}^\dagger}^{d\in\mathbb{J}^\dagger *} \varphi_{l,d}\, g^\alpha_{n-l,j-d}\, v^0_{l,d}. \tag{4.3}$$

Then, we have

$$|u^1_{n,j}| \;\le\; \Delta x\Delta y \sum_{l\in\mathbb{N}^\dagger}^{d\in\mathbb{J}^\dagger *} \varphi_{l,d}\, g^\alpha_{n-l,j-d} |v^0_{l,d}| \;\le\; \Delta x\Delta y \sum_{l\in\mathbb{N}^\dagger}^{d\in\mathbb{J}^\dagger *} \varphi_{l,d}\, g^\alpha_{n-l,j-d} ||v^0||_\infty \;\le\; e^{1\epsilon_g} ||v^0||_\infty, \tag{4.4}$$

where the last inequality is due to (4.1). Since $v^1_{n,j} = \max_\alpha u^1_{n,j}$, we have

$$|v^1_{n,j}| = |\max_\alpha u^1_{n,j}| \;\le\; \max_\alpha |u^1_{n,j}| \;\le\; e^{1\epsilon_g} ||v^0||_\infty,$$

as wanted for the base case. For the hypothesis, assume that (4.2) hold for $m = m'$, $1 \le m' \le M - 1$

$$|v^{m'}_{n,j}| < e^{m'\epsilon_g} ||v^0||_\infty, \qquad (n, j) \in \mathbb{N} \times \mathbb{J}. \tag{4.5}$$

In the induction step, we need show that (4.2) also holds for $m = m' + 1$, i.e.

$$|v_{n,j}^{m'+1}| < e^{(m'+1)\epsilon_g}||v^0||_\infty. \tag{4.6}$$

To show (4.6), recalling $u_{n,j}^{m'+1}$ from (3.28) gives

$$\begin{aligned} |u_{n,j}^{m'+1}| \;&\leq\; \Delta x \Delta y \sum_{l\in\mathbb{N}^\dagger}^{d\in\mathbb{J}^\dagger *} \varphi_{l,d}\; g_{n-l,j-d}^{\alpha} |v_{l,d}^{m'}| \;\overset{\text{(i)}}{\leq}\; \Delta x \Delta y \sum_{l\in\mathbb{N}^\dagger}^{d\in\mathbb{J}^\dagger *} \varphi_{l,d}\; g_{n-l,j-d}^{\alpha}\; e^{m'\epsilon_g}||v^0||_\infty \\ &\leq\; e^{\epsilon_g} e^{m'\epsilon_g}||v^0||_\infty \;=\; e^{(m'+1)\epsilon_g}||v^0||_\infty. \end{aligned} \tag{4.7}$$

Here, (i) is due to the hypothesis (4.5) together with the fact that the scheme for Ω_{out}, captured by equation (3.27), is also ℓ_∞-stable as shown earlier. Hence, $|v_{n,j}^{m'+1}| = |\max_\alpha u_{n,j}^{m'+1}| \;\leq\; e^{(m'+1)\epsilon_g}||v^0||_\infty$, proving (4.6) for $m = m' + 1$. This concludes the proof. □

4.2 Consistency

While equations (3.26), (3.27), and (3.29) are convenient for computation, they are not in a form amendable for analysis. For purposes of verifying consistency, it is more convenient to rewrite them in a single equation. To this end, for $(x_n, y_j, \tau_{m+1}) \in \Omega_{\text{in}}$, i.e. $n \in \mathbb{N}$ and $j \in \mathbb{J}$, we define operator $\mathcal{C}_{n,j}^{m+1}(\cdot)$, where

$$\mathcal{C}_{n,j}^{m+1}(\cdot) \equiv \mathcal{C}_{n,j}^{m+1}\left(h, v_{n,j}^{m+1}, \{v_{l,d}^m\}_{\substack{l\in\mathbb{N}^\dagger \\ d\in\mathbb{J}^\dagger}}\right) = \frac{1}{\Delta\tau}\left[v_{n,j}^{m+1} - \max_{\alpha\in\mathcal{A}_h}\; \left\{\Delta x \Delta y \sum_{l\in\mathbb{N}^\dagger}^{d\in\mathbb{J}^\dagger *} \varphi_{l,d}\; g_{n-l,j-d}^{\alpha}\; v_{l,d}^m\right\}\right]. \tag{4.8}$$

Using $\mathcal{C}_{n,j}^{m+1}(\cdot)$ defined in (4.8), our numerical scheme at the reference node $\mathbf{x} = (x_n, y_j, \tau_{m+1})$ can be rewritten in an equivalent form as follows

$$0 = \mathcal{H}_{n,j}^{m+1}\left(h, v_{n,j}^{m+1}, \{v_{l,d}^m\}_{\substack{l\in\mathbb{N}^\dagger \\ d\in\mathbb{J}^\dagger}}\right) \equiv \begin{cases} \mathcal{C}_{n,j}^{m+1}(\cdot) & \mathbf{x} \in \Omega_{\text{in}}, \\ v_{n,j}^{m+1} - p(e^{x_n}, e^{y_j}) & \mathbf{x} \in \Omega_{\tau_0}, \\ v_{n,j}^{m+1} - p(e^{x_n}, e^{y_j})e^{-r\tau_{m+1}} & \mathbf{x} \in \Omega_{\text{out}}, \end{cases} \tag{4.9}$$

where the sub-domains are defined in (2.8), and $p(\cdot,\cdot)$ is the terminal condition.

To demonstrate the consistency in viscosity sense of (4.9), we need an intermediate result given in Lemma 4.2 below.

Lemma 4.2 (Two dimensional - Ω_{in})**.** *Let ϕ be a test function in $\mathcal{C}^\infty(\Omega)$. For fixed $\alpha \in \mathcal{A}$ and $\mathbf{x}_{n,j}^m \in \Omega$, where $n, j \in \mathbb{N}$ and $m \in \{1, \ldots, M\}$, with $\phi_{n,j}^m = \phi\big(\mathbf{x}_{n,j}^m\big)$, and for sufficiently small h, we have*

$$\Delta x \Delta y \sum_{l\in\mathbb{N}^\dagger}^{d\in\mathbb{J}^\dagger *} \varphi_{l,d}\; g_{n-l,j-d}^{\alpha}\; \phi_{l,d}^m \;=\; \phi_{n,j}^m + \Delta\tau\,[\mathcal{L}_\alpha\phi]_{n,j}^m + \mathcal{O}(h^2). \tag{4.10}$$

Here, $\big[\mathcal{L}_\alpha\phi\big]_{n,j}^m = \big[\mathcal{L}_\alpha\phi\big]\big(\mathbf{x}_{n,j}^m\big)$, and the differential operator $\mathcal{L}_\alpha$ are defined in (2.7).

Proof of Lemma 4.2. Starting from the discrete convolution on the left-hand-side (lhs) of (4.10), we need to recover an associated convolution integral of the form (3.5) which is posed on an infinite integration region. Since for an arbitrary fixed τ_m, $\phi(x, y, \tau_m)$ is not necessarily in $L^1(\mathbb{R}^2)$, standard mollification techniques can be used to obtain a mollifier $\chi(x, y, \tau_m) \in L^1(\mathbb{R}^2)$ which agrees with $\phi(x, y, \tau_m)$ on $\mathbf{D}^\dagger$ [19], and has bounded derivatives up to second order across $\mathbb{R}^2$. For brevity, instead

of $\chi(x,y,\tau_m)$, we will write $\chi(x,y)$, which is a smooth bivariate function of $(x,y)\in\mathbb{R}^2$. We have

$$\begin{aligned}\Delta x\Delta y\sum_{l\in\mathbb{N}^\dagger}^{d\in\mathbb{N}^\dagger *}\varphi_{l,d}\; g^{\alpha}_{n-l,j-d}\;\phi^m_{l,d} &\overset{\text{(i)}}{=}\iint_{\mathbf{D}^\dagger} g_\alpha\left(x_n-x,y_j-y;\Delta\tau\right)\;\phi(x,y)\;dx\;dy+\mathcal{O}(h^2)\\ &\overset{\text{(ii)}}{=}\iint_{\mathbb{R}^2} g_\alpha\left(x_n-x,y_j-y;\Delta\tau\right)\;\chi(x,y)\;dx\;dy+\mathcal{O}(h^2)+\mathcal{O}\big(he^{-1/h}\big)\\ &\overset{\text{(iii)}}{=}[\chi*g](x_n,y_j)+\mathcal{O}(h^2).\\ &=\mathcal{F}^{-1}\left[\mathcal{F}\left[\chi\right](\eta,\zeta)\;G\left(\eta,\zeta;\Delta\tau\right)\right](x_n,y_j)+\mathcal{O}(h^2).\end{aligned}\tag{4.11}$$

Here, in (i), the $\mathcal{O}(h^2)$ is due to error in the composite trapezoidal rule, noting that ϕ has bounded derivatives of all orders in Ω because Ω is a bounded domain; in (ii) the boundary truncation error is $\mathcal{O}\big(he^{-1/h}\big)$, due to Lemma 3.2, and in (iii) $[\chi*g]$ denotes the convolution of $\chi(x,y)$ and $g_\alpha(x,y;\Delta\tau)$.

In (4.11), with $\Psi(\eta,\zeta)$ given in (3.10), expanding $G(\eta,\zeta;\Delta\tau)=e^{\Psi(\eta,\zeta)\Delta\tau}$ using a Taylor series gives

$$G(\eta,\zeta;\Delta\tau)\approx 1+\Psi(\eta,\zeta)\Delta\tau+\mathcal{R}(\eta,\zeta)\Delta\tau^2,\quad \mathcal{R}(\eta,\zeta)=\frac{\Psi(\eta,\zeta)^2e^{\xi\Psi(\eta,\zeta)}}{2},\quad \xi\in(0,\Delta\tau).\tag{4.12}$$

Therefore,

$$\begin{aligned}[\chi*g]\left(x_n,y_j\right)&=\mathcal{F}^{-1}\left[\mathcal{F}\left[\chi\right](\eta,\zeta)\;\left(1+\Psi(\eta,\zeta)\Delta\tau+\mathcal{R}(\eta,\zeta)\Delta\tau^2\right)\right](x_n,y_j)\\ &=\chi(x_n,y_j)+\Delta\tau\mathcal{F}^{-1}\left[\mathcal{F}\left[\chi\right](\eta,\zeta)\;\Psi\left(\eta,\zeta\right)\right](x_n,y_j)\\ &\quad+\Delta\tau^2\mathcal{F}^{-1}\left[\mathcal{F}\left[\chi\right](\eta,\zeta)\;\mathcal{R}(\eta,\zeta)\right](x_n,y_j).\end{aligned}\tag{4.13}$$

Here, the first term in (4.13), namely $\chi(x_n,y_j)\equiv\chi(x_n,y_j,\tau_m)$ is simply $\phi^m_{n,j}$ by construction of $\chi(\cdot)$. For the second term in (4.13), we focus on $\mathcal{F}\left[\chi\right](\eta,\zeta)\;\Psi\left(\eta,\zeta\right)$. Recalling the closed-form expression for $\Psi(\eta,\zeta)$ in (3.10), we obtain

$$\mathcal{F}[\chi](\eta,\zeta)\Psi(\eta,\zeta)=\mathcal{F}\big[\frac{\sigma_x^2}{2}\chi_{xx}+\frac{\sigma_y^2}{2}\chi_{yy}+(r-\frac{\sigma_x^2}{2})\chi_x+(r-\frac{\sigma_y^2}{2})\chi_y+\rho\sigma_x\sigma_y\chi_{xy}-r\chi\big](\eta,\zeta)=\mathcal{F}\left[\mathcal{L}_\alpha\chi\right](\eta,\zeta).$$

Therefore, the second term in (4.13) becomes

$$\Delta\tau\mathcal{F}^{-1}\left[\mathcal{F}\left[\chi\right](\eta,\zeta)\;\Psi\left(\eta,\zeta\right)\right](x_n,y_j)=\Delta\tau\left[\mathcal{L}_\alpha\chi\right](\mathbf{x}^m_{n,j})=\Delta\tau\left[\mathcal{L}_\alpha\chi\right]^m_{n,j}.\tag{4.14}$$

For the third term $\Delta\tau^2\mathcal{F}^{-1}\left[\mathcal{F}\left[\chi\right](\eta,\zeta)\;\mathcal{R}(\eta,\zeta)\right](x_n,y_j)$ in (4.13), we have

$$\begin{aligned}&\Delta\tau^2\left|\mathcal{F}^{-1}\left[\mathcal{F}[\chi](\eta,\zeta)\;\mathcal{R}(\eta,\zeta)\right](x_n,y_j)\right|\\ &\quad=\frac{\Delta\tau^2}{(2\pi)^2}\left|\iint_{\mathbb{R}^2}e^{i(\eta x_n+\zeta y_j)}\mathcal{R}(\eta,\zeta)\left[\iint_{\mathbb{R}^2}e^{-i(\eta x+\zeta y)}\chi(x,y)\;dx\;dy\right]d\eta d\zeta\right|\\ &\quad\le\Delta\tau^2\iint_{\mathbb{R}^2}|\chi(x,y)|\;dxdy\;\iint_{\mathbb{R}^2}|\mathcal{R}(\eta,\zeta)|\;d\eta d\zeta.\end{aligned}\tag{4.15}$$

Noting $\mathcal{R}(\eta,\zeta)=\dfrac{\Psi(\eta,\zeta)^2e^{\xi\Psi(\eta,\zeta)}}{2}$, as shown in (4.12), where a closed-form expression for $\Psi(\eta,\zeta)$ is given in (3.10), we obtain

$$|\mathcal{R}(\eta,\zeta)|=\frac{|(\Psi(\eta,\zeta))^2|}{2}\exp\big(\xi\big(-\frac{\sigma_x^2\eta^2}{2}-\frac{\sigma_y^2\zeta^2}{2}-\rho\sigma_x\sigma_y\eta\zeta-r\big)\big).$$

The term $|(\Psi(\eta,\zeta))^2|$ can be written in the form $|\Psi|^2=\sum_{\substack{k+q=4\\k,q\ge0}}C_{kq}\eta^k\zeta^q$, where C_{kq} are bounded coefficients. This is a quartic polynomial in η and ζ. Furthermore, the exponent of exponential term is bounded by

$$-\frac{1}{2}\sigma_x^2\eta^2-\frac{1}{2}\sigma_y^2\zeta^2-\rho\sigma_x\sigma_y\eta\zeta-r\le-\frac{1}{2}\sigma_x^2\eta^2-\frac{1}{2}\sigma_y^2\zeta^2+|\rho|\sigma_x\sigma_y|\eta\zeta|$$

For $|\rho| < 1$, we have $|\rho|\sigma_x\sigma_y|\eta\zeta| < \frac{1}{2}(\sigma_x^2\eta^2 + \sigma_y^2\zeta^2)$. Therefore, $\iint_{\mathbb{R}^2} |\mathcal{R}(\eta,\zeta)|\ d\eta d\zeta$ is bounded since

$$\iint_{\mathbb{R}^2} |\eta|^k|\zeta|^q\ e^{-\frac{1}{2}\sigma_x^2\eta^2 - \frac{1}{2}\sigma_y^2\zeta^2 - \rho\sigma_x\sigma_y\eta\zeta}\ d\eta\ d\zeta, \quad k+q=4,\ k,q\geq 0,$$

is also bounded. Together with $\chi(x,y) \in L^1(\mathbb{R}^2)$, the rhs of (4.15) is $\mathcal{O}(\Delta\tau^2)$, i.e.

$$\Delta\tau^2 \left|\mathcal{F}^{-1}\left[\mathcal{F}[\chi](\eta,\zeta)\ \mathcal{R}(\eta,\zeta)\right](x_n,y_j)\right| = \mathcal{O}(\Delta\tau^2). \tag{4.16}$$

Substituting (4.14) and (4.16) into (4.13), noting (4.11) and $\chi(x,y) = \phi(x,y)$ for $(x,y) \in \mathbf{D}^\dagger$ gives

$$\Delta x\Delta y \sum_{l\in\mathbb{N}^\dagger}^{d\in\mathbb{N}^\dagger *} \varphi_{l,d}\ g^{\alpha}_{n-l,j-d}\ \phi^m_{l,d} \ = \ \phi^m_{n,j} + \Delta\tau\left[\mathcal{L}_\alpha\phi\right]^m_{n,j} + \mathcal{O}(h^2).$$

This concludes the proof. □

To establish the consistency in the viscosity sense of our scheme as presented in (4.9), it is essential to first examine the local consistency. This requires revisiting the operator $F_{\text{in}}(\cdot)$ defined in (2.12). In the context of a discretized control set $\mathcal{A}_h$, we introduce a modified operator that aligns with the piecewise constant control approach.

Definition 4.1. *Given a discretization parameter $h > 0$, we define the operator F^h_{in} for each $\alpha \in \mathcal{A}_h \subseteq \mathcal{A}$ as follows:*

$$F^h_{in} := F_{in},\ \alpha \in \mathcal{A}_h \subseteq \mathcal{A}, \tag{4.17}$$

Building on this definition, Lemma 4.3 presents an important result regarding the approximation error bound when implementing the piecewise constant control technique.

Lemma 4.3. *For any $\mathbf{x} \in \Omega_{in}$, and for a test function $\phi \in \mathcal{C}^\infty(\Omega)$ and a constant ξ, we have*

$$\left|F_{in}(\mathbf{x},\phi(\mathbf{x}),D\phi(\mathbf{x}),D^2\phi(\mathbf{x})) - F^h_{in}(\mathbf{x},(\phi+\xi)(\mathbf{x}),D(\phi+\xi)(\mathbf{x}),D^2(\phi+\xi))\right| \leq Ch + r\xi, \tag{4.18}$$

where $C > 0$ is a bounded constant independently of h.

Proof of Lemma 4.3. By insertion and the triangle inequality, the lhs of (4.18) is bounded as follows

$$\begin{aligned}\left|F_{\text{in}}(\cdot) - F^h_{\text{in}}(\cdot)\right| &\leq \left|\sup_{\alpha\in\mathcal{A}_h}\mathcal{L}_\alpha(\phi+\xi) - \sup_{\alpha\in\mathcal{A}}\mathcal{L}_\alpha(\phi+\xi)\right| + \left|\sup_{\alpha\in\mathcal{A}}\mathcal{L}_\alpha(\phi+\xi) - \sup_{\alpha\in\mathcal{A}}\mathcal{L}_\alpha(\phi)\right| \\ &= \left|\sup_{\alpha\in\mathcal{A}_h}\mathcal{L}_\alpha\phi - \sup_{\alpha\in\mathcal{A}}\mathcal{L}_\alpha\phi\right| + r\xi. \end{aligned} \tag{4.19}$$

Due to the compactness of $\mathcal{A}$, the supremum of $\mathcal{L}_\alpha(\phi)$ is attainable at, say $\alpha^* \equiv (\sigma_x^*, \sigma_y^*, \rho^*) \in \mathcal{A}$. By (3.1), there exists $\alpha'^* \equiv (\sigma_x'^*, \sigma_y'^*, \rho'^*) \in \mathcal{A}_h$ with $\|\alpha'^* - \alpha^*\|_2 \leq h$. Therefore, the first term in (4.19) becomes $\left|\sup_{\alpha\in\mathcal{A}_h}\mathcal{L}_\alpha(\phi) - \sup_{\alpha\in\mathcal{A}}\mathcal{L}_\alpha(\phi)\right| = |\mathcal{L}_{\alpha'^*}\phi - \mathcal{L}_{\alpha^*}\phi| = \ldots$

$$\begin{aligned}\ldots &\overset{\text{(i)}}{=} \left|\frac{1}{2}((\sigma_x'^*)^2 - (\sigma_x^*)^2)(\phi_{xx} - \phi_x) + \frac{1}{2}((\sigma_y'^*)^2 - (\sigma_y^*)^2)(\phi_{yy} - \phi_y) + (\rho'^*\sigma_x'^*\sigma_y'^* - \rho^*\sigma_x^*\sigma_y^*)\phi_{xy}\right| \\ &\leq \frac{1}{2}\left|((\sigma_x'^*)^2 - (\sigma_x^*)^2)\right|(|\phi_{xx}| + |\phi_x|) + \frac{1}{2}\left|((\sigma_y'^*)^2 - (\sigma_y^*)^2)\right|(|\phi_{yy}| + |\phi_y|) + \left|(\rho'^*\sigma_x'^*\sigma_y'^* - \rho^*\sigma_x^*\sigma_y^*)\right||\phi_{xy}| \\ &\overset{\text{(ii)}}{\leq} Ch,\end{aligned}$$

where $C > 0$ is a bounded constant independently of h. Here, in (i), we first insert the $\mathcal{L}_\alpha(\cdot)$ operator (2.7) and then combine similar terms; (ii) due to the $\|\alpha'^* - \alpha^*\|_2 \leq h$, together with the compactness of the admissible control set $\mathcal{A}$ and the fact that the test function ϕ has continuous bounded derivatives in Ω since Ω is bounded. This concludes the proof. □

Below, we state the key supporting lemma related to local consistency of our numerical scheme (4.9).

Lemma 4.4 (Local consistency)**.** *Suppose that (i) the discretization parameter h satisfies* (3.24)*. Then, for any test function $\phi \in \mathcal{C}^\infty(\Omega)$, with $\phi_{n,j}^m = \phi\left(\mathbf{x}_{n,j}^m\right)$ and $\mathbf{x} \coloneqq (x_n, y_j, \tau_{m+1}) \in \Omega$, and for a sufficiently small h, we have*

$$\mathcal{H}_{n,j}^{m+1}\left(h, \phi_{n,j}^{m+1}+\xi, \left\{\phi_{l,d}^m+\xi\right\}_{\substack{l\in\mathbb{N}^\dagger \\ d\in\mathbb{J}^\dagger}}\right) = \begin{cases} F_{in}^h(\cdot,\cdot) + c(\mathbf{x})\xi + \mathcal{O}(h) & \mathbf{x}\in\Omega_{in}, \\ F_{out}(\cdot,\cdot) + c(\mathbf{x})\xi & \mathbf{x}\in\Omega_{out}; \\ F_{\tau_0}(\cdot,\cdot) + c(\mathbf{x})\xi & \mathbf{x}\in\Omega_{\tau_0}. \end{cases} \tag{4.20}$$

Here, ξ is a constant, and $c(\cdot)$ is a bounded function satisfying $|c(\mathbf{x})| \le \max(r,1)$ for all $\mathbf{x} \in \Omega$. The operators F_{in}^h, defined in (4.17)*, and $F_{out}(\cdot)$, and $F_{\tau_0}(\cdot)$, respectively defined in* (2.13)-(2.14)*, are functions of $(\mathbf{x}, \phi(\mathbf{x}))$.*

Proof of Lemma 4.4. Since $\phi \in \mathcal{C}^\infty(\Omega)$ and Ω is bounded, ϕ has continuous and bounded derivatives of up to second-order in Ω. We now show that the first equation of (4.20) is true, that is,

$$\mathcal{H}_{n,j}^{m+1}(\cdot) = \mathcal{C}_{n,j}^{m+1}(\cdot) = F_{\text{in}}^h(\mathbf{x}, \phi(\mathbf{x})) + c(\mathbf{x})\xi + \mathcal{O}(h)$$
$$\text{if } x_{\min} < x_n < x_{\max},\ y_{\min} < y_j < y_{\max},\ 0 < \tau_{m+1} \le T.$$

where operators $\mathcal{C}_{n,j}^{m+1}(\cdot)$ is defined in (4.8). In this case, operator $\mathcal{C}_{n,j}^{m+1}(\cdot)$ is written as follows

$$\begin{aligned} \mathcal{C}_{n,j}^{m+1}(\cdot) &= \frac{1}{\Delta\tau}\left[\phi_{n,j}^{m+1} + \xi - \max_{\alpha\in\mathcal{A}_h}\left\{\Delta x\Delta y \sum_{l\in\mathbb{N}^\dagger}^{d\in\mathbb{J}^\dagger{}^*} \varphi_{l,d}\, g_{n-l,j-d}^\alpha\, (\phi_{l,d}^m + \xi)\right\}\right] \\ &= \frac{1}{\Delta\tau}\left[\phi_{n,j}^{m+1} - \max_{\alpha\in\mathcal{A}_h}\left\{\Delta x\Delta y \sum_{l\in\mathbb{N}^\dagger}^{d\in\mathbb{J}^\dagger{}^*} \varphi_{l,d}\, g_{n-l,j-d}^\alpha\, \phi_{l,d}^m\right\} + \xi\left(1 - \max_{\alpha\in\mathcal{A}_h}\Delta x\Delta y \sum_{l\in\mathbb{N}^\dagger}^{d\in\mathbb{J}^\dagger{}^*} \varphi_{l,d}\, g_{n-l,j-d}^\alpha\right)\right] \\ &\overset{\text{(i)}}{=} \frac{\phi_{n,j}^{m+1} - \phi_{n,j}^m}{\Delta\tau} - \max_{\alpha\in\mathcal{A}_h}[\mathcal{L}_\alpha\phi]_{n,j}^m + \frac{\xi}{\Delta\tau}\left(1 - \max_{\alpha\in\mathcal{A}_h}\left\{\Delta x\Delta y \sum_{l\in\mathbb{N}^\dagger}^{d\in\mathbb{J}^\dagger{}^*} \varphi_{l,d}\, g_{n-l,j-d}^\alpha\right\}\right) + \mathcal{O}(h). \end{aligned} \tag{4.21}$$

Here, (i) is due to use Lemma 4.2. Regarding the term $\frac{\xi}{\Delta\tau}\left(1 - \max_{\alpha\in\mathcal{A}_h}\{\cdot\}\right)$, suppose that $\max_{\alpha\in\mathcal{A}_h}\{\cdot\}$ is attainable at α'. Then, $|1 - \max_{\alpha\in\mathcal{A}_h}\{\cdot\}| = \left|1 - \Delta x\Delta y \sum_{l\in\mathbb{N}^\dagger}^{d\in\mathbb{J}^\dagger{}^*} \varphi_{l,d}\, g_{n-l,j-d}^{\alpha'}\right| \le \dots$

$$\dots \le \left|1 - \iint_{\mathbb{R}^2} g_{\alpha'}(x_n - x, y_j - y; \Delta\tau)dxdy\right| + \left|\iint_{\mathbb{R}^2} g_{\alpha'}(\cdot,\cdot;\Delta\tau)dxdy - \Delta x\Delta y \sum_{l\in\mathbb{N}^\dagger}^{d\in\mathbb{J}^\dagger{}^*} \varphi_{l,d}\, g_{n-l,j-d}^{\alpha'}\right|. \tag{4.22}$$

The first term of (4.22) is simply $1 - e^{-r\Delta\tau} = r\Delta\tau + \mathcal{O}(h^2)$, noting $\iint_{\mathbb{R}^2} g_\alpha(\cdot,\cdot;\Delta\tau)dxdy = e^{-r\Delta\tau}$ for any $\alpha \in \mathcal{A}$. The second term of (4.22) is simply $\mathcal{O}(h^2) + \mathcal{O}(he^{-1/h}) = \mathcal{O}(h^2)$ due to numerical integration error and boundary truncation error, as noted earlier. With this in mind, we have

$$\frac{\xi}{\Delta\tau}\left(1 - \max_{\alpha\in\mathcal{A}_h}\Delta x\Delta y \sum_{l\in\mathbb{N}^\dagger}^{d\in\mathbb{J}^\dagger{}^*} \varphi_{l,d}\, g_{n-l,j-d}^\alpha\right) = r\xi + \mathcal{O}(h).$$

Substituting this result into (4.21) gives

$$\mathcal{C}_{n,j}^{m+1}(\cdot) = \frac{\phi_{n,j}^{m+1} - \phi_{n,j}^m}{\Delta\tau} - \max_{\alpha\in\mathcal{A}_h}[\mathcal{L}_\alpha\phi]_{n,j}^m + r\xi + \mathcal{O}(h) \overset{\text{(i)}}{=} \left[\phi_\tau - \max_{\alpha\in\mathcal{A}_h}\mathcal{L}_\alpha\phi\right]_{n,j}^{m+1} + r\xi + \mathcal{O}(h).$$

Here, in (i), we use $(\phi_\tau)_{n,j}^m = (\phi_\tau)_{n,j}^{m+1} + \mathcal{O}(h)$, $(\phi_z)_{n,j}^m = (\phi_z)_{n,j}^{m+1} + \mathcal{O}(h)$, $z \in \{x, y\}$, and for the cross derivative term $(\phi_{xy})_{n,j}^m = (\phi_{xy})_{n,j}^{m+1} + \mathcal{O}(h)$. This proves the first equation in (4.20). The remaining equations in (4.20) can be proved using similar arguments with the first equation, and hence omitted for brevity. This concludes the proof. □

We now verify the consistency of the numerical scheme $\mathcal{H}_{n,j}^{m+1}(\cdot)$ as defined in (4.9). We first define the notion of consistency in the viscosity sense below.

Definition 4.2 (Consistency in viscosity sense)**.** *Suppose the discretization parameter h satisfies* (3.24)*. The numerical scheme* (4.9) *is consistent in the viscosity sense if, for all $\hat{\mathbf{x}} = (\hat{x}, \hat{y}, \hat{\tau}) \in \Omega$, and for any $\phi \in \mathcal{C}^\infty(\Omega)$, with $\phi_{n,j}^m = \phi\big(\mathbf{x}_{n,j}^m\big)$ and $\mathbf{x} = (x_n, y_j, \tau_{m+1})$, we have both of the following*

$$\limsup_{\substack{h\to 0,\ \mathbf{x}\to\hat{\mathbf{x}}\\ \xi\to 0}} \mathcal{H}_{n,j}^{m+1}\left(h, \phi_{n,j}^{m+1} + \xi, \left\{\phi_{l,d}^m + \xi\right\}_{\substack{l\in\mathbb{N}^\dagger\\ d\in\mathbb{J}^\dagger}}\right) \le (F_\Omega)^*\left(\hat{\mathbf{x}}, \phi(\hat{\mathbf{x}}), D\phi(\hat{\mathbf{x}}), D^2\phi(\hat{\mathbf{x}})\right), \tag{4.23}$$

$$\liminf_{\substack{h\to 0,\ \mathbf{x}\to\hat{\mathbf{x}}\\ \xi\to 0}} \mathcal{H}_{n,j}^{m+1}\left(h, \phi_{n,j}^{m+1} + \xi, \left\{\phi_{l,k}^m + \xi\right\}_{\substack{l\in\mathbb{N}^\dagger\\ d\in\mathbb{J}^\dagger}}\right) \ge (F_\Omega)_*\left(\hat{\mathbf{x}}, \phi(\hat{\mathbf{x}}), D\phi(\hat{\mathbf{x}}), D^2\phi(\hat{\mathbf{x}})\right). \tag{4.24}$$

Here, $(F_\Omega)^*(\cdot)$ and $(F_\Omega)_*(\cdot)$ respectively are the u.s.c. and the l.s.c. envelop of the operator $F_\Omega(\cdot)$ defined in (2.10)*.*

Below, we state and prove the main lemma on consistency of the numerical scheme (4.9).

Lemma 4.5 (Consistency)**.** *Suppose the discretization parameter h satisfies* (3.24)*. Then, the numerical scheme* (4.9) *is consistent with the two-factor uncertain volatility pricing problem* (2.1) *in Ω in the sense of Definition 4.2.*

Proof of Lemma 4.5. We first prove (4.23). Let $\hat{\mathbf{x}} \equiv (\hat{x}, \hat{y}, \hat{\tau})$ be an arbitrary, but fixed, point in Ω. Consider $h \to 0$. There exists sequences of $\{h_i\}$, $\{m_i\}$, $\{\mathbf{x}_i\}$, and $\{\xi_i\}$, such that

$$\text{as } \ i \to \infty, \quad h_i \to 0,\ \xi_i \to 0,\ \mathbf{x}_i \equiv (x_{n_i}, y_{j_i}, \tau_{m_i+1}) \to \hat{\mathbf{x}} \equiv (\hat{x}, \hat{y}, \hat{\tau}), \tag{4.25}$$

and

$$\limsup_{i\to\infty} \mathcal{H}_{n_i,j_i}^{m_i+1}\left(h_i, \phi_{n_i,j_i}^{m_i+1} + \xi_i, \left\{\phi_{l_i,d_i}^{m_i} + \xi_i\right\}\right) = \limsup_{\substack{h\to 0,\ \mathbf{x}\to\hat{\mathbf{x}}\\ \xi\to 0}} \mathcal{H}_{n,j}^{m+1}\left(h, \phi_{n,j}^{m+1} + \xi, \left\{\phi_{l,d}^m + \xi\right\}\right). \tag{4.26}$$

Now, we consider the case $\hat{\mathbf{x}} \in \Omega_{\text{in}}$. According to the first equation of (4.20) (Lemma 4.4), we have

$$\mathcal{H}_{n_i,j_i}^{m_i+1}\left(h_i, \phi_{n_i,j_i}^{m_i+1} + \xi_i, \left\{\phi_{l_i,k_i}^{m_i} + \xi_i\right\}\right) = F_{\text{in}}^{h_i}\left(\mathbf{x}_i, \phi(\mathbf{x}_i), D\phi(\mathbf{x}_i), D^2\phi(\mathbf{x}_i)\right) + r\xi_i + \mathcal{O}(h_i) \tag{4.27}$$

Using (4.18) and (4.27) gives, for each i,

$$\left|F(\mathbf{x}_i, \phi(\mathbf{x}_i), \cdot, \cdot) - \mathcal{H}_{n_i,j_i}^{m_i+1}\big(h_i, \phi_{n_i,j_i}^{m_i+1} + \xi_i, \{\phi_{l_i,k_i}^{m_i} + \xi_i\}\big)\right| \le C_i h_i + (r + c(\mathbf{x}_i))\xi_i + \mathcal{O}(h_i) \tag{4.28}$$

Here, $C_i > 0$ is a bounded constant and $|c(\mathbf{x}_i)| \le \max(r, 1)$ for all i. Thus, from (4.28), we have

$$\mathcal{H}_{n_i,j_i}^{m_i+1}\big(h_i, \phi_{n_i,j_i}^{m_i+1} + \xi_i, \{\phi_{l_i,k_i}^{m_i} + \xi_i\}\big) \le F\big(\mathbf{x}_i, \phi(\mathbf{x}_i), \cdot, \cdot\big) + C_i h_i + (r + c(\mathbf{x}_i))\xi_i + \mathcal{O}(h_i) \tag{4.29}$$

Combining (4.26) and (4.29), with continuity of $F(\cdot)$, we obtain

$$\begin{aligned}
\limsup_{\substack{h\to 0,\ \mathbf{x}\to\hat{\mathbf{x}}\\ \xi\to 0}} \mathcal{H}_{n,j}^{m+1}\left(h, \phi_{n,j}^{m+1} + \xi, \left\{\phi_{l,d}^m + \xi\right\}\right) &= \limsup_{i\to\infty} \mathcal{H}_{n_i,j_i}^{m_i+1}\left(h_i, \phi_{n_i,j_i}^{m_i+1} + \xi_i, \left\{\phi_{l_i,d_i}^{m_i} + \xi_i\right\}\right)\\
&\le \limsup_{i\to\infty} F\left(\mathbf{x}_i, \phi(\mathbf{x}_i), D\phi(\mathbf{x}_i), D^2\phi(\mathbf{x}_i)\right) + \limsup_{i\to\infty}(C_i h_i + (r + c(\mathbf{x}_i))\xi_i)\\
&= F^*\left(\hat{\mathbf{x}}, \phi(\hat{\mathbf{x}}), D\phi(\hat{\mathbf{x}}), D^2\phi(\hat{\mathbf{x}})\right).
\end{aligned}$$

This proves (4.23) for $\hat{\mathbf{x}} \in \Omega_{\text{in}}$. The case (4.23) for other sub-domains as well as the case (4.24) can be proved in a similar fashion. This concludes the proof. □

4.3 Monotonicity

We present a result on the monotonicity of scheme (4.9).

Lemma 4.6. *(Monotonicity) Scheme* (4.9) *satisfies*

$$\mathcal{H}_{n,j}^{m+1}\left(h, v_{n,j}^{m+1}, \{u_{l,d}^{m}\}\right) \leq \mathcal{H}_{n,j}^{m+1}\left(h, v_{n,j}^{m+1}, \{z_{l,d}^{m}\}\right) \tag{4.30}$$

for bounded $\left\{u_{l,d}^{m}\right\}$ *and* $\left\{z_{l,d}^{m}\right\}$ *having* $\left\{u_{l,d}^{m}\right\} \geq \left\{z_{l,d}^{m}\right\}$*, where the inequality is understood in the component-wise sense.*

Proof of Lemma 4.6. Since scheme (4.9) is defined case-by-case, to establish (4.30), we will show that each case satisfies (4.30). It is straightforward that the scheme satisfies (4.30) in Ω_{τ_0}) and Ω_{out}. Now we establish that $\mathcal{C}_{n,j}^{m+1}(\cdot)$, as defined in (4.8) for Ω_{in}, also satisfies (4.30). We have

$$\begin{aligned}
&\mathcal{C}_{n,j}^{m+1}\left(h, v_{n,j}^{m+1}, \{u_{l,d}^{m}\}\right) - \mathcal{C}_{n,j}^{m+1}\left(h, v_{n,j}^{m+1}, \{z_{l,d}^{m}\}\right) \\
&= \frac{1}{\Delta\tau}\left[\max_{\alpha} \Delta x \Delta y \sum_{l\in\mathbb{N}^{\dagger}}^{d\in\mathbb{J}^{\dagger}*} \varphi_{l,d}\; g_{n-l,j-d}^{\alpha}\; z_{l,d}^{m} - \max_{\alpha} \Delta x \Delta y \sum_{l\in\mathbb{N}^{\dagger}}^{d\in\mathbb{J}^{\dagger}*} \varphi_{l,d}\; g_{n-l,j-d}^{\alpha}\; u_{l,d}^{m}\right] \\
&\overset{\text{(i)}}{\leq} \frac{1}{\Delta\tau}\left[\max_{\alpha} \Delta x \Delta y \sum_{l\in\mathbb{N}^{\dagger}}^{d\in\mathbb{N}^{\dagger}*} \varphi_{l,d}\; g_{n-l,j-d}^{\alpha}\; \left(z_{l,d}^{m} - u_{l,d}^{m}\right)\right] \leq 0.
\end{aligned} \tag{4.31}$$

Here, (i) is due to the fact that, $\max_{\alpha\in\mathcal{A}} f_1(\alpha) - \max_{\alpha\in\mathcal{A}} f_2(\alpha) \leq \max_{\alpha}(f_1(\alpha) - f_2(\alpha))$ for two real-valued functions f_1, f_2 of α. This concludes the proof. □

Theorem 4.1 (Convergence to viscosity solution in Ω_{in})**.** *Suppose that all the conditions for Lemmas 4.1), 4.4 and 4.6 are satisfied. Our scheme* (4.9) *converges in* Ω_{in} *to the unique continuous viscosity solution of the two-factor uncertain volatility model pricing problem given in Definition* (2.2)*.*

Proof of Theorem 4.1. Our scheme is ℓ_∞-stable (Lemma 4.1), and consistent in the viscosity sense (Lemma 4.4) and monotone (Lemma 4.6). Since a strong comparison holds in Ω_{in} (Remark 2.2), by [3], convergence in Ω_{in} to the unique continuous vicosity solution of the HJB equation is ensured. □

5 Numerical experiments

This section presents the selected numerical results of our monotone piecewise constant control integration method (MPCCI) applied to the two-factor uncertain volatility model pricing problem. The modelling parameters for the tests carried out are given in Table 5.1, reproduced from [22][Table 3]. We note that specific ranges for σ_x, σ_y, and the correlation coefficient ρ are given.

5.1 Preliminary

Prior to initiating our experiments, it is essential to define a sufficiently large computational domain, guided by the boundary truncation error bound provided in Lemma 3.2. Specifically, we follow steps outlined in Remark 3.2 to determine $x_{\min}^{\dagger}$, $x_{\max}^{\dagger}$, $y_{\min}^{\dagger}$ and $y_{\max}^{\dagger}$. In particular, in (3.17), $\epsilon = 10^{-10}$ is used. With the model parameters given Table 5.1, this procedure gives $x_{\min} = \ln(X_0) - 1.2$, $x_{\max} = \ln(X_0) + 1.2$, $y_{\min} = \ln(Y_0) - 1.2$, $y_{\max} = \ln(Y_0) + 1.2$. Furthermore, for $z \in \{x, y\}$, the values of $z_{\min}^{\dagger}$, $z_{\max}^{\dagger}$, $z_{\min}^{\ddagger}$ and $z_{\max}^{\ddagger}$ are determined as in (3.19)-(3.20). Extensive testing indicates that larger intervals have negligible impact on numerical solutions, whereas smaller domains exhibit minor variations. These findings are numerically validated in Subsection 5.3 Unless noted otherwise, the specifics of mesh size and timestep refinement levels utilized in all experiments are detailed in Table 5.2.

Parameter	Value/ Range	Parameter	Value/ Range
T	0.25 (years)	X_0	40
r	0.05	Y_0	40
σ_x	[0.3, 0.5]	K	40
σ_y	[0.3, 0.5]	K_1	34
ρ	[0.3, 0.5]	K_2	46

TABLE 5.1: *Model parameters used in numerical experiments for two-factor uncertain volatility model-reproduced from [22] Table 3.*

Level	N (x)	J (y)	M (τ)	Q (α)
0	2^7	2^7	50	8
1	2^8	2^8	100	24
2	2^9	2^9	200	56
3	2^{10}	2^{10}	400	120
4	2^{11}	2^{11}	800	248

TABLE 5.2: *Grid and timestep refinement levels for numerical tests.*

Our MPCCI numerical prices are verified against those produced by: (i) closed-form solutions (for certain European rainbow options), (ii) FD methods reported in the literature, particularly the unconditionally monotone FD method of [22], (iii) tree-grid (TG) methods of [17], and (iii) Monte Carlo (MC) simulation. The Monte Carlo validation is carried out in two steps

1. Step 1: we solve the two-factor uncertain volatility pricing problem using the proposed MPCCI on a fine computational grid (comprising of 2^{10} x-nodes, 2^{10} y-nodes, and 400 timesteps). At each time-τ_m, we store the optimal controls or all pair of discrete states (x_n, y_j), denoted as $\{(\alpha^*)^m_{n,j}\} \equiv \{(\sigma_x^*, \sigma_y^*, \rho^*)^m_{n,j}\}$, where $n \in \mathbb{N}^\dagger$, $j \in \mathbb{J}^\dagger$, and $m = 0, \ldots, M$.

2. Step 2: we conduct Monte Carlo simulations of the two-dimensional dynamics (2.1) from $t = 0$ to $t = T$, following the stored MPCCI-computed optimal controls. For a given pair of simulated values of X and Y, linear interpolation, if necessary, is used to determine the control. Specifically, for the γ-th X and Y simulated values at time-τ_m, denoted by $\hat{X}_m^{(\gamma)}$ and $\hat{Y}_m^{(\gamma)}$, and given $x_{n'} \le \hat{X}_m^{(\gamma)} \le x_{n'+1}$ and $y_{j'} \le \hat{Y}_m^{(\gamma)} \le y_{j'+1}$, we interpolate the optimal control $(\alpha^*)^m_{n',j'}$, $(\alpha^*)^m_{n'+1,j}$, $(\alpha^*)^m_{n',j'+1}$, and $(\alpha^*)^m_{n'+1,j'+1}$ to determine the volatilities $\tilde{\sigma}_x^m$ and $\tilde{\sigma}_y^m$ and the correlation coefficient $\tilde{\rho}_m$ for the interval $[\tau_m, \tau_{m+1}]$. The Euler-Maruyama discretization is then applied for each the interval $[\tau_m, \tau_{m+1}]$ as follows:

$$\hat{X}_{m+1}^{(\gamma)} = \hat{X}_m^{(\gamma)} \left(1 + r\Delta t + \tilde{\sigma}_x^m \sqrt{\Delta t}\, \xi_x^{(\gamma)}\right),$$
$$\hat{Y}_{m+1}^{(\gamma)} = \hat{Y}_m^{(\gamma)} \left(1 + r\Delta t + \tilde{\sigma}_y^m \sqrt{\Delta t}\, (\tilde{\rho}_m \xi_x^{(\gamma)} + \sqrt{1 - (\tilde{\rho}_m)^2}\, \xi_y^{(\gamma)})\right),$$

where $\xi_x^{(\gamma)}$ and $\xi_y^{(\gamma)}$ are independent standard normal random variables. The option value is approximated by $\frac{e^{-rT}}{\Gamma} \sum_{\gamma=1}^{\Gamma} p\big(\hat{X}_M^{(\gamma)}, \hat{Y}_M^{(\gamma)}\big)$, with $p(\cdot,\cdot)$ as the payoff function, using a total of $\Gamma = 10^6$ simulation paths.

5.2 Validation examples

5.2.1 European call options

Our first test case evaluates a European call option on the maximum of two assets, as described in [22]. The payoff function $p(e^x, e^y)$ is given by

$$p(e^x, e^y) = \max(\max(e^x, e^y) - K, 0), \quad K > 0. \tag{5.1}$$

For worst-case for the short position, the optimal control $\{(\alpha^*)\}$ at any point (x, y, τ) is mathematically determined to be $\sigma_x^* = \sigma_{\max}^x$, $\sigma_y^* = \sigma_{\max}^y$, and $\rho^* = \rho_{\min}$. Given the parameters in (5.1), this translates to $\sigma_x^* = \sigma_y^* = 0.5$ and $\rho^* = 0.3$. The exact option price is thus analytically computable using the closed-form expression from [32].

Level	Price	Abs. error	Ratio
0	6.84492756	2.77e-03	
1	6.84700690	6.93e-04	4.00
2	6.84752662	1.73e-05	4.00
3	6.84765654	4.33e-05	4.00
4	6.84768902	1.08e-05	4.00
Ref. [32]	6.84769986		
MC: 95%-CI	[6.8319, 6.8618]		

TABLE 5.3: *Convergence study for a European call option on the maximum of two risky assets under two-factor uncertain volatility model (worst-case) - payoff function in* (5.1). *The closed-form solution is obtained using [32] with fixed parameters* $\sigma_x^* = 0.5$, $\sigma_y^* = 0.5$ *and* $\rho^* = 0.3$.

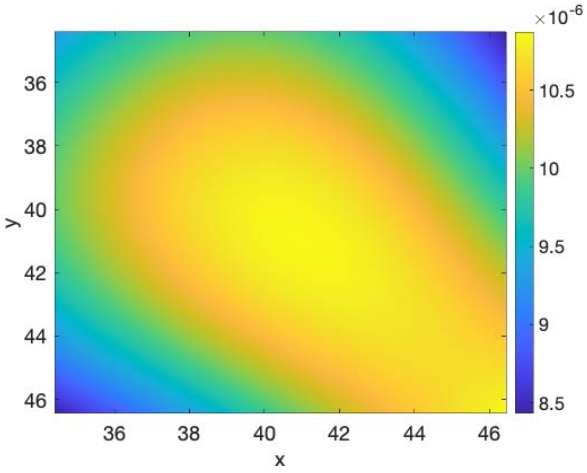

FIGURE 5.1: *Absolute error on* Ω *associated with test case reported in Table 5.3.*

Despite knowing the optimal control, we discretize the admissible control set in our experiments for generality, i.e. $\mathcal{A}_h$ is used and it contains the optimal control $\{(\alpha^*)\} = \{\sigma^x_{\max}, \sigma^y_{\max}, \rho_{\min}\}$ at all refinement levels. It is observed that the proposed MPCCI scheme accurately yielded the aforementioned optimal control for all (x_n, y_j, τ_m). Table 5.3 shows the convergence results for the time $t = 0$ option price at $e^x = X_0$, $e^y = Y_0$. To provide an estimate of the convergence rate of the proposed MPCCI method, we compute the "Abs. error" as the absolute error between the exact option [32] and numerical option prices, and the "Ratio" as the ratio of successive absolute errors. These results indicate excellent agreement with the analytic solution from [32], as do the results from MC simulations. Notably, the MPCCI method exhibits second-order convergence.

We further demonstrates the accuracy of the MPCCI method in the entire domain Ω. In Figure 5.1, we present the absolute error at time $t = 0$ on grid points obtained with refinement Level 4. The absolute error, computed as

$$\left|v(x_n, y_j, \tau_M) - v^M_{n,j}\right|, \quad n \in \mathbb{N}^\dagger, \quad j \in \mathbb{J}^\dagger, \quad \tau_M = T,$$

is very small across the computational domain, typically of the order of 10^{-5} or less, with higher errors concentrated near the strike $K = 40$, as expected.

In Table 5.4 and Figure 5.2, we display the best-case results for the short position. The outcomes closely mirror those of the worst-case scenario, showing excellent agreement with the closed-form solution, exhibiting second-order of convergence.

The observed second-order convergence in Tables 5.3 and 5.4 can be explained as followed. Recall that in the worst-case scenario, the optimal control α^* at any point (x, y, τ) is always $\{\sigma^x_{\max}, \sigma^y_{\max}, \rho_{\min}\}$, while for the best-case, it becomes $\{\sigma^x_{\min}, \sigma^y_{\min}, \rho_{\max}\}$. Since α^* is included in $\mathcal{A}_h$ (at every refinement level) and is obtained by solving the optimization problem at each timestep, the discretization error from approximating $\mathcal{A}$ by $\mathcal{A}_h$ is eliminated. In addition, for a refinement level, the constancy of α^* results in a single Green's function, namely $g_{\alpha^*}(\cdot; \Delta\tau)$, for all timesteps, eliminating timestepping error thanks to the additive nature of integration. Therefore, the only error stems from numerical integration, where the second-order accuracy is observed. This second-order convergence is expected from the composite trapezoidal rule because gridpoints aligns with the kinks of the payoff function around the line $x = y$, $x = K$ and $y = K$.

Level	Price	Abs. error	Ratio
0	3.96880850	4.80e-03	
1	3.97240621	1.20e-03	4.00
2	3.97330502	3.00e-04	4.00
3	3.97352968	7.49e-05	4.00
4	3.97358584	1.87e-05	4.00
MC: 95%-CI	[3.9657, 3.9840]		
Ref. [32]	3.97360457		

TABLE 5.4: *Convergence study of an European call option on the maximum of two risky assets under two-factor uncertain volatility model (best-case) - payoff function in* (5.1)*. The closed-form solution is obtained using [32] with fixed parameters* $\sigma_x = 0.3$, $\sigma_y = 0.3$ *and* $\rho = 0.5$.

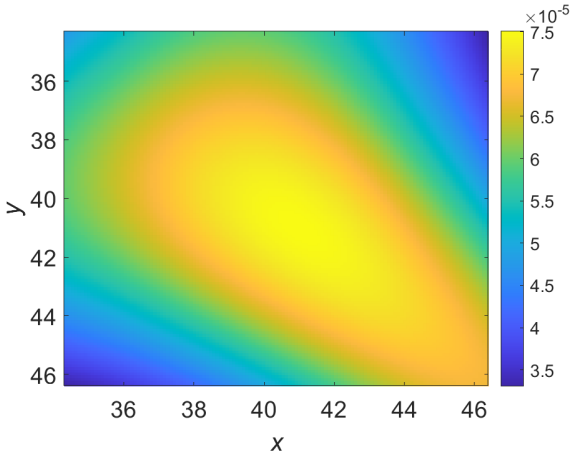

FIGURE 5.2: *Absolute error on* Ω *associated with test case reported in Table 5.4.*

Level	Price	Change	Ratio
0	2.65092717		
1	2.66374754	0.0128	
2	2.67280480	0.0091	1.42
3	2.67793762	0.0051	1.76
4	2.68070303	0.0028	1.86
MC: 95%-CI	[2.6735, 2.6832]		
FD [22]	2.6744		
TG [17]	2.6784		

TABLE 5.5: *Convergence study for a butterfly option (worst-case) under a two-factor uncertain volatility model - payoff function in* (5.2)*. Reference prices: by FD method is 2.6744 [22] (finest level in Table 6 therein, Pure wide stencil (with rotation)), by TG method is 2.6784 [17] (finest level in Table 3 therein)*

Level	Price	Change	Ratio
0	0.94015237		
1	0.92418409	-0.0138	
2	0.91794734	-0.0062	2.56
3	0.91473085	-0.0032	1.94
4	0.91308945	-0.0016	1.96
MC : 95%-CI	[0.9120, 0.9182]		
FD [22]	0.9148		
TG [17]	0.9173		

TABLE 5.6: *Convergence study for a butterfly option (best-case) under a two-factor uncertain volatility model - payoff function in* (5.2)*. Reference prices: (i) by FD method is 0.9148 [22] (finest level in Table 8 therein, Pure wide stencil (with rotation)), (ii) by TG method is 0.9173 [17] (finest level in Table 4 therein).*

5.2.2 Butterfly options

In the second test, we consider a butterfly option on the maximum of two assets. For this option, the payoff function $p(e^x, e^y)$ is given by

$$p(e^x, e^y) = \max\big(\max(e^x, e^y) - K_1, 0\big) - 2\max\big(\max(e^x, e^y) - (K_1 + K_2)/2, 0\big) + \max\big(\max(e^x, e^y) - K_2, 0\big), \quad K_1, K_2 > 0. \tag{5.2}$$

For the butterfly payoff function (5.2), a closed-form expression for the option price is unknown. To estimate the convergence rate of the proposed MPCCI method, we calculate the "Change" as the difference in values from coarser to finer grids and the "Ratio" as the ratio of changes between successive grids. We compare our prices against reference prices obtained by a FD method with pure wide stencil rotation developed in [22], and by a tree-grid (TG) method of [17].

Tables 5.5 and 5.6 display the numerical prices for both the worst-case and best-case scenarios (short position), indicating that our method approximates first-order convergence. The comparison

with reference prices shows minimal differences: against FD method prices, the discrepancies are around 6×10^{-3} for the worst-case and 1×10^{-3} for the best-case scenarios. When compared with TG prices, the differences are 2×10^{-3} and 4×10^{-3}, respectively, highlighting the MPCCI method's precision.

5.3 Impact of spatial domain sizes

In this subsection, we numerically validate the adequacy of our selected spatial domain for the experiments. We revisit the scenarios from Tables 5.3, 5.5, and 5.6, this time doubling the lengths of the spatial domains. Specifically, we extend the spatial domain boundaries to $x_{\min} = \ln(X_0) - 2.4$, $x_{\max} = \ln(X_0) + 2.4$, $y_{\min} = \ln(Y_0) - 2.4$, $y_{\max} = \ln(Y_0) + 2.4$, with the number of intervals N and J also doubled to maintain the same Δx and Δy.

The numerical prices from this extended domain, shown in Table 5.7, are virtually identical with those obtained from the original smaller domain (reproduced under columns marked Tab. 5.3, Tab. 5.5, and Tab. 5.6). This indicates that enlarging the spatial computational domain further has a negligible effect on the numerical prices. Additionally, for a comprehensive analysis, we conducted tests on smaller spatial domains with the boundaries set to $x_{\min} = \ln(X_0) - 0.9$, $x_{\max} = \ln(X_0) + 0.9$, $y_{\min} = \ln(Y_0) - 0.9$, and $y_{\max} = \ln(Y_0) + 0.9$, using the same Δx and Δy as in previous tests. The prices, presented in Table 5.8, show slight discrepancies (from the fourth decimal digits) when compared to those obtained from original domain size.

These findings affirm the adequacy of our computational domain, whose size was carefully chosen based on the upper bound for the boundary truncation error of the Green's function provided in (3.16). This approach effectively balance the need for demonstrating theoretical convergence and computational efficiency in our analysis.

Level	Two-factor uncertain volatility model					
	European		Butterfly (worst)		Butterfly (best)	
	Price	Price (Tab. 5.3)	Price	Price (Tab. 5.5)	Price	Price (Tab. 5.6)
0	6.84492758	6.84492756	2.65092717	2.65092717	0.94015237	0.94015237
1	6.84700691	6.84700690	2.66374754	2.66374754	0.92418409	0.92418409
2	6.84752663	6.84752662	2.67280480	2.67280480	0.91794734	0.91794734
3	6.84765655	6.84765654	2.67793762	2.67793762	0.91473085	0.91473085
4	6.84768903	6.84768902	2.68070303	2.68070303	0.91308945	0.91308945

TABLE 5.7: *Prices obtained using a larger spatial computational domain:* $x_{\min} = \ln(X_0) - 2.4$, $x_{\max} = \ln(X_0) + 2.4$, $y_{\min} = \ln(Y_0) - 2.4$, $y_{\max} = \ln(Y_0) + 2.4$, *in comparison with prices in Table 5.3, 5.5, 5.6 obtained with the original smaller domain* $z_{\min} = \ln(Z_0) - 1.2$, $z_{\max} = \ln(Z_0) + 1.2$, *for* $z \in \{x, y\}$.

5.4 Impact of boundary conditions

In this subsection, we numerically demonstrate that our straightforward approach of employing discounted payoffs for boundary sub-domains is adequate. We revisited previous experiments reported in Tables 5.3, 5.5, and 5.6, introducing sophisticated boundary conditions based on the asymptotic behavior of the HJB equation (2.6) as $z \to -\infty$ and $z \to \infty$ for $z \in \{x, y\}$ as proposed in [22]. Specifically, the HJB equation (2.6) simplifies to the 1D forms shown in (5.3) when x or y tends to $-\infty$:

$$\begin{aligned} v_\tau - \sup_{\sigma_y \in \mathcal{A}_y} \left\{ \left(r - (\sigma_y)^2/2\right) v_y + (\sigma_y)^2/2 v_{yy} \right\} + rv = 0, \quad x \to -\infty, \\ v_\tau - \sup_{\sigma_x \in \mathcal{A}_x} \left\{ \left(r - (\sigma_x)^2/2\right) v_x + (\sigma_x)^2/2 v_{xx} \right\} + rv = 0, \quad y \to -\infty. \end{aligned} \tag{5.3}$$

Level	Two-factor uncertain volatility model					
	European		Butterfly (worst)		Butterfly (best)	
	Price	Price (Tab. 5.3)	Price	Price (Tab. 5.5)	Price	Price (Tab. 5.6)
0	6.84490660	6.84492756	2.65092891	2.65092717	0.94014513	0.94015237
1	6.84698614	6.84700690	2.66375021	2.66374754	0.92417329	0.92418409
2	6.84750559	6.84752662	2.67280820	2.67280480	0.91793374	0.91794734
3	6.84763508	6.84765654	2.67794155	2.67793762	0.91471524	0.91473085
4	6.84766711	6.84768902	2.68070724	2.68070303	0.91307245	0.91308945

TABLE 5.8: *Prices obtained using a smaller computational domain:* $x_{\min} = \ln(X_0) - 0.9$, $x_{\max} = \ln(X_0) + 0.9$. $y_{\min} = \ln(Y_0) - 0.9$, *and* $y_{\max} = \ln(Y_0) + 0.9$. *Compare with prices in Table 5.3, 5.5, 5.6, where* $z_{\min} = \ln(Z_0) - 1.2$, $z_{\max} = \ln(Z_0) + 1.2$, *for* $z \in \{x, y\}$.

As $x, y \to -\infty$, the HJB equation (2.6) simplifies to the ordinary differential equation $v_\tau + rv = 0$.

To adhere to these asymptotic boundary conditions, we choose a much large spatial domain: $x_{\min} = \ln(X_0) - 9.6$, $x_{\max} = \ln(X_0) + 9.6$, $y_{\min} = \ln(Y_0) - 9.6$, $y_{\max} = \ln(Y_0) + 9.6$, and adjust the number of intervals N and J accordingly to maintain the same grid resolution (Δx and Δy). Employing the monotone piecewise constant control integration technique, tailored for the 1D case, we effectively solved the 1D HJB equations in (5.3). The ordinary differential equation $v_\tau + rv = 0$ is solved directly and efficiently. The scheme's convergence to the viscosity solution can be rigorously established in the same fashion as the propose scheme.

The resulting option prices, listed in Table 5.9, are virtually identical with those from the original settings (under columns marked with Tab. 5.3, Tab. 5.5, and Tab. 5.6). These results confirm the effectiveness of our simple boundary conditions, demonstrating that they are both easy to implement and sufficient for the theoretical and practical demands of our numerical experiments.

Level	Two-factor uncertain volatility model					
	European		Butterfly (worst)		Butterfly (best)	
	Price	Price (Tab. 5.3)	Price	Price (Tab. 5.5)	Price	Price (Tab. 5.6)
0	6.84492760	6.84492756	2.65092717	2.65092717	0.94015237	0.94015237
1	6.84700690	6.84700690	2.66374754	2.66374754	0.92418410	0.92418409
2	6.84752663	6.84752662	2.67280480	2.67280480	0.91794734	0.91794734
3	6.84765655	6.84765654	2.67793762	2.67793762	0.91473085	0.91473085
4	6.84768903	6.84768902	2.68070303	2.68070303	0.91308945	0.91308945

TABLE 5.9: *Results using sophisticated boundary conditions* (5.3). *Compare with results in Table 5.3, 5.5, 5.6 where simple boundary conditions based on discounted payoffs are used.*

6 Conclusion

In this paper, we have presented a novel and streamlined approach to solving two-dimensional (2D) Hamilton-Jacobi-Bellman (HJB) Partial Differential Equations (PDEs) arising from two-factor uncertain volatility models with uncertain correlation.

Departing from the traditional "discretize, then optimize" strategy, our "decompose and integrate, then optimize" method utilizes a piecewise constant control technique to decompose the HJB PDE into independent 2D linear PDEs at each timestep, significantly simplifying the optimization process. Our main contributions include developing a monotone piecewise constant control numerical integration

scheme that employs Green's functions for solving independent linear 2D PDEs. This approach simplifies the treatment of cross-derivative terms, offering an advantage over traditional finite difference methods. We have implemented our scheme efficiently using FFTs and circulant convolution, utilizing the Toeplitz matrix structure to expand convolution kernels into circulant matrices for both inner and double summations. This enables efficient computation through 2D FFTs. We mathematically demonstrate of the stability and consistency of the scheme in the viscosity sense, along with its pointwise convergence to the viscosity solution of the HJB PDE. Extensive numerical results showcasing the effectiveness of our approach align remarkably with benchmark solutions.

While our focus has been on models with uncertain volatilities, our comprehensive and systematic approach provides a framework for developing similar integration methods for other HJB PDEs in finance, demonstrating its broad applicability and potential for future exploration.

References

[1] M. Avellaneda, A. Levy, and A. Parás. Pricing and hedging derivative securities in markets with uncertain volatilities. *Applied Mathematical Finance*, 2:73–88, 1995.

[2] Marco Avellaneda and Robert Buff. Combinatorial implications of nonlinear uncertain volatility models: the case of barrier options. *Applied Mathematical Finance*, 6(1):1–18, 1999.

[3] G. Barles and P.E. Souganidis. Convergence of approximation schemes for fully nonlinear equations. *Asymptotic Analysis*, 4:271–283, 1991.

[4] O. Bokanowski, S. Maroso, and H. Zidani. Some convergence results for Howard's algorithm. *SIAM Journal on Numerical Analysis*, 47:3001–3026, 2009.

[5] J Frédéric Bonnans and Housnaa Zidani. Consistency of generalized finite difference schemes for the stochastic HJB equation. *SIAM Journal on Numerical Analysis*, 41(3):1008–1021, 2003.

[6] K. Debrabant and E.R. Jakobsen. Semi-Lagrangian schemes for linear and fully non-linear diffusion equations. *Mathematics of Computation*, 82:1433–1462, 2013.

[7] Nikolai G Dokuchaev and Andrey V Savkin. The pricing of options in a financial market model with transaction costs and uncertain volatility. *Journal of Multinational Financial Management*, 8(2-3):353–364, 1998.

[8] Dean G. Duffy. *Green's Functions with Applications*. Chapman and Hall/CRC, New York, 2nd edition, 2015.

[9] P. A. Forsyth and G. Labahn. Numerical methods for controlled Hamilton-Jacobi-Bellman PDEs in finance. *Journal of Computational Finance*, 11(2):1, 2007.

[10] M. G. Garroni and J. L. Menaldi. *Green functions for second order parabolic integro-differential problems*. Number 275 in Pitman Research Notes in Mathematics. Longman Scientific and Technical, Harlow, Essex, UK, 1992.

[11] Andrés Alayón Glazunov and Jie Zhang. A note on the bivariate distribution representation of two perfectly correlated random variables by Dirac's δ-function. *arXiv preprint arXiv:1205.0933*, 2012.

[12] J. Guyon and P. Henry-Labordere. Uncertain volatility model: a Monte-Carlo approach. *Journal of Computational Finance*, 14:37–74, 2011.

[13] Patrick S Hagan, Deep Kumar, Andrew S Lesniewski, and Diana E Woodward. Managing smile risk. *The Best of Wilmott*, 1:249–296, 2002.

[14] S. Heston. A closed form solution for options with stochastic volatility with applications to bond and currency options. *Review of Financial Studies*, 6:327–343, 1993.

[15] Bartosz Jaroszkowski and Max Jensen. Valuation of European options under an uncertain market price of volatility risk. *Applied Mathematical Finance*, 29(3):213–226, 2022.

[16] Bartosz Jaroszkowski and Max Jensen. Valuation of European options under an uncertain market price of volatility risk. *Applied Mathematical Finance*, 29(3):213–226, 2022.

[17] Igor Kossaczkỳ, Matthias Ehrhardt, and Michael Günther. The 2D tree–grid method. *Journal of Computational Finance*, 22(12):29–57, 2019.

[18] N.V. Krylov. Approximating value functions for controlled degenerate diffusion processes by using piecewise constant policies. *Electronic Journal of Probability*, 4(2):1–19, 1999.

[19] J.M. Lee. *Introduction to smooth manifolds*. Springer-Verlag, New York, 2 edition, 2012.

[20] T. Lyons. Uncertain volatility and the risk free synthesis of derivatives. *Applied Mathematical Finance*, 2:117–133, 1995.

[21] T. J. Lyons. Uncertain volatility and the risk-free synthesis of derivatives. *Applied Mathematical Finance*, 2(2):117–133, 1995.

[22] K. Ma and P.A. Forsyth. An unconditionally monotone numerical scheme for the two-factor uncertain volatility model. *IMA Journal of Numerical Analysis*, 37(2):905–944, 2017.

[23] A.M. Oberman. Convergent difference schemes for degenerate elliptic and parabolic equations: Hamilton–Jacobi Equations and free boundary problems. *SIAM Journal Numerical Analysis*, 44(2):879–895, 2006.

[24] H. Pham. On some recent aspects of stochastic control and their applications. *Probability Surveys*, 2:506–549, 2005.

[25] David M Pooley, Peter A Forsyth, and Ken R Vetzal. Two factor option pricing with uncertain volatility. In *International Conference on Computational Science and Its Applications*, pages 158–167. Springer, 2003.

[26] D.M. Pooley, P.A. Forsyth, and K.R. Vetzal. Numerical convergence properties of option pricing PDEs with uncertain volatility. *IMA Journal of Numerical Analysis*, 23:241–267, 2003.

[27] C. Reisinger and P.A. Forsyth. Piecewise constant policy approximations to Hamilton-Jacobi-Bellman equations. *Applied Numerical Mathematics*, 103:27–47, 2016.

[28] Walter Rudin. *Principles of mathematical analysis*. 1953.

[29] Marjon J Ruijter and Cornelis W Oosterlee. Two-dimensional Fourier cosine series expansion method for pricing financial options. *SIAM Journal on Scientific Computing*, 34(5):B642–B671, 2012.

[30] Adam T Smith. American options under uncertain volatility. *Applied Mathematical Finance*, 9(2):123–141, 2002.

[31] Jerome Spanier, Keith B Oldham, and Robert H Romer. An atlas of functions, 1988.

[32] R. Stulz. Options on the minimum or the maximum of two risky assets: Analysis and applications. *Journal of Financial Economics*, 10(2):161–185, 1982.

[33] A.Dembo W. Bryc and T. Jiang. Spectral measure of large random Hankel, Markov and Toeplitz matrices. 2006.

[34] J. Wang and P.A. Forsyth. Maximal use of central differencing for Hamilton-Jacobi-Bellman PDEs in finance. *SIAM Journal on Numerical Analysis*, 46:1580–1601, 2008.

[35] Xavier Warin. Some non-monotone schemes for time dependent Hamilton-Jacobi-Bellman equations in stochastic control. *Journal of Scientific Computing*, 66(3):1122–1147, 2016.

[36] R. Willink. Bounds on the bivariate normal distribution function. *Communications in Statistics-Theory and Methods*, 33(10):2281–2297, 2005.

[37] H. Zhang and D.M. Dang. A monotone numerical integration method for mean-variance portfolio optimization under jump-diffusion models. *Mathematics and Computers in Simulation*, 219:112–140, 2024.

Appendices

A Special case $\rho = \pm 1$

A.1 Approximation of $\delta(\cdot)$

In this appendix, we detail key elements of the proposed scheme for the case $\rho = \pm 1$ as highlighted in Remark 3.1, and provide selected numerical results. The key challeng is that, for computational purposes, the Dirac delta function $\delta(y - (a + \rho bx))$, $a = \mu_y - \rho b \mu_x$ with $b = \frac{\sigma_y}{\sigma_x}$, needs to be approximated. We focus on the case $\rho = 1$. The analysis for $\rho = -1$ follows similarly and is omitted. Using a conditional density approach, $\delta(y-(a+\rho bx))$ is approximated by a Gaussian (a conditional density) when the correlation coefficient is $\hat{\rho}$ with $\hat{\rho} \nearrow 1$ [11]:

$$\delta(y - (a + \rho bx)) = \lim_{\hat{\rho} \nearrow 1} \delta_{\hat{\rho}}(y - (a + \rho bx)), \text{ where } \delta_{\hat{\rho}}(\gamma) = \frac{\exp\left(-\frac{\gamma^2}{2\kappa_y^2(1-\hat{\rho}^2)}\right)}{\sqrt{2\pi}\kappa_y\sqrt{1-\hat{\rho}^2}}, \text{ and } \kappa_y = \sigma_y\sqrt{\Delta\tau}. \tag{A.1}$$

For a fixed α, recall the exact Green's function $g_\alpha(x, y; \Delta\tau)$ defined in (3.13). We define by $\hat{g}_\alpha(x, y; \Delta\tau)$ an approximation to $g_\alpha(x, y; \Delta\tau)$ obtained by replacing $\delta(y - (a + \rho bx))$ by $\delta_{\hat{\rho}}(y - (a + \rho bx))$. Formally,

$$\hat{g}_\alpha(x, y; \Delta\tau) = e^{-r\Delta\tau} \frac{1}{\sqrt{2\pi}\kappa_x} \exp\left(-\frac{(x - \mu_x)^2}{2\kappa_x^2}\right) \delta_{\hat{\rho}}(y - (a + \rho bx)), \tag{A.2}$$

where $\delta_{\hat{\rho}}(\cdot)$ is defined in (A.1), and $a = \mu_y - \rho b \mu_x$ and $b = \frac{\sigma_y}{\sigma_x}$. The function $\hat{g}_\alpha(x, y; \Delta\tau)$ is the weight function for our scheme. In this case, our scheme is monotone, and it is straightforward to show that it is ℓ_∞-stable. The selection of $\hat{\rho}$ is crucial for the scheme's consistency. Below, we show that choosing $\hat{\rho}$ appropriately can achieve first-order consistency for the scheme.

A.2 Consistency

For the rest of the proof, we let C be generic bounded constant independent of the discretization parameter h, which may take different values from line to line. We re-examine the proof of Lemma 4.2, now utilising $\hat{g}_\alpha(x, y; \Delta\tau)$ from (A.2) instead of $g_\alpha(x, y; \Delta\tau)$. For a smooth test function ϕ, and recalling the smooth function $\chi \in L^1(\mathbb{R}^2)$ with bounded derivatives up to second-order in $\mathbb{R}^2$, a mollified version of ϕ, we have

$$\begin{aligned}
\Delta x \Delta y \sum_{l \in \mathbb{N}^\dagger}^{d \in \mathbb{N}^\dagger *} \varphi_{l,d}\ \hat{g}^\alpha_{n-l,j-d}\ \phi^m_{l,d} \ &= \iint_{\mathbb{R}^2} \hat{g}_\alpha\left(x_n - x, y_j - y; \Delta\tau\right)\ \chi(x, y)\ dx\ dy + \mathcal{O}(h^2) + \mathcal{O}\big(he^{-1/h}\big) \\
&= \iint_{\mathbb{R}^2} g_\alpha\left(x_n - x, y_j - y; \Delta\tau\right)\ \chi(x, y)\ dx\ dy \\
&+ \iint_{\mathbb{R}^2} \left(\hat{g}_\alpha(x_n - x, y_j - y; \Delta\tau) - g_\alpha(x_n - x, y_j - y; \Delta\tau)\right)\ \chi(x, y)\ dx\ dy + \mathcal{O}(h^2). \qquad \text{(A.3)}
\end{aligned}$$

We now focus on the error term (the second term) in (A.3), expressed through substitutions as

$$\int_{\mathbb{R}} \frac{e^{-r\Delta\tau}}{\sqrt{2\pi}\kappa_x} \exp\left(-\frac{(x - \mu_x)^2}{2\kappa_x^2}\right) \left(\int_{\mathbb{R}} \left(\delta_{\hat{\rho}}(y - (a + \rho b x)) - \delta(y - (a + \rho b x))\right)\ \chi(x_n - x, y_j - y) dy \right) dx. \qquad \text{(A.4)}$$

Regarding the inner integral, we have $\int_{\mathbb{R}} \left(\delta_{\hat{\rho}}(y - (a + \rho b x)) - \delta(y - (a + \rho b x))\right)\ \chi(x_n - x, y_j - y) dy = \ldots$

$$\ldots = \int_{\mathbb{R}} \delta_{\hat{\rho}}(y - (a + \rho b x))\ \chi(x_n - x, y_j - y)\ dy - \chi(x_n - x, y_j - (a + \rho b x)). \qquad \text{(A.5)}$$

Here, the second term in (A.5) is due to from the sifting property of the Delta function. Letting $\gamma = y_j - (a + \rho b x)$ and applying a change of variables, the integral in (A.5) is reformulated as

$$\int_{\mathbb{R}} \delta_{\hat{\rho}}\big(y - \gamma\big) \chi(x_n - x, y) dy. \qquad \text{(A.6)}$$

By Taylor's series expansion, we have

$$\chi(\cdot, y) = \chi(\cdot, \gamma + (y - \gamma)) = \chi(\cdot, \gamma) + (y - \gamma)\frac{\partial \chi}{\partial y}(\cdot, \gamma) + \frac{(y - \gamma)^2}{2}\frac{\partial^2 \chi}{\partial y^2}(\cdot, \gamma) + o((y - \gamma)^2).$$

So $\int_{\mathbb{R}} \delta_{\hat{\rho}}\big(y - \gamma\big) \chi(x_n - x, y) dy = \ldots$

$$\ldots = \int_{\mathbb{R}} \delta_{\hat{\rho}}\big(y - \gamma\big) \left(\chi(x_n - x, \gamma) + (y - \gamma)\frac{\partial \chi}{\partial y}(x_n - x, \gamma) + \frac{(y - \gamma)^2}{2}\frac{\partial^2 \chi}{\partial y^2}(x_n - x, \gamma) + o((y - \gamma)^2) \right) dy. \qquad \text{(A.7)}$$

Terms in (A.7) are further simplified as follows

$$\begin{aligned}
\int_{\mathbb{R}} \delta_{\hat{\rho}}\big(y - \gamma\big)\ \chi(\cdot, \gamma)\ dy &= \chi(\cdot, \gamma) \int_{\mathbb{R}} \delta_{\hat{\rho}}\big(y - \gamma\big)\ dy = \chi(\cdot, \gamma), \\
\int_{\mathbb{R}} \delta_{\hat{\rho}}\big(y - \gamma\big)(y - \gamma)\frac{\partial \chi}{\partial y}(\cdot, \gamma)\ dy &= \frac{\partial \chi}{\partial y}(\cdot, \gamma) \int_{\mathbb{R}} \delta_{\hat{\rho}}\big(y - \gamma\big)(y - \gamma)\ dy = 0, \qquad \text{(A.8)} \\
\int_{\mathbb{R}} \delta_{\hat{\rho}}\big(y - \gamma\big)\frac{(y - \gamma)^2}{2}\frac{\partial^2 \chi}{\partial y^2}(\cdot, \gamma)\ dy &= \frac{\partial^2 \chi}{\partial y^2}(x, \gamma) \int_{\mathbb{R}} \delta_{\hat{\rho}}\big(y - \gamma\big)\frac{(y - \gamma)^2}{2}\ dy = \frac{\kappa_y^2(1 - \hat{\rho}^2)}{4}\ \frac{\partial^2 \chi}{\partial y^2}(\cdot, \gamma).
\end{aligned}$$

Next, we substitute (A.8) into (A.6) which is the first term in (A.5), noting that the term $\chi(\cdot, \gamma)$ in (A.8) is indeed $\chi(x_n - x, y_j - (a + \rho b x))$ and it cancels with the term $\chi(x_n - x, y_j - (a + \rho b x))$ in (A.5). Therefore, due to boundedness of the derivatives of $\chi(\cdot)$, the error term (A.4) becomes $C\kappa_y^2(1 - \hat{\rho}^2)$. Therefore, noting $\kappa_y = \sigma_y\sqrt{\Delta\tau}$, (A.3) becomes

$$\Delta x \Delta y \sum_{l \in \mathbb{N}^\dagger}^{d \in \mathbb{N}^\dagger *} \varphi_{l,d}\ \hat{g}^\alpha_{n-l,j-d}\ \phi^m_{l,d} \ = \iint_{\mathbb{R}^2} g_\alpha\left(x_n - x, y_j - y; \Delta\tau\right)\ \chi(x, y)\ dx\ dy + C\Delta\tau(1 - \hat{\rho}^2) + \mathcal{O}(h^2). \qquad \text{(A.9)}$$

Now we re-examine Lemma 4.4 with (A.9) in mind. Here, since we need to achieve $C\Delta\tau(1 - \hat{\rho}^2)/h \to 0$ as $h \to 0$, $\hat{\rho}$ needs to be such that $(1 - \hat{\rho}^2) \to 0$ as $h \to 0$. A possible choice is $\hat{\rho} = \sqrt{1 - Ch}$, which gives $(1 - \hat{\rho}^2) = \mathcal{O}(h)$, and we obtain the same overall $\mathcal{O}(h)$ error as in Lemma 4.4 for scenarios $|\rho| < 1$.

A.3 Select numerical experiments

Level	Price	Error	Ratio
0	8.41173784	3.67e-03	
1	8.41450094	9.07e-04	4.00
2	8.41519144	2.16e-04	4.19
3	8.41536400	5.15e-05	4.20
4	8.41540714	1.17e-05	4.40
Ref. [32]	8.41540757		
MC: 95%-CI	[8.3991, 8.4314]		

TABLE A.1: *Convergence study for a European call option on the maximum of two risky assets under two-factor uncertain volatility model (worst case) with $\rho \in [-1, 1]$ - payoff function in* (5.1)*. The closed form solution is obtained using [32] with fixed parameters $\sigma_x^* = 0.5$, $\sigma_y^* = 0.5$ and $\rho^* = -1$.*

Level	Price	Error	Ratio
0	4.20586189	1.84e-03	
1	4.20724358	4.60e-04	4.00
2	4.20758890	1.15e-04	4.00
3	4.20767522	2.86e-05	4.02
4	4.20769680	7.02e-06	4.07
Ref. [32]	4.20770382		
MC: 95%-CI	[4.2026, 4.2208]		

TABLE A.2: *Convergence study for a European call option on the maximum of two risky assets under two-factor uncertain volatility model (best case) with $\rho \in [-1, 1]$ - payoff function in* (5.1)*. The closed form solution is obtained using [32] with fixed parameters $\sigma_x^* = 0.5$, $\sigma_y^* = 0.5$ and $\rho^* = 1$.*

In the above, we show that a possible choice for $\hat{\rho}$ is $\sqrt{1 - Ch}$, where C is a bounded constant independently of h. We now present a heuristic method to determine C. In the numerical experiments for these special cases, we choose $\Delta y = 6\kappa_y\sqrt{1-\hat{\rho}^2}$, resulting in $\hat{\rho} = \sqrt{1 - \frac{\Delta y^2}{(6\kappa_y)^2}}$. To avoid the necessity for interpolation, for a given $\alpha \in \mathcal{A}_h$, we adjust the partition for y-direction so that each of pair $(x_n,\ a + \rho b x_n)$, where $n \in \mathbb{N}$, aligns with the grid points. In Tables A.1 and A.2, we present the worst-case and best-case prices for the short position in the case of European rainbow options with payoff function in (5.1) with $\rho \in [-1, 1]$. Other parameters given in Table 5.1, and the mesh size and timestep refinement levels are in Table 5.2. The closed-form solution is obtained using [32] with fixed parameters $\sigma_x = 0.5$, $\sigma_y = 0.5$ and $\rho = \{-1, 1\}$. It is evident that the numerical prices show excellent agreement with the closed-form solutions, and also exhibit approximately second-order of convergence, which aligns with our explanations in Section 5.2.1.

B Details of padding matrices

In this appendix, we provide details of the padding matrices $\tilde{\mathbf{g}}_{-1,0}^{q,\alpha}$, $\tilde{\mathbf{g}}_{1,0}^{q,\alpha}$, $\tilde{\mathbf{g}}_{-1,1}^{q,\alpha}$, $\tilde{\mathbf{g}}_{0,1}^{q,\alpha}$ and $\tilde{\mathbf{g}}_{1,1}^{q,\alpha}$ for the circulant matrix $\tilde{\mathbf{g}}_q^\alpha$ defined in (3.32). These padding matrices are defined as follows

$$\tilde{\mathbf{g}}_{-1,0}^{q,\alpha} = \begin{bmatrix} g^\alpha_{-N/2+1,q} & g^\alpha_{-N/2,q} & \cdots & g^\alpha_{-3N/2+1,q} & g^\alpha_{3N/2-1,q} & g^\alpha_{3N/2-2,q} & \cdots & g^\alpha_{N/2,q} \\ g^\alpha_{-N/2+2,q} & g^\alpha_{-N/2+1,q} & \cdots & g^\alpha_{-3N/2+2,q} & g^\alpha_{-3N/2+1,q} & g^\alpha_{3N/2-1,q} & \cdots & g^\alpha_{N/2+1,q} \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ g^\alpha_{N/2,q} & g^\alpha_{N/2-1,q} & \cdots & g^\alpha_{-N/2,q} & g^\alpha_{-N/2-1,q} & g^\alpha_{-N/2-2,q} & \cdots & g^\alpha_{3N/2-1,q} \end{bmatrix}_{N\times(2N+1)},$$

$$\tilde{\mathbf{g}}_{1,0}^{q,\alpha} = \begin{bmatrix} g^\alpha_{-3N/2+1,q} & g^\alpha_{3N/2-1,q} & g^\alpha_{3N/2-2,q} & \cdots & g^\alpha_{N/2+1,q} & g^\alpha_{N/2,q} & \cdots & g^\alpha_{-N/2,q} \\ g^\alpha_{-3N/2+2,q} & g^\alpha_{-3N/2+1,q} & g^\alpha_{3N/2-1,q} & \cdots & g^\alpha_{N/2+2,q} & g^\alpha_{N/2+1,q} & \cdots & g^\alpha_{-N/2+1,q} \\ \vdots & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ g^\alpha_{-N/2,q} & g^\alpha_{-N/2-1,q} & g^\alpha_{-N/2-2,q} & \cdots & g^\alpha_{-3N/2+1,q} & g^\alpha_{3N/2-1,q} & \cdots & g^\alpha_{N/2-1,q} \end{bmatrix}_{N\times(2N+1)},$$

$$\tilde{\mathbf{g}}_{-1,1}^{q,\alpha} = \begin{bmatrix} g^\alpha_{N/2-1,q} & g^\alpha_{N/2-2,q} & \cdots & g^\alpha_{-N/2+2,q} \\ g^\alpha_{N/2,q} & g^\alpha_{N/2-1,q} & \cdots & g^\alpha_{-N/2+3,q} \\ \vdots & \vdots & & \vdots \\ g^\alpha_{3N/2-2,q} & g^\alpha_{3N/2-3,q} & \cdots & g^\alpha_{N/2+1,q} \end{bmatrix}_{N\times(N-2)},$$

$$\tilde{\mathbf{g}}_{0,1}^{q,\alpha} = \begin{bmatrix} g^{\alpha}_{3N/2-1,q} & g^{\alpha}_{3N/2-2,q} & \cdots & g^{\alpha}_{N/2+3,q} & g^{\alpha}_{N/2+2,q} \\ g^{\alpha}_{-3N/2+1,q} & g^{\alpha}_{3N/2-1,q} & \cdots & g^{\alpha}_{N/2+4,q} & g^{\alpha}_{N/2+3,q} \\ \vdots & \vdots & & \vdots & \vdots \\ g^{\alpha}_{-N/2-2,q} & g^{\alpha}_{-N/2-3,q} & \cdots & g^{\alpha}_{-3N/2+2,q} & g^{\alpha}_{-3N/2+1,q} \end{bmatrix}_{(N-1)\times(N-2)},$$

$$\tilde{\mathbf{g}}_{1,1}^{q,\alpha} = \begin{bmatrix} g^{\alpha}_{-N/2-1,q} & g^{\alpha}_{-N/2-2,q} & \cdots & g^{\alpha}_{-3N/2+2,q} \\ g^{\alpha}_{-N/2,q} & g^{\alpha}_{-N/2-1,q} & \cdots & g^{\alpha}_{-3N/2+3,q} \\ \vdots & \vdots & & \vdots \\ g^{\alpha}_{N/2-2,q} & g^{\alpha}_{N/2-1,q} & \cdots & g^{\alpha}_{-N/2+1,q} \end{bmatrix}_{N\times(N-2)}.$$

Next, we provide details of padding matrices $\tilde{\mathbf{g}}^{\alpha}_{-1,0}$, $\tilde{\mathbf{g}}^{\alpha}_{1,0}$, $\tilde{\mathbf{g}}^{\alpha}_{-1,1}$, $\tilde{\mathbf{g}}^{\alpha}_{0,1}$ and $\tilde{\mathbf{g}}^{\alpha}_{1,1}$ for the 2D circulant matrix $\tilde{\mathbf{g}}^{\alpha}$ defined in (3.33). These matrices are defined as follows

$$\tilde{\mathbf{g}}^{\alpha}_{-1,0} = \begin{bmatrix} \tilde{\mathbf{g}}^{\alpha}_{-J/2+1} & \tilde{\mathbf{g}}^{\alpha}_{-J/2} & \cdots & \tilde{\mathbf{g}}^{\alpha}_{-3J/2+1} & \tilde{\mathbf{g}}^{\alpha}_{3J/2-1} & \tilde{\mathbf{g}}^{\alpha}_{3J/2-2} & \cdots & \tilde{\mathbf{g}}^{\alpha}_{J/2} \\ \tilde{\mathbf{g}}^{\alpha}_{-J/2+2} & \tilde{\mathbf{g}}^{\alpha}_{-J/2+1} & \cdots & \tilde{\mathbf{g}}^{\alpha}_{-3J/2+2} & \tilde{\mathbf{g}}^{\alpha}_{-3J/2+1} & \tilde{\mathbf{g}}^{\alpha}_{3J/2-1} & \cdots & \tilde{\mathbf{g}}^{\alpha}_{J/2+1} \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ \tilde{\mathbf{g}}^{\alpha}_{J/2} & \tilde{\mathbf{g}}^{\alpha}_{J/2-1} & \cdots & \tilde{\mathbf{g}}^{\alpha}_{-J/2} & \tilde{\mathbf{g}}^{\alpha}_{-J/2-1} & \tilde{\mathbf{g}}^{\alpha}_{-J/2-2} & \cdots & \tilde{\mathbf{g}}^{\alpha}_{3J/2-1} \end{bmatrix}_{(3N-1)J\times(3N-1)(2J+1)},$$

$$\tilde{\mathbf{g}}^{\alpha}_{1,0} = \begin{bmatrix} \tilde{\mathbf{g}}^{\alpha}_{-3J/2+1} & \tilde{\mathbf{g}}^{\alpha}_{3J/2-1} & \tilde{\mathbf{g}}^{\alpha}_{3J/2-2} & \cdots & \tilde{\mathbf{g}}^{\alpha}_{J/2+1} & \tilde{\mathbf{g}}^{\alpha}_{J/2} & \cdots & \tilde{\mathbf{g}}^{\alpha}_{-J/2} \\ \tilde{\mathbf{g}}^{\alpha}_{-3J/2+2} & \tilde{\mathbf{g}}^{\alpha}_{-3J/2+1} & \tilde{\mathbf{g}}^{\alpha}_{3J/2-1} & \cdots & \tilde{\mathbf{g}}^{\alpha}_{J/2+2} & \tilde{\mathbf{g}}^{\alpha}_{J/2+1} & \cdots & \tilde{\mathbf{g}}^{\alpha}_{-J/2+1} \\ \vdots & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ \tilde{\mathbf{g}}^{\alpha}_{-J/2} & \tilde{\mathbf{g}}^{\alpha}_{-J/2-1} & \tilde{\mathbf{g}}^{\alpha}_{-J/2-2} & \cdots & \tilde{\mathbf{g}}^{\alpha}_{-3J/2+1} & \tilde{\mathbf{g}}^{\alpha}_{3J/2-1} & \cdots & \tilde{\mathbf{g}}^{\alpha}_{J/2-1} \end{bmatrix}_{(3N-1)J\times(3N-1)(2J+1)},$$

$$\tilde{\mathbf{g}}^{\alpha}_{-1,1} = \begin{bmatrix} \tilde{\mathbf{g}}^{\alpha}_{J/2-1} & \tilde{\mathbf{g}}^{\alpha}_{J/2-2} & \cdots & \tilde{\mathbf{g}}^{\alpha}_{-J/2+2} \\ \tilde{\mathbf{g}}^{\alpha}_{J/2} & \tilde{\mathbf{g}}^{\alpha}_{J/2-1} & \cdots & \tilde{\mathbf{g}}^{\alpha}_{-J/2+3} \\ \vdots & \vdots & & \vdots \\ \tilde{\mathbf{g}}^{\alpha}_{3J/2-2} & \tilde{\mathbf{g}}^{\alpha}_{3J/2-3} & \cdots & \tilde{\mathbf{g}}^{\alpha}_{J/2+1} \end{bmatrix}_{(3N-1)J\times(3N-1)(J-2)},$$

$$\tilde{\mathbf{g}}^{\alpha}_{0,1} = \begin{bmatrix} \tilde{\mathbf{g}}^{\alpha}_{3J/2-1} & \tilde{\mathbf{g}}^{\alpha}_{3J/2-2} & \cdots & \tilde{\mathbf{g}}^{\alpha}_{J/2+3} & \tilde{\mathbf{g}}^{\alpha}_{J/2+2} \\ \tilde{\mathbf{g}}^{\alpha}_{-3J/2+1} & \tilde{\mathbf{g}}^{\alpha}_{3J/2-1} & \cdots & \tilde{\mathbf{g}}^{\alpha}_{J/2+4} & \tilde{\mathbf{g}}^{\alpha}_{J/2+3} \\ \vdots & \vdots & & \vdots & \vdots \\ \tilde{\mathbf{g}}^{\alpha}_{-J/2-2} & \tilde{\mathbf{g}}^{\alpha}_{-J/2-3} & \cdots & \tilde{\mathbf{g}}^{\alpha}_{-3J/2+2} & \tilde{\mathbf{g}}^{\alpha}_{-3J/2+1} \end{bmatrix}_{(3N-1)(J-1)\times(3N-1)(J-2)},$$

$$\tilde{\mathbf{g}}^{\alpha}_{1,1} = \begin{bmatrix} \tilde{\mathbf{g}}^{\alpha}_{-J/2-1} & \tilde{\mathbf{g}}^{\alpha}_{-J/2-2} & \cdots & \tilde{\mathbf{g}}^{\alpha}_{-3J/2+2} \\ \tilde{\mathbf{g}}^{\alpha}_{-J/2} & \tilde{\mathbf{g}}^{\alpha}_{-J/2-1} & \cdots & \tilde{\mathbf{g}}^{\alpha}_{-3J/2+3} \\ \vdots & \vdots & & \vdots \\ \tilde{\mathbf{g}}^{\alpha}_{J/2-2} & \tilde{\mathbf{g}}^{\alpha}_{J/2-1} & \cdots & \tilde{\mathbf{g}}^{\alpha}_{-J/2+1} \end{bmatrix}_{(3N-1)J\times(3N-1)(J-1)}.$$