An unconditionally monotone numerical scheme for the two factor uncertain volatility model *†

K. Ma ‡ P. A. Forsyth §

December 16, 2015

Abstract

Under the assumption that two asset prices follow an uncertain volatility model, the maximal and minimal solution values of an option contract are given by a two dimensional Hamilton-Jacobi-Bellman (HJB) PDE. A fully implicit, unconditionally monotone finite difference numerical scheme is developed in this paper. Consequently, there are no time step restrictions due to stability considerations. The discretized algebraic equations are solved using policy iteration. Our discretization method results in a local objective function which is a discontinuous function of the control. Hence some care must be taken when applying policy iteration. The main difficulty in designing a discretization scheme is development of a monotonicity preserving approximation of the cross derivative term in the PDE. We derive a hybrid numerical scheme which combines use of a fixed point stencil and a wide stencil based on a local coordinate rotation. The algorithm uses the fixed point stencil as much as possible to take advantage of its accuracy and computational efficiency. The analysis shows that our numerical scheme is l_∞ stable, consistent in the viscosity sense, and monotone. Thus, our numerical scheme guarantees convergence to the viscosity solution.

Keywords: Monotone scheme, Fully implicit, Uncertain volatility, HJB equation, Policy iteration, Hybrid scheme

Version: December 16, 2015

1 Introduction

1.1 Overview

A key sufficient requirement for ensuring convergence to the viscosity solution of multidimensional HJB equations is that the discretization be monotone (Barles et al., 1995; Barles and Souganidis, 1991). We are particularly interested in optimal stochastic control problems where the control appears in the diffusion tensor. In this case, construction of a monotone scheme is a non-trivial matter. Previous work has focused on explicit wide stencil schemes (Bonnans and Zidani, 2003; Debrabant and Jakobsen, 2013). In this paper, we focus on fully implicit methods (hence avoiding timestep restrictions due to stability considerations). In addition, we attempt to minimize the use of a wide stencil discretization. To provide a concrete application of our method, we focus on the well known uncertain volatility model for pricing multi-factor contingent claims. However, the reader should have no difficulty applying the techniques in this paper to other optimal stochastic control problems formulated as HJB equations.

*This work was supported by the Bank of Nova Scotia and the Natural Sciences and Engineering Research Council of Canada
†The authors would like to thank P. Azimzadeh for many useful discussions
‡Cheriton School of Computer Science, University of Waterloo, Waterloo ON, Canada N2L 3G1 `k26ma@uwaterloo.ca`
§Cheriton School of Computer Science, University of Waterloo, Waterloo ON, Canada N2L 3G1 `paforsyt@uwaterloo.ca`

1.2 Uncertain volatility model

The uncertain volatility model was first independently developed by Lyons (Lyons, 1995) and Avellaneda *et al.* (Avellaneda and Paras, 1996). In this case, the volatility of the risky asset is assumed to lie within a range of values. As such, prices obtained under a no-arbitrage analysis are no longer unique. All that can be computed are the best-case and worst-case prices, for a specified long or short position. By assuming the worst case, an investor can hedge his/her position and obtain a non-negative balance in the hedging portfolio, regardless of the actual volatility movement, provided that volatility remains within the specified range.

Several studies have already considered the uncertain volatility for one factor problems. A European call option with transaction cost and uncertain volatility is considered in Dokuchaev and Savkin (1998). Barrier options under uncertain volatility were studied in Avellaneda and Buff (1999) and Buff (2002), as well as American options and a portfolio of uncertain volatility options. American options were also studied in Smith (2002). Using market bid-ask spreads, an uncertain volatility calibration method was suggested in Coleman et al. (2010). A fully implicit PDE scheme is developed for discretely observed barrier options in Forsyth and Vetzal (2001). These studies are all based on numerical solution of the HJB equations.

In the one-dimensional (single factor) case, it has been shown in Pooley et al. (2003a) that seemingly reasonable discretizations of the uncertain volatility PDE may not converge to the viscosity solution, which is the financially relevant solution. Consequently, it is important to ensure that the numerical scheme is l_∞ stable, consistent in the viscosity sense, and monotone. These properties guarantee convergence to the viscosity solution (Barles and Souganidis, 1991).

Two factor uncertain volatility models were discussed in Pooley et al. (2003b), however, the scheme was not guaranteed to be monotone. The main difficulty in constructing compact multi-dimensional monotone schemes is the presence of the mixed derivative term, which appears in any case where there is a non-zero correlation between the two underlying assets. In certain cases, monotone schemes can be constructed for very restrictive grid spacing conditions and for certain classes of diffusion tensors (Øksendal and Sulem, 2005), but this approach is not very general.

In general, no fixed point stencil finite difference scheme can produce a monotone scheme for arbitrary two factor diffusion tensors (Dong and Krylov, 2006). To ensure monotonicity for problems with non-constant diffusion tensors, first order wide stencil methods have been suggested. That is, the stencil increases in size (relative to the node spacing) as the grid is refined. In this paper, we will primarily use a wide stencil based on a local coordinate rotation. An alternative approach is based on factoring the diffusion tensor. This idea has a long history in stochastic control, see for example (Menaldi, 1989; Camilli and Falcone, 1995; Kushner and Dupuis, 2001). For a recent overview of these methods, we refer the reader to Debrabant and Jakobsen (2013). Another variant of the wide stencil method is discussed in Bonnans and Zidani (2003) and Bonnans et al. (2004). However, as noted in Debrabant and Jakobsen (2013), the complexity of computing the coefficients of the wide stencil technique in Bonnans and Zidani (2003) is quite large, which leads to problems if the coefficients need to be recomputed at every node and every policy iteration (as would be required in our implicit approach).

1.3 Main results

- A fully implicit, consistent, unconditionally monotone numerical scheme is first developed for a two factor uncertain volatility model. The discretized algebraic equations are solved using policy iteration. Our discretization method results in a local objective function which can be a discontinuous function of the control. Hence some care must be taken when applying policy iteration (Huang et al., 2012). Since we use implicit timestepping, there are no time step restrictions due to stability considerations, an advantage over the method in Debrabant and Jakobsen (2013).

- Each policy iteration requires solution of an unstructured sparse M-matrix at each iterate. Since the stencil potentially changes at each policy iteration, this means that the data structure of the sparse matrix must be recomputed at each policy iteration. In this paper, we use a preconditioned

Bi-CGSTAB iterative method for solving the sparse matrix (Saad, 2004). However, the cost of constructing the data structure and solving the matrix is in fact negligible in comparison to the cost of solving the local optimization problem at each grid node. Assuming that the number of policy iterations is bounded as the mesh size tends to zero (which is in fact observed experimentally) the fully implicit method has essentially the same complexity per step as the explicit method in Debrabant and Jakobsen (2013).

- A monotone scheme is constructed by factoring the diffusion tensor in Debrabant and Jakobsen (2013). We compare this approach to a method based on a local coordinate system rotation. Although both of these wide stencils are first order, our numerical experiments indicate that the use of the locally rotated coordinate system seems to perform better than constructing a local coordinate system based on factoring the diffusion tensor.

- We also derive a hybrid numerical scheme that combines use of a fixed point stencil (Clift and Forsyth, 2008; Øksendal and Sulem, 2005) with a wide stencil. The fixed point stencil is a second-order approximation (for smooth test functions), but this discretization cannot ensure monotonicity at every node in general. We propose an algorithm which uses the fixed point stencil as much as possible to take advantage of its accuracy and computational efficiency, while still keeping the numerical scheme monotone. This can be viewed as the multi-dimensional generalization of the standard "central differencing as much as possible" scheme in one dimension (Wang and Forsyth, 2008). Our tests show that this hybrid technique is generally more smoothly convergent and more accurate than a pure wide stencil scheme. Note that use of an explicit scheme coupled with the hybrid discretization would not result in a practical method, due to the small timesteps required for stability.

2 Formulation

Let $\mathcal{U}(S_1, S_2, \tau)$ be the value of a European option contract written on asset prices S_1 and S_2, which both follow the stochastic processes under the risk neutral measure

$$\begin{aligned} \mathrm{d}S_1 &= (r - q_1)S_1\mathrm{d}t + \sigma_1 S_1 \mathrm{d}W_1, \\ \mathrm{d}S_2 &= (r - q_2)S_2\mathrm{d}t + \sigma_2 S_2 \mathrm{d}W_2, \end{aligned} \tag{2.1}$$

where r is the risk-free interest rate, $q_i,\ i = 1, 2$ are the dividend yields for S_i. $\sigma_i,\ i = 1, 2$ are volatilities, and $W_i,\ i = 1, 2$ are Wiener processes with $\mathrm{d}W_1\mathrm{d}W_2 = \rho\ \mathrm{d}t$.

We consider the uncertain volatility model that was first developed in Avellaneda and Paras (1996) and Lyons (1995). That is, σ_i is an uncertain volatility in the processes (2.1), but lies within a range, e.g., $\sigma_1 \in [\sigma_{1,\min}, \sigma_{1,\max}]$ and $\sigma_2 \in [\sigma_{2,\min}, \sigma_{2,\max}]$. In addition, uncertain correlation between the two underlying assets is permitted, e.g., $\rho_{\min} \le \rho \le \rho_{\max}$. When the volatilities σ_1, σ_2, and the correlation ρ are uncertain, the the price of an option contract is no longer unique. However, in the event of uncertain parameters, we can determine the worst case hedging costs for long and short positions.

These maximal and minimal values of an option contract are given by the following Hamilton-Jacobi-Bellman (HJB) PDEs

$$\begin{aligned} \mathcal{U}_\tau = \sup_{Q \in Z} \mathcal{L}^Q \mathcal{U} \ \ ; \quad \text{or} \quad \mathcal{U}_\tau = \inf_{Q \in Z} \mathcal{L}^Q \mathcal{U}, \\ \mathcal{U}(S_1, S_2, 0) = \mathcal{W}(S_1, S_2), \end{aligned} \tag{2.2}$$

which is defined over $(S_1, S_2, \tau) \in [0, +\infty) \times [0, +\infty) \times [0, T]$. $\mathcal{W}(S_1, S_2)$ is the terminal payoff of the option contract. The sup in equation (2.2) corresponds to the worst case short position, while the inf corresponds to the worst case long position. The differential operator $\mathcal{L}^Q$ is defined as

$$
\begin{aligned}
&\mathcal{L}^Q\mathcal{U} = \mathbf{V}\cdot\nabla\mathcal{U} + (\mathbf{D}\nabla)\cdot\nabla\mathcal{U} - r\mathcal{U},\\
&\mathbf{D}\in\mathbb{R}^2\times\mathbb{R}^2;\quad \mathbf{V}\in\mathbb{R}^2;\\
&\nabla = \begin{pmatrix} \frac{\partial}{\partial S_1} \\ \frac{\partial}{\partial S_2} \end{pmatrix},\quad \mathbf{V} = \begin{pmatrix} (r-q_1)S_1 \\ (r-q_2)S_2 \end{pmatrix},\quad \mathbf{D} = \frac{1}{2}\begin{pmatrix} \sigma_1^2 S_1^2 & \rho\sigma_1\sigma_2 S_1 S_2 \\ \rho\sigma_1\sigma_2 S_1 S_2 & \sigma_2^2 S_2^2 \end{pmatrix},
\end{aligned}
\tag{2.3}
$$

where ∇ is the gradient operator, $\mathbf{V}$ is the drift tensor, and $\mathbf{D}$ is the diffusion tensor.

Note that the notation $(\mathbf{D}\nabla)\cdot\nabla\mathcal{U}$ is to be interpreted as

$$
(\mathbf{D}\nabla)\cdot\nabla\mathcal{U} = \frac{\sigma_1^2 S_1^2}{2}\,\mathcal{U}_{S_1S_1} + \rho\sigma_1\sigma_1 S_1 S_2\,\mathcal{U}_{S_1S_2} + \frac{\sigma_2^2 S_2^2}{2}\,\mathcal{U}_{S_2S_2}\ . \tag{2.4}
$$

The control $Q = (\sigma_1,\ \sigma_2,\ \rho)$, and the admissible set of the controls is given by

$$
\begin{aligned}
Z = [\sigma_{1,\min},\sigma_{1,\max}] \times [\sigma_{2,\min},\sigma_{2,\max}] \times [\rho_{\min},\rho_{\max}],\\
\sigma_{1,\min}\geq 0,\quad \sigma_{2,\min}\geq 0,\quad -1\leq\rho_{\min}\leq 1,\quad -1\leq\rho_{\max}\leq 1.
\end{aligned}
\tag{2.5}
$$

Without loss of generality, we only consider sup problem in the following discussion. All the results of this paper hold in the inf case as well.

3 Restriction of control set Z

Before we introduce our discretization method, we take a short digression here to discuss the maximization of the right hand side of equation (2.2). We consider (for the time being) that all the derivatives which appear on the right hand side of equation (2.2) are constructed from known, smooth functions. Since consistency in the viscosity sense is defined in terms of smooth test functions (Barles and Souganidis, 1991), this will be relevant to our discretization approach.

To maximize the solution value in equation (2.2), it suffices to maximize the diffusion terms. Let $\Gamma_{kl} \equiv \frac{\partial^2\mathcal{U}}{\partial S_k \partial S_l},\ k,l = 1,2$. Assume for the moment that Γ_{kl} is known, independent of the control. In this notation, the diffusion terms in (2.2) become

$$
\sup_{Q\in Z}\left((\mathbf{D}\nabla)\cdot\nabla\mathcal{U}\right) = \max_{(\sigma_1,\sigma_2,\rho)\in Z}\left(\frac{\sigma_1^2 S_1^2}{2}\Gamma_{11} + \rho\sigma_1\sigma_2 S_1 S_2\Gamma_{12} + \frac{\sigma_2^2 S_2^2}{2}\Gamma_{22}\right). \tag{3.1}
$$

Since Z (2.5) is a compact set, the supremum is simply the maximum value.

It is easy to see that the optimal correlation value is a bang-bang control. That is, the optimal $\rho \in \{\rho_{\min},\ \rho_{\max}\}$, depends only on the sign of the cross derivative Γ_{12}.

$$
\rho\left(\Gamma_{12}\right) = \begin{cases} \rho_{\max} & \Gamma_{12}\geq 0 \\ \rho_{\min} & \Gamma_{12} < 0. \end{cases} \tag{3.2}
$$

With ρ given from (3.2), a quadratic-form optimization with linear constraints needs to be solved. The problem is formulated as

$$
\max_{\sigma}\sigma^T\mathbf{M}\sigma \equiv \max_{\sigma_1,\sigma_2}\begin{pmatrix}\sigma_1 & \sigma_2\end{pmatrix}\begin{pmatrix} \frac{s_1^2}{2}\Gamma_{11} & \rho(\Gamma_{12})\frac{s_1 s_2}{2}\Gamma_{12} \\ \rho(\Gamma_{12})\frac{s_1 s_2}{2}\Gamma_{12} & \frac{s_2^2}{2}\Gamma_{22} \end{pmatrix}\begin{pmatrix}\sigma_1 \\ \sigma_2\end{pmatrix}, \tag{3.3}
$$

subject to

$$
\sigma_{1,\min}\leq\sigma_1\leq\sigma_{1,\max},\quad \sigma_{2,\min}\leq\sigma_2\leq\sigma_{2,\max}. \tag{3.4}
$$

Proposition 3.1. *Suppose that Γ_{ik} exist $\forall i,k$. The optimal value of the objective function in (3.1) can be determined by examining values only on the boundary of Z, denoted by ∂Z.*

$$\sup_{Q\in Z}\left((\mathbf{D}\nabla)\cdot\nabla\mathcal{U}\right)=\sup_{Q\in\partial Z}\left((\mathbf{D}\nabla)\cdot\nabla\mathcal{U}\right). \tag{3.5}$$

Proof. From equation (3.2), the choice of the optimal correlation ρ is either $\rho_{\max}$ or $\rho_{\min}$, depending on the sign of the cross derivative term. Thus, the optimal correlation is always either end of its range $[\rho_{\min},\rho_{\max}]$.

The quadratic form in equation (3.3) is $\sigma^T\mathbf{M}\sigma$. A critical point is such that $\mathbf{M}\sigma=0$. When $\mathbf{M}$ is a non-singular, the critical point is $(0,0)$, which is either outside Z or on the boundary of Z. When $\mathbf{M}$ is singular, the critical points are on the line $\left\{(\sigma_1,\sigma_2)\middle|\frac{S_1^2}{2}\Gamma_{11}\sigma_1+\frac{\rho(\Gamma_{12})S_1S_2}{2}\Gamma_{12}\sigma_2=0\right\}$. If this line intersects Z, then the optimal value is attained at ∂Z. If this line does not intersect Z, then the optimal value is also on ∂Z. Hence, in all cases, the optimal value can be attained by examining the objective function on ∂Z. □

Remark 3.1. *Proposition 3.1 will prove useful when we design a numerical scheme. In the case when the discretization stencil depends on the control, no closed form expression is available for the optimal value. We can then discretize the control set and search over the boundary ∂Z, instead of the entire three dimensional set Z. Consistency in the viscosity sense is defined in terms of smooth test functions, hence our assumption that Γ_{ik} exist is not restrictive and we can then use Proposition 3.1 to prove that this is a consistent discretization (in the viscosity sense).*

4 Discretization

In this paper, we develop an unconditionally monotone finite difference numerical scheme for the two factor uncertain volatility model. However, a standard finite difference scheme cannot ensure monotonicity due to the cross derivative term. For example, the fixed point stencil method in Øksendal and Sulem (2005) requires a restrictive grid spacing, which cannot always be satisfied, to preserve monotonicity. In our problem, the tensor diffusion is non-constant and non-diagonally dominant. We will focus mainly on a wide stencil method based on a local coordinate rotation, but we include some comparisons with the factoring technique in Debrabant and Jakobsen (2013). Furthermore, we propose a hybrid algorithm which combines use of a fixed point stencil (Clift and Forsyth, 2008; Øksendal and Sulem, 2005) with a wide stencil. This algorithm uses the fixed point stencil as much as possible to take advantage of its accuracy and computational efficiency, but still keeping the numerical scheme monotone.

We discretize equation (2.2) over a finite grid $N=N_1\times N_2$ in the plane (S_1,S_2). Define a set of nodes $\{(S_1)_1,(S_1)_2,\ldots,(S_1)_{N_1}\}$ in S_1 direction and $\{(S_2)_1,(S_2)_2,\ldots,(S_2)_{N_2}\}$ in S_2 direction. Denote the n^{th} time step by $\tau^n=n\Delta\tau,\ n=0,\ldots,N_\tau$, with $N_\tau=\frac{T}{\Delta\tau}$. Let $\mathcal{U}^n_{i,j}$ be the approximate solution of the equation (2.2) at $((S_1)_i,(S_2)_j,\tau^n)$.

It will be convenient to define

$$\begin{aligned}\Delta(S_1)_{\max}&=\max_i\left((S_1)_{i+1}-(S_1)_i\right), \quad \Delta(S_1)_{\min}=\min_i\left((S_1)_{i+1}-(S_1)_i\right),\\ \Delta(S_2)_{\max}&=\max_i\left((S_2)_{i+1}-(S_2)_i\right), \quad \Delta(S_2)_{\min}=\min_i\left((S_2)_{i+1}-(S_2)_i\right).\end{aligned} \tag{4.1}$$

We assume that there is a mesh discretization parameter h such that

$$\Delta(S_1)_{\max}=C_1h,\ \Delta(S_2)_{\max}=C_2h,\ \Delta(S_1)_{\min}=C_1'h,\ \Delta(S_2)_{\min}=C_2'h,\ \Delta\tau=C_3h, \tag{4.2}$$

where C_1,C_2,C_1',C_2',C_3 are constants independent of h.

4.1 The fixed point stencil

We use a seven-point stencil (Clift and Forsyth, 2008; Øksendal and Sulem, 2005) to discretize the cross-partial derivative $\frac{\partial^2 \mathcal{U}}{\partial S_1 \partial S_2}$. Denote

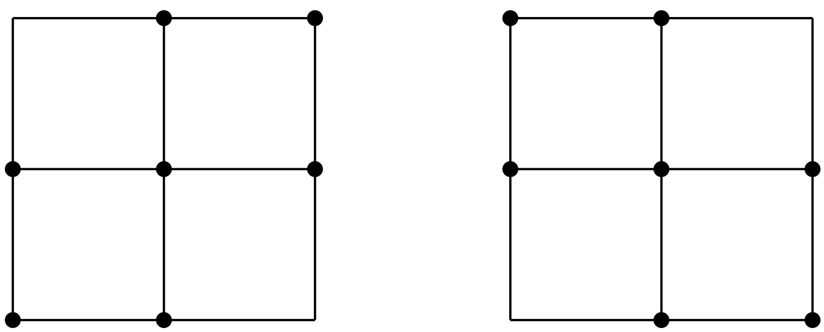
$$\begin{aligned} \Delta^+(S_1)_i &= (S_1)_{i+1} - (S_2)_i, \quad \Delta^-(S_1)_i = (S_1)_i - (S_1)_{i-1}, \\ \Delta^+(S_2)_j &= (S_2)_{j+1} - (S_2)_j, \quad \Delta^-(S_2)_j = (S_2)_j - (S_2)_{j-1}. \end{aligned} \tag{4.3}$$

We approximate the cross-partial derivative at $((S_1)_i, (S_2)_j, \tau^n)$ using one of the following stencils, as illustrated in Figure 4.1, depending on the sign of ρ. For $\rho \geq 0$, we use

$$\frac{\partial^2 \mathcal{U}}{\partial S_1 \partial S_2} \approx \frac{2\mathcal{U}^n_{i,j} + \mathcal{U}^n_{i+1,j+1} + \mathcal{U}^n_{i-1,j-1}}{\Delta^+(S_1)_i \Delta^+(S_2)_j + \Delta^-(S_1)_i \Delta^-(S_2)_j} - \frac{\mathcal{U}^n_{i+1,j} + \mathcal{U}^n_{i-1,j} + \mathcal{U}^n_{i,j+1} + \mathcal{U}^n_{i,j-1}}{\Delta^+(S_1)_i \Delta^+(S_2)_j + \Delta^-(S_1)_i \Delta^-(S_2)_j}. \tag{4.4}$$

For $\rho < 0$, we use

$$\frac{\partial^2 \mathcal{U}}{\partial S_1 \partial S_2} \approx -\frac{2\mathcal{U}^n_{i,j} + \mathcal{U}^n_{i+1,j-1} + \mathcal{U}^n_{i-1,j+1}}{\Delta^+(S_1)_i \Delta^-(S_2)_j + \Delta^-(S_1)_i \Delta^+(S_2)_j} + \frac{\mathcal{U}^n_{i+1,j} + \mathcal{U}^n_{i-1,j} + \mathcal{U}^n_{i,j+1} + \mathcal{U}^n_{i,j-1}}{\Delta^+(S_1)_i \Delta^-(S_2)_j + \Delta^-(S_1)_i \Delta^+(S_2)_j}. \tag{4.5}$$



(a) $\rho \geq 0$ (b) $\rho < 0$

Figure 4.1: The seven-point stencil for $\rho \geq 0$ and $\rho < 0$. The seven points used in the stencil depend on the sign of ρ.

Standard three point differences are used for the $\frac{\partial^2 \mathcal{U}}{\partial S_1 \partial S_1}$ and $\frac{\partial^2 \mathcal{U}}{\partial S_2 \partial S_2}$ terms. First order partial derivatives in (2.2) are approximated with second order central differencing as much as possible (see Appendix A). We select central, forward and backward differencing to minimize the appearance of negative coefficients in the discretization (Wang and Forsyth, 2008). The linear differential operator $\mathcal{L}$ in (2.2) is discretized to form the discrete linear operator L^Q_f.

$$\begin{aligned} L^Q_f \mathcal{U}^n_{i,j} = &(\alpha^{S_1}_{i,j} - \gamma_{i,j})\mathcal{U}^n_{i-1,j} + (\beta^{S_1}_{i,j} - \gamma_{i,j})\mathcal{U}^n_{i+1,j} + (\alpha^{S_2}_{i,j} - \gamma_{i,j})\mathcal{U}^n_{i,j-1} + (\beta^{S_2}_{i,j} - \gamma_{i,j})\mathcal{U}^n_{i,j+1} \\ &+ 1_{\rho \geq 0}(\gamma_{i,j}\mathcal{U}^n_{i+1,j+1} + \gamma_{i,j}\mathcal{U}^n_{i-1,j-1}) + 1_{\rho < 0}(\gamma_{i,j}\mathcal{U}^n_{i+1,j-1} + \gamma_{i,j}\mathcal{U}^n_{i-1,j+1}) \\ &- (\alpha^{S_1}_{i,j} + \beta^{S_1}_{i,j} + \alpha^{S_2}_{i,j} + \beta^{S_2}_{i,j} - 2\gamma_{i,j} + r)\mathcal{U}_{i,j}, \end{aligned} \tag{4.6}$$

where $\alpha^{S_1}_{i,j}, \beta^{S_1}_{i,j}, \alpha^{S_2}_{i,j}, \beta^{S_2}_{i,j}$, and $\gamma_{i,j}$ are defined in Appendix A. The notation L^Q_f indicates that the equation coefficients are functions of the control Q.

The positive coefficient condition (Forsyth and Labahn, 2007) is

$$\begin{aligned} &\alpha^{S_1}_{i,j} - \gamma_{i,j} \geq 0, \quad \beta^{S_1}_{i,j} - \gamma_{i,j} \geq 0, \quad \alpha^{S_2}_{i,j} - \gamma_{i,j} \geq 0, \quad \beta^{S_2}_{i,j} - \gamma_{i,j} \geq 0, \\ &\gamma_{i,j} \geq 0, \quad \alpha^{S_1}_{i,j} + \beta^{S_1}_{i,j} + \alpha^{S_2}_{i,j} + \beta^{S_2}_{i,j} - 2\gamma_{i,j} + r \geq 0, \quad 1 \leq i < N_1, \ 1 \leq j < N_2. \end{aligned} \tag{4.7}$$

Due to the presence of the $\gamma_{i,j}$ term in (4.6), the discretization does not ensure that the positive coefficient condition (4.7) is satisfied even if our choice of the seven-point operator ensures that $\gamma_{i,j} \geq 0$. However, our algorithm makes the positive coefficient condition hold on as many grid nodes as possible with a fixed stencil. Only when the cross derivative term disappears in the HJB equation (2.2) can we guarantee that the positive coefficient condition always holds for a fixed point stencil.

Remark 4.1. *It is possible to carry out a logarithmic transformation on equation (2.2). In the new coordinate system* $(\log S_1, \log S_2)$*, the diffusion tensor becomes constant for a fixed control. If we discretize the PDE on the space* $(\log S_1, \log S_2)$*, a positive coefficient discretization can be constructed for a very restrictive grid spacing condition (Clift and Forsyth, 2008), but this approach is not very general, and the diffusion tensor is not constant if local volatility surfaces are used, which is common in practice. Consequently, we prefer to use the more meaningful discretization in* (S_1, S_2) *coordinates.*

4.2 Local coordinate rotation: the wide stencil

We now consider the wide stencil discretization method. Suppose we discretize equation (2.2) at grid node (i,j) for a fixed control. Consider a virtual rotation of the local coordinate system clockwise by

$$\theta_{i,j} = \frac{1}{2}\tan^{-1}\left(\frac{2\rho\sigma_1\sigma_2(S_1)_i(S_2)_j}{(\sigma_1(S_1)_i)^2 - (\sigma_2(S_2)_j)^2}\right). \tag{4.8}$$

That is, (y_1, y_2) in the transformed coordinate system is obtained by using the following matrix multiplication

$$\begin{pmatrix} S_1 \\ S_2 \end{pmatrix} = \begin{pmatrix} \cos\theta_{i,j} & -\sin\theta_{i,j} \\ \sin\theta_{i,j} & \cos\theta_{i,j} \end{pmatrix}\begin{pmatrix} y_1 \\ y_2 \end{pmatrix}. \tag{4.9}$$

We denote the rotation matrix in (4.9) as $\mathbf{R}_{i,j}$. This rotation operation will result in a zero correlation in the diffusion tensor of the rotated system. That is, the cross derivative term will be eliminated. Under this grid rotation, the second order terms in equation (2.2) are, in the transformed coordinate system (y_1, y_2),

$$a_{i,j}\frac{\partial^2 \mathcal{V}}{\partial y_1^2} + b_{i,j}\frac{\partial^2 \mathcal{V}}{\partial y_2^2}, \tag{4.10}$$

where $\mathcal{V}$ is the value function $\mathcal{V}(y_1, y_2, \tau)$ in the transformed coordinate system, and

$$\begin{aligned} a_{i,j} &= \left(\frac{\left(\sigma_1\cos(\theta_{i,j})(S_1)_i\right)^2}{2} + \rho\sigma_1\sigma_2(S_1)_i(S_2)_j\sin(\theta_{i,j})\cos(\theta_{i,j}) + \frac{\left(\sigma_2\sin(\theta_{i,j})(S_2)_j\right)^2}{2}\right), \\ b_{i,j} &= \left(\frac{\left(\sigma_1\sin(\theta_{i,j})(S_1)_i\right)^2}{2} - \rho\sigma_1\sigma_2(S_1)_i(S_2)_j\sin(\theta_{i,j})\cos(\theta_{i,j}) + \frac{\left(\sigma_2\cos(\theta_{i,j})(S_2)_j\right)^2}{2}\right). \end{aligned} \tag{4.11}$$

The diffusion tensor in (4.10) is diagonally dominant with no off-diagonal terms, and consequently a standard finite difference discretization for the second partial derivatives is a positive coefficient scheme. The rotation angle $\theta_{i,j}$ depends on the grid node and the control, therefore it is impossible to rotate the global coordinate system by a constant angle and build a grid over the space (y_1, y_2). The local coordinate system rotation is only used to construct a virtual grid which overlays the original mesh. We have to approximate the values of $\mathcal{U}$ on our virtual local grid using an interpolant $\mathcal{J}_h\mathcal{U}$ on the original mesh. To keep the numerical scheme monotone, linear interpolation is the most accurate interpolation we can use. Thus, $\mathcal{J}_h$ is a linear interpolation operator. Moreover, to keep the numerical scheme consistent, we need to use the points on our virtual grid whose Euclidean distances are $O(\sqrt{h})$ from the central node, where h is the mesh discretization parameter (4.2). This results in a wide stencil method since the relative stencil length increases as the grid is refined ($\frac{\sqrt{h}}{h} \to \infty$ as $h \to 0$). The wide stencil method is illustrated in Figure 4.2. With a slight abuse of notation, we define the following

$$\mathcal{U}^n(\mathbf{S}) \equiv \mathcal{U}(S_1, S_2, \tau^n), \quad \mathbf{S} = \begin{pmatrix} S_1 \\ S_2 \end{pmatrix}, \quad \mathcal{V}^n(\mathbf{y}) \equiv \mathcal{V}(y_1, y_2, \tau^n), \quad \mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}. \tag{4.12}$$

Then, the second order terms in equation (2.2) at $((S_1)_i, (S_2)_j, \tau^n)$ are approximated as

$$
\begin{aligned}
& a_{i,j} \frac{\mathcal{V}^n\left(\mathbf{y}_{i,j} + \sqrt{h}\mathbf{e}_1\right) + \mathcal{V}^n\left(\mathbf{y}_{i,j} - \sqrt{h}\mathbf{e}_1\right) - 2\mathcal{V}^n\left(\mathbf{y}_{i,j}\right)}{h} \\
& \quad + b_{i,j} \frac{\mathcal{V}^n\left(\mathbf{y}_{i,j} + \sqrt{h}\mathbf{e}_2\right) + \mathcal{V}^n\left(\mathbf{y}_{i,j} - \sqrt{h}\mathbf{e}_2\right) - 2\mathcal{V}^n\left(\mathbf{y}_{i,j}\right)}{h} \\
& \approx a_{i,j} \frac{\mathcal{J}_h\mathcal{U}^n(\mathbf{S}_{i,j} + \sqrt{h}(\mathbf{R}_{i,j})_1) + \mathcal{J}_h\mathcal{U}^n(\mathbf{S}_{i,j} - \sqrt{h}(\mathbf{R}_{i,j})_1) - 2\mathcal{U}^n(\mathbf{S}_{i,j})}{h} \\
& \quad + b_{i,j} \frac{\mathcal{J}_h\mathcal{U}^n(\mathbf{S}_{i,j} + \sqrt{h}(\mathbf{R}_{i,j})_2) + \mathcal{J}_h\mathcal{U}^n(\mathbf{S}_{i,j} - \sqrt{h}(\mathbf{R}_{i,j})_2) - 2\mathcal{U}^n(\mathbf{S}_{i,j})}{h},
\end{aligned}
\tag{4.13}
$$

where $\mathbf{S}_{i,j} = ((S_1)_i, (S_2)_j)$, $\mathbf{y}_{i,j} = \mathbf{R}_{i,j}^T\mathbf{S}_{i,j}$, $(\mathbf{R}_{i,j})_k$ is k-th column of the rotation matrix $\mathbf{R}_{i,j}$ (4.9), and

$$
\mathbf{e}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \mathbf{e}_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.
$$

To satisfy the positive coefficient condition, we then use an upstream finite differencing to discretize the first order derivatives.

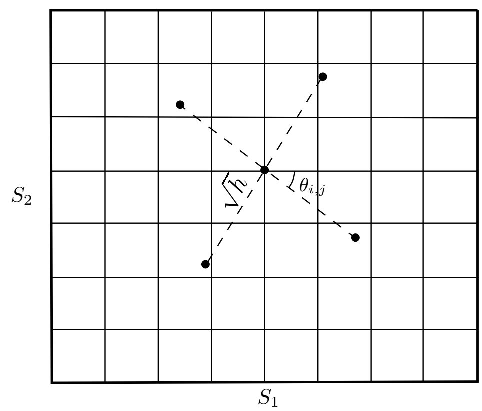


Figure 4.2: The wide stencil method based on local coordinate rotation.

4.3 Boundary conditions

We shall assume that the discretization is posed on a bounded domain for computational purposes. The discretization is applied to the localized finite region $(S_1, S_2) \in [0, S_{1,\max}] \times [0, S_{2,\max}]$.

No boundary condition is needed on the lower boundaries $S_1 = 0$ or $S_2 = 0$. The equation (2.2) reduces to

$$
\frac{\partial \mathcal{U}}{\partial \tau} = \begin{cases}
(r - q_2)S_2 \dfrac{\partial \mathcal{U}}{\partial S_2} + \dfrac{S_2^2\sigma_2^2}{2}\dfrac{\partial^2 \mathcal{U}}{\partial S_2^2} - r\mathcal{U}, & \text{for } (S_1, S_2) \in \{0\} \times (0, S_{2,\max}), \\
(r - q_1)S_1 \dfrac{\partial \mathcal{U}}{\partial S_1} + \dfrac{S_1^2\sigma_1^2}{2}\dfrac{\partial^2 \mathcal{U}}{\partial S_1^2} - r\mathcal{U}, & \text{for } (S_1, S_2) \in (0, S_{1,\max}) \times \{0\}, \\
-r\mathcal{U}, & \text{at } (S_1, S_2) = (0, 0).
\end{cases}
\tag{4.14}
$$

The cross derivative term vanishes on the lower boundaries. Thus, we can use a standard finite difference stencil to construct a monotone scheme on the lower boundaries.

In order to preserve monotonicity of the discretization, a Dirichlet boundary condition is imposed on the upper boundaries $S_1 = S_{1,\max}$ or $S_2 = S_{2,\max}$. As pointed out in Barles et al. (1995), we can expect any errors incurred by imposing approximate boundary conditions at $S_1 = S_{1,\max}$ or $S_2 = S_{2,\max}$ to be small in areas of interest if $S_{1,\max}$ or $S_{2,\max}$ is sufficiently large. As $S_1 \to \infty$ or $S_2 \to \infty$, we normally use financial reasoning to determine the asymptotic form of the solution. The upper boundary may be approximated by a time-dependent value

$$\mathcal{U}_A(S_1, S_2, \tau) \approx c_0(\tau) + c_1(\tau)S_1 + c_2(\tau)S_2. \tag{4.15}$$

4.4 Avoid using points below the lower boundaries

To make the numerical scheme consistent in a wide stencil method, the stencil length needs to be increased to use the points beyond the nearest neighbors of the original grid. As shown in Section 4.2, we use the four points $\mathbf{S}_{i,j} \pm \sqrt{h}(\mathbf{R}_{i,j})_k,\ k = 1, 2$ in (4.13), when we approximate the second order terms (4.10). Therefore, when solving the PDE on a bounded region, this numerical discretization (4.13) may require points outside the computational domain.

When a candidate point we use is outside the computational region at the upper boundaries, we directly use the asymptotic solution as specified in (4.15) at the point. However, we have to take special care when we may use a point below the lower boundaries $S_1 = 0$ or $S_2 = 0$. The possibility of using points below the lower boundaries only occurs when the node (i, j) falls in the region

$$[h, \sqrt{h}] \times (0, S_{2,\max}] \cup (0, S_{1,\max}] \times [h, \sqrt{h}]. \tag{4.16}$$

We propose a simple method to avoid this problem, which retains consistency. That is, when one of the four candidate points $\mathbf{S}_{i,j} \pm \sqrt{h}(\mathbf{R}_{i,j})_k,\ k = 1, 2$ is below the lower boundaries, we then shrink its corresponding distance to h, instead of $\sqrt{h}$. This treatment ensures that all data required is within the computational domain. The details of the method are given in Algorithm 4.1. We will prove that this simple idea retains consistency in Section 5.2.

Algorithm 4.1 Avoid using the points below the lower boundaries when approximating the $\frac{\partial^2 \mathcal{V}}{\partial y_k^2},\ k = 1, 2$

1: Let $\mathbf{S}_{k,left} = \mathbf{S}_{i,j} - \sqrt{h}(\mathbf{R}_{i,j})_k$ and $h_{k,left} = \sqrt{h}$
2: **if** $\mathbf{S}_{k,left}$ below the lower boundaries **then**
3: $\quad h_{left} = h$
4: **end if**
5: Let $\mathbf{S}_{k,right} = \mathbf{S}_{i,j} + \sqrt{h}(\mathbf{R}_{i,j})_k$ and $h_{k,right} = \sqrt{h}$
6: **if** $\mathbf{S}_{k,right}$ below the lower boundaries **then**
7: $\quad h_{right} = h$
8: **end if**
9: The second derivative term $\frac{\partial^2 \mathcal{V}}{\partial y_k^2}$ at $\mathbf{y}_{i,j} = \mathbf{R}_{i,j}^T \mathbf{S}_{i,j}$ are approximated as
10:
$$\frac{\frac{\mathcal{J}_h \mathcal{U}(\mathbf{S}_{i,j} - h_{k,left}(\mathbf{R}_{i,j})_k) - \mathcal{U}(\mathbf{S}_{i,j})}{h_{k,left}} + \frac{\mathcal{J}_h \mathcal{U}(\mathbf{S}_{i,j} + h_{k,right}(\mathbf{R}_{i,j})_k) - \mathcal{U}(\mathbf{S_{i,j}})}{h_{k,right}}}{\frac{h_{k,left} + h_{k,right}}{2}}. \tag{4.17}$$

4.5 Factoring the diffusion tensor

In Debrabant and Jakobsen (2013), the wide stencil method based on factoring the diffusion tensor is surveyed. For the convenience of the reader, we briefly summarize this method here. For more details we refer readers to Debrabant and Jakobsen (2013). Let the diffusion tensor in (2.2) be

$$\mathbf{D} = \frac{1}{2}\mathbf{C}^T\mathbf{C}.$$

Then, the second order terms in (2.2) are approximated as

$$((\mathbf{D}\nabla)\cdot\nabla\mathcal{U}) \approx \frac{1}{2}\Bigg(\frac{\mathcal{J}_h\mathcal{U}(\mathbf{S}+\sqrt{h}\mathbf{C}_1)+\mathcal{J}_h\mathcal{U}(\mathbf{S}-\sqrt{h}\mathbf{C}_1)-2\mathcal{U}(\mathbf{S})}{h} + \frac{\mathcal{J}_h\mathcal{U}(\mathbf{S}+\sqrt{h}\mathbf{C}_2)+\mathcal{J}_h\mathcal{U}(\mathbf{S}-\sqrt{h}\mathbf{C}_2)-2\mathcal{U}(\mathbf{S})}{h}\Bigg) + O(h), \tag{4.18}$$

where $\mathbf{C}_k$ is k-th column of $\mathbf{C}$. From the stochastic processes of the two asset prices (2.1), it is natural to choose

$$\mathbf{C} = \begin{pmatrix} \sigma_1 S_1 & 0 \\ \sigma_2\rho S_2 & \sigma_2\sqrt{1-\rho^2}S_2 \end{pmatrix}.$$

That is, $\mathbf{C}$ is the lower triangular matrix associated with the Cholesky decomposition of the diffusion tensor.

This consistent approximation is also a first order approximation and compatible with a monotone numerical scheme. Although the defining ideas, between this method and the local coordinate rotation introduced in Section 4.2, are different, we can relate them by re-interpreting the approximation (4.18). Firstly, we virtually transform the coordinate system as follows:

$$\begin{pmatrix} S_1 \\ S_2 \end{pmatrix} = \mathbf{C}\begin{pmatrix} y_1 \\ y_2 \end{pmatrix}. \tag{4.19}$$

This transformation will result in a zero correlation in the diffusion tensor of the transformed system. After applying this local virtual coordinate transformation, we then construct a local discretization in a manner similar to the method used for the rotation method in Section 4.2. The transformation (4.19) is both a stretching and rotation of the coordinate system, not an orthogonal rotation (4.9) as in Section 4.2. Thus, in (4.18), we shall use points whose Euclidean distance from (S_1, S_2) are $|\sqrt{h}\mathbf{C}_k|$, $k = 1, 2$, which is state dependent on S_1 and S_2. For example, the points we use may be *far* away from the central node (i, j), especially when the grid state $(S_1)_i$ or $(S_2)_j$ is large. However, as noted in Bonnans and Zidani (2003) and Kushner and Dupuis (2001), it is highly desirable to limit the use of points that are *far* away from the central node. When we use the method of locally rotating coordinate system, the candidate points are always $\sqrt{h}|(\mathbf{R}_{i,j})_k| = \sqrt{h}$ away from the central node. In our numerical experiments, we will compare the performance of these two methods.

4.6 Maximal use of a fixed point stencil

We will derive a hybrid scheme which combines use of the fixed point stencil (Section 4.1) with the wide stencil based on a local coordinate rotation (Section 4.2). The fixed point stencil is a second-order approximation of the diffusion terms, but this discretization cannot ensure a positive coefficient method at every node in general. The computational cost is also highly increased when we use a wide stencil. This is due to the fact that we have an analytical solution for the local optimization problem for the fixed point stencil case. On the other hand, when using a wide stencil, we need to discretize the control set and then perform a linear search to find the optimal value for the control. We propose an algorithm which uses the fixed point stencil as much as possible to take advantage of its accuracy and computational efficiency, while still satisfying the positive coefficient condition. Note that our algorithm is also applicable if we factor the diffusion tensor, as in Debrabant and Jakobsen (2013).

Lemma 4.1. *The positive coefficient condition (4.7) for a fixed point stencil is satisfied for an arbitrary* $Q = (\sigma_1, \sigma_2, \rho)$*, if the following constraints hold*

(1) We must select equation (4.4) if $\rho \geq 0$ *and equation (4.5) if* $\rho < 0$ *to approximate the cross derivative term.*

(2) The following sufficient conditions are satisfied,

for $\rho \geq 0$

$$\frac{(S_2)_j \max(\Delta^+(S_1)_i, \Delta^-(S_1)_i)}{(S_1)_i} \frac{\Delta^+(S_1)_i + \Delta^-(S_1)_i}{\Delta^+(S_1)_i\Delta^+(S_2)_j + \Delta^-(S_1)_i\Delta^-(S_2)_j} \leq \frac{\sigma_1}{\sigma_2\rho}, \tag{4.20a}$$

$$\frac{(S_2)_j}{(S_1)_i \max(\Delta^+(S_2)_j, \Delta^-(S_2)_j)} \frac{\Delta^+(S_1)_i\Delta^+(S_2)_j + \Delta^-(S_1)_i\Delta^-(S_2)_j}{\Delta^+(S_2)_j + \Delta^-(S_2)_j} \geq \frac{\sigma_1\rho}{\sigma_2}, \tag{4.20b}$$

for $\rho < 0$

$$\frac{(S_2)_j \max(\Delta^+(S_1)_i, \Delta^-(S_1)_i)}{(S_1)_i} \frac{\Delta^+(S_1)_i + \Delta^-(S_1)_i}{\Delta^+(S_1)_i\Delta^-(S_2)_j + \Delta^-(S_1)_i\Delta^+(S_2)_j} \leq \frac{\sigma_1}{\sigma_2|\rho|}, \tag{4.21a}$$

$$\frac{(S_2)_j}{(S_1)_i \max(\Delta^+(S_2)_j, \Delta^-(S_2)_j)} \frac{\Delta^+(S_1)_i\Delta^-(S_2)_j + \Delta^-(S_1)_i\Delta^+(S_2)_j}{\Delta^+(S_2)_j + \Delta^-(S_2)_j} \geq \frac{\sigma_1|\rho|}{\sigma_2}. \tag{4.21b}$$

Proof. We select equation (4.4) if $\rho \geq 0$ and equation (4.5) if $\rho < 0$ to approximate the cross derivative term, this choice then ensures $\gamma_{i,j} \geq 0$. The condition (2) makes the following inequities hold

$$\alpha_{i,j}^{S_1} - \gamma_{i,j} \geq 0,\ \beta_{i,j}^{S_1} - \gamma_{i,j} \geq 0,\ \alpha_{i,j}^{S_2} - \gamma_{i,j} \geq 0,\ \alpha_{i,j}^{S_2} - \gamma_{i,j} \geq 0.$$

For more details see Øksendal and Sulem (2005, Chapter 9.4). □

Theorem 4.1. *Assume that*

(1) We must select equation (4.4) if $\rho \geq 0$ *and equation (4.5) if* $\rho < 0$ *to approximate the cross derivative term.*

(2) The grid spacings satisfy the following conditions in terms of extreme values of the control $Q = (\sigma_1, \sigma_2, \rho)$*.*

$$\begin{cases} (4.20a) \text{ for } (\sigma_{1,\min}, \sigma_{2,\max}, \rho_{\max}) \text{ and } (4.20b) \text{ for } (\sigma_{1,\max}, \sigma_{2,\min}, \rho_{\max}), & \text{if } \rho_{\min} \geq 0, \\ (4.21a) \text{ for } (\sigma_{1,\min}, \sigma_{2,\max}, \rho_{\min}) \text{ and } (4.21b) \text{ for } (\sigma_{1,\max}, \sigma_{2,\min}, \rho_{\min}), & \text{if } \rho_{\max} \leq 0, \\ (4.20a) \text{ for } (\sigma_{1,\min}, \sigma_{2,\max}, \rho_{\max}),\ (4.20b) \text{ for } (\sigma_{1,\max}, \sigma_{2,\min}, \rho_{\max}),\ (4.21a) \text{ for } (\sigma_{1,\min}, \sigma_{2,\max}, \rho_{\min}), & \\ \quad \text{and } (4.21b) \text{ for } (\sigma_{1,\max}, \sigma_{2,\min}, \rho_{\min}), & \text{if } \rho_{\min} \leq 0 \leq \rho_{\max}. \end{cases} \tag{4.22}$$

With these conditions, we can select a differencing scheme (see Appendix A) so that the positive coefficient condition (4.7) is satisfied for $\forall Q \in Z$*. We denote the domain where the conditions (4.22) are satisfied by* Ω_f*.*

Proof. For the case $\rho_{\min} \geq 0$, if the constraint (4.20) holds for all $Q \in Z$, we have

$$\begin{aligned} &\frac{(S_2)_j \max(\Delta^+(S_1)_i, \Delta^-(S_1)_i)}{(S_1)_i} \frac{\Delta^+(S_1)_i + \Delta^-(S_1)_i}{\Delta^+(S_1)_i\Delta^+(S_2)_j + \Delta^-(S_1)_i\Delta^-(S_2)_j} \\ &\leq \inf_{Q \in Z} \frac{\sigma_1}{\sigma_2\rho} = \frac{\sigma_{1,\min}}{\sigma_{2,\max}\rho_{\max}}, \\ &\frac{(S_2)_j}{(S_1)_i \max(\Delta^+(S_2)_j, \Delta^-(S_2)_j)} \frac{\Delta^+(S_1)_i\Delta^+(S_2)_j + \Delta^-(S_1)_i\Delta^-(S_2)_j}{\Delta^+(S_2)_j + \Delta^-(S_2)_j} \\ &\geq \sup_{Q \in Z} \frac{\sigma_1\rho}{\sigma_2} = \frac{\sigma_{1,\max}\rho_{\max}}{\sigma_{2,\min}}. \end{aligned} \tag{4.23}$$

The proof is similar for the other two cases. □

We select central/upstream differencing (forward or backward differencing) for the first order derivative terms. When the conditions in Theorem 4.1 are satisfied, upstream differencing ensures that the positive coefficient condition holds. However, central differencing is used as much as possible to minimize discretization error. Consequently, given a control Q, if central differencing satisfies the positive coefficient condition, central differencing will be preferred.

Remark 4.2. *Grid spacing conditions in Theorem 4.1 depend on the space state* (S_1, S_2)*, thus the structure of a grid is not always such that these conditions are met everywhere. We shall not enforce these conditions, but indeed check whether they are satisfied at a given grid node.*

Our algorithm is summarized as follows. The domains are defined in Table 4.1. The fixed point stencil introduced in Section 4.1 is used in the domain Ω_f. For the case $\big((S_1)_i, (S_2)_j, \tau^{n+1}\big) \in \Omega_w$, we need to use a wide stencil based on a local coordinate rotation to discretize the second derivative terms $(\mathbf{D}\nabla) \cdot \nabla \mathcal{U}$ in the HJB equation (2.2). When using the wide stencil discretization, we use an upstream finite differencing for the first order derivatives. We avoid using points below the lower boundaries for $\big((S_1)_i, (S_2)_j, \tau^{n+1}\big) \in \Omega_{w^*}$. We use the asymptotic solution (4.15) of the HJB equation at a point outside the computational region at the upper boundaries. From the discretization (4.13), we can see that the measure of Ω_{out} converges to zero as $h \to 0$ (4.2). Lastly, fully implicit time-stepping is used to ensure the unconditional monotonicity of our numerical scheme.

Ω	$[0, S_{1,\max}] \times [0, S_{2,\max}] \times [0, T]$
Ω_{τ_0}	$[0, S_{1,\max}] \times [0, S_{2,\max}] \times \{0\}$
Ω_{up}	$\{S_{1,\max}\} \times (0, S_{2,\max}] \times (0, T] \cup (0, S_{1,\max}] \times \{S_{2,\max}\} \times (0, T]$
Ω_{in}	$\Omega/\Omega_{\tau_0}/\Omega_{up}$
Ω_f	The region in Ω_{in} where conditions (4.22) in Theorem 4.1 hold.
Ω_b	$[h, \sqrt{h}] \times (0, S_{2,\max}] \times (0, T] \cup (0, S_{1,\max}] \times [h, \sqrt{h}] \times (0, T]$.
Ω_{w^*}	The region in Ω_b that does not satisfy the condition (4.22).
Ω_w	$\Omega_{in}/\Omega_f/\Omega_{w^*}$
Ω_{out}	$(S_{1,\max}, S_{1,\max} + \sqrt{h}] \times [0, S_{2,\max} + \sqrt{h}] \times (0, T] \cup [0, S_{1,\max}] \times (S_{2,\max}, S_{2,\max} + \sqrt{h}] \times (0, T]$

Table 4.1: The domain definitions.

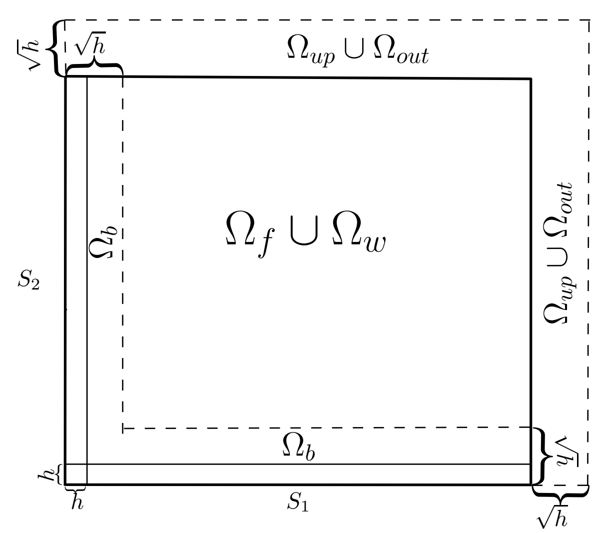


Figure 4.3: The domain descriptions.

4.7 Discretization form

We will give details of the discretization for the HJB equation (2.2) in Ω_{in} in this section. For the case $\left((S_1)_i, (S_2)_j, \tau^{n+1}\right) \in \Omega_f$ where the fixed point stencil is used, the HJB equation (2.2) has the following discretized form

$$\frac{\mathcal{U}_{i,j}^{n+1} - \mathcal{U}_{i,j}^{n}}{\Delta\tau} = \sup_{Q\in\partial Z}\left(L_f^Q \mathcal{U}_{i,j}^{n+1}\right), \tag{4.24}$$

where the discretized linear operator L_f^Q is defined in (4.6).

Remark 4.3. *(Restricting the control to the boundary) In the discrete equations $L_f^Q\mathcal{U}_{i,j}^{n+1}$, the numerical approximations of first order derivatives depend on the stencil, backward, forward or central differencing, which depend on the control. Thus, the discrete first order derivatives are also involved in the optimization of the discrete equations. In addition, the numerical approximation of the cross derivative term in (4.6) is dependent on the sign of the correlation ρ. In Proposition 3.1, the objective function contains just the diffusion terms, and we assume that $\Gamma_{kl},\ k,l = 1,2$ are constant and independent of the control. Therefore, the optimal value of the discrete equations is not necessarily attained at the boundary ∂Z. However, Proposition 3.1 holds for a smooth test function. Consequently, restricting the control to the boundary of the control set is a consistent approximation in the viscosity sense. Note that we also have an analytic expression for the optimal control for the discrete equations $L_f^Q\mathcal{U}_{i,j}^{n+1}$ when restricting $Q \in \partial Z$. See details in Section E.*

For the case $\left((S_1)_i, (S_2)_j, \tau^{n+1}\right) \in \Omega_w$ where the wide stencil is used, the discretized form of the linear differential operator $\mathcal{L}$ (2.3) is denoted by L_w^Q.

$$\begin{aligned}
L_w^Q\mathcal{U}_{i,j}^{n+1} &= \frac{a_{i,j}}{h}\mathcal{J}_h\mathcal{U}^{n+1}\left(\mathbf{S}_{i,j} + \sqrt{h}(\mathbf{R}_{i,j})_1\right) + \frac{a_{i,j}}{h}\mathcal{J}_h\mathcal{U}^{n+1}\left(\mathbf{S}_{i,j} - \sqrt{h}(\mathbf{R}_{i,j})_1\right)\\
&+ \frac{b_{i,j}}{h}\mathcal{J}_h\mathcal{U}^{n+1}\left(\mathbf{S}_{i,j} + \sqrt{h}(\mathbf{R}_{i,j})_2\right) + \frac{b_{i,j}}{h}\mathcal{J}_h\mathcal{U}^{n+1}\left(\mathbf{S}_{i,j} - \sqrt{h}(\mathbf{R}_{i,j})_2\right)\\
&+ 1_{(r-q_1)\geq 0}\frac{(r-q_1)(S_1)_i}{\Delta^+(S_1)_i}\mathcal{U}_{i+1,j}^{n+1} - 1_{(r-q_1)<0}\frac{(r-q_1)(S_1)_i}{\Delta^-(S_1)_i}\mathcal{U}_{i-1,j}^{n+1} + 1_{(r-q_2)\geq 0}\frac{(r-q_2)(S_2)_j}{\Delta^+(S_2)_j}\mathcal{U}_{i,j+1}^{n+1}\\
&- 1_{(r-q_2)<0}\frac{(r-q_2)(S_2)_j}{\Delta^-(S_2)_j}\mathcal{U}_{i,j-1}^{n+1} - \Bigg(1_{(r-q_1)\geq 0}\frac{(r-q_1)(S_1)_i}{\Delta^+(S_1)_i} - 1_{(r-q_1)<0}\frac{(r-q_1)(S_1)_i}{\Delta^-(S_1)_i}\\
&+ 1_{(r-q_2)\geq 0}\frac{(r-q_2)(S_2)_j}{\Delta^+(S_2)_j} - 1_{(r-q_2)<0}\frac{(r-q_2)(S_2)_j}{\Delta^-(S_2)_j} + \frac{2a_{i,j}}{h} + \frac{2b_{i,j}}{h} + r\Bigg)\mathcal{U}_{i,j}^{n+1},
\end{aligned} \tag{4.25}$$

where $a_{i,j}$ and $b_{i,j}$ are given in (4.11), and the presence of $\mathcal{J}_h\mathcal{U}^{n+1}\left(\mathbf{S}_{i,j} \pm \sqrt{h}(\mathbf{R}_{i,j})_k\right)$, $k = 1,2$ is due to the discretization of the second derivative terms (4.13). As defined in (4.12), $\mathcal{U}^n(\mathbf{S}) \equiv \mathcal{U}(S_1, S_2, \tau^n)$, $\mathbf{S} = (S_1, S_2)$ and $\mathbf{S}_{i,j} = ((S_1)_i, (S_2)_j)$.

Remark 4.4. *The points $\mathbf{S}_{i,j} \pm \sqrt{h}(\mathbf{R}_{i,j})_k,\ k = 1,2$ used in (4.25) are control Q dependent. Therefore, the discretization in this case will depend on the control. We indicate this fact in the notation of the discrete linear operator L_w^Q.*

Since the numerical approximations of the diffusion terms depend on the control in the discrete equations $L_w^Q\mathcal{U}_{i,j}^{n+1}$, there is no simple analytic expression which can be used to maximize the discrete equations (4.25). We also do not have any known convexity properties of (4.25). For a compact set of the controls, we must find the global maximum of (4.25) to ensure that our policy iteration algorithm converges. Hence, we discretize the control set Z (2.5), and maximize by linear search.

As explained in Remark 4.3, we will maximize the discrete equations $L_w^Q\mathcal{U}_{i,j}^{n+1}$ restricting the control to ∂Z. This significantly reduces the computational cost. We denote ∂Z_h as the discrete approximation of ∂Z

$$\partial Z_h = \{(\sigma_1)_1, \ldots, (\sigma_1)_{l_{\max}}\} \times \{(\sigma_2)_1, \ldots, (\sigma_2)_{k_{\max}}\} \times \{\rho_{\min}, \rho_{\max}\}, \tag{4.26}$$

where $(\sigma_1)_1 = \sigma_{1,\min}$, $(\sigma_1)_{l_{\max}} = \sigma_{1,\max}$, $(\sigma_2)_1 = \sigma_{2,\min}$, and $(\sigma_2)_{k_{\max}} = \sigma_{2,\max}$. Let

$$\max_i \left((\sigma_1)_i - (\sigma_1)_{i-1}\right) = C_4 h \text{ and } \max_i \left((\sigma_2)_i - (\sigma_2)_{i-1}\right) = C_5 h, \tag{4.27}$$

where h (4.2) is the mesh discretization parameter.

Finally, using fully implicit timestepping, the HJB equation (2.2) has the following discretized form for this case

$$\frac{\mathcal{U}^{n+1} - \mathcal{U}^n}{\Delta\tau} = \sup_{Q \in \partial Z_h} \left(L_w^Q \mathcal{U}_{i,j}^{n+1}\right). \tag{4.28}$$

For the case $\left((S_1)_i, (S_2)_j, \tau^{n+1}\right) \in \Omega_{w^*}$, we need to adapt the discretized linear operator L_w^Q to avoid using points below the lower boundaries as described in Algorithm 4.1. The details of discretized equation for this case are given in Appendix B.

4.8 The matrix form of the discrete equations

It is convenient to use a matrix form to represent the discretized equations for computational purposes. In this section we define a number of matrices and vectors to represent the discretized PDEs in (4.24), (4.28) and (B.2). Let $\mathcal{U}_{i,j}^n$ be the approximate solution of the equation (2.2) at $((S_1)_i, (S_2)_j, \tau^n)$, $1 \le i \le N_1$, $1 \le j \le N_2$ and $0 \le \tau^n \le N_\tau$, and form the solution vector

$$\mathbf{U}^n = \left(\mathcal{U}_{1,1}^n, \mathcal{U}_{2,1}^n, \ldots, \mathcal{U}_{N_1,1}^n, \ldots, \mathcal{U}_{1,N_2}^n, \ldots, \mathcal{U}_{N_1,N_2}^n\right). \tag{4.29}$$

It will sometimes be convenient to use a single index when referring to an entry of the solution vector

$$\mathcal{U}_\ell^n = \mathcal{U}_{i,j}^n, \quad \ell = i + (j-1)N_1.$$

Let $N = N_1 \times N_2$, and we define the $N \times N$ matrix $\mathbf{L}^{n+1}(\mathcal{Q})$, where

$$\mathcal{Q} = \{Q_1, \ldots, Q_N\} \tag{4.30}$$

is an indexed set of N controls, and each Q_ℓ is in the set of admissible controls. $\mathbf{L}_{\ell,k}^{n+1}(\mathcal{Q})$ is the entry on the ℓ-th row and k-th column, where $\ell = i + (j-1)N_1,\ i = 1, \ldots, N_1,\ j = 1, \ldots, N_2$.

For the case $\left((S_1)_i, (S_2)_j, \tau^{n+1}\right) \in \Omega_{up}$ where the Dirichlet boundary condition (4.15) is imposed, and we then have

$$\mathbf{L}_{\ell,k}^{n+1}(\mathcal{Q}) = 0, \quad k = 1, \ldots, N, \tag{4.31}$$

and define the vector $\mathbf{F}^{n+1}$ with entries

$$\mathbf{F}_\ell^{n+1} = \begin{cases} \mathcal{U}_A\left((S_1)_i, (S_2)_j, \tau^{n+1}\right), & \left((S_1)_i, (S_2)_j, \tau^{n+1}\right) \in \Omega_{up}, \\ 0, & \text{otherwise.} \end{cases} \tag{4.32}$$

For the case $\left((S_1)_i, (S_2)_j, \tau^{n+1}\right) \in \Omega_f$, the entries $\mathbf{L}_{\ell,k}^{n+1}(\mathcal{Q})$ are constructed from the discrete linear operator L_f^Q (4.6). That is,

$$[\mathbf{L}^{n+1}(\mathcal{Q})\mathbf{U}^{n+1}]_\ell = L_f^Q \mathcal{U}_{i,j}^{n+1}. \tag{4.33}$$

For the case $\left((S_1)_i, (S_2)_j, \tau^{n+1}\right) \in \Omega_w$, we need to use the values at the following four off-grid points $\mathbf{S}_{i,j} \pm \sqrt{h}(\mathbf{R}_{i,j})_k,\ k = 1, 2$ in the discrete linear operator L_w^Q (4.25). Let these four points denoted as $P_{i,j}^m,\ m = 1, 2, 3, 4$, respectively. Note that these points may be outside the bounded domain Ω_{in}. When $P_{i,j}^m \in \Omega_{in}$, using linear interpolation, values at these four points are approximated as follows

$$\mathcal{J}_h \mathcal{U}^{n+1}\left(P_{i,j}^m\right) = \begin{cases} \displaystyle\sum_{\substack{d=0,1 \\ e=0,1}} \omega_{i,j}^{p_m+d, q_m+e} \mathcal{U}_{p_m+d, q_m+e}^{n+1}, & P_{i,j}^m \in \Omega_{in} \\ 0, & \text{Otherwise} \end{cases}. \tag{4.34}$$

For linear interpolation, we have that $\omega_{i,j}^{p_m+d,q_m+e} \geq 0$ and $\sum_{\substack{d=0,1\\e=0,1}} \omega_{i,j}^{p_m+d,q_m+e} = 1$. By inserting (4.34) in (4.25), the entries $\mathbf{L}_{\ell,k}^{n+1}(\mathcal{Q})$ on ℓ-th row are then specified. When a point $P_{i,j}^m$ is outside the domain Ω_{in} and inside the domain Ω_{out}, we then use its asymptotic solution at the point without extrapolating its value. We need to define the vector $\mathbf{B}^{n+1}(\mathcal{Q})$ to facilitate the construction of the matrix form in this situation when we use a point in the domain Ω_{out}.

$$\mathbf{B}_\ell^{n+1}(\mathcal{Q}) = \begin{cases} 1_{P_{i,j}^1 \in \Omega_{out}} \frac{a_{i,j}}{h} \mathcal{U}_A(P_{i,j}^1) + 1_{P_{i,j}^2 \in \Omega_{out}} \frac{a_{i,j}}{h} \mathcal{U}_A(P_{i,j}^2) & \\ \qquad + 1_{P_{i,j}^3 \in \Omega_{out}} \frac{b_{i,j}}{h} \mathcal{U}_A(P_{i,j}^3) + 1_{P_{i,j}^4 \in \Omega_{out}} \frac{b_{i,j}}{h} \mathcal{U}_A(P_{i,j}^4), & \big((S_1)_i, (S_2)_j, \tau^{n+1}\big) \in \Omega_w \cup \Omega_{w^*} \\ 0, & \text{otherwise} \end{cases}, \tag{4.35}$$

where $\mathcal{U}_A(P_{i,j}^m)$ is the asymptotic solution (4.15) at the point. As a result, for the case $\big((S_1)_i, (S_2)_j, \tau^{n+1}\big) \in \Omega_w$, we have

$$[\mathbf{L}^{n+1}(\mathcal{Q})\mathbf{U}^{n+1}]_\ell + \mathbf{B}_\ell^{n+1}(\mathcal{Q}) = L_w^Q \mathcal{U}_{i,j}^{n+1}. \tag{4.36}$$

Lastly, for $\big((S_1)_i, (S_2)_j, \tau^{n+1}\big) \in \Omega_{w^*}$, using the corresponding discrete linear operator $L_{w^*}^Q$ (B.1), the entries $\mathbf{L}_{\ell,k}^{n+1}(\mathcal{Q})$ are constructed similarly to the previous case where $\big((S_1)_i, (S_2)_j, \tau^{n+1}\big) \in \Omega_w$.

Let

$$\mathbf{A}(\mathcal{Q}) \equiv [\mathbf{I} - \Delta\tau \mathbf{L}^{n+1}(\mathcal{Q})], \tag{4.37}$$

and

$$\mathbf{C}(\mathcal{Q}) \equiv \mathbf{U}^n + \mathbf{F}^{n+1} - \mathbf{F}^n + \Delta\tau \mathbf{B}^{n+1}(\mathcal{Q}). \tag{4.38}$$

so that the discretized equations are written in the compact form

$$\sup_{\mathcal{Q} \in \hat{Z}} \big\{ -\mathbf{A}(\mathcal{Q})\mathbf{U}^{n+1} + \mathbf{C}(\mathcal{Q}) \big\} = 0, \tag{4.39}$$

where we define $\hat{Z}$ as

$$\hat{Z} = \begin{cases} \partial Z, & \big((S_1)_i, (S_2)_j, \tau^{n+1}\big) \in \Omega_f, \\ \partial Z_h, & \big((S_1)_i, (S_2)_j, \tau^{n+1}\big) \in \Omega_w \cup \Omega_{w^*}, \end{cases}. \tag{4.40}$$

5 Convergence to the viscosity solution

In general, we cannot expect solutions to the HJB equation (2.2) to be smooth. Hence, we seek the viscosity solution of the equation (2.2). From Barles et al. (1995), we find that a sufficient condition which guarantees convergence to the viscosity solution is that the numerical scheme is ℓ_∞ stable, consistent in the viscosity sense, and monotone. In the following sections, we will verify each of the properties in turn for our numerical scheme.

5.1 Viscosity solution for the localized problem

To make the statement of the problem more precise in the context of viscosity solutions, we now write the localized problem in a compact form, which includes the terminal and boundary equations in a single equation. Let us define

$$\mathbf{x} = (S_1, S_2, \tau), \quad D\mathcal{U}(\mathbf{x}) = (\frac{\partial \mathcal{U}}{\partial S_1}, \frac{\partial \mathcal{U}}{\partial S_2}), \quad D^2\mathcal{U}(\mathbf{x}) = \begin{pmatrix} \frac{\partial^2 \mathcal{U}}{\partial S_1^2} & \frac{\partial^2 \mathcal{U}}{\partial S_1 \partial S_2} \\ \frac{\partial^2 \mathcal{U}}{\partial S_1 \partial S_2} & \frac{\partial^2 \mathcal{U}}{\partial S_2^2} \end{pmatrix}.$$

The HJB equation for the value function (2.2) on the localized domain $\Omega \cup \Omega_{out}$ is given by

$$F\mathcal{U} \equiv F\left(\mathbf{x}, \mathcal{U}(\mathbf{x}), D\mathcal{U}(\mathbf{x}), D^2\mathcal{U}(\mathbf{x})\right) = 0, \tag{5.1}$$

where the operator $F\mathcal{U}$ is defined by

$$F\mathcal{U} = \begin{cases} F_{in}\mathcal{U} \equiv F_{in}\left(\mathbf{x}, \mathcal{U}(\mathbf{x}), D\mathcal{U}(\mathbf{x}), D^2\mathcal{U}(\mathbf{x})\right), & \mathbf{x} \in \Omega_{in} = \Omega_f \cup \Omega_w \cup \Omega_{w^*}, \\ F_{\tau_0}\mathcal{U} \equiv F_{\tau_0}\left(\mathbf{x}, \mathcal{U}(\mathbf{x})\right), & \mathbf{x} \in \Omega_{\tau_0}, \\ F_{\max}\mathcal{U} \equiv F_{\max}\left(\mathbf{x}, \mathcal{U}(\mathbf{x})\right), & \mathbf{x} \in \Omega_{up} \cup \Omega_{out}. \end{cases} \tag{5.2}$$

Here,

$$\begin{aligned} F_{in}\mathcal{U} &= \mathcal{U}_\tau - \max_{Q \in Z}\left(\mathcal{L}\mathcal{U}\right),\ (2.2) \\ F_0\mathcal{U} &= \mathcal{U} - \mathcal{W}(S_1, S_2), \\ F_{\max}\mathcal{U} &= \mathcal{U} - \mathcal{U}_A(S_1, S_2, \tau)\ , \end{aligned} \tag{5.3}$$

where $\mathcal{U}_A$ is the asymptotic form of the solution, as in equation (4.15).

Before defining the viscosity solution of equation (5.1), we first recall the definitions of upper and lower semi-continuous envelopes. Given a function $f : \tilde{\Omega} \to \mathbb{R},\ \tilde{\Omega} \subseteq \mathbb{R}^n$, the upper semi-continuous envelope of f, denoted by f^*, is defined as

$$f^*(\tilde{x}) = \lim_{\tilde{r} \to 0^+}\left[\sup\left\{f(y)\ \middle|\ y \in B(\tilde{x}, \tilde{r}) \cap \tilde{\Omega}\right\}\right], \tag{5.4}$$

where $B(\tilde{x}, r) = \{y \in \mathbb{R}^n \mid |\tilde{x} - y| < \tilde{r}\}$. We also have the obvious definition for a lower semi-continuous envelope $f_*(\tilde{x})$.

We also define

$$\limsup_{y \to \tilde{x}} f(y) = \lim_{\tilde{r} \to 0^+}\left[\sup\left\{f(y)\ \middle|\ y \in B(\tilde{x}, \tilde{r}) \cap \tilde{\Omega} - \tilde{x}\right\}\right], \tag{5.5}$$

with the corresponding definition of $\liminf$.

Definition 5.1. *(Viscosity solution of equation 5.1) A locally bounded function $\mathcal{U} : \Omega \cup \Omega_{out} \to \mathbb{R}$ is a viscosity sub-solution (resp. super-solution) of equation (5.1) if, for all test functions $\phi(\mathbf{x}) \in C^\infty(\Omega \cup \Omega_{out})$, and all $\mathbf{x}$, such that $\mathcal{U} - \phi$ has a strict global maximum (resp. minimum) with $\phi(\mathbf{x}) = \mathcal{U}^*(\mathbf{x})$ (resp. $\mathcal{U}_*(\mathbf{x})$), we have*

$$\begin{aligned} &F_*\left(\mathbf{x}, \phi(\mathbf{x}), D\phi(\mathbf{x}), D^2\phi(\mathbf{x})\right) \le 0, \\ &\left(\ \textit{resp. } F^*\left(\mathbf{x}, \phi(\mathbf{x}), D\phi(\mathbf{x}), D^2\phi(\mathbf{x})\right) \ge 0\right), \end{aligned} \tag{5.6}$$

where $F_*(\cdot)$ is the lower semi-continuous envelope of F (resp. the upper semi-continuous envelope F^*). $\mathcal{U}$ is a viscosity solution if it is both a viscosity sub-solution and a viscosity super-solution.

Proposition 5.1. *(Strong comparison) Suppose the payoff function $\mathcal{W}(S_1, S_2)$ at expiry time T is continuous with quadratic growth, then the value function satisfies a strong comparison result, hence there exists an unique continuous viscosity solution of the problem (2.2) (Pham, 2005; Guyon and Henry-Labordere, 2011).*

Proof. See Pham (2005). ☐

Corollary 5.1. *Note that we restrict ourselves to a finite domain $\Omega \cup \Omega_{out}$ for the HJB equation $F\mathcal{U}$ defined in (5.1), hence the value function (5.1) satisfies a strong comparison result.*

5.2 Consistency

For the purpose of proving convergence to the viscosity solution, it is more convenient to rewrite equations (4.24), (4.28) and (B.2) in an equivalent form. Let $\mathcal{G}(\cdot)$ be the discrete approximation to F_{in} for $\mathbf{x} \in \Omega_{in}$, and $\mathbf{x}_{i,j}^{n+1} = ((S_1)_i, (S_2)_j, \tau^{n+1})$. For $\mathbf{x}_{i,j}^{n+1} \in \Omega_f$, from (4.24), we have

$$\mathcal{G}\left(h, \mathbf{x}_{i,j}^{n+1}, \mathcal{U}_{i,j}^{n+1}, \left\{\mathcal{U}_{a,b}^{n+1}\right\}_{\substack{a \ne i \\ \text{or } b \ne j}}, \{\mathcal{U}_{k,l}^n\}\right) = \frac{\mathcal{U}_{i,j}^{n+1} - \mathcal{U}_{i,j}^n}{\Delta\tau} - \sup_{Q \in \partial Z}\left(L_f^Q \mathcal{U}_{i,j}^{n+1}\right) = 0. \tag{5.7}$$

For $\mathbf{x}_{i,j}^{n+1} \in \Omega_w$, from (4.28), we have

$$\mathcal{G}\left(h, \mathbf{x}_{i,j}^{n+1}, \mathcal{U}_{i,j}^{n+1}, \left\{\mathcal{U}_{a,b}^{n+1}\right\}_{\substack{a\neq i \\ \text{or } b\neq j}}, \{\mathcal{U}_{k,l}^{n}\}\right) = \frac{\mathcal{U}_{i,j}^{n+1} - \mathcal{U}_{i,j}^{n}}{\Delta\tau} - \sup_{Q\in\partial Z_h}\left(L_w^Q \mathcal{U}_{i,j}^{n+1}\right) = 0. \tag{5.8}$$

For $\mathbf{x}_{i,j}^{n+1} \in \Omega_{w^*}$, from (B.2), we have

$$\mathcal{G}\left(h, \mathbf{x}_{i,j}^{n+1}, \mathcal{U}_{i,j}^{n+1}, \left\{\mathcal{U}_{a,b}^{n+1}\right\}_{\substack{a\neq i \\ \text{or } b\neq j}}, \{\mathcal{U}_{k,l}^{n}\}\right) = \frac{\mathcal{U}_{i,j}^{n+1} - \mathcal{U}_{i,j}^{n}}{\Delta\tau} - \sup_{Q\in\partial Z_h}\left(L_{w^*}^Q \mathcal{U}_{i,j}^{n+1}\right) = 0. \tag{5.9}$$

Finally, we have

$$\mathcal{G}(\cdot) = 0 = \begin{cases} \mathcal{U}((S_1)_i, (S_2)_j, 0) - \mathcal{W}((S_1)_i, (S_2)_j), & \mathbf{x}_{i,j}^{n+1} \in \Omega_{\tau_0}, \\ \mathcal{U}((S_1)_i, (S_2)_j, \tau^{n+1}) - \mathcal{U}_A((S_1)_i, (S_2)_j, \tau^{n+1}), & \mathbf{x}_{i,j}^{n+1} \in \Omega_{up} \cup \Omega_{out}. \end{cases} \tag{5.10}$$

The domains $\Omega_f, \ldots, \Omega_{out}$ are defined in Table 4.1, and $\mathcal{U}_A$ is defined in equation (4.15).

Definition 5.2. *(Consistency) For any C^∞ function $\phi(S_1, S_2, \tau)$ in $\Omega \cup \Omega_{out}$, with $\phi_{i,j}^{n+1} = \phi(\mathbf{x}_{i,j}^{n+1}) = \phi\left((S_1)_i, (S_2)_j, \tau^{n+1}\right)$, the numerical scheme $\mathcal{G}(\cdot)$ is consistent in the viscosity sense, if, $\forall \hat{\mathbf{x}} = (\hat{S}_1, \hat{S}_2, \hat{\tau})$ with $\mathbf{x}_{i,j}^{n+1} = \left((S_1)_i, (S_2)_j, \tau^{n+1}\right)$, the following holds*

$$\limsup_{\substack{h\to 0 \\ \psi\to 0 \\ \mathbf{x}_{i,j}^{n+1}\to\hat{\mathbf{x}}}} \mathcal{G}\left(h, \mathbf{x}_{i,j}^{n+1}, \phi_{i,j}^{n+1} + \psi, \left\{\phi_{a,b}^{n+1} + \psi\right\}_{\substack{a\neq i \\ or\ b\neq j}}, \left\{\phi_{k,l}^{n} + \psi\right\}\right) \le F^*\left(\hat{\mathbf{x}}, \phi(\hat{\mathbf{x}}), D\phi(\hat{\mathbf{x}}), D^2\phi(\hat{\mathbf{x}})\right), \tag{5.11}$$

and

$$\liminf_{\substack{h\to 0 \\ \psi\to 0 \\ \mathbf{x}_{i,j}^{n+1}\to\hat{\mathbf{x}}}} \mathcal{G}\left(h, \mathbf{x}_{i,j}^{n+1}, \phi_{i,j}^{n+1} + \psi, \left\{\phi_{a,b}^{n+1} + \psi\right\}_{\substack{a\neq i \\ or\ b\neq j}}, \left\{\phi_{k,l}^{n} + \psi\right\}\right) \ge F_*\left(\hat{\mathbf{x}}, \phi(\hat{\mathbf{x}}), D\phi(\hat{\mathbf{x}}), D^2\phi(\hat{\mathbf{x}})\right). \tag{5.12}$$

Lemma 5.1. *(Local consistency). Suppose the mesh discretization parameter h is defined in (4.2) and the control discretization satisfies equation (4.27), then for any C^∞ function $\phi(S_1, S_2, \tau)$ in $\Omega \cup \Omega_{out}$, with $\phi_{i,j}^{n+1} = \phi\left((S_1)_i, (S_2)_j, \tau^{n+1}\right) = \phi(\mathbf{x}_{i,j}^{n+1})$, and for h, ψ sufficiently small, ψ a constant, we have that*

$$\begin{aligned} &\mathcal{G}\left(h, \mathbf{x}_{i,j}^{n+1}, \phi_{i,j}^{n+1} + \psi, \left\{\phi_{a,b}^{n+1} + \psi\right\}_{\substack{a\neq i \\ or\ b\neq j}}, \left\{\phi_{k,l}^{n} + \psi\right\}\right) \\ &= \begin{cases} F_{in}\phi_{i,j}^{n+1} + O(h) + O(\psi), & \mathbf{x}_{i,j}^{n+1} \in \Omega_f, \\ F_{in}\phi_{i,j}^{n+1} + O(h) + O(\psi), & \mathbf{x}_{i,j}^{n+1} \in \Omega_w, \\ F_{in}\phi_{i,j}^{n+1} + O(\sqrt{h}) + O(\psi), & \mathbf{x}_{i,j}^{n+1} \in \Omega_{w^*}, \\ F_{\tau_0}\phi_{i,j}^{n+1} + O(\psi), & \mathbf{x}_{i,j}^{n+1} \in \Omega_{\tau_0}, \\ F_{\max}\phi_{i,j}^{n+1} + O(\psi), & \mathbf{x}_{i,j}^{n+1} \in \Omega_{up} \cup \Omega_{out}. \end{cases} \end{aligned} \tag{5.13}$$

Proof. To be precise, define the following

$$\begin{aligned} \mathcal{L}\phi_{i,j}^{n+1} &\equiv \mathcal{L}\phi((S_1)_i, (S_2)_j, \tau^{n+1}), \\ (\phi_\tau)_{i,j}^{n+1} &\equiv \phi_\tau((S_1)_i, (S_2)_j, \tau^{n+1}). \end{aligned} \tag{5.14}$$

For the case $\mathbf{x}_{i,j}^{n+1} \in \Omega_f$, $L_f^Q \phi_{i,j}^{n+1}$ (4.6) is a locally consistent discretization of the linear operator $\mathcal{L}$ (2.3), that is,

$$L_f^Q \phi_{i,j}^{n+1} = \mathcal{L}\phi_{i,j}^{n+1} + O(h), \tag{5.15}$$

which is easily proved by Taylor series, and note that

$$\begin{aligned} L_f^Q \left(\phi_{i,j}^{n+1} + \psi\right) &= L_f^Q \phi_{i,j}^{n+1} - r\psi, \\ \frac{\phi_{i,j}^{n+1} - \phi_{i,j}^n}{\Delta\tau} &= (\phi_\tau)_{i,j}^{n+1} + O(h). \end{aligned} \tag{5.16}$$

Since ϕ is a smooth test function, and $\frac{\partial^2 \phi}{\partial S_k \partial S_l}$, $k, l = 1, 2$ are independent of the control, then, by Proposition 3.1, we have

$$\sup_{Q \in \partial Z} \left(\mathcal{L}\phi_{i,j}^{n+1}\right) = \sup_{Q \in Z} \left(\mathcal{L}\phi_{i,j}^{n+1}\right), \tag{5.17}$$

and from equation (5.7) and (5.17), we then have the result

$$\begin{aligned} &\mathcal{G}\left(h, \mathbf{x}_{i,j}^{n+1}, \phi_{i,j}^{n+1} + \psi, \left\{\phi_{a,b}^{n+1} + \psi\right\}_{\substack{a \neq i \\ \text{or } b \neq j}}, \left\{\phi_{k,l}^n + \psi\right\}\right) \\ &= \frac{\phi_{i,j}^{n+1} - \phi_{i,j}^n}{\Delta\tau} - \sup_{Q \in \partial Z} \left(L_f^Q \phi_{i,j}^{n+1}\right) + O(\psi) \\ &= (\phi_\tau)_{i,j}^{n+1} - \sup_{Q \in \partial Z} \left(\mathcal{L}\phi_{i,j}^{n+1}\right) + O(\psi) + O(h) \\ &= (\phi_\tau)_{i,j}^{n+1} - \sup_{Q \in Z} \left(\mathcal{L}\phi_{i,j}^{n+1}\right) + O(\psi) + O(h) \\ &= F_{in}\phi_{i,j}^{n+1} + O(\psi) + O(h), \quad \mathbf{x}_{i,j}^{n+1} \in \Omega_f \end{aligned} \tag{5.18}$$

For the case where $\mathbf{x}_{i,j}^{n+1} \in \Omega_w$, $L_w^Q \phi_{i,j}^{n+1}$ (4.25) is also locally consistent,

$$L_w^Q \phi_{i,j}^{n+1} = \mathcal{L}\phi_{i,j}^{n+1} + O(h), \tag{5.19}$$

and note that

$$\begin{aligned} L_w^Q \left(\phi_{i,j}^{n+1} + \psi\right) &= L_w^Q \phi_{i,j}^{n+1} - r\psi, \\ \frac{\phi_{i,j}^{n+1} - \phi_{i,j}^n}{\Delta\tau} &= (\phi_\tau)_{i,j}^{n+1} + O(h). \end{aligned} \tag{5.20}$$

From equation (5.8), we then have

$$\begin{aligned} &\mathcal{G}\left(h, \mathbf{x}_{i,j}^{n+1}, \phi_{i,j}^{n+1} + \psi, \left\{\phi_{a,b}^{n+1} + \psi\right\}_{\substack{a \neq i \\ \text{or } b \neq j}}, \left\{\phi_{k,l}^n + \psi\right\}\right) \\ &= \frac{\phi_{i,j}^{n+1} - \phi_{i,j}^n}{\Delta\tau} - \sup_{Q \in \partial Z_h} \left(L_w^Q \phi_{i,j}^{n+1}\right) + O(\psi) \\ &= (\phi_\tau)_{i,j}^{n+1} - \sup_{Q \in \partial Z_h} \left(\mathcal{L}\phi_{i,j}^{n+1}\right) + O(\psi) + O(h). \end{aligned} \tag{5.21}$$

We discretize the set ∂Z and maximize the discrete equations by linear search. If the discretization step for the control is also $O(h)$, then this is a consistent approximation (Wang and Forsyth, 2008), since the equation coefficients are Lipschitz continuous functions of the controls. That is, using equation (5.17),

$$\sup_{Q \in \partial Z_h} \left(\mathcal{L}\phi_{i,j}^{n+1}\right) = \sup_{Q \in \partial Z} \left(\mathcal{L}\phi_{i,j}^{n+1}\right) + O(h) = \sup_{Q \in Z} \left(\mathcal{L}\phi_{i,j}^{n+1}\right) + O(h). \tag{5.22}$$

Using equation (5.22) in equation (5.21), we then have the final result

$$\begin{aligned}\mathcal{G}\left(h, \mathbf{x}_{i,j}^{n+1}, \phi_{i,j}^{n+1}+\psi, \left\{\phi_{a,b}^{n+1}+\psi\right\}_{\substack{a\neq i\\ \text{or } b\neq j}}, \left\{\phi_{k,l}^{n}+\psi\right\}\right)&\\ = (\phi_\tau)_{i,j}^{n+1} - \sup_{Q\in Z}\left(\mathcal{L}\phi_{i,j}^{n+1}\right) + O(\psi) + O(h),&\\ = F_{in}\phi_{i,j}^{n+1} + O(\psi) + O(h), \quad \mathbf{x}_{i,j}^{n+1}\in\Omega_w.&\end{aligned} \tag{5.23}$$

For the case $\mathbf{x}_{i,j}^{n+1}\in\Omega_{w^*}$, the proof is similar to the case $\mathbf{x}_{i,j}^{n+1}\in\Omega_w$, but the consistency of the discrete linear operator $L_{w^*}^{Q}$ is perhaps not obvious. A possible inconsistency may arise when we shrink the stencil length from $O(\sqrt{h})$ to $O(h)$ to avoid using points below the lower boundaries. However, consistency still holds for $L_{w^*}^{Q}$ (see the proof in Appendix C).

$$L_{w^*}^{Q}\phi_{i,j}^{n+1} = \mathcal{L}\phi_{i,j}^{n+1} + O(\sqrt{h}).$$

Following the same steps as the case $\mathbf{x}_{i,j}^{n+1}\in\Omega_w$, we finally have

$$\mathcal{G}\left(h, \mathbf{x}_{i,j}^{n+1}, \phi_{i,j}^{n+1}+\psi, \left\{\phi_{a,b}^{n+1}+\psi\right\}_{\substack{a\neq i\\ \text{or } b\neq j}}, \left\{\phi_{k,l}^{n}+\psi\right\}\right) = F_{in}\phi_{i,j}^{n+1} + O(\psi) + O(\sqrt{h}), \quad \mathbf{x}_{i,j}^{n+1}\in\Omega_{w^*}. \tag{5.24}$$

The remaining results in (5.13) can be proven using similar arguments. □

Lemma 5.2. *(Consistency) Provided that all conditions in Lemma 5.1 are satisfied, then scheme (5.7-5.10) is consistent according to Definition (5.2).*

Proof. This follows in straightforward fashion from Lemma 5.1, using the same steps as in, for example, Huang and Forsyth (2012). □

5.3 Stability

Definition 5.3. *(M-matrix) If a matrix* $\mathbf{A}$ *has elements* $a_{ii} > 0$ *and* $a_{ij} < 0$ *for* $i \neq j$ *and every row sum is non-negative with at least one row sum positive in each connected part of* $\mathbf{A}$*, then* $\mathbf{A}$ *is an M-matrix (Varga, 2009).*

Remark 5.1. *We remind the reader that a sufficient condition for a matrix* $\mathbf{A}$ *to be an M-matrix is that* $\mathbf{A}$ *has positive diagonals, non-positive offdiagonals, and is diagonally dominant (Varga, 2009).*

Lemma 5.3. *Providing the following conditions hold*

- *We only use the discrete linear operator* L_f^Q *(4.6) in the domain* Ω_f*,*
- *A linear interpolation operator* $\mathcal{J}_h$ *is used in (4.25) and (B.1).*

Then, $\mathbf{A}(\mathcal{Q}) = [\mathbf{I} - \Delta\tau\mathbf{L}^{n+1}(\mathcal{Q})]$ *(4.39) is an M-matrix, with*

$$\sum_k [\mathbf{I} - \Delta\tau\mathbf{L}^{n+1}(\mathcal{Q})]_{\ell,k} \geq 1 \ . \tag{5.25}$$

Proof. From the formation of matrix $\mathbf{L}$ in (4.31), (4.33) and (4.36), it is easily seen that $[\mathbf{I} - \Delta\tau\mathbf{L}^{n+1}(\mathcal{Q})]$ has positive diagonals, non-positive offdiagonals, and the ℓ-th row sums for the matrix is

$$\sum_k \left[\mathbf{I} - \Delta\tau\mathbf{L}^{n+1}(\mathcal{Q})\right]_{\ell,k} = \begin{cases} 1 + r\Delta\tau & i = 1,\ldots,N_1-1, \ , j = 1,\ldots,N_2-1, \\ 1 & i = N_1 \text{ or } j = N_2, \end{cases} \tag{5.26}$$

where $\ell = i + (j-1)N_1$. Thus, the matrix $[\mathbf{I} - \Delta\tau\mathbf{L}^{n+1}(\mathcal{Q})]$ is diagonally dominant. □

Lemma 5.4. *(Stability) If the conditions for Lemma 5.3 are satisfied, the discretization (4.39), equivalently (5.7-5.10), is unconditionally l_∞ stable, as mesh discretization parameter (4.2) $h \to 0$, satisfying*

$$\|\mathbf{U}^n\|_\infty \le \max\left(\|\mathbf{U}^0\|_\infty,\ C_6\right), \tag{5.27}$$

where $C_6 = \max_n \|\mathbf{F}^n\|_\infty$, where $\mathbf{F}^n$ is determined by the asymptotic boundary condition (4.15).

Proof. By Lemma 5.3, and using a straightforward maximum analysis as in d'Halluin et al. (2004), the result follows. □

Remark 5.2. *From the properties of M-matrices and equation (5.26) we have that*

$$\|\mathbf{A}(\mathcal{Q})^{-1}\|_\infty = \|[\mathbf{I} - \Delta\tau\mathbf{L}^{n+1}(\mathcal{Q})]^{-1}\|_\infty \le \max_\ell \frac{1}{rowsum([\mathbf{I} - \Delta\tau\mathbf{L}^{n+1}(\mathcal{Q})]_\ell)} \le 1 \tag{5.28}$$

5.4 Monotonicity

Definition 5.4. *(Monotonicity) The discrete scheme is monotone if for all $\mathcal{Y}^n_{i,j} \ge \mathcal{X}^n_{i,j},\ \forall i,j,n$*

$$\mathcal{G}\left(h, \mathbf{x}^{n+1}_{i,j}, \mathcal{U}^{n+1}_{i,j}, \left\{\mathcal{Y}^{n+1}_{a,b}\right\}_{\substack{a\neq i \\ or\ b\neq j}}, \left\{\mathcal{Y}^n_{k,l}\right\}\right) \le \mathcal{G}\left(h, \mathbf{x}^{n+1}_{i,j}, \mathcal{U}^{n+1}_{i,j}, \left\{\mathcal{X}^{n+1}_{a,b}\right\}_{\substack{a\neq i \\ or\ b\neq j}}, \left\{\mathcal{X}^n_{k,l}\right\}\right). \tag{5.29}$$

Lemma 5.5. *(Monotonicity) If the scheme (5.7-5.10) satisfies the conditions required for Lemma 5.3, then the discretization is monotone, according to Definition 5.4.*

Proof. Since our discretization is a positive coefficient scheme $\forall Q \in \hat{Z}$ (4.40), monotonicity follows using the same steps as in Forsyth and Labahn (2007). □

5.5 Convergence

Theorem 5.1. *(Convergence) Assume that discretization (5.7-5.10) satisfies all the conditions required by Lemma 5.2, 5.4 and 5.5, and that Proposition 5.1 holds, then numerical scheme (5.7-5.10) converges to the unique continuous viscosity solution of the problem* (5.1).

Proof. Since the scheme is monotone, consistent and ℓ_∞-stable, this follows from the results in Barles and Souganidis (1991). □

6 Solution of the nonlinear discrete algebraic equations

Although we have established that discretization (4.39) is consistent, ℓ_∞ stable and monotone, fully implicit timestepping requires solution of highly nonlinear algebraic equations at each timestep. For the applications addressed in Forsyth and Labahn (2007) an efficient method for solving the associated algebraic systems made use of a policy iteration scheme. However, our discretization method is control dependent, and consequently the local objective function may be a discontinuous function of the control (Wang and Forsyth, 2008; Huang et al., 2012). Hence some care must be taken when applying policy iteration. Recall that at every timestep τ^n, the nonlinear algebraic linear equations (4.39) can be represented as in the form

$$\sup_{\mathcal{Q}\in\hat{Z}} \left\{-\mathbf{A}(\mathcal{Q})\mathbf{U}^{n+1} + \mathbf{C}(\mathcal{Q})\right\} = 0, \tag{6.1}$$

where $\mathcal{Q} \in \hat{Z}$ (see the definition of $\hat{Z}$ in (4.40)) denotes that each $Q_\ell \in \hat{Z},\ \ell = 1, \ldots, N$. Equation (6.1) is to be understood in the row-wise sense, i.e. $\sup_{\mathcal{Q}\in\hat{Z}}[\ \cdot\ ]_\ell = 0\ ;\ \ell = 1, \cdots, N_1N_2$.

Before proceeding with a discussion of Policy Iteration, for solution of equation (6.1), we list here a set of properties of $\mathbf{A}(\mathcal{Q})$, $\mathbf{C}(\mathcal{Q})$, $\hat{Z}$, which will prove useful in later sections.

Properties 6.1. *(Properties of* $\mathbf{A}(\mathcal{Q})$*,* $\mathbf{C}(\mathcal{Q})$*,* $\hat{Z}$*)*

(i) The set of controls $\hat{Z}$ *(4.40) is compact.*

(ii) The matrices and vectors have the property that $\mathbf{A}_{\ell,k}(\mathcal{Q})$ *and* $\mathbf{C}_\ell(\mathcal{Q})$ *depend only on* Q_ℓ*. That is,* $\mathbf{A}_{\ell,k}(\mathcal{Q}) = \mathbf{A}_{\ell,k}(Q_\ell)$ *and* $\mathbf{C}_\ell(\mathcal{Q}) = \mathbf{C}_\ell(Q_\ell)$*.*

(iii) $\mathbf{A}(\mathcal{Q})$ *is a diagonally dominant M-matrix* $\forall \mathcal{Q}$*, and* $\sum_k \mathbf{A}_{\ell,k}(\mathcal{Q}) \geq C_r > 0$*, where* C_r *is independent of* $\mathcal{Q}$ *and row* ℓ*.*

(iv) $\|\mathbf{A}(\mathcal{Q})\|_\infty$*,* $\|\mathbf{C}(\mathcal{Q})\|_\infty$*, and* $\|\mathbf{A}(\mathcal{Q})^{-1}\|_\infty$ *are bounded uniformly w.r.t.* $\mathcal{Q}$*.*

Lemma 6.1 (Verification of Properties 6.1)**.** *The discretization (4.39) satisfies Properties 6.1.*

Proof. Property (i) holds from the definition of $Z, \hat{Z}$, see equation (2.5) and equation (4.40). From the definitions of $\mathbf{A}$ and $\mathbf{C}$, in equations (4.37-4.38), (ii) follows from the fact that the control at discrete node ℓ depends only on the discretized equation at node ℓ. (iii) holds from Lemma 5.3, with $C_r = 1$ (equation (5.26)). From (i) and the definitions of $\mathbf{A}$ and $\mathbf{C}$, we have that $\|\mathbf{A}(\mathcal{Q})\|$ and $\|\mathbf{C}(\mathcal{Q})\|$ are bounded independent of $\mathcal{Q}$. From equation (5.28), it follows that $\|\mathbf{A}(\mathcal{Q})^{-1}\|$ is bounded independent of $\mathcal{Q}$ as well, hence (iv) is satisfied. $\square$

Fix a vector $\mathbf{W}$. From Properties 6.1, there exists a sequence $\mathcal{Q}^k$, such that

$$\lim_{k\to\infty}\Big(-\mathbf{A}(\mathcal{Q}^k)\mathbf{W} + \mathbf{C}(\mathcal{Q}^k)\Big) = \sup_{\mathcal{Q}\in\hat{Z}} \left\{-\mathbf{A}(\mathcal{Q})\mathbf{W} + \mathbf{C}(\mathcal{Q})\right\} \ . \tag{6.2}$$

Since $\mathbf{A}(\mathcal{Q})$, $\mathbf{C}(\mathcal{Q})$ are bounded, then there is a convergent subsequence $\{\mathcal{Q}^{k_j}\}$ such that $\mathbf{A}(\mathcal{Q}^{k_j}) \to \widehat{\mathbf{A}}(\mathbf{W})$ and $\mathbf{C}(\mathcal{Q}^{k_j}) \to \widehat{\mathbf{C}}(\mathbf{W})$, for some $\widehat{\mathbf{A}}(\mathbf{W})$, $\widehat{\mathbf{C}}(\mathbf{W})$, satisfying

$$-\widehat{\mathbf{A}}(\mathbf{W})\mathbf{W} + \widehat{\mathbf{C}}(\mathbf{W}) = \sup_{\mathcal{Q}\in\hat{Z}} \left\{-\mathbf{A}(\mathcal{Q})\mathbf{W} + \mathbf{C}(\mathcal{Q})\right\} \ . \tag{6.3}$$

We also have the following result

Proposition 6.1. *If Properties 6.1 hold, with* $\widehat{\mathbf{A}}(\mathbf{W})$ *and* $\widehat{\mathbf{C}}(\mathbf{W})$ *defined in equation (6.3), then* $\widehat{\mathbf{A}}(\mathbf{W})$ *is an M-matrix, and* $\|\widehat{\mathbf{C}}(\mathbf{W})\|_\infty$ *and* $\|\widehat{\mathbf{A}}(\mathbf{W})^{-1}\|_\infty$ *are bounded uniformly w.r.t.* $\mathbf{W}$*.*

Proof. From Properties 6.1, every matrix in the sequence $\mathbf{A}(\mathcal{Q}^{k_j})$ has non-positive off-diagonals, and has $\sum_k \mathbf{A}_{\ell,k}(\mathcal{Q}^{k_j}) \geq C_r > 0$, independent of $\mathcal{Q}^{k_j}$, hence the limit of the sequence $\widehat{\mathbf{A}}(\mathbf{W})$ has these properties as well, and thus $\widehat{\mathbf{A}}(\mathbf{W})$ is an M-matrix with $\sum_k \widehat{\mathbf{A}}_{\ell,k}(\mathbf{W}) \geq C_r > 0$. Since $\|\widehat{\mathbf{A}}(\mathbf{W})^{-1}\|_\infty \leq 1/C_r$, then $\|\widehat{\mathbf{A}}(\mathbf{W})^{-1}\|_\infty$ is bounded independent of $\mathbf{W}$ (see equation (5.28)). Similarly, since $\widehat{\mathbf{C}}(\mathbf{W})$ is the limit of a sequence of $\mathbf{C}(\mathcal{Q}^{k_j})$, which are bounded independent of $\mathcal{Q}^{k_j}$, then $\widehat{\mathbf{C}}(\mathbf{W})$ is bounded independent of $\mathbf{W}$. $\square$

Policy iteration is a well known iterative method for solution of problems of type (6.1) (Howard, 1960). The policy iteration approach for solution of equation (6.1) is given in Algorithm 6.1.

The term *scale* in Algorithm 6.1 is used to ensure that unrealistic levels of accuracy are not required when the value is very small (typically *scale* for an option priced in dollars is unity). There are several possibilities for solving the linear system in the policy iteration method. In this paper, we use a preconditioned **Bi-CGSTAB** iterative method for solving the sparse matrix (Saad, 2004). We use a level one ILU preconditioner. Note that in general, the stencil changes at each policy iteration, hence we must recompute the symbolic ILU at each policy iteration.

Algorithm 6.1 Policy Iteration

Let $\mathbf{W}^0$ = Initial solution vector $\mathbf{U}^n$; given $scale > 0,\ tolerance > 0$
for $k = 0, 1, 2, \ldots$ until converge **do**
  $-\widehat{\mathbf{A}}(\mathbf{W}^k)\mathbf{W}^k + \widehat{\mathbf{C}}(\mathbf{W}^k) = \sup_{\mathcal{Q} \in \hat{Z}} \left\{ -\mathbf{A}(\mathcal{Q})\mathbf{W}^k + \mathbf{C}(\mathcal{Q}) \right\}$
  Solve the linear system $\widehat{\mathbf{A}}\left(\mathbf{W}^k\right)\mathbf{W}^{k+1} = \widehat{\mathbf{C}}\left(\mathbf{W}^k\right)$
  if $\max_\ell \dfrac{|\mathbf{W}^{k+1} - \mathbf{W}^k|}{\max\left[scale, |(\mathbf{W}^{k+1}|\right]} < tolerance$ **then**
    break from the iteration
  end if
end for
$\mathbf{U}^{n+1} = \mathbf{W}^{k+1}$

6.1 Convergence of the policy iteration

If $\mathbf{A}(\mathcal{Q}), \mathbf{C}(\mathcal{Q})$ are continuous functions of the control $\mathcal{Q}$, then convergence of the policy iteration is well known, see for example (Kushner and Dupuis, 2001). In fact, for the continuous case, superlinear convergence can be established (Bokanowski et al., 2009). However, we remind the reader that use of *central difference as much as possible* methods result in $\mathbf{A}(\mathcal{Q}), \mathbf{C}(\mathcal{Q})$ being possibly discontinuous functions of the control. Hence, in order to ensure convergence of Algorithm 6.1 in the general case, we follow along the lines in Huang et al. (2012).

Theorem 6.1. *(Convergence of policy iteration) If Properties 6.1 are satisfied, then Algorithm 6.1 converges to the unique solution of equation (6.1), for any initial iterate* $\mathbf{U}^n$*.*

Proof. See Appendix D. □

Remark 6.1. *For nodes where* $\mathbf{A}(\mathcal{Q}), \mathbf{C}(\mathcal{Q})$ *are continuous functions of* $\mathcal{Q}$*, or where the control set* $\hat{Z}$ *is finite (i.e. the control set is discretized) then trivially*

$$\begin{aligned} \widehat{\mathbf{A}}(\mathbf{W}) = \mathbf{A}(\hat{\mathcal{Q}}) \quad ; \quad & \widehat{\mathbf{C}}(\mathbf{W}) = \mathbf{C}(\hat{\mathcal{Q}}) \\ & \hat{\mathcal{Q}} \in \underset{\mathcal{Q} \in \hat{Z}}{\arg\max} \left\{ -\mathbf{A}(\mathcal{Q})\mathbf{W} + \mathbf{C}(\mathcal{Q}) \right\} \ . \end{aligned} \tag{6.4}$$

More generally, since $\hat{Z}$ *is compact, we can define the optimal control as*

$$\hat{\mathcal{Q}} \in \underset{\mathcal{Q} \in \hat{Z}}{\arg\max} \left\{ \left(-\mathbf{A}(\mathcal{Q})\mathbf{W} + \mathbf{C}(\mathcal{Q}) \right)^* \right\} \ . \tag{6.5}$$

where $(\cdot)^*$ *refers to the upper semi-continuous envelope of the argument (as a function of* $\mathcal{Q}$ *for fixed* $\mathbf{W}$*). We give the details of the method used to determine* $\hat{\mathcal{Q}}$ *in Appendix E. Note that in our case, we have only a finite number of possible discontinuities in* $\mathbf{A}(\mathcal{Q}), \mathbf{C}(\mathcal{Q})$*.*

7 Complexity: Comparison of Implicit and Explicit Methods

Each time step requires the solution of a local optimization problem at each grid node. We consider the worst case where the wide stencil is used and the control is discretized. We have shown that the numerical scheme only needs to perform a linear search along the boundary of the control set, instead of the entire three dimensional space Z. This finding decreases the complexity of evaluating the objective function from $O(\frac{1}{h^3})$ to $O(\frac{1}{h})$ for each node. Thus, with total a $O(\frac{1}{h^2})$ nodes, this gives a complexity $O(\frac{1}{h^3})$ for solving the local optimization problems at each time step. When using a fully implicit timestepping method, we also need to use policy iterations to advance time. The time complexity of solving the sparse M-matrix in each policy iteration is $O((\frac{1}{h^2})^{5/4})$ (Saad, 2004). Assuming that the number of policy iterations is bounded, as the mesh size tends to zero, which is in fact observed in our experiments, the complexity of the time advance is thus dominated by the solutions of the local optimization problems. Finally, the total complexity is $O(\frac{1}{h^4})$ with the number of time steps $O(\frac{1}{h})$.

In the existing literature (Debrabant and Jakobsen, 2013; Bonnans and Zidani, 2003), the wide stencil method and an explicit timestepping technique is typically used to solve HJB equations. The complexity of our numerical scheme in the worst case is the same as for an explicit method, using a wide stencil method, since the spatial derivatives are computed on a mesh spacing of size $\sqrt{h}$ (Debrabant and Jakobsen, 2013). However, the complexity estimate also holds for the hybrid scheme, whereby a mixture of fixed and wide stencils are used, since fully implicit timestepping does not have any stability restrictions. On the contrary, if a fixed point stencil is used at even a single node, the number of time steps for an explicit method becomes $O(\frac{1}{h^2})$ instead of $O(\frac{1}{h})$ (for a pure wide stencil scheme). Note that for nodes where a fixed point stencil is used, the analytical solution of the local optimization problem has $O(1)$ complexity.

The worst case for the implicit method compared to an explicit method (e.g. see Debrabant and Jakobsen (2013)) results in both methods having the same complexity per timestep. The implicit methods will undoubtedly have a larger constant in the order relation compared to an explicit method. Hence the overall efficiency will be purely dependent on the total number of timesteps. Since the number of timesteps for an implicit method is completely decoupled from the mesh size parameter h, we can certainly envision cases (e.g. barrier options) where a small spatial mesh parameter is required for accuracy. In this case, an explicit method would require that timesteps be directly tied to this mesh size, which may be very small, while the implicit method may use only the timestep required to minimize time truncation error. Of course, these effects will be highly problem dependent. Finally, we note that an implicit method, which is unconditionally stable, may be preferred in a production environment with inexperienced users.

8 Numerical results

Our first test case is for a European call option on the maximum of two assets with a payoff

$$\max(\max(S_1, S_2) - K, 0), \tag{8.1}$$

All model parameters are given in Table 8.1. We consider the worst-case option value for a short position. In this case, since the payoff is convex, and convexity is preserved (Janson and Tysk, 2004), the worst case price can be analytically obtained for the value with the fixed parameters $\sigma_1 = \sigma_{1,\max}$, $\sigma_2 = \sigma_{2,\max}$, $\rho = \rho_{2,\min}$. The closed-form solution (Stulz, 1982) with these volatility and correlation values is $\mathcal{U}(S_1 = 40, S_2 = 40, K = 40, t = 0) = 6.8477$. Thus, it is the solution to the HJB equation (2.2).

The numerical solutions were computed on a sequence of uniformly refined grids, starting with 91×91 grid nodes. The initial discretization parameter h (4.2) is 0.4, and the initial timestep size is 0.01. At each grid refinement, the timestep is halved. The relative convergence tolerance for nonlinear policy iteration is 10^{-6} (see Algorithm 6.1). We use $(S_1)_{\max} = (S_2)_{\max} = 400$ (i.e. about ten times the asset values of interest). We carried out some tests using $(S_1)_{\max} = (S_2)_{\max} = 2000$. The solutions at $(S_1, S_2) = (40, 40)$ were the same to six digits.

Convergence results using a pure wide stencil method based on a local coordinate system and the hybrid scheme which uses the fixed point stencil as much as possible are given in Table 8.2. Both the numerical results seem to be convergent to the benchmark. However, the hybrid scheme results are more accurate than those results obtained by the pure wide stencil method. We also carried out numerical experiments for the wide stencil based on factoring the diffusion tensor as shown in Table 8.3. The numerical results in Table 8.3 have larger errors than those in Table 8.2. Especially at the first two refinements, the pure wide stencil based on the factoring diffusion tensor performs poorly. Furthermore, the hybrid scheme significantly improves the accuracy of this pure wide stencil method. Table 8.2 and Table 8.3 also list computing time. The computer used is a standard desktop PC with a Intel Xeon E5440 CPU at 2.83GHz. The hybrid scheme requires less CPU time compared to the pure wide stencil method, at each refinement level.

Table 8.4 gives the average number of the policy iterations per time step in both the pure wide and the hybrid scheme method, which is about three. This result verifies our assumption that the number of the policy iterations is bounded as $h \to 0$, and hence the fully implicit method has the same complexity per step as an explicit method (for the pure wide stencil methods). Table 8.4 gives the ratio of the grid nodes where the fixed point stencil are used to the total number of nodes in the hybrid scheme. The ratio shows that the fixed point stencil method cannot ensure monotonicity in general.

Note that the analytical result for the worst-case value is not immediately obvious, since even though Γ_{11} and Γ_{22} (3.1) are both non-negative, Γ_{12} is non-positive for a European call option on the maximal of two asset prices. Hence, maximizing or minimizing (3.1) is not necessarily trivial, although in this case it turns out that the same volatility ($\sigma_1 = 0.5,\ \sigma_2 = 0.5$) and correlation values ($\rho = 0.3$) should be chosen for the worst-case value in theory. Further, the numerical scheme did not always set the optimal controls to the same values as for the analytical values at all grid nodes for each time step. That is, the optimal controls for the discrete equations (4.39) are not the same as values obtained in (3.1). For example, the numerical approximations of the diffusion terms sometimes had different signs than would be expected from the theoretical values. Nevertheless, by optimizing the discrete equations, the numerical solution converges to the correct solution.

Parameter	Value
Type	Call
Time to expiry (T)	0.25
r	0.05
$\sigma_{1,\min}$	0.3
$\sigma_{2,\max}$	0.5
$\sigma_{2,\min}$	0.3
$\sigma_{2,\max}$	0.5
$\rho_{\min}$	0.3
$\rho_{\max}$	0.5

Table 8.1: Model parameters for the max of two asset call option.

		Hybrid Scheme (with rotation)				Pure Wide Stencil (rotation)			
Time steps	Nodes	Value	Diff	Ratio	CPU Time	Value	Diff	Ratio	CPU Time
25	91×91	6.9182			21.01s	7.4556			31.30s
50	181×181	6.8638	0.0544		303.67s	7.1452	0.310		425.14s
100	361×361	6.8542	0.00962	5.62	4300.73s	6.9892	0.156	1.98	7209.09s
200	721×721	6.8506	0.00361	2.66	41046.12s	6.9208	0.0684	2.28	97918.79s

Table 8.2: Convergence results for an at-the-money European call option with the payoff (8.1) and parameters as given in Table 8.1. $S_1 = 40,\ S_2 = 40,\ K = 40$. Pure Wide stencil shows the numerical solutions given by a wide stencil method based on a local coordinate rotation, and Hybrid Scheme shows results obtained using the fixed point stencil as much as possible. Diff is the value of the change in the solution as the grid refined. Ratio is the ratio of successive differences. Analytic solution in this case is 6.8477. Worst case short.

Our next test uses the same parameters as in Table 8.1. The payoff has been changed to a butterfly on the maximum of two assets. In particular, the payoff is

$$\begin{aligned} S_{\max} &= \max(S_1, S_2), \\ \mathcal{W}(S_1, S_2) &= \max(S_{\max} - K_1, 0) + \max(S_{\max} - K_2, 0) - 2\max(S_{\max} - (K_1 + K_2)/2, 0). \end{aligned} \tag{8.2}$$

This test is more challenging, since the payoff of the butterfly option is no longer convex, and thus the signs of the second order derivative terms change over the solution domain. Convergence results for the worst-case

		Hybrid Scheme (with factoring)				Pure Wide Stencil (factoring)			
Time steps	Nodes	Value	Diff	Ratio	CPU Time	Value	Diff	Ratio	CPU Time
25	91×91	6.9639			30.40s	5.9476			29.42s
50	181×181	6.9302	0.0437		411.28s	6.4910	0.543		432.37s
100	361×361	6.8966	0.0336	1.30	5741.64s	6.7168	0.226	2.40	8593.83s
200	721×721	6.8746	0.0221	1.52	54789.17s	6.7942	0.0774	2.92	116443.90s

Table 8.3: Convergence results for an at-the-money European call option with the payoff (8.1) and parameters as given in Table 8.1. $S_1 = 40,\ S_2 = 40,\ K = 40$. Pure Wide Stencil shows the numerical solutions given by a wide stencil method based on factoring the diffusion tensor, and Hybrid Scheme shows results obtained using the fixed point stencil as much as possible. Diff is the value of the change in the solution as the grid refined. Ratio is the ratio of successive differences. Analytic solution in this case is 6.8477. Worst case short.

	Average Iterations		
Time steps	Hybrid Scheme	Pure Wide	Fraction Fixed
25	3.3	3.1	0.38
50	3.3	2.9	0.42
100	3.0	2.5	0.44
200	2.8	2.4	0.45

Table 8.4: The test case of a European call option on the maximum of two assets. Average Iterations is the average number of the policy iterations per time step. Pure Wide stands for the wide stencil method based on a local coordinate rotation, while Hybrid Scheme stands for the hybrid scheme using the fixed point stencil as much as possible. Fraction Fixed gives the ratio of the grid nodes where the fixed point stencil is used to the total number of nodes in the hybrid scheme.

and best-case (short position) values are given in Tables 8.5 to 8.8. The numerical results in Table 8.5 and Table 8.7 are given by the wide stencil based on a local coordinate rotation. As shown in the tables, the convergence ratio of the pure wide stencil method does not seem to be smooth. The best-case results seem to oscillate at the last two refinements. However, when we combine the wide stencil with use of a fixed point stencil as much as possible, the solution converges more smoothly.

Compared to the results in Table 8.6 and 8.8, which are given by the wide stencil method based on factoring the diffusion tensor, the performance of the wide stencil based on a local rotation seems to be superior. Both in the worst case and the best case scenarios, the errors of the pure wide stencil based on the factoring diffusion tensor are very large, especially at the first two refinements. Again, the hybrid scheme significantly improves the performance of the factoring method.

The average number of the policy iterations per time step is shown in Table 8.9 for the butterfly test case. The trends are the same as in Table 8.4, although both pure wide and hybrid stencil method tend to require more iterations on average. This is a direct result of this problem being truly nonlinear.

For comparison, Table 8.10 gives prices of the butterfly options on maximal of two assets using fixed volatility and correlation values. We see that the uncertain worst-case and best-case values form an upper and lower bound for the fixed parameter prices.

		Hybrid Scheme (with rotation)			Pure Wide Stencil (rotation)		
Time steps	Nodes	Value	Diff	Ratio	Value	Diff	Ratio
25	91×91	2.7160			2.6371		
50	181×181	2.6946	0.0214		2.6397	0.00261	
100	361×361	2.6880	0.00655	3.27	2.6650	0.0252	0.10
200	721×721	2.6862	0.00184	3.60	2.6744	0.00940	2.67

Table 8.5: Convergence results for a worst-case (short) butterfly option with parameters as given in Table 8.1 and payoff specified by equation (8.2). $S_1 = 40,\ S_2 = 40,\ K_1 = 34,\ K_2 = 46$. Pure Wide Stencil shows the numerical solutions given by a wide stencil method based on a local coordinate rotation, and Hybrid Scheme shows results obtained using of the fixed point stencil as much as possible. Diff is the value of the change in the solution as the grid refined. Ratio is the ratio of successive differences.

		Hybrid Scheme (with factoring)			Pure Wide Stencil (factoring)		
Time steps	Nodes	Value	Diff	Ratio	Value	Diff	Ratio
25	91×91	2.8518			3.1129		
50	181×181	2.7733	0.0885		2.6121	0.501	
100	361×361	2.7282	0.0452	1.96	2.6083	0.00372	135
200	721×721	2.7085	0.0196	2.31	2.6196	-0.0113	-0.32

Table 8.6: Convergence results for a worst-case (short) butterfly option with parameters as given in Table 8.1 and payoff specified by equation (8.2). $S_1 = 40,\ S_2 = 40,\ K_1 = 34,\ K_2 = 46$. Pure Wide Stencil shows the numerical solutions given by a wide stencil method based on factoring the diffusion tensor, and Hybrid Scheme shows results obtained using of the fixed point stencil as much as possible. Diff is the value of the change in the solution as the grid refined. Ratio is the ratio of successive differences.

9 Conclusions

We have developed a fully implicit, unconditionally monotone finite difference numerical scheme for the two dimensional uncertain volatility HJB equation (2.2).

		Hybrid Scheme (with rotation)			Pure Wide Stencil (rotation)		
Time steps	Nodes	Value	Diff	Ratio	Value	Diff	Ratio
25	91×91	0.9751			0.9787		
50	181×181	0.9420	0.0331		0.9213	0.0574	
100	361×361	0.9227	0.0193	1.72	0.9129	0.00842	1.69
200	721×721	0.9183	0.00435	4.44	0.9148	−0.00943	−0.89

Table 8.7: Convergence results for a best-case (short) butterfly option with parameters as given in Table 8.1 and payoff specified by equation (8.2). $S_1 = 40,\ S_2 = 40,\ K_1 = 34,\ K_2 = 46$. Pure Wide Stencil shows the numerical solutions given by a wide stencil method based on a local coordinate rotation, and Hybrid Scheme shows results obtained using of the fixed point stencil as much as possible. Diff is the value of the change in the solution as the grid refined. Ratio is the ratio of successive differences.

		Hybrid Scheme (with factoring)			Pure Wide Stencil (factoring)		
Time steps	Nodes	Value	Diff	Ratio	Value	Diff	Ratio
25	91×91	0.6448			2.3915		
50	181×181	0.7621	0.117		1.5937	0.796	
100	361×361	0.8621	0.0999	1.17	1.1287	0.465	1.71
200	721×721	0.8913	0.0293	3.41	1.0273	0.101	4.60

Table 8.8: Convergence results for a best-case (short) butterfly option with parameters as given in Table 8.1 and payoff specified by equation (8.2). $S_1 = 40,\ S_2 = 40,\ K_1 = 34,\ K_2 = 46$. Pure Wide Stencil shows the numerical solutions given by a wide stencil method based on factoring the diffusion tensor, and Hybrid Scheme shows results obtained using of the fixed point stencil as much as possible. Diff is the value of the change in the solution as the grid refined. Ratio is the ratio of successive differences.

	Average Iterations		
Time steps	Hybrid Scheme	Pure Wide	Fraction Fixed
25	4.0	3.7	0.38
50	3.8	3.7	0.42
100	3.6	3.6	0.44
200	3.3	3.3	0.45

Table 8.9: The test case for a worst-case (short) butterfly option on maximal of two assets. Average Iterations is the average number of the policy iterations per time step. Pure Wide stands for the wide stencil based on a local coordinate rotation, while Hybrid Scheme stands for the hybrid scheme using the fixed point stencil as much as possible. Fraction Fixed gives the ratio of the grid nodes where the fixed point stencil are used to the total number of nodes in the hybrid scheme.

Test	Value
Uncertain worst-case	2.6862
$\sigma_1 = 0.3,\ \sigma_2 = 0.3,\ \rho = 0.3$	2.1910
$\sigma_1 = 0.3,\ \sigma_2 = 0.3,\ \rho = 0.5$	2.1891
$\sigma_1 = 0.4,\ \sigma_2 = 0.4,\ \rho = 0.4$	1.7404
$\sigma_1 = 0.5,\ \sigma_2 = 0.5,\ \rho = 0.3$	1.4480
$\sigma_2 = 0.5,\ \sigma_2 = 0.5,\ \rho = 0.5$	1.4364
Uncertain best-case	0.9183

Table 8.10: Option values for various parameter choices with a butterfly payoff. $S_1 = 40,\ S_2 = 40,\ K_1 = 34,\ K_2 = 46,\ T = 0.25$. The worst-case and best-case (short position) are obtained by the hybrid scheme using the fixed point stencil as much as possible and the wide stencil based on a local coordinate rotation.

In general, we cannot expect solutions to HJB equations to be smooth. Hence, we seek the viscosity solution of the equation (2.2). Given a monotone scheme, it is straightforward to show that our scheme is ℓ_∞ stable (d'Halluin et al., 2004). We also prove that our numerical scheme is consistent in the viscosity sense. Consequently, we can prove that our scheme guarantees convergence to the viscosity solution. Due to the presence of the cross derivative term, a fixed point stencil will not, in general, produce a monotone discretization. We have derived a hybrid scheme which uses a fixed point stencil as much as possible and a wide stencil method as a complement to ensure monotonicity. Our numerical experiments showed that our hybrid scheme performs better than a pure wide stencil. Our numerical experiments indicated that a wide stencil scheme based on a local grid rotation seems to be superior to a scheme based on factoring the diffusion tensor.

We used fully implicit timestepping to build an unconditionally monotone numerical scheme. Implicit timestepping then requires solution of highly nonlinear algebraic equations at each time step, which are solved using the policy iteration algorithm. Our numerical discretization depends on the control, and thus results in a locally discontinuous function of the control. However, we can prove that policy iteration is still guaranteed to converge.

In our numerical scheme, the cost of constructing the data structure and solving the matrix at each timestep is dominated by the cost of solving the local optimization problems at each grid node. Therefore, the total complexity is the same as for an explicit method at each timestep using a wide stencil discretization, but there are no time step restrictions due to stability considerations. Unconditional stability also permits efficient use of the hybrid scheme (fixed point stencil as much as possible).

A Discrete equation coefficients in the fixed point stencil

The coefficients in the linear operator (4.6) are given in the following. We use three point operators for the first and second derivatives. Central Differencing in S_1 and S_2 direction:

$$\begin{aligned}
\alpha_{i,j}^{S_1,central} &= \left[\frac{(\sigma_1(S_1)_i)^2}{((S_1)_i - (S_1)_{i-1})((S_1)_{i+1} - (S_1)_{i-1})} - \frac{(r-q_1)(S_1)_i}{(S_1)_{i+1} - (S_1)_{i-1}}\right], \\
\beta_{i,j}^{S_1,central} &= \left[\frac{(\sigma_1(S_1)_i)^2}{((S_1)_{i+1} - (S_1)_i)((S_1)_{i+1} - (S_1)_{i-1})} + \frac{(r-q_1)(S_1)_i}{(S_1)_{i+1} - (S_1)_{i-1}}\right], \\
\alpha_{i,j}^{S_2,central} &= \left[\frac{(\sigma_2(S_2)_j)^2}{((S_2)_j - (S_2)_{j-1})((S_2)_{j+1} - (S_2)_{j-1})} - \frac{(r-q_2)(S_2)_j}{(S_2)_{j+1} - (S_2)_{j-1}}\right], \\
\beta_{i,j}^{S_2,central} &= \left[\frac{(\sigma_2(S_2)_j)^2}{((S_2)_{j+1} - (S_2)_j)((S_2)_{j+1} - (S_2)_{j-1})} + \frac{(r-q_2)(S_2)_j}{(S_2)_{j+1} - (S_2)_{j-1}}\right].
\end{aligned} \tag{A.1}$$

Forward/Backward Differencing in S_1 and S_2 direction (upstream):

$$
\begin{aligned}
\alpha_{i,j}^{S_1,ups} &= \left[\frac{(\sigma_1 (S_1)_i)^2}{((S_1)_i - (S_1)_{i-1})((S_1)_{i+1} - (S_1)_{i-1})} + \max\left(0, -\frac{(r-q_1)(S_1)_i}{(S_1)_i - (S_1)_{i-1}}\right)\right], \\
\beta_{i,j}^{S_1,ups} &= \left[\frac{(\sigma_1 (S_1)_i)^2}{((S_1)_{i+1} - (S_1)_i)((S_1)_{i+1} - (S_1)_{i-1})} + \max\left(0, \frac{(r-q_1)(S_1)_i}{(S_1)_{i+1} - (S_1)_i}\right)\right], \\
\alpha_{i,j}^{S_2,ups} &= \left[\frac{(\sigma_2 (S_2)_j)^2}{((S_2)_j - (S_2)_{j-1})((S_2)_{j+1} - (S_2)_{j-1})} + \max\left(0, -\frac{(r-q_2)(S_2)_j}{(S_2)_j - (S_2)_{j-1}}\right)\right], \\
\beta_{i,j}^{S_2,ups} &= \left[\frac{(\sigma_2 (S_2)_j)^2}{((S_2)_{j+1} - (S_2)_j)((S_2)_{j+1} - (S_2)_{j-1})} + \max\left(0, \frac{(r-q_2)(S_2)_j}{(S_2)_{j+1} - (S_2)_j}\right)\right].
\end{aligned} \tag{A.2}
$$

$$
\gamma_{i,j} = \begin{cases} \frac{\rho (S_1)_i (S_2)_j \sigma_1 \sigma_2}{((S_1)_{i+1} - (S_1)_i)((S_2)_{j+1} - (S_2)_j) + ((S_1)_i - (S_1)_{i-1})((S_2)_j - (S_2)_{j-1})}, & \text{if } \rho >= 0, \\ -\frac{\rho (S_1)_i (S_2)_j \sigma_1 \sigma_2}{((S_1)_{i+1} - (S_1)_i)((S_2)_j - (S_2)_{j-1}) + ((S_1)_i - (S_1)_{i-1})((S_2)_{j+1} - (S_2)_j)}, & \text{if } \rho < 0. \end{cases} \tag{A.3}
$$

B The discretized equation for the case $\big((S_1)_i, (S_2)_j, \tau^{n+1}\big) \in \Omega_{w^*}$

For the case $\big((S_1)_i, (S_2)_j, \tau^{n+1}\big) \in \Omega_{w^*}$, using Algorithm 4.1 to avoid using points below the lower boundaries, the discrete linear operator L_w^Q (4.25) needs to be modified to the form $L_{w^*}^Q$.

$$
\begin{aligned}
L_{w^*}^Q \mathcal{U}_{i,j}^{n+1} = & \frac{a_{i,j}}{h_{1,left}(h_{1,left} + h_{1,right})} \mathcal{J}_h \mathcal{U}^{n+1}\left(\mathbf{S}_{i,j} - h_{1,left} (\mathbf{R}_{i,j})_1\right) \\
& + \frac{a_{i,j}}{h_{1,right}(h_{1,left} + h_{1,right})} \mathcal{J}_h \mathcal{U}^{n+1}\left(\mathbf{S}_{i,j} + h_{1,right} (\mathbf{R}_{i,j})_1\right) \\
& + \frac{b_{i,j}}{h_{2,left}(h_{2,left} + h_{2,right})} \mathcal{J}_h \mathcal{U}^{n+1}\left(\mathbf{S}_{i,j} - h_{1,left} (\mathbf{R}_{i,j})_2\right) \\
& + \frac{b_{i,j}}{h_{2,right}(h_{2,left} + h_{2,right})} \mathcal{J}_h \mathcal{U}^{n+1}\left(\mathbf{S}_{i,j} + h_{2,right} (\mathbf{R}_{i,j})_2\right) \\
& + 1_{(r-q_1)\geq 0} \frac{(r-q_1)(S_1)_i}{\Delta^+ (S_1)_i} \mathcal{U}_{i+1,j}^{n+1} - 1_{(r-q_1)<0} \frac{(r-q_1)(S_1)_i}{\Delta^- (S_1)_i} \mathcal{U}_{i-1,j}^{n+1} \\
& + 1_{(r-q_2)\geq 0} \frac{(r-q_2)(S_2)_j}{\Delta^+ (S_2)_j} \mathcal{U}_{i,j+1}^{n+1} - 1_{(r-q_2)<0} \frac{(r-q_2)(S_2)_j}{\Delta^- (S_2)_j} \mathcal{U}_{i,j-1}^{n+1} \\
& - \Bigg(1_{(r-q_1)\geq 0} \frac{(r-q_1)(S_1)_i}{\Delta^+ (S_1)_i} - 1_{(r-q_1)<0} \frac{(r-q_1)(S_1)_i}{\Delta^- (S_1)_i} + 1_{(r-q_2)\geq 0} \frac{(r-q_2)(S_2)_j}{\Delta^+ (S_2)_j} \\
& + 1_{(r-q_2)<0} \frac{(r-q_2)(S_2)_j}{\Delta^- (S_2)_j} + \frac{a_{i,j}}{h_{1,left}(h_{1,left} + h_{1,right})} + \frac{a_{i,j}}{h_{1,right}(h_{1,left} + h_{1,right})} \\
& + \frac{b_{i,j}}{h_{2,left}(h_{2,left} + h_{2,right})} + \frac{b_{i,j}}{h_{2,right}(h_{2,left} + h_{2,right})} + r \Bigg) \mathcal{U}_{i,j}^{n+1},
\end{aligned} \tag{B.1}
$$

where $h_{k,left},\ h_{k,right},\ k = 1, 2$ are determined by Algorithm 4.1. Then, using fully implicit timestepping, the HJB equation (2.2) has the following discretized equation for this case

$$
\frac{\mathcal{U}^{n+1} - \mathcal{U}^n}{\Delta \tau} = \sup_{Q \in \partial Z_h} \left(L_{w^*}^Q \mathcal{U}_{i,j}^{n+1} \right). \tag{B.2}
$$

C Proof of the local consistency of $L_{w^*}^Q$

Proof. We use the discrete linear operator $L_{w^*}^Q$ (B.1) in the region $\mathbf{x}_{i,j}^{n+1} \in \Omega_{w^*}$. Ω_{w^*} is the region in Ω_b where the conditions (4.22) are not satisfied and then the wide stencil is used. As defined in Table 4.1, Ω_b is

$$\Omega_b \equiv [h, \sqrt{h}] \times (0, S_{2,\max}] \times (0, T] \cup (0, S_{1,\max}] \times [h, \sqrt{h}] \times (0, T], \tag{C.1}$$

where h (4.2) is a mesh discretization parameter.

We divide this region Ω_b into two parts. The first part Ω_{b_1} is defined as

$$\Omega_{b_1} \equiv [h, \sqrt{h}] \times [h, \sqrt{h}] \times (0, T], \tag{C.2}$$

and $\Omega_{b_2} = \Omega_b / \Omega_{b_1}$.

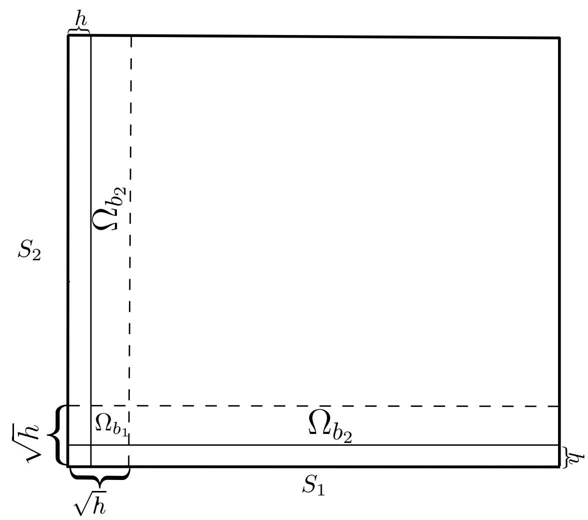


Figure C.1: The region Ω_b.

Algorithm 4.1 guides us as to how to shrink the stencil length to avoid using points below the lower boundaries when approximating the second order terms $\frac{\partial^2 \mathcal{V}}{\partial y_k^2}$, $k = 1, 2$ (4.10). If $\mathbf{x}_{i,j}^{n+1} \in \Omega_{w^*} \cap \Omega_{b_2}$, we only need to change either the value of $h_{k,left}$ or $h_{k,right}$ from $\sqrt{h}$ to h, but not both. Only if $\mathbf{x}_{i,j}^{n+1} \in \Omega_{w^*} \cap \Omega_{b_1}$, we may shrink $h_{k,left}$ and $h_{k,right}$ to h simultaneously.

For the case $\mathbf{x}_{i,j}^{n+1} \in \Omega_{w^*} \cap \Omega_{b_2}$, without loss of generality, let $h_{k,left} = h$ and $h_{k,right} = \sqrt{h}$. Suppose ϕ is a smooth test function and we use linear interpolation operator $\mathcal{J}_h$, then we have

$$\begin{aligned}
&\frac{\frac{\mathcal{J}_h \phi^{n+1}(\mathbf{S}_{i,j} - h(\mathbf{R}_{i,j})_k) - \phi^{n+1}(\mathbf{S}_{i,j})}{h} + \frac{\mathcal{J}_h \phi^{n+1}\left(\mathbf{S}_{i,j} + \sqrt{h}(\mathbf{R}_{i,j})_k\right) - \phi^{n+1}(\mathbf{S}_{i,j})}{\sqrt{h}}}{\frac{h+\sqrt{h}}{2}} \\
&= \frac{\frac{\phi^{n+1}(\mathbf{y}_{i,j} - h\mathbf{e}_k) - \phi^{n+1}(\mathbf{y}_{i,j}) + O(h^2)}{h} + \frac{\phi^{n+1}\left(\mathbf{y}_{i,j} + \sqrt{h}\mathbf{e}_k\right) - \phi^{n+1}(\mathbf{y}_{i,j}) + O(h^2)}{\sqrt{h}}}{\frac{h+\sqrt{h}}{2}} \\
&= \frac{\frac{\phi^{n+1}(\mathbf{y}_{i,j} - h\mathbf{e}_k) - \phi^{n+1}(\mathbf{y}_{i,j})}{h} + \frac{\phi^{n+1}\left(\mathbf{y}_{i,j} + \sqrt{h}\mathbf{e}_k\right) - \phi^{n+1}(\mathbf{y}_{i,j})}{\sqrt{h}}}{\frac{h+\sqrt{h}}{2}} + O(\sqrt{h}) \\
&= \frac{\partial^2 \phi}{\partial y_k^2} + O(\sqrt{h}) + O(\sqrt{h}), \quad k = 1, 2
\end{aligned} \tag{C.3}$$

which follows from Taylor series expansion and that the error of linear interpolation for a smooth function ϕ is $O(h^2)$. Thus, our discretization to the second order terms at $\mathbf{x}_{i,j}^{n+1}$ is consistent.

For the case $\mathbf{x}_{i,j}^{n+1} \in \Omega_{w^*} \cap \Omega_{b_1}$, when we shrink $h_{k,left}$ and $h_{k,right}$ to h simultaneously, following the same steps in the previous case, we have

$$\frac{\frac{\mathcal{J}_h \phi^{n+1}(\mathbf{S}_{i,j} - h(\mathbf{R}_{i,j})_k) - \phi^{n+1}(\mathbf{S}_{i,j})}{h} + \frac{\mathcal{J}_h \phi^{n+1}(\mathbf{S}_{i,j} + h(\mathbf{R}_{i,j})_k) - \phi^{n+1}(\mathbf{S}_{i,j})}{h}}{\frac{h+h}{2}} = \frac{\partial^2 \phi}{\partial y_k^2} + O(1). \tag{C.4}$$

In this case, the approximation of $\frac{\partial^2 \phi}{\partial y_k^2}$ is locally inconsistent. However, by observing the fact that the value of $a_{i,j}$ and $b_{i,j}$ in the region Ω_{b_1} is $O(h)$ (see equation (4.11)), our discretization at $\mathbf{x}_{i,j}^{n+1}$ is still locally consistent. That is,

$$a_{i,j} \underbrace{\left(\frac{\partial^2 \phi}{\partial y_1^2}\bigg|_{\mathbf{y}_{i,j}} + O(1) \right)}_{\text{approximation of } \frac{\partial^2 \phi}{\partial y_1^2}} + b_{i,j} \underbrace{\left(\frac{\partial^2 \phi}{\partial y_2^2}\bigg|_{\mathbf{y}_{i,j}} + O(1) \right)}_{\text{approximation of } \frac{\partial^2 \phi}{\partial y_2^2}} = \left(a_{i,j} \frac{\partial^2 \phi}{\partial y_1^2} + b_{i,j} \frac{\partial^2 \phi}{\partial y_2^2} \right) + O(h) = ((\mathbf{D}\nabla) \cdot \nabla \phi)\big|_{\mathbf{x}_{i,j}^{n+1}} + O(h). \tag{C.5}$$

In $L_{w^*}^Q$, we use the standard forward or backward finite differencing, depending on the sign of drift $r - q_k,\ k = 1, 2$ to discretize the first order derivatives in (2.2). The approximations of the first order derivatives are clearly locally consistent to $O(h)$. Finally, we have, in the worst case,

$$L_{w^*}^Q \phi_{i,j}^{n+1} = \mathcal{L}\phi_{i,j}^{n+1} + O(\sqrt{h}). \tag{C.6}$$

$\square$

D Proof of Theorem 6.1

For the convenience of the reader, we give a brief sketch of the proof of convergence of Policy Iteration here. Note that step 4 in Algorithm 6.1 is

$$\widehat{\mathbf{A}}\left(\mathbf{W}^k\right)\mathbf{W}^{k+1} \quad = \quad \widehat{\mathbf{C}}\left(\mathbf{W}^k\right) \tag{D.1}$$

From Proposition 6.1, $\|\widehat{\mathbf{A}}\left(\mathbf{W}\right)^{-1}\|_\infty$, and $\|\widehat{\mathbf{C}}\left(\mathbf{W}\right)\|_\infty$ are bounded independent of $\mathbf{W}$. Then, from equation (D.1), we have that $\mathbf{W}^k$ is bounded $\forall k$.

Subtract $\widehat{\mathbf{A}}\left(\mathbf{W}^k\right)\mathbf{W}^k$ from both sides of equation (D.1) to give

$$\begin{aligned}
\widehat{\mathbf{A}}\left(\mathbf{W}^k\right)\left(\mathbf{W}^{k+1} - \mathbf{W}^k\right) &= -\widehat{\mathbf{A}}\left(\mathbf{W}^k\right)\mathbf{W}^k + \widehat{\mathbf{C}}\left(\mathbf{W}^k\right) \\
&= \sup_{\mathcal{Q}\in\hat{Z}} \left\{-\mathbf{A}(\mathcal{Q})\mathbf{W}^k + \mathbf{C}(\mathcal{Q})\right\} \\
&\geq -\widehat{\mathbf{A}}\left(\mathbf{W}^{k-1}\right)\mathbf{W}^k + \widehat{\mathbf{C}}\left(\mathbf{W}^{k-1}\right) \\
&= 0
\end{aligned} \tag{D.2}$$

where the last line follows from writing equation (D.1) for $k-1$.

Since $\widehat{\mathbf{A}}\left(\mathbf{W}^k\right)$ is an M-matrix, from equation (D.2), it follows that $\mathbf{W}^{k+1} - \mathbf{W}^k \geq 0$. Since $\mathbf{W}^{k+1}$ are nondecreasing and bounded, then the iteration converges to a vector $\mathbf{W}^\infty$. Since $\widehat{\mathbf{A}}$ is bounded, we have

$$\begin{aligned}
\lim_{k\to\infty} \widehat{\mathbf{A}}\left(\mathbf{W}^k\right)\left(\mathbf{W}^{k+1} - \mathbf{W}^k\right) &= 0 \\
&= \lim_{k\to\infty} \sup_{\mathcal{Q}\in\hat{Z}} \left\{-\mathbf{A}(\mathcal{Q})\mathbf{W}^k + \mathbf{C}(\mathcal{Q})\right\} \\
&= \sup_{\mathcal{Q}\in\hat{Z}} \left\{-\mathbf{A}(\mathcal{Q})\mathbf{W}^\infty + \mathbf{C}(\mathcal{Q})\right\}\ ,
\end{aligned} \tag{D.3}$$

since $\sup(\cdot)$ is uniformly continuous w.r.t. $\mathbf{W}^k$. Hence $\mathbf{W}^\infty$ is a solution to equation (D.3). Suppose we have two solutions to (D.3), $\mathbf{X}$ and $\mathbf{Y}$, then

$$0 = \sup_{\mathcal{Q}\in\hat{Z}} \{-\mathbf{A}(\mathcal{Q})\mathbf{Y} + \mathbf{C}(\mathcal{Q})\} - \sup_{\mathcal{Q}\in\hat{Z}} \{-\mathbf{A}(\mathcal{Q})\mathbf{X} + \mathbf{C}(\mathcal{Q})\} \leq \sup_{\mathcal{Q}\in\hat{Z}} \{\mathbf{A}(\mathcal{Q})(\mathbf{X} - \mathbf{Y})\} \tag{D.4}$$

Since $\mathbf{A}(\mathcal{Q})$ is bounded, $\exists$ a sequence $\mathcal{Q}^j$ such that $\mathbf{A}(\mathcal{Q}^j) \rightarrow \bar{\mathbf{A}}$, and

$$\lim_{j\to\infty} \mathbf{A}(\mathcal{Q}^j)(\mathbf{X} - \mathbf{Y}) = \sup_{\mathcal{Q}\in\hat{Z}} \{\mathbf{A}(\mathcal{Q})(\mathbf{X} - \mathbf{Y})\} = \bar{\mathbf{A}}(\mathbf{X} - \mathbf{Y}) \geq 0 \tag{D.5}$$

Using the same steps as in the proof of Proposition 6.1, $\bar{\mathbf{A}}$ is an M-matrix, hence $\mathbf{X} \geq \mathbf{Y}$. Interchanging $\mathbf{X}$ and $\mathbf{Y}$ gives $\mathbf{Y} \geq \mathbf{X}$, hence $\mathbf{X} = \mathbf{Y}$.

E The optimal value for $\hat{Q}_\ell^k$

We give here some details of the method used to determine the optimal control. Recall that the optimal control can be defined in general as in Remark 6.1

$$\hat{\mathcal{Q}} \in \arg\max_{\mathcal{Q}\in\hat{Z}} \left\{ \left(-\mathbf{A}(\mathcal{Q})\mathbf{W} + \mathbf{C}(\mathcal{Q})\right)^* \right\} , \tag{E.1}$$

given a policy iterate $\mathbf{W}$.

In our case, we have only simple discontinuities in $\mathbf{A}(\mathcal{Q}), \mathbf{C}(\mathcal{Q})$ which occur when the discretization changes from central to forward/backward or vice versa. Consequently, we can determine $\widehat{\mathbf{A}}$ and $\widehat{\mathbf{C}}$ by first determining the optimal point $\hat{\mathcal{Q}}$, and, if this corresponds to a point of discontinuity, we take the appropriate limiting value of $\mathbf{A}(\mathcal{Q}), \mathbf{C}(\mathcal{Q})$.

For $\left((S_1)_i, (S_2)_j, \tau^{n+1}\right) \in \Omega_w \cup \Omega_{w^*}$, we have to discretize the set ∂Z (2.5), and determine the optimal value for $\hat{Q}_\ell$ by using linear search over the discrete set ∂Z_h (4.26).

For $\left((S_1)_i, (S_2)_j, \tau^{n+1}\right) \in \Omega_f$, we firstly determine the optimal $\hat{\rho}_\ell$. The discretized cross derivative term $\left(\Gamma_{12}^h(\rho)\right)_\ell$ (either (4.4) or (4.5)) depends on the sign of the correlation ρ. The choice of the optimal $\hat{\rho}_\ell$ is as follows:

$$\hat{\rho}_\ell = \begin{cases} \rho_{\max}, & \rho_{\max} \left(\Gamma_{12}^h(\rho_{\max})\right)_\ell \geq \rho_{\min} \left(\Gamma_{12}^h(\rho_{\min})\right)_\ell , \\ \rho_{\min}, & \rho_{\max} \left(\Gamma_{12}^h(\rho_{\max})\right)_\ell < \rho_{\min} \left(\Gamma_{12}^h(\rho_{\min})\right)_\ell . \end{cases} \tag{E.2}$$

Given an arbitrary pair of the volatility values (σ_1, σ_2), this choice maximizes the objective function.

Then, suppose that we only preselect a forward or backward difference depending on the sign of drift term terms (2.1) in order to discretize first order derivative terms. Then, the form of the discretized linear operator L_f^Q (4.6) is independent of the volatilities, and $\mathbf{A}(Q_\ell)$ is a continuous function of the volatilities. In addition, $\mathbf{C}_\ell(Q_\ell)$ (4.38) is constant with respect to Q_ℓ in this case. Therefore, we can determine the optimal volatilities $((\hat{\sigma}_1)_\ell, (\hat{\sigma}_2)_\ell)$ in a straightforward fashion. By inserting the optimal $\hat{\rho}_\ell$ and the discrete diffusion terms $\left(\Gamma_{kl}^h\right)_\ell$, $k, l = 1, 2$ into (E.1), a quadratic-form optimization with linear constraints needs to be solved. The form is equivalent to inserting $\hat{\rho}_\ell$ and $\left(\Gamma_{kl}^h\right)_\ell$ into (3.3). Restricting the control set to ∂Z, then the linear constraint is

$$\begin{aligned}(\sigma_1, \sigma_2) \in \Sigma \equiv \{\sigma_{1,\min} \times [\sigma_{2,\min}, \sigma_{2,\max}]\} \cup \{\sigma_{1,\max} \times [\sigma_{2,\min}, \sigma_{2,\max}]\} \\ \cup \{\sigma_{2,\min} \times (\sigma_{1,\min}, \sigma_{1,\max})\} \cup \{\sigma_{2,\max} \times (\sigma_{1,\min}, \sigma_{1,\max})\} .\end{aligned} \tag{E.3}$$

We then can obtain an analytical solution to a quadratic optimization problem.

However, if central weighting for the first order derivative terms is used as much as possible in L_f^Q in order to discretize the first order derivative terms, the form of the discretization at $\left((S_1)_i, (S_2)_j, \tau^{n+1}\right)$ is dependent on the volatilities, thus $\mathbf{A}_{\ell,k}(Q_\ell)$ (4.37) will not, in general, be a continuous of function of the volatilities. However, as shown in the last section, the proof of the convergence of the policy iterative algorithm does not require continuity of the local objective function. As in Wang and Forsyth (2008), we use Algorithm E.1 to determine the optimal volatility values. Considering node $\big((S_1)_i, (S_2)_j, \tau^{n+1}\big)$, with the current solution estimate $\mathbf{W}$ in Algorithm 6.1, the optimal $\hat{\rho}_\ell$ is determined as in (E.2). Suppose the subsets of (σ_1, σ_2), which give a positive coefficient discretization, for central, forward and backward differencing respectively, are $\Sigma_\ell^{forward}$, $\Sigma_\ell^{backward}$ and $\Sigma_\ell^{central}$. Without loss of generality, suppose the sign of the drift terms are positive in (2.1), thus we only need to select between forward and central differencing. Since central differencing is the most accurate, it should be used as much as possible. That is, $\Sigma_\ell^{forward} = \Sigma - \Sigma_\ell^{central}$.

Algorithm E.1 Determining the Optimal Control $\hat{Q}_\ell$ and the Differencing Method

Determine the optimal $\hat{\rho}_\ell = \begin{cases} \rho_{\max}, & \rho_{\max}\left(\Gamma_{12}^h(\rho_{\max})\right)_\ell \geq \rho_{\min}\left(\Gamma_{12}^h(\rho_{\min})\right)_\ell \\ \rho_{\min}, & \rho_{\max}\left(\Gamma_{12}^h(\rho_{\max})\right)_\ell < \rho_{\min}\left(\Gamma_{12}^h(\rho_{\min})\right)_\ell \end{cases}$

Compute the positive coefficient sets $\Sigma_\ell^{central}$ and $\Sigma_\ell^{forward}$ for (σ_1, σ_2).

differencing = central, $((\hat{\sigma}_1)_\ell, (\hat{\sigma}_2)_\ell) = (0, 0)$, $F_{\max} = -\infty$

for d = central, forward **do**

  Solve $(\sigma_1^d, \sigma_2^d) \in \arg\max_{(\sigma_1,\sigma_2)\in\bar{\Sigma}_\ell^d} \left[-\mathbf{A}\left(\sigma_1, \sigma_2, \hat{\rho}_\ell\right)\mathbf{W} + \mathbf{C}(\sigma_1, \sigma_2, \hat{\rho}_\ell)\right]_\ell^*$

  if $\left[-\mathbf{A}(\sigma_1^d, \sigma_2^d, \hat{\rho}_\ell)\mathbf{W} + \mathbf{C}(\sigma_1^d, \sigma_2^d, \hat{\rho}_\ell)\right]_\ell^* > F_{\max}$ **then**

    differencing = d, $((\hat{\sigma}_1)_\ell, (\hat{\sigma}_2)_\ell) = (\sigma_1^d, \sigma_2^d)$,

  end if

  $\hat{Q}_\ell = ((\hat{\sigma}_1)_\ell, (\hat{\sigma}_2)_\ell, \hat{\rho}_\ell)$

end for

In Algorithm E.1, we compute the positive coefficients set $\Sigma_\ell^{central}$ and $\Sigma_\ell^{forward}$. For a given differencing method, the range of possible values of the volatilities is divided into segments where the objective function is smooth. That is, central differencing or forward differencing can be used on disjoint intervals of Σ (E.3). On each of the subintervals, we need to maximize a quadratic problem with a linear constraint. Thus, standard methods are then used to determine the maximum within each interval, and an analytic expression for the local objective function is available. Note that in Algorithm E.1, we compute the maximum on the closure of the sets $\Sigma_\ell^{central}$, $\Sigma_\ell^{forward}$, which we denote by $\bar{\Sigma}_\ell^{central}$, $\bar{\Sigma}_\ell^{forward}$, which ensures that the maximum of the upper semi-continuous envelope is attained.

Remark E.1. *For each spatial node (i, j), we can pre-compute the range of Σ (E.3), where central, forward and backward differencing give rise to a positive coefficient method, and use the precomputed ranges $\Sigma_\ell^{central}$, $\Sigma_\ell^{forward}$ and $\Sigma_\ell^{backward}$ at each step in the policy iteration.*

References

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

Avellaneda, M. and A. Paras (1996). Managing the volatility risk of portfolios of derivative securities: the lagrangian uncertain volatility model. *Applied Mathematical Finance* 3(1), 21 – 52.

Barles, G., C. Daher, and M. Romano (1995). Convergence of numerical schemes for parabolic equations arising in finance theory. *Mathematical models and methods in applied Sciences* 5(1), 125–143.

Barles, G. and P. E. Souganidis (1991). Convergence of approximation schemes for fully nonlinear second order equations. *Asymptotic analysis* 4(3), 271–283.

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

Bonnans, J. F., E. Ottenwaelter, and H. Zidani (2004). A fast algorithm for the two dimensional HJB equation of stochastic control. *M2AN Math. Model. Numer. Anal.* 38(4), 723–735.

Bonnans, J. F. and H. Zidani (2003). Consistency of generalized finite difference schemes for the stochastic HJB equation. *SIAM Journal on Numerical Analysis* 41(3), 1008–1021.

Buff, R. (2002). *Uncertain Volatility Models: Theory and Application*, volume 1. Springer-Verlag, Berlin.

Camilli, F. and M. Falcone (1995). An approximation scheme for the optimal control of diffusion processes. *RAIRO, Modelisation mathematique et analyse numerique* 29(1), 97–122.

Clift, S. S. and P. A. Forsyth (2008). Numerical solution of two asset jump diffusion models for option valuation. *Applied Numerical Mathematics* 58(6), 743–782.

Coleman, T. F., C. He, and Y. Li (2010). Calibrating volatility function bounds for an uncertain volatility model. *Journal of Computational Finance* 13(4), 63.

Debrabant, K. and E. Jakobsen (2013). Semi-Lagrangian schemes for linear and fully non-linear diffusion equations. *Mathematics of Computation* 82(283), 1433–1462.

d'Halluin, Y., P. A. Forsyth, and G. Labahn (2004). A penalty method for American options with jump diffusion processes. *Numerische Mathematik* 97(2), 321–352.

Dokuchaev, N. G. and A. V. Savkin (1998). The pricing of options in a financial market model with transaction costs and uncertain volatility. *Journal of Multinational Financial Management* 8(2), 353–364.

Dong, H. and N. V. Krylov (2006). On the rate of convergence of finite-difference approximations for Bellman equations with constant coefficients. *St Petersburg Mathematical Journal* 17(2), 295–314.

Forsyth, P. A. and G. Labahn (2007). Numerical methods for controlled Hamilton-Jacobi-Bellman PDEs in finance. *Journal of Computational Finance* 11(2), 1–44.

Forsyth, P. A. and K. R. Vetzal (2001). Implicit solution of uncertain volatility/transaction cost option pricing models with discretely observed barriers. *Applied Numerical Mathematics* 36(4), 427–445.

Guyon, J. and P. Henry-Labordere (2011). The uncertain volatility model: a Monte Carlo approach. *Journal of Computational Finance* 14(3), 37–71.

Howard, R. A. (1960). *Dynamic Programming and Markov Processes.* MIT Press, Boston.

Huang, Y. and P. A. Forsyth (2012). Analysis of a penalty method for pricing a guaranteed minimum withdrawal benefit (GMWB). *IMA Journal of Numerical Analysis* 32(1), 320–351.

Huang, Y., P. A. Forsyth, and G. Labahn (2012). Combined fixed point and policy iteration for Hamilton–Jacobi–Bellman equations in finance. *SIAM Journal on Numerical Analysis* 50(4), 1861–1882.

Janson, S. and J. Tysk (2004). Preservation of convexity of solutions to parabolic equations. *Journal of Differential Equations* 206(1), 182–226.

Kushner, H. J. and P. G. Dupuis (2001). *Numerical methods for stochastic control problems in continuous time*, volume 24 of *Applications of Mathematics*. Springer-Verlag, New York.

Lyons, T. J. (1995). Uncertain volatility and the risk-free synthesis of derivatives. *Applied mathematical finance* 2(2), 117–133.

Menaldi, J. L. (1989). Some estimates for finite difference approximations. *SIAM Journal on Control and Optimization* 27(3), 579–607.

Øksendal, B. K. and A. Sulem (2005). *Applied stochastic control of jump diffusions.* Springer-Verlag, Berlin.

Pham, H. (2005). On some recent aspects of stochastic control and their applications. *Probability Surveys* 2(1–2), 506–549.

Pooley, D. M., P. A. Forsyth, and K. R. Vetzal (2003a). Numerical convergence properties of option pricing PDEs with uncertain volatility. *IMA Journal of Numerical Analysis* 23(2), 241–267.

Pooley, D. M., P. A. Forsyth, and K. R. Vetzal (2003b). Two factor option pricing with uncertain volatility. In *Computational Science and Its Applications ICCSA 2003*, volume 2669 of *Lecture Notes in Computer Science*, pp. 158–167. Springer-Verlag, Berlin.

Saad, Y. (2004). *Iterative Methods for Sparse Linear Systems*. SIAM, Philadelphia.

Smith, A. T. (2002). American options under uncertain volatility. *Applied Mathematical Finance* 9(2), 123–141.

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

Varga, R. S. (2009). *Matrix Iterative Analysis*. Springer-Verlag, Berlin.

Wang, J. and P. A. Forsyth (2008). Maximal use of central differencing for Hamilton-Jacobi-Bellman PDEs in finance. *SIAM Journal on Numerical Analysis* 46(3), 1580–1601.